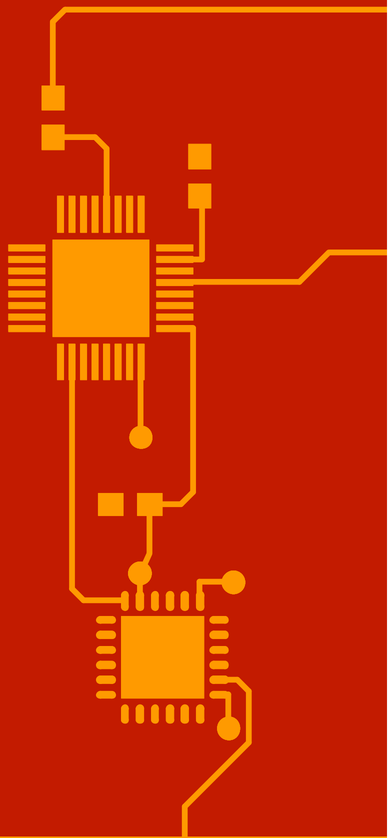


Controlling of industrial ROBOTIC MANIPULATORS operating with flexible tools

RASMUS B. B. OLESEN

Group 11gr938/12gr1038
2012, Aalborg University

AALBORG UNIVERSITY
STUDENT REPORT

AALBORG UNIVERSITY
STUDENT REPORT

The Faculty of Engineering and Science (tek-nat.aau.dk)
Department of Electronic Systems (es.aau.dk)

Section for Automation and Control (control.aau.dk)
Fredrik Bajers Vej 7, 9220 Aalborg Øst
Phone 9940 8600 (department)

Project title
Controlling of industrial robotic manipulators operating with flexible tools

Project theme
Autonomous systems

Specialization
Intelligent Autonomous Systems

Project supervisor
Henrik Schiøler

Project period
P9 (fall semester 2011)
P10 (spring semester 2012)

Project group
11gr938 (2011)
12gr1038 (2012)

Pages
130 (main report) and 142 (appendix)

Copies
6

Project initiation
5th of September 2011

Project hand-in
31st of May 2012

Participant
Rasmus B. B. Olesen

Synopsis

The objective of this Master's thesis was to improve end-effector control for an industrial robotic manipulator operating with flexible tools. A REIS RV15 manipulator including different sensors was at disposal for experiments. Improving the control involves damping the oscillations of the flexible tool, which will increase the accuracy of the TCP position and decrease the task completion time.

A dynamic model was derived for both the manipulator and the flexible tool. A sliding mode controller structure was selected to damp the tool oscillations, and the controller applied the dynamic model in the process. System identification methods were used to adapt the model to the practical tool configuration using sensor measurement. Both offline and online methods were described. Sensor information fusion was also considered to improve model state estimates. A Kalman filter was used to combine the estimates and various sensor measurements including strain of the tool.

The control system was empirically tested to measure its performance regarding TCP accuracy and settling time of the strain response. All parts were tested separately. The relay tuning method estimated the eigenfrequency of the tool to within 8-11 % of the correct value, and the Kalman filter showed noise removal properties. The controller reduced the settling time from 50,85 s to 3,85 s in one case, but did not reach sliding mode during the test. The bandwidth of the manipulator was not adequate, and the control signal had to be limited.

Det Teknisk-Naturvidenskabelige Fakultet (tek-nat.aau.dk)
Institut for Elektroniske Systemer (es.aau.dk)

Sektion for Automation og Regulering (control.aau.dk)
Fredrik Bajers Vej 7, 9220 Aalborg Øst
Tlf. 9940 8600 (institut)

Projekttitel
Regulering af industrirobotter der anvender fleksible værktøjer

Projekttema
Autonome systemer

Specialeretning
Intelligente autonome systemer

Projektvejleder
Henrik Schiøler

Projektperiode
P9 (efterårssemester 2011)
P10 (forårssemester 2012)

Projektgruppe
11gr938 (2011)
12gr1038 (2012)

Sidetal
130 (hovedrapport) and 142 (appendix)

Oplag
6

Projektstart
5. september 2011

Projektaflevering
31. maj 2012

Deltager
Rasmus B. B. Olesen

Synopsis

Formålet med dette masterprojekt var at forbedre reguleringen af end-effectoren for en industriel robotmanipulator, der anvender fleksible værktøjer. En REIS RV15 robot samt forskellige sensorer var til rådighed for forsøg. Forbedring af reguleringen involverer dæmpning af svingninger i det fleksible værktøj, hvilket vil forøge nøjagtigheden af TCP-positionen og nedbringe tiden, det tager at færdiggøre opgaven.

Der blev udviklet en model for både robotten og det fleksible værktøj. En sliding mode regulator blev valgt for at dæmpe svingningerne i værktøjet, og regulatoren anvendte den dynamiske model i processen. Methoder til systemidentifikation blev anvendt for at tilpasse modellen til den praktiske opstilling under anvendelse af sensormålinger. Både offline og online metoder blev beskrevet. Sensor information fusion blev deslige taget i betragtning for at forbedre tilstandsestimaterne. Et Kalman filter blev anvendt til at kombinere modelestimater og sensormålinger heriblandt strain for værktøjet.

Reguleringssystemet blev testet i praksis for at måle dets ydeevne mht. nøjagtigheden af TCP position og indsvingningstid for strain-målingerne. Alle delsystemer blev testet separat. Relætuning estimerede værktøjets egenfrekvens inden for 8-11 % af den korrekt værdi, og støj kunne fjernes med Kalman filteret. Regulatoren reducerede indsvingningstiden fra 50,85 s til 3,85 s i et af tilfældene, men nåede ikke i sliding mode under testen. Robottens båndbredde var ikke tilstrækkelig, og reguleringssignalet måtte desuden begrænses.

Controlling of industrial ROBOTIC MANIPULATORS operating with flexible tools

RASMUS B. B. OLESEN

Group 11gr938/12gr1038
2012, Aalborg University

To my family

Til min familie

Preface

This report covers the work from the Master's thesis of Rasmus Olesen (group 11gr938/12gr1038) at Aalborg University 2011/2012 that combines both 9-th and 10-th semesters for a thorough work on the subject. The area of specialization is Intelligent Autonomous Systems (IAS) at the section of Control Engineering under the Department of Electronic Systems.

The report is divided into the four parts *manipulator modeling*, *manipulator control*, *system evaluation* and *appendix*. Each of the parts include several chapters describing the underlying theory in details. The appendix contains supporting information, which is referred to when needed. A list of figures, a list of tables and a bibliography is included in the appendix as well.

References to the bibliography list are given by a square bracket notation like e.g. [2], which in this case refers to item number 2 in the bibliography list. A notation like e.g. [2, 3] refers to the sources 2 and 3 at the same time. Images may be inspired by other image sources, which will be clarified in the image caption. The caption will be supported by a figure number from the particular source and the text *inspired by* if the figure has been used as inspiration for an image in the report. If no sources are stated, the image is generated by the author of this report. The university logo on the cover and the title pages is from the local source [1].

An acknowledge chapter is given immediately after this preface. It lists the names of the people that the author acknowledges for their support regarding technical/theoretical issues. A nomenclature is given after the acknowledge to make an overview of the different variables and abbreviations used throughout the report. Terms listed in the nomenclature will not be attached with a source (reference to the bibliography), since they are derived from text in the report. In order to see the specific sources, the terms must be located within the report.

If units are to be indicated after an equation, they are located to the right of the equation. All units are SI-units if not otherwise noted. All other mathematical punctuations are respecting common standards in typing mathematics if not otherwise noted. Boxed equations are considered of great importance of the math in context or as a final statement to a chapter/section.

Two types of computational software has been applied in the work, including MATLAB/Simulink[1] from The MathWorks, Inc. and LabVIEW[2] from National Instruments Corporation. Both types of software will be referred to using their names, but without their registered trademark icons, which is stated here as a footnote. Different paragraphs are used throughout the report to state important issues.

Assumption: Specific hardware limitations or model simplifications
Convention: Special conventions like $j \triangleq i+1$
Example: Concepts clarified by an example, analytically or numerically
Notation: Special mathematical or linguistic notations
Definition: Something stricter than a convention

[1] MATLAB® /Simulink® are registered trademarks for The MathWorks, Inc.
[2] LabVIEW® is a registered trademark for National Instruments Corporation

Acknowledgements

A number of people have been contributing to this master thesis. Theoretical questions have been answered comprehensively and technical assistance has been offered to hardware/software related inquiries. The names of the people and their respective contributions are listed below. Academic titles for AAU employees and corresponding departments are gathered from aau.dk. Thanks to all contributors for their time and expertise.

Henrik Schiøler (associate professor, supervisor)
Department of Electronic Systems
Several different methods have been considered for this project, and Henrik provided constructive suggestions on how to apply them. He encouraged the author to apply new methods, which advanced the technical level of the thesis and introduced the author to a number of new fields within control and modeling of dynamic systems.

Ole Madsen (M.Sc., Ph.D.)
Department of Production and Mechanical Engineering
An industrial robotic manipulator was needed for the project, and Ole Madsen was contacted. He provided access to the REIS RV15 manipulator and corresponding laboratory as well as answering robotics related questions.

Leif Jakobsen (assistant engineer)
Department of Production and Mechanical Engineering
Leif installed the all important strain gauges on the flexible tool as well as answering questions about strain measurements. He also supplied equipment for measuring the eigenfrequencies of the tool.

Daniel Winther Uhrenholt (workman)
Department of Electronic Systems
Several editions of the flexible tool used for this project has been designed, and they have all been constructed by Daniel.

(*to be continued on the next page*)

Esben Toke Christensen and Jeppe Bjørn Jørgensen (students, mech. engineering)
The capabilities of the REIS RV15 manipulator was firstly demonstrated by Esben and Jeppe, who had been using the manipulator in a former student project.

Group MP 3.026/pon105-27b (5./6. semester students, mechanical engineering)[3]
In order to control the REIS RV15 manipulator, group MP 3.026/pon105-27b supplied a working platform written in the LabVIEW programming language. Members of the group, Casper Abildgaard Pedersen, Jonas Laustsen, Jens Grandjean Jørgensen and Kasper Mygind Madsen, have further demonstrated the different functions of the software as well as answering hardware/software related questions throughout the project period.

Mohsen Soltani (Ph.D., M.Sc., E.E.)
Department of Energy Technology
The finite element method, as a method for solving dynamics of a flexible beam, was introduced by Mohsen and literature was recommended.

[3]Two group names are used to distinguish between 5. semester (group MP 3.026) and 6. semester (group pon105-27b), but the members of the two group are the same.

Nomenclature

Symbol	Description
α	Actuator model constant, $\alpha_i = R_i/L_i$ [1/s]
$\boldsymbol{\alpha}$	Diagonal matrix of α_i constants
α_{cc}	Convex combination factor
β	Actuator model constant, $\beta_i = K_{ei}/L_i$ [Arad]
$\boldsymbol{\beta}$	Diagonal matrix of β_i constants
γ	Actuator model constant, $\gamma_i = K_{mi}/J_i$ [Vsrad / kgm^2]
$\boldsymbol{\gamma}$	Diagonal matrix of γ_i constants
δ	Actuator model constant, $\delta_i = B_i/J_i$ [Nms / kgm^2]
	Dirac delta function (not to be confused with actuator model constant)
Δ	Determinant of second-order polynomial
Δ_s	Limit of saturation function
$\boldsymbol{\delta}$	Diagonal matrix of δ_i constants
ϵ	General strain
ϵ_s	Gain of saturation function
$\varepsilon^{\mathsf{ja}}$	Random variable for encoder measurements [rad]
$\varepsilon^{\mathsf{gyro}}$	Random variable for gyroscope measurements [V]
$\varepsilon_c^{\mathsf{gyro}}$	Random variable for gyroscope measurements (converted) [rad/s]
$\varepsilon^{\mathsf{acc}}$	Random variable for accelerometer measurements [V]
$\varepsilon_c^{\mathsf{acc}}$	Random variable for accelerometer measurements (converted) [g]
$\varepsilon^{\mathsf{str}}$	Random variable for strain measurements [V]
ζ	General damping ratio
ζ_i	Damping ratio of i-th eigenmode
η	Scale factor for Laplace transforms
θ	Generalized coordinate vector [rad]
$\dot{\theta}$	First derivative of generalized coordinate vector [rad/s]
$\ddot{\theta}$	Second derivative of generalized coordinate vector [rad/s^2]
θ_i	Joint angle for the i-th joint (moving link i) [rad]
θ_a	Generalized coordinate vector on actuator side [rad]
$\theta_{i,\mathsf{max}}$	Maximum swivel range of i-th manipulator joint [rad]
$\dot{\theta}_{i,\mathsf{max}}$	Maximum angular velocity of i-th manipulator joint [rad/s]
$\theta_{i,0}$	Zero reference angle of i-th manipulator joint [rad]
$\theta_{i,\mathsf{SWmax},+}$	Maximum positive swivel angle of i-th joint [rad]
$\theta_{i,\mathsf{SWmax},\div}$	Maximum negative swivel angle of i-th joint [rad]
$\theta_{i,\mathsf{off}}$	Offset on i-th encoder measurement [rad]
$\breve{\theta}_i$	Measurement of i-th manipulator joint angle [rad]
$\bar{\theta}$	Generalized coordinate vector operation point
ϑ	System parameter vector

Symbol	Description
$\boldsymbol{\vartheta}$	N4SID matrix complete model estimate
κ	Gauge factor and Actuator model constant, $\kappa_i = 1/L_i$ [A/Vs] (context specific)
κ_w	Curvature of beam deflection $w(x,t)$ [1/m]
$\boldsymbol{\kappa}$	Diagonal matrix of κ_i constants
Λ	Roots to second-order polynomial
λ_r	Real part of root to second-order polynomial, $\lambda_r = \Re\{\Lambda\}$
λ_c	Complex part of root to second-order polynomial, $\lambda_c = \Im\{\Lambda\}$
λ_i	i-th eigenvalue [1/m]
μ	Laplace shift parameter
$\hat{\mu}$	Sample mean
${}^j\nu_i$	Definition of origo for i-th frame in reference to j-th frame [m]
ξ	Phasor angle [rad] and force vector [N] (context specific)
ϖ	Angle between frames $\mathbb{F}_\tau$ and $\mathbb{F}_e$ [rad]
ρ	Density [kg/m^3]
σ^{gyro}	Standard deviation of gyroscope measurements [V]
σ^{acc}	Standard deviation of accelerometer measurements [V]
τ	Index representing tool frame $\mathbb{F}_\tau$
τ_e	Excitation torque [Nm]
$\tau_{f,i}$	Nonlinear friction force on link i, Nm
$\boldsymbol{\tau}_i$	Torque on i-th link [Nm]
τ_i'	Torque on i-th link on actuator side [Nm]
τ_a	Torque from actuator [Nm]
$\tau_{\mathsf{cor},i}$	Coriolis torque affecting link i [Nm]
$\tau_{\mathsf{cen},i}$	Centrifugal torque affecting link i [Nm]
υ	Linear vertical acceleration of tool frame, $\upsilon(\dot{\theta},\ddot{\theta}) \triangleq {}^0\ddot{\nu}_\tau(\dot{\theta},\ddot{\theta})$ [m/s^2]
ϕ_i	Mode shape of i-th eigenmode [m]
$\varphi^{\mathsf{DH}}_{X_i}$	DH-parameter describing rotation of frames around X_i [rad]
$\varphi^{\mathsf{DH}}_{Z_j}$	DH-parameter describing rotation of frames around Z_j [rad]
ψ	Polynomial coefficient
ω_i	Eigenfrequency of i-th eigenmode [rad]
ω_{lc}	Limit cycle frequency
ω_{res}	Resulting frequency from relay tuning method
$\bar{\omega}_i$	Angular velocity vector of i-th link
a_b	Cross sectional area of beam [m^2]
a_c	Constant for strain measurement
A_{lc}	Limit cycle amplitude
$\mathbf{A}_s$	Continuous Kalman filter system matrix
$\mathbf{A}$	Continuous system matrix
$\mathcal{A}$	Discrete system matrix
b	Index representing base frame $\mathbb{F}_b$
B	Friction constant [Nms]
$\mathbf{B}_s$	Continuous Kalman filter input matrix
$\mathbf{B}$	Continuous input matrix
$\mathcal{B}$	Discrete input matrix
c_i	Digital count from i-th rotary encoder
$\mathbf{C}_s$	Continuous Kalman filter output matrix
C	General mapping
cm	Center of gravity / center of mass denotation
$\text{co}(\cdot)$	Convex hull
$\mathbf{C}$	Continuous out matrix
$\mathcal{C}$	Discrete out matrix
$\mathbf{C}$	Coriolis/centrifugal matrix
$\mathbb{C}^n$	n-dimensional complex value space
d	Constant for solving second-order polynomial

Symbol	Description
d_s	System disturbance and smoothing parameter for sig(s)-function
d_m	Measurement disturbance
$\mathrm{diag}(\cdot)$	Constructor of diagonal matrix
$\mathrm{dis}(v,d)$	Mapping, displacement of d in v-direction
$\mathbf{D}$	Continuous direct term matrix
$\mathcal{D}$	Discrete direct term matrix
e	Index representing end-effector frame $\mathbb{F}_e$
$\mathbf{e}_1$	Elementary x-axis vector
$\mathbf{e}_2$	Elementary y-axis vector
$\mathbf{e}_3$	Elementary z-axis vector
E	Exponential notation, e.g. $10^5 \sim 1\mathsf{E}5$
E	Young's modulus [N/m^2]
$\mathbb{E}^n$	n-dimensional Euclidean space (special case of $\mathbb{R}^n$)
f_M	Nonlinear mass matrix vector field
f_b	Force from beam self-weight [N/m]
f	General vector field or distributed force (context specific)
f_i	Eigenfrequency of i-th eigenmode [Hz]
f_l	Force from beam tip load [N/m]
f_+	Vector field for subspace $s > 0$
f_-	Vector field for subspace $s < 0$
f_{cc}	Convex combination of fields f_- and f_+
f_s	Sampling frequency
$\hat{f}$	Nominal system dynamics without uncertainties
$\mathbb{F}_i$	Frame i
F	Point force [N]
$\mathbf{F}$	Friction vector with entries F_i
$\mathbb{F}_i$	Frame i notation
G_a	Instrumental amplifier gain
$\mathbf{G}$	Gravity "vector" (see appendix I)
g	Gravitational constant, $g = 9{,}82$ [m/s^2]
g_i	Gravity vector from center of gravity of link i
$\bar{g}$	Inertial frame gravity vector
h	Height of flexible beam and relay hysteresis (context specific)[m]
h_b	Height of beam [m]
H	Hilbert operator
i, j	Frame indices and general indices
$I_{\mathsf{cm},k,i}$	Inertia of i-th link around center of gravity and around k-th axis [kgm^2]
$\mathcal{I}_i$	Inertia tensor matrix of link i [kgm^2]
I	Second moment of inertia [m^4]
I_a	Actuator current vector [A]
k	General index
ja	Joint angle denotation
J	Inertia [kgm^2]
$\mathbf{J}$	Jacobian matrix
K_e	Electrical actuator constant [Vsrad]
K_m	Mechanical actuator constant [Vsrad]
K_{eq}	Equivalent gain
L	Inductance [H]
ℓ_i	Length of i-th link [m]
ℓ_i^{cog}	Length from i-th joint to center of gravity point [m]
ℓ^{str}	Length between tool frame $\mathbb{F}_\tau$ and gauges [m]
ℓ_{off}	Offset between frame $\mathbb{F}_b$ and $\mathbb{F}_0$ [m]
$\ell_{X_i}^{\mathsf{DH}}$	DH-parameter describing translation between frames in X_i-direction [mm]
$\ell_{Z_j}^{\mathsf{DH}}$	DH-parameter describing translation between frames in Z_j-direction [mm]
ℓ	Length of beam [m]

Symbol	Description
$\mathcal{L}$	Laplace transform
$\mathcal{L}^{-1}$	Inverse Laplace transform
L_i	4-tuple DH-parameter link description
m_i	Mass of i-th manipulator joint [kg]
m_i'	Mass of i-th actuator [kg]
m_l	Mass of payload [kg]
m_b	Mass of beam [kg]
m_t	Mass of tool, $m_t \triangleq m_l + m_b$ [kg]
M	Number of modes in solution
$\mathbf{M}$	Mass matrix
n	Number of degrees of freedom
n_A	Size of system matrix $\mathbf{A}$
n_u	Number of inputs to general system identification model
n_s	Number of strain gauges in Wheatstone bridge
n_x	Number of states in general system identification model
n_y	Number of outputs from general system identification model
$\hat{N}$	Normal vector
$\mathcal{N}$	Normal (Gaussian) distribution function
N_i	Gear ratio of i-th drive train
$\mathbf{N}$	Diagonal matrix of gear ratios N_i
o	Index representing workbench frame $\mathbb{F}_o$
p	Index representing production cell frame $\mathbb{F}_p$
p_i	Center of gravity position vector of link i
p_{tcp}	Spatial position of TCP
P	General polynomial
$\mathcal{P}_i$	Generalized polynomial of order i
q_c	Component tolerance measure
q_i	Modal coordinate for i-th eigenmode
q	Scaled tool strain state
q_{ref}	Scaled tool strain state reference
q^+	Constant acceleration in subspace $s > 0$
q^-	Constant acceleration in subspace $s < 0$
$\tilde{q}$	Tracking error of scaled tool strain, $\tilde{q} = q - q_{\mathsf{ref}}$
Q_k	Constants for derivation of PDE solution
Q	Process noise covariance matrix
r	Mass ratio between beam and payload
r_{gyro}	Actual angular velocity to measure by gyroscope [rad/s]
r_{acc}	Actual acceleration to measure by accelerometer [g]
r_w	Radius of osculating circle on beam deflection $w(x,t)$ [m]
r_1, r_2	Roots in characteristic polynomial (RT)
${}^i_j\mathbf{R}$	Pure rotation from i-th frame to j-th frame
R	Resistance [Ω] and sensor noise covariance matrix (context specific)
R_g	Instrumental amplifier gain resistance [Ω]
R_i	Strain gauge resistance of i-th gauge [Ω]
$\mathrm{rot}(v, \theta)$	Mapping, rotation of θ rad around v
$\mathbb{R}_+$	Real number set $[0, \infty)$
$\mathbb{R}^n$	n-dimensional real value space
$\mathrm{sgn}()$	Signum function
$\mathrm{sat}()$	Saturation function
$\mathrm{sig}()$	Sigmoid function
s	Laplace operator and sliding function (context specific)
$\mathrm{srw}(v, d, \theta)$	Mapping, rotation of θ rad around v and displacement of d in v-direction
$\mathcal{S}_p$	Set of stabilizing control signals yielding specific performance
S	Wheatstone bridge sensitivity
$\mathcal{S}_u$	Set of possible control signals

Symbol	Description
$\mathcal{S}_s$	Set of stabilizing control signals
t	Time [s]
t_{set}	Settling time [s]
t_{nom}	Nominal settling time without controller[s]
t_r	Reaching phase time [s]
t_s	Sliding mode time [s]
t_{shift}	Phase shift for mapping function (RT)
T_s	Sampling time
T	Time delay for Dirac delta and Heaviside step function [s]
${}^j_i\mathbf{T}$	Rotational transformation from i-th frame to j-th frame
u	Control signal to servo amplifiers, $u \triangleq \dot{\boldsymbol{\tau}} = \mathbf{NF}(\ddot{\theta})$ [V]
u_+	Control signal in subspace $s > 0$
u_-	Control signal in subspace $s < 0$
u_{eq}	Equivalent control signal
U_a	Actuator voltage vector [V]
$\mathbf{U}$	Input data series for N4SID method
v	General vector
$v_{\mathsf{o}}^{\mathsf{str}}$	Voltage signal from Wheatstone bridge [V]
$v_{\mathsf{o2}}^{\mathsf{str}}$	Voltage signal from Wheatstone bridge (amplified) [V]
v_s^{str}	Voltage supply to strain measurement bridge [V]
$v_{\mathsf{o}}^{\mathsf{acc}}$	Voltage signal from accelerometer [V]
$v_{\mathsf{o}}^{\mathsf{gyro}}$	Voltage signal from gyroscope [V]
$\bar{v}_i$	Linear velocity vector of i-th frame
$\bar{v}_{\mathsf{cm},i}$	Linear velocity of center of gravity of link i [m/s]
w	Beam deflection $w(x,t)$ depending on location and time [m]
w_b	Width of beam [m]
w_h	Homogeneous part of deflection [m]
w_p	Particular part of deflection [m]
w_b	Deflection from self-weight [m]
w_l	Deflection from tip load [m]
$w^{(k)}$	k-th derivative of deflection w
W	General Laplace transformation of deflection w
W_k	Constants for derivation of PDE solution
x	Longitudinal position on beam [m]
x, y	Denotations of X_0 and Z_0, respectively (for simplicity)
x_0	Initial state vector for PEM
x^{IMU}	Distance from tool frame origo to center of IMU (x-direction) [m]
X, X_i	General/local x-axis in i-th frame
$\mathbf{X}$	State data series for N4SID method
y	Vertical position of tool frame $\mathbb{F}_\tau$, $y \triangleq -Y_\tau$ 6 m
y^{IMU}	Distance from tool frame origo to center of IMU (y-direction) [m]
Y, Y_i	General/local y-axis in i-th frame
y	Vertical position of tool frame origo
$\mathbf{Y}$	Output data series for N4SID method
Z, Z_i	General/local z-axis in i-th frame
Z_o	Instrumental amplifier input impedance [Ω]
*	Complex conjugated
!	Factorial
$^\intercal$	Matrix transpose
$\mapsto$	Linear mapping
$\therefore$	Therefore
$\mathbf{I}_k$	$k \times k$ identity matrix
$\mathbf{0}_k$	$k \times k$ zero matrix
$\bar{0}$, $\bar{1}$	Zero and one column vector, respectively
$\triangleq$	Definition

Symbol	Description
$\Re$	Real part of complex number
$\Im$	Imaginary part of complex number
$\mathbf{j}$	Imaginary number indicator, $\mathbf{j} \triangleq \sqrt{-1}$
$\forall$	For any
$\in$	Included in
∇_s	Gradient of s
$\square$	End of example and definition
$\diamond$	End of convention, assumption and notation
$\langle f(x), g(x) \rangle$	Inner product brackets of functions
$A \subset B$	Set A a subset of B
DSMC	Discrete sliding mode controller
DOF	Degrees of freedom
EKF	Extended Kalman filter
FPGA	Field programmable gate array
IMU	Inertial measuring unit
mod	Modulus notation
ODE	Ordinary differential equation
PDE	Partial differential equation
PEM	Prediction error method
PRBS	Pseudo random binary sequence
QSMB	Quasi sliding mode band
RT	Relay tuning
SMC	Sliding mode controller
TCP	Tool center point

- Clashing variables will not be mixed up in the specific context where it is used
- Variables missing from the nomenclature are explained when used
- i and j may be used at random as indices when they are not used in the same equation
- Some variables are indexed to emphasize the fact that they are related to a manipulator link
- Variables like W and W_i can mean two different things

The following short nomenclature is reserved for the describing function part of the chatter suppression section in chapter 9. This is introduced to avoid renaming common notations or creating confusing notations. If the variables are used elsewhere in the report, it will be announced beforehand.

Symbol	Description
φ	Nonlinear dynamics phase function
ϕ	Phase shift
ψ	Linear dynamics phase function
ω	Response frequency
A	Amplitude (gain of linear dynamics)
B	Amplitude (gain of nonlinear dynamics)
G	Linear dynamics amplitude function
N	Nonlinear dynamics amplitude function
$u(t)$	Response of nonlinear dynamics (Similar to control signal $u(t)$)

Appendix I does also have its own nomenclature list.

Symbol	Description
α_i	Angular acceleration of link i [rad/s^2]
a_i	Linear acceleration of link i (at the bottom of the link)
f_i	Force propagation down to link i [N]
F_i	Force on link i center of gravity [N]
$\mathcal{L}$	Lagrangian, $\mathcal{L} = \mathcal{T} - \mathcal{U}$ [J]
$\mathcal{M}_k$	Generalized inertia tensor matrix of link k
n_i	Torque propagation down to link i [Nm]
N_i	Torque on link i center of gravity [Nm]
$\mathcal{N}$	Index representing Newtonian approach
$\mathcal{U}$	Potential energy [J]
$\mathcal{U}_m$	Potential energy of manipulator alone [J]
$\mathcal{U}_t$	Potential energy of tool alone [J]
$\mathcal{T}$	Kinetic energy [J]
$\mathcal{T}_m$	Kinetic energy of manipulator alone [J]
$\mathcal{T}_t$	Kinetic energy of tool alone [J]

Contents

Chapter 1

Introduction

Industries relying on robotics to perform various repetitive and/or specialized tasks must ensure, that the systems are flexible and quickly readjustable to new productions in order for the manufactures to benefit from them. If the reconfiguration cost exceeds the profit from investing in robotic equipment in the first place there is no need for the equipment. This is only applicable to productions/operations where the robots are performing tasks accomplished by humans. Dangerous or highly complex tasks may in some cases only be ascribed for robots to perform.

A number of different elements can be used to characterize the efficiency of a production cell, and also to assist the optimization process. Time, energy consumption, acceleration of load, distance traveled, mechanical wear and so on can be defined prior to installation of the production cell. Depending on the specific requirements of the process considered, different subsets of the parameters can be selected as parts of the optimization problem. Since the parameters are somewhat related, the underlaying dependency must be included within the optimizer, such that the parameters cannot be selected independently by the user. The definition of a production cell used in this report is a local, possible enclosed, environment in a production factory including a number of machines focused on performing a certain subtask of a production.

Introducing intelligent control and optimization of a production cell work flow decreases reconfiguration time from one production to another as well as ensuring optimal handling of objects according to operator specified instructions. The objects may be known in details or automatically identified by a vision system or other system identification methods.

Two different interpretations of the term flexible are considered. The first interpretation deals with a robotic production cell being flexible in relation with reconfiguration. Whenever the task changes and/or the items are changing, the cell must be either reconfigured with the robotic equipment being relocated or the programming must be updated. Different topics can be considered in this case. The location of the robotics equipment to minimize reconfiguration costs in case of changes in production. Also, the control of one or multiple robots to intelligently adapt to changes in production may be considered, where vision systems are included to allow for automatic trajectory planning and handling of unknown objects. Intelligent controlling techniques are also beneficial if they are capable of performing complex real-time analysis of the production environment used for minimizing the energy consumption along with maintaining a fast unit production time and producing a high quality product. Second interpretation of the term flexible deals with robotics being flexible in construction. The flexibility has a number of advantages as well as drawbacks. One drawback emerges, when the mass of the manipulator payload is opposing the mass of the manipulator. This causes bending of the manipulator links, and the payload is no longer at the expected spatial location as estimated by a rigid model. Non-rigid controllers are therefore not applicable to this kind of construction. Oscillations can also arise in the links, making the location of the payload a function not only of the actuator excitation but also depending on dynamics of the manipulator links themselves. Even with these drawbacks, the advantages enables manipulators to operate in formerly impassible spaces as well as twisting through corners due to an increase in the number of links possible. Other benefits include faster response and safer operation near humans. Also the decrease in weight makes it consume less energy, which is also beneficial [95].

A project description follows this introduction in the sequel chapter and describes the general purpose of this project. Because the scope of robotics and control is large a project delimitation will ensure, that the project can be completed before submission date.

Chapter 2
Project description

Industrial robotic manipulators are used in many different areas, but efficient trajectory planning and control is common for them all. Production cells may include static and/or dynamic obstacles. Stationary equipment is referred to as static obstacles whereas dynamic obstacles covers moving objects, other manipulators and human workers. A certain schedule must be followed by the manipulator in order to keep the flow running in the production chain, and the trajectory must be optimized to achieve efficient motion. A schedule is defined as an ordered list of tasks to execute and will in most cases be of circular shape for repetitive productions.

Based on a dynamic model of the different manipulators, payloads and static obstacles an optimal schedule can be designed. Performance can then be bounded from above, which can be compared to given requirements for that particular configuration. By performance is included accuracy of *end-effector* positioning, mechanical stress requirements, motion of the load etc. An end-effector is defined as the tool at the end of a manipulator. To avoid decreasing performance according to the given requirements, numerous elements within the schedule can be considered.

One special relationship involves time and accuracy. Increasing production throughput (shorter production time) and requiring increased accuracy as well (larger energy consumption) stresses each manipulator in the production chain. It is therefore important to define a ratio between energy consumption and production time, because they are inversely proportional. Another element that may be considered is the topic of rigidity. If manipulators are considered rigid structures, the end-effector position is absolutely known in Euclidean space. However, if the manipulator is fitted with a tool that shows flexible behavior, the performance will be decreased if the flexible dynamics are not included in the control. Non-rigidity contributes with more complex system dynamics and oscillations that must be counteracted when positioning the end-effector.

2.1 Problem formulation

Based on this short introduction to manipulator optimization, the following general question can be asked, which has been selected as the overall goal of the project. It includes the relevant elements of task optimization and forms the basis of the thesis described in this report.

> *How to improve end-effector position/orientation control of industrial robotic manipulators in terms of accuracy and operating time when handling flexible tools?*

The term operating time is referring to the time from when then manipulator is ordered to move from one steady state to another (possible also back to the starting point). In order to proceed with the development, three methods are listed below. The methods are expected to answer the question on how to improve performance when handling flexible tools. All methods will comply with the project theme *autonomous systems* aiming for systems capable of working independently. Later on, a definition will describe the meaning of the term *improvement* when dealing with manipulator control.

- Tool/load dynamics must be estimated from analyzing the response to various excitations
- Measurements from several sensors must be combined to improve the estimation of end-effector dynamics
- Oscillations in flexible tools must be reduced by a controller

Each of the methods include the load of the manipulator. If the tool is flexible, the load specifications and possible dynamics are important for the overall performance. Flexible behavior must be counteracted by the controller based on the load specifications. When the tool/load dynamics are unknown in advance, they must be estimated from analyzing the system response using different sensors.

The control improvement implies achieving a better performance regarding faster response, improved repeatability/precision combined with minimizing the trajectory length for each joint. Accuracy however, will always affect the pace of the manipulator if energy must be minimized simultaneously. Other production requirements/restrictions will provide further constraint to the optimization problem.

Previous work has been performed on similar projects considering identifying the parameters of industrial robotics equipment, adaption of controllers to varying loads and trajectory optimization. A selection of papers concerning these similar subjects are described in the sequel section. Later on, the delimitation of this project will be outlined, which will contain some of the elements from previous work.

2.2 Previous work on the subject

The field of robotics is well-studied, and a lot of material is thus available on the subject. Three significant parts have so far been considered for this project as introduced in the latter section. They include *trajectory optimization* (increase pace), *manipulator/load dynamics estimation* (improve model for control) and *adaptive manipulator control* (improve control for different loads). The first part is described in a number of different ways using either a high order polynomial to represent the trajectory [10] or transforming the trajectory optimization problem before solving it in order to apply different solving techniques [101, 75].

A set of other techniques involves cubicsplines [28, 29], Béziercurves [23] or B-splines + Rational Bézier curves [106] to approximate the trajectory with a smooth path, which can be described parametrically and then outlined as an optimization problem. Although within the same type of problem, the paper [23] is based on mobile robotics. This paper tries to use a number of chained line segments to complete the trajectory. The advantage is, that the straight line is always shorter than a curved one and faster to compute. By setting up requirements to the gradient variation at the point of discontinuity, the segmented path will at some point form a trajectory that will seem smooth to the manipulator and still be capable of clearing obstacle spaces.

The second part was the estimation of manipulator/load dynamics through response analysis. A number of papers work with making parameter estimation algorithms faster when working in real-time [92], identifying the parameters of robots being described by integral models [34] and constructing linear manipulator models for specific manipulator models on the market [77]. An integral model is derived using the concept of energy equations (Lagrange) [55].

A handful of papers consider the friction parameters, that are often difficult to model precisely. They may be nonlinear and they significantly influences the precision of robotic motion. The paper [37] considers identifying the friction parameters of a manipulator, whereas [44, 45] models and identifies the friction parameters depending on the angular velocity of the robot joints and a varying load mass. Lastly, [3] describes the estimation of a load mass.

The last part of the project is the adaptive control part, which enables the manipulator controller to adapt to changes in the manipulator response and thus maintain an optimal closed-loop response. The paper [39] considers 1-DOF flexible manipulators and adapts the controller according to the load dynamics. Different techniques can be used whether they work online (while the manipulator is operating) or offline (while setting up the manipulator).

This project will not contain all the elements introduced above, but will be delimited. However, the scope of the project will still be sufficiently enough to answer the proposed question. The previous work within the field of robotics described above will contribute with methods and ideas to complete the work. A delimitation of the project is given in the following section.

2.3 Project delimitation

In order to answer the main question of the project within the project time window the project must be delimited. It was considered to optimize the joint trajectories in an attempt to exclude trajectories that will cause the flexible tool to oscillate. However, oscillations are only to be damped close to the destination point. If the oscillations are damped during the entire trajectory travel, the time performance will be decreased. A trajectory model consisting of a simple path is therefore developed, which is moving the end-effector from an initial state to the destination point without any obstacle avoidance involved. The project will be considering positioning/orienting the end-effector, which can be translated to a *tool center point* (TCP) to define a specific path for a specific point on the end-effector [22]. A flexible tool designed for the purpose of this project will simulate an end-effector during testing.

Secondly, the load dynamics are considered constant during each individual test, and therefore not able to change mass/shape during operation. Further, the mass is estimated as a point mass and inertia is therefore not included. The main focus of the project involves damping of oscillations in the tool, and the work will be fully applicable to handling tasks. Whenever a known item is lifted, the tool eigenfrequencies are changed, which the controller must adapt to. In this case, it can be performed by switching between two controllers. If the new loads on the other hand are unknown, the tool eigenfrequencies can be estimated using techniques introduced later in this report. Time varying loads in general will not be considered, but through the change in flexible tool dynamics the change in payload can be given as a model state.

As was prepared for at the beginning of the project description, measurements must be used to improve the estimate of the end-effector response. An operating time requirement is difficult to determine, because this project is not based on an actual production. However, based on a practical example of loading items into a press a requirement can be given. Different sensor information will be fused together to improve the model in general, and the model parameters will be estimated using system identification techniques. The model in question will be derived using general manipulator dynamics and partial differential equations describing the flexible tool behavior. Optimally, the operator of the manipulator should only be obliged to state the following parameters to the control system

- Pick-up location of load
- Destination of load
- Process time window

with the latter item defining the time from load pick-up to hand-over at the destination. This process involves both the pace of the manipulator when moving on the predefined joint trajectory as well as the time it takes to counteract the flexible tool oscillations.

A general model of a robot manipulator will be derived initially, which will provide both position and orientation of the end-effector. Based on this model, a system identification algorithm will be designed to estimate the model parameters describing the system behavior. This includes changes in payload and general changes in manipulator parameters such as friction.

Even though a number of elements has been removed from the project work, it is assumed to fully meet the task of damping unwanted flexible tool oscillations, which in turn will increase accuracy and pace performance. The end-effector can be positioned faster, and the accuracy can be improved by using different measurements to estimate model states. If more of the previous work is used throughout the report, it will be referred to when needed. A definition of the technical requirements follows based on the general requirements as well as the technical approaches needed to complete the project described in the above delimitation.

2.4 Technical requirements

Only generalized requirements were given from the first sections of this chapter. Within this section technical requirements are derived, which the final product must respect in order for the project to be positively accepted and answer the proposed question. Also, the requirements can be used in the technical development to describe mathematical solutions and selection of proper methods. The requirements are subdivided into the topics *modeling* and *control* as given below

Manipulator modeling requirements

- The model must include a certain amount of complexity to allow a parameter estimation algorithm to fit system states, outputs and control signals to the model structure to provide the control algorithm with the best conditions for estimating the manipulator behavior and respect given controller requirements
- Generalized kinematics must be derived to allow for arbitrary manipulator location within a production cell as well as arbitrary workbench coordinates relative to a local coordinate system
- Dynamics of the flexible tool must be modeled in order to determine the TCP trajectory from strain measurements performed on the flexible tool

If the dynamics are undermodeled, the goodness of fit between measurements and model will decrease. Generally, all models of physical systems are undermodeled representations of the actual dynamics. Therefore, applying a great deal of research in this part will form a model including various elements of the physics involved.

Manipulator control requirements

- The improved controller must perform equal or better than a controller based on a rigid manipulator model regarding oscillation settling time. This corresponds to a termination of the controller after the manipulator joints have reached their reference values

- Tool oscillations must be damped to allow fast TCP positioning. A practical example (given by Ole Madsen, see personal profile in acknowledgements) has a 10-12 s task time and a settling time of $t_{\mathsf{set}} < 5$ s is selected on the basis of this time interval
- Tool tip deflection must be below the level of the path accuracy of $\pm 0{,}1$ mm from the default manipulator controller [80]
- The controller structure must be based on nonlinear control theory

The latter item is selected to exploit the possibilities of nonlinear control in this correlation. Numerical requirements are only possible in the cases, where the user has specified a set of constraints/locations, which will only be used in examples throughout the report. A graphical representation of what needs to be achieved is shown in figure 2.1 and 2.2.

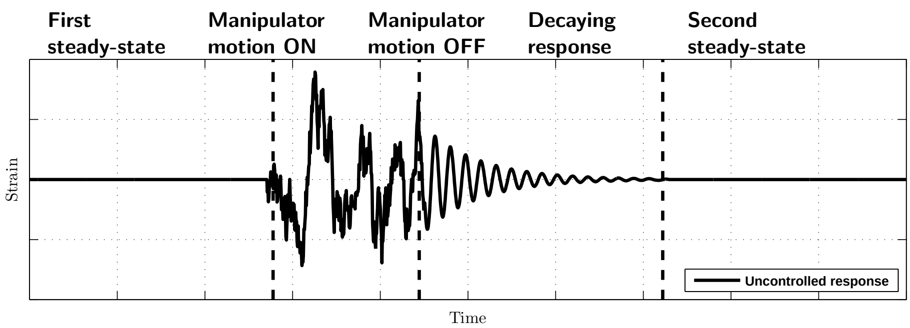


Figure 2.1: *Arbitrary strain response when moving manipulator (uncontrolled)*

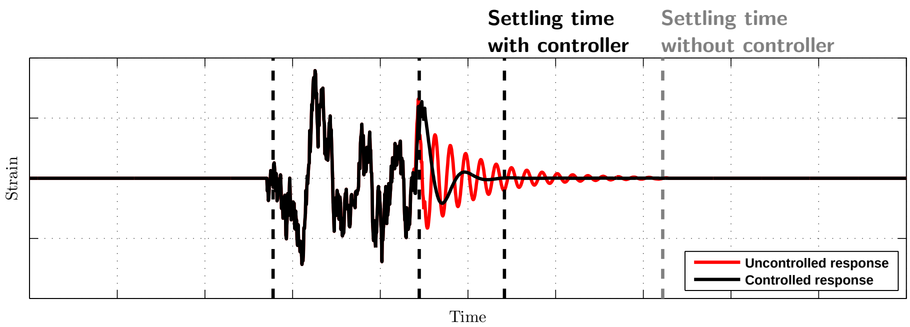


Figure 2.2: *Arbitrary strain response when moving manipulator (controlled)*

The first figure shows an example of the strain measurements during manipulator movement (example has been constructed to explicitly show the point). When the manipulator is activated from steady-state it will move to the specified destination. The decaying response is the tool oscillating after the manipulator has stopped moving. Second figure shows an example response after the system has been added to a control loop with a damping controller. The system has been added additional damping, and the oscillations are decaying more rapidly. Further details on the controller performance is provided in chapter 9. All technical requirements will be referred to throughout the report, and a summary will be given in the conclusion. Methods applied to the development process will be discussed next.

2.5 Methods applied to the development process

This project can be divided into two main categories *manipulator modeling* and *manipulator control*, which will each include a number of subelements. The order of the elements is of great importance for the progress of the project. By organizing the subjects prior to project initiation it can be ensured that they are handled in the expected order. The first category is the manipulator modeling, which will include the following subelements

- Modeling of manipulator kinematics
- Augmentation of kinematic model with dynamics
- Verification of model experimentally
- Application of the model in production cell equivalent

Derivation of a complete system model is thus the objective of the first part of the project. This model can be used in the sequel part of the project, which is concerning the manipulator control and will include the following

- Definition of controller structure
- Derivation of controller based on given structure and requirements
- Verification of model experimentally in production cell equivalent

The derived model is forming the base of the controller design, which aims to fulfill the requirements given in the above sections. By using the listed methods, the project will at all time include the necessary information to proceed

In order to verify the model and controller, sensor measurements will be logged for comparison with the theory. A detailed description of the applied methods as well as a description of the verification methods will be given in the respective chapters. The entire system is evaluated in the system evaluation part. A number of case descriptions will be given as part of the conclusion describing how the project may be used in reality.

Before ending this project description, the practical methods for showing the performance of the constructed control system will be described. An industrial manipulator will be fitted with a flexible tool analog. This analog is a flexible device with a tip mass. The mass of the tool simulates the actual weight of the tool, whereas the flexible beam between the manipulator and the tip mass simulates the mounting bracket on the manipulator. By installing strain gauges in the mounting bracket (onto the flexible beam in this case), the dynamics of the entire tool can be estimated from these measurements. It is then possible to counteract the motion of the tool in order to damp the oscillations. The tool will be described in details in the following chapter and system dynamics will be derived in the following part. A short summary of the project description will be given next.

2.6 Summary

The project described in the previous sections contains the elements depicted in figure 2.3. A reference signal to a control loop determines the trajectory of the TCP. If the tool is non-rigid, the final position can only be guaranteed in steady state (or at least after the settling time has passed), which may be undesirable. Using a controller module with prior knowledge of the system response makes it possible to predict the behavior of the manipulator and the flexible tool, and thus counteract the undesirable flexible behavior. Numerous sensors are mounted onto the manipulator structure to provide measurements of the actual system behavior.

As the system is running, the manipulator model parameters are updated based on information from the measured responses to given input signals. A system identification module solves the model equations with identical excitation signals and fits the model parameters optimally according to some pre-defined rules. This ensures, that model uncertainties and model insufficiencies are taken into account to obtain the best possible manipulator control.

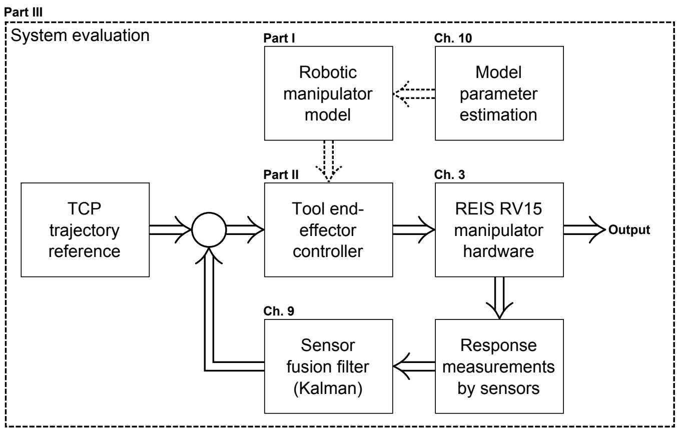


Figure 2.3: *Project configuration with report references*

The short list below summarizes the project by defining the content of the final product. The specific content of each specific item will be described in details in the subsequent chapters.

- System identification
- Sensor fusion
- Controller

Together they form an autonomous system, which is capable of automatically estimate the dynamics of the system including the flexible tool and a load. The controller is adapted to the model to achieve the necessary performance. This will in turn ensure, that the requirements are fulfilled even if the hardware configuration changes. Figure 2.4 shows the path from overall project formulation through various methods to solve the problem and finally the delimited project description and corresponding methods. The modeling is introduced in the next chapter.

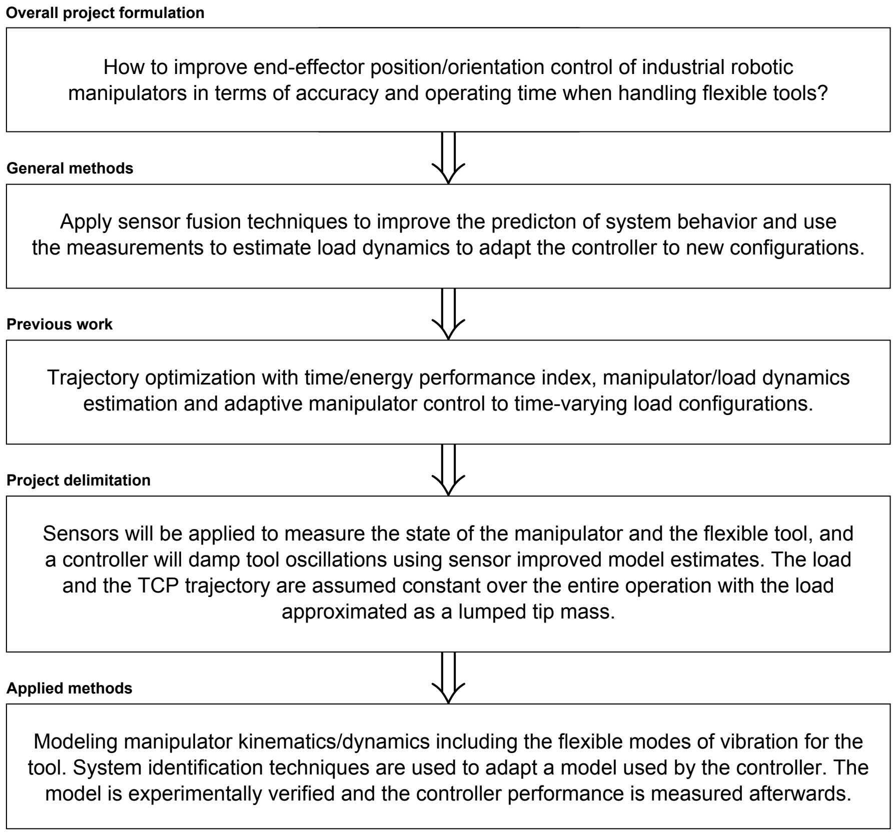


Figure 2.4: *Path from overall project formulation to final project delimitation*

PART I
Manipulator modeling

Chapter 3

Modeling introduction

Optimizing the motion of a manipulator requires knowledge of the physics involved and the hardware configuration of the production cell. The sensitivity and capability of the system must be investigated prior to applying control, since the different manipulator joints have different physical specifications and therefore show different dynamic behavior. Verification of the physics will also be considered later.

Before initiating the modeling process a number of requirements are listed to guide the process, which will ensure the expected outcome from the model. The following elements must be included in the model

- Kinematic part to determine orientations and positions
- Dynamic behavior to include inertia and friction terms
- Actuator dynamics and limitations
- Flexible tool dynamics

All properties listed will be verified at the end of the modeling part to ensure that they are all fulfilled sufficiently enough for the modeling to be successfully completed. The kinematics and dynamics (first three bullets in the above list) will be summarized in sections 3.3 and 3.4, while the actual derivation of the kinematics and dynamics of the manipulator itself is located in appendices H and I, respectively. Modeling the tool dynamics will though be given explicitly within chapter 4. From the project description in chapter 2 a set of requirements were given in section 2.4 for the modeling part specifically. These requirements are re-written below for convenience

- The model must include a certain amount of complexity to allow a parameter estimation algorithm to fit system states, outputs and control signals to the model structure to provide the control algorithm with the best conditions for estimating the manipulator behavior and respect given controller requirements
- Generalized kinematics must be derived to allow for arbitrary manipulator location within a production cell as well as arbitrary workbench coordinates relative to a local coordinate system
- Dynamics of the flexible tool must be modeled in order to determine the TCP trajectory from strain measurements performed on the flexible tool

The first requirement implies, that the complexity of the model must be of an order, which will enable a parameter fitting process to find a suitable solution to the fitting problem. If the friction for example is not included in the model, a set of measurements including this friction cannot be fitted properly within the entire range of data. By maintaining a balance between the complexity of the model and the available data for fitting, the model will provide a result close to the actual system behavior.

The second requirement states that the manipulator can be located arbitrary within the production cell, as long as the spatial coordinates and orientation can be determined from measurements. A general model must be capable of supporting different configurations. Also, the workbench coordinate system in which operations such as drilling, milling, pressing etc. can take place, is selected arbitrarily. Deriving a kinematic model with this amount of freedom allows for arbitrary manipulator configurations and a quick re-configuration in case the production flow must be changed.

Lastly, the flexible tool calls for a non-rigid model in order for the controller to position the manipulator end-effector/TCP accurately. A rigid connection between the end-effector and the TCP is assumed, and the two terms will therefore be used at random. The end-effector terminologi is only used when orientation is also an issue. As described in appendix G, strain gauges will be applied to the flexible tool to measure the strain, which is then translated into deflection. This in turn allows for determination of the end-effector/TCP position.

Before summarizing the important parts of the kinematics and dynamics from appendices H and I, respectively, a number of mathematical notations and conventions applied to the remainder of the report are listed and described in the sequel.

3.1 Mathematical notations and conventions

Before initiating the modeling work a set of mathematical notations and conventions are outlined for the convenience of the reader and to avoid misinterpretations. In order to describe the model mathematically a number of terms and expressions are applied when denoting frame orientation and origo, which will respect the following conventions throughout the report. The position of a frame origo in reference to other frames is needed and is expressed as given in notation 1. Vectors are in general described in this way.

> **Notation 1:** *Notation of j-th frame origo in reference with i-th frame origo is ${}^{i}\nu_{j}$, with ν being reserved for this purpose in this report. Vectors in general can be denoted only with v_j if they are defined in frame j by implicit.* ◇

Orientation of frames attached to a specific point on the manipulator is denoted as given by notation 2.

> **Notation 2:** *Frame orientation ${}^{i}_{j}\mathbf{T}$ denotes a description of objects in the j-th frame in reference with the i-th frame (notation from [30])* ◇

Convention 1 defines the structure of a transformation matrix and is based on DH-parameters, which are introduced in details in the kinematics section of this chapter.

Convention 1: *The order of rotation and displacement involves the following steps. Firstly, the origo of the j-th frame is displaced by ℓ^{DH}_{Zj} and subsequently rotated by $\varphi^{\mathsf{DH}}_{Zj}$. Secondly, this new j-th frame is displaced to the origo of the i-th frame by ℓ^{DH}_{Xi} and finally rotated by $\varphi^{\mathsf{DH}}_{Xi}$ to coincide the new frame with the i-th frame to complete the motion of the frame (rotation/displacement order from [30]). Mathematically it is defined as*

$$ {}^{i}_{j}\mathbf{T} = \mathrm{rot}(X_i, \varphi^{\mathsf{DH}}_{Xi})\,\mathrm{dis}(X_i, \ell^{\mathsf{DH}}_{Xi})\,\mathrm{rot}(Z_j, \varphi^{\mathsf{DH}}_{Z_j})\,\mathrm{dis}(Z_j, \ell^{\mathsf{DH}}_{Zj}) $$

with $\mathrm{rot}(\cdot)$ *and* $\mathrm{dis}(\cdot)$ *being a rotation operator and a displacement operator, respectively. The first argument defines the axis of operation, and the second argument the rate (rotation or displacement). Other conventions using different orders of transformation are possible, which provide similar results as used in [86]. A sketch of the DH-frames is depicted in figure 3.3.* $\diamond$

Details about the transformation is given when deriving the kinematics. A number of general notations are further needed. Vectors are denoted as e.g. v using lower-case letters while matrices are denoted with upper-case bold type letters like e.g. $\mathbf{M}$ except when expressed by calligraphic letters such as $\mathcal{M}$ (special cases will be denoted when used). Special vectors can be denoted with a bar like the zero $\bar{0}$ and the one $\bar{1}$ vector. Other cases will be explained when applied. As the mathematical expressions can be expanded beyond the width of this book, abbreviations for mathematical functions are introduced as well, see notation 3

Notation 3: *Trigonometric functions can be shortened in the following ways:* $\mathrm{c}\,\theta \sim \cos\theta$, $\mathrm{s}\,\theta \sim \sin\theta$, $\mathrm{t}\,\theta \sim \tan\theta$. *Sum of products of trigonometric functions* $\mathrm{c}_{12} = \cos(\theta_1 + \theta_2)$ *are also used in short form [30]. Special cases of this convention including other trigonometric functions or algebraic expressions will be specified when needed.* $\diamond$

When dealing with several variables at ones it is useful to use a shorthand description. The vector θ will be reserved for denoting the *generalized coordinate vector*, which includes variables to describe displacements and/or angles depending on the type of manipulator. It will be used frequently in the remainder of this report and is defined formally in convention 2

Convention 2: *The generalized coordinate vector θ contains joint angles and/or translations for each joint in the manipulator model. $\theta \in \mathbb{R}^n$ with n denoting the number of degrees of freedom described by the model. Commonly (in this report) it can be referred to as the joint angle space, since the REIS RV15 has no prismatic joints.* $\diamond$

A special convention regarding parts of the model describing relations between consecutive links is given by convention 3. It is introduced to simplify the model equations and ease the reading of them as well.

> **Convention 3:** *The indices i and j used for frame indexing are related by $j = i + 1$ throughout the modeling part. When the indices are used on their own they need not necessarily refer to a frame index, but can be used to indicate a conventional index.* ◇

Convention 4 provides a description of direction when working with scalar components θ_j and unit direction vectors.

> **Convention 4:** *The mathematical notation $\ddot{\theta}_j \mathbf{e}_3$ implies that $\ddot{\theta}_j$ is the vector component magnitude described in frame j in the direction of the third axis given also in frame j. There will not be given any prefixes to the unit vector like ${}^j\mathbf{e}_3$ to denote this, as it will be given by implicit* ◇

A last note before initiating the modeling of the manipulator is a remark on the number of degrees of freedom necessary to achieve satisfactory results. It is important for the work described in this report.

> **Remark 1** (model reduction)**:** *It is only possible to measure the strain of the flexible tool in one direction, and the need for making a regular manipulator model with 6 degrees of freedom is therefore unnecessary. All entries in the generalized coordinate vector θ that are causing the manipulator to work outside the XZ-plane will be replaced by a constant in practice and removed in the mathematical description. The re-definition $n \triangleq 3$ will be used in the remainder of the report, and the manipulator can therefore be referred to as a 3DOF-manipulator* □

The important mathematical notations and conventions along with an important remark on model size have been explained in the above. A short list of common terms follows in the next section.

3.2 Common terms

Before initiating the modeling of the system a short list of common terms used throughout this part is provided. This is included to avoid misinterpretations.

Destination
Final location of the item being handled by the manipulator or the TCP, and it can be stated as a position + an orientation or simply by a position depending on the tool

End-effector
Tool at the end of a manipulator.

Pick-up location
Location at which the manipulator picks up a specific item. Similar to the definition of the destination, it can be stated as a position + an orientation or simply a position depending on the tool.

Operating time
Completion time for a single task e.g. lifting an item from a conveyor belt to a machine and then return to the initial position.

Task
A schedule of specific jobs to complete by the manipulator within the operating time window.

In the sequel sections, the kinematics and dynamics will be summarized based on the theory from appendices H and I, respectively. The complete model will be verified through experiments conducted on the mechanical hardware, which is treated in the system evaluation part. Next chapter introduces a model of the flexible tool, and a controller will be developed in following part on the basis of the modeling work from this part.

Before initiating the modeling in section 3.3, figure 3.1 and 3.2 show the manipulator and the flexible tool, respectively, which must be modeled in this chapter. A detailed description of the hardware used in this project is given in appendix G. The tool specifications are referred to as the Ξ-system and are given in section G.4.

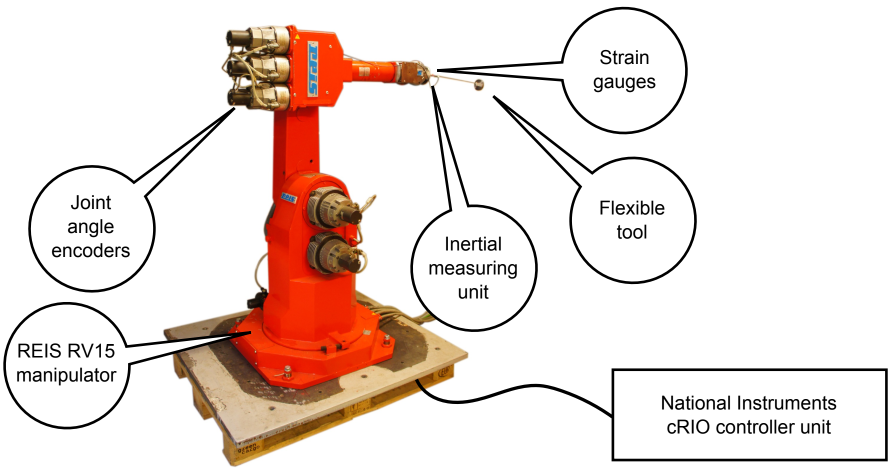


Figure 3.1: *Overall hardware configuration*

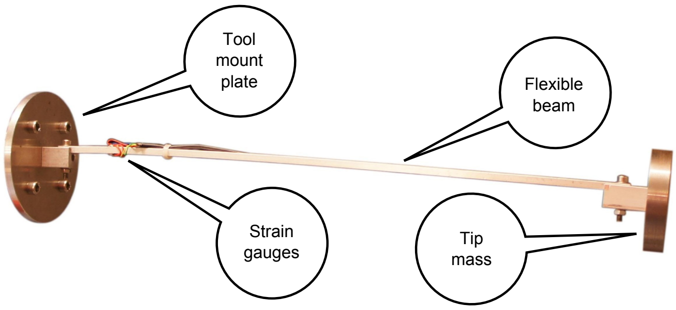


Figure 3.2: *Flexible tool configuration (image from laboratory, repeated from figure G.12)*

3.3 Manipulator kinematics model

Based on the technical requirements for the modeling part given in section 2.4 a general manipulator model must be derived. It is constructed from a kinematics model expressing positions and orientations of joints and links along with a dynamics model taking limitations of each link due to friction, inertia etc. into account. This section describes the kinematics involved in the model, that allows for mapping between different frames defined on the manipulator structure. The content of this section is given on the basis of the theory derived in appendix H, which goes into details with some of the subparts including derivation of DH-frames.

Robotics constructed from a number of serially coupled links of various configuration and motion (prismatic/revolute) can be described by the so-called *Denavit-Hartenberg parameters* [30], and the robot type is often referred to as a *manipulator*. This will also be the most widely used term to describe the mechanical construction. The parameters denote the geometry of a manipulator mathematically and together with the joint angle space θ, the position of a point or the orientation of a link can be determined. A 4-tuple collects the necessary parameters describing each link of the manipulator in terms of the previous one. It is specified for the j-th link as

$$L_j = \{\ell^{\text{DH}}_{X_i}, \varphi^{\text{DH}}_{X_i}, \ell^{\text{DH}}_{Z_j}, \varphi^{\text{DH}}_{Z_j}\} \tag{3.1}$$

where $j = i + 1$ (from convention 3) and post-superscript DH denoting that the parameters are used specific for the DH-parameter notation. Indices X_i and Z_j denotes the specific local axis of rotation/translation. The term $\ell^{\text{DH}}_{X_i}$ denotes a translation in the X_i direction, $\varphi^{\text{DH}}_{X_i}$ a rotation around the X_i-axis and so forth. The parameters are further limited by $\ell^{\text{DH}}_{X_i}, \ell^{\text{DH}}_{Z_j} \in \mathbb{R}$ and $\varphi^{\text{DH}}_{X_i}, \varphi^{\text{DH}}_{Z_j} \in [-\pi/2, \pi/2]$. A sketch of the DH-parameters is given in figure 3.3.

The DH-parameters are only used for deriving transformation matrices at the beginning of the modeling part. After that the parameters will be given implicitly through the transformations ${}^{i}_{j}\mathbf{T}$. Also, the DH-parameters are not necessarily describing actual locations on the manipulator, since frames may be coinciding. In that case,

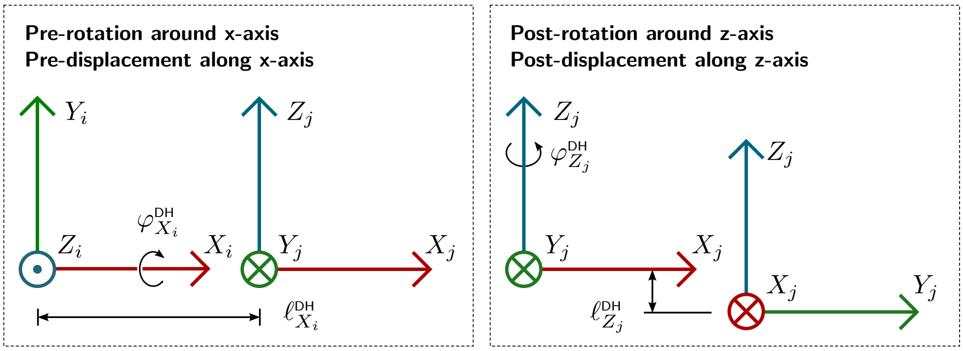


Figure 3.3: *Graphical interpretation of DH-parameters (inspired by fig. 3.15 in [30])*

special conversions must be used. It is assumed, that the rotation of a link is always around the Z_j-axis, and that all X_i-axes are pointing in the same direction. This simplifies the model as it excludes a number of redundant parameters.

The derivation of kinematics in appendix H has been based on the DH-parameters of the REIS RV15 manipulator as given in table 3.1. The parameters are from [25], and minor adjustments have been performed to fit this model. Notice that only three lengths are needed to describe the position of the origin of the last frame as the origins for frame 4 through 6 are assumed coinciding.

		$\ell_{X_i}^{\mathsf{DH}}$	$\varphi_{X_i}^{\mathsf{DH}}$	$\ell_{Z_j}^{\mathsf{DH}}$	$\varphi_{Z_j}^{\mathsf{DH}}$
Axis 1	$L_1 = \{\ell_{X_0}^{\mathsf{DH}}, \varphi_{X_0}^{\mathsf{DH}}, \ell_{Z_1}^{\mathsf{DH}}, \varphi_{Z_1}^{\mathsf{DH}}\}$	0	0	0,75	θ_1
Axis 2	$L_2 = \{\ell_{X_1}^{\mathsf{DH}}, \varphi_{X_1}^{\mathsf{DH}}, \ell_{Z_2}^{\mathsf{DH}}, \varphi_{Z_2}^{\mathsf{DH}}\}$	0	$-\pi/2$	0	θ_2
Axis 3	$L_3 = \{\ell_{X_2}^{\mathsf{DH}}, \varphi_{X_2}^{\mathsf{DH}}, \ell_{Z_3}^{\mathsf{DH}}, \varphi_{Z_3}^{\mathsf{DH}}\}$	0,60	0	0	θ_3
Axis 4	$L_4 = \{\ell_{X_3}^{\mathsf{DH}}, \varphi_{X_3}^{\mathsf{DH}}, \ell_{Z_4}^{\mathsf{DH}}, \varphi_{Z_4}^{\mathsf{DH}}\}$	0	$-\pi/2$	0,55	θ_4
Axis 5	$L_5 = \{\ell_{X_4}^{\mathsf{DH}}, \varphi_{X_4}^{\mathsf{DH}}, \ell_{Z_5}^{\mathsf{DH}}, \varphi_{Z_5}^{\mathsf{DH}}\}$	0	$-\pi/2$	0	θ_5
Axis 6	$L_6 = \{\ell_{X_5}^{\mathsf{DH}}, \varphi_{X_5}^{\mathsf{DH}}, \ell_{Z_6}^{\mathsf{DH}}, \varphi_{Z_6}^{\mathsf{DH}}\}$	0	$+\pi/2$	0	θ_6

Table 3.1: *Denavit-Hartenberg parameters for the REIS RV15 from [25] (units in m and rad)*

A frequently used DH-parameter is $\varphi_{Z_j}^{\mathsf{DH}}$, which is therefore defined as $\varphi_{Z_j}^{\mathsf{DH}} \equiv \theta_j$. The angle θ_j describes the angular displacement of link j (will be used after the derivation of the general transformation) according to some initial configuration. Positive direction of rotation is defined according to the usual cross product rule $Z = X \times Y$. A positive angle is given when the first elementary axis $\mathbf{e}_1$ is rotating towards the second elementary axis $\mathbf{e}_2$.

Due to remark 1 on page 18, not all DH-parameters from table 3.1 are needed. The altered parameters are given in table 3.2 and includes only three links (for details on the derivation, see appendix H).

		$\ell^{\mathsf{DH}}_{X_i}$	$\varphi^{\mathsf{DH}}_{X_i}$	$\ell^{\mathsf{DH}}_{Z_j}$	$\varphi^{\mathsf{DH}}_{Z_j}$
Axis 1	$L_1 = \{\ell^{\mathsf{DH}}_{X_0}, \varphi^{\mathsf{DH}}_{X_0}, \ell^{\mathsf{DH}}_{Z_1}, \varphi^{\mathsf{DH}}_{Z_1}\}$	0	$-\pi/2$	0	θ_1
Axis 2	$L_2 = \{\ell^{\mathsf{DH}}_{X_1}, \varphi^{\mathsf{DH}}_{X_1}, \ell^{\mathsf{DH}}_{Z_2}, \varphi^{\mathsf{DH}}_{Z_2}\}$	ℓ_1	0	0	θ_2
Axis 3	$L_3 = \{\ell^{\mathsf{DH}}_{X_2}, \varphi^{\mathsf{DH}}_{X_2}, \ell^{\mathsf{DH}}_{Z_3}, \varphi^{\mathsf{DH}}_{Z_3}\}$	ℓ_2	0	0	θ_3

Table 3.2: *Denavit-Hartenberg parameters (angles in rad) for re-configured setup [30]*

Numerical values have been substituted with variables to simplify the expressions. The variables are given as $\ell_1 = 0{,}6$ and $\ell_2 = 0{,}684$. Figure 3.4 shows the location of the new frames graphically.

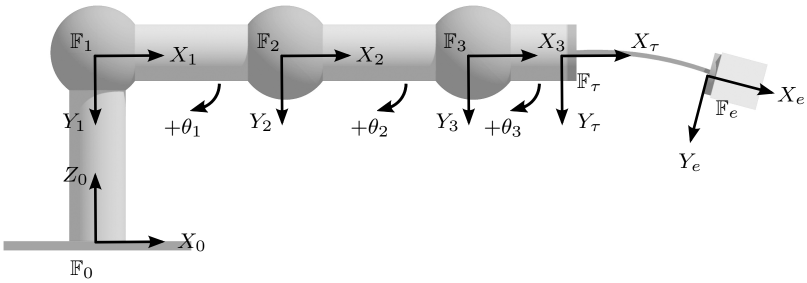


Figure 3.4: *Frame orientations on 3DOF manipulator configuration*

Frames $\mathbb{F}_\tau$ and $\mathbb{F}_e$ denote the tool frame and the end-effector frame, respectively. Whenever the position of a point on the manipulator can be expressed in terms of the joint angles, the linear velocity and linear acceleration can be determined for the point as well. This enables the expression of a transfer function from applied actuator torque to resulting motion of the end-effector in an arbitrary direction when deriving the dynamics. A number of matrices are resulting from the kinematics work to perform a *homogeneous transformation*. They are derived on the basis of the DH-parameters of table 3.2 and the general transformation (H.4). The joint angle dependent matrices and a constant offset matrix are expressed as

$$
{}^{b}_{0}\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \ell_{\text{off}} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^{0}_{1}\mathbf{T}(\theta_1) = \begin{bmatrix} \mathrm{c}\,\theta_1 & -\mathrm{s}\,\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\mathrm{s}\,\theta_1 & -\mathrm{c}\,\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{1}_{2}\mathbf{T}(\theta_2) = \begin{bmatrix} \mathrm{c}\,\theta_2 & -\mathrm{s}\,\theta_2 & 0 & \ell_1 \\ \mathrm{s}\,\theta_2 & \mathrm{c}\,\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad {}^{2}_{3}\mathbf{T}(\theta_3) = \begin{bmatrix} \mathrm{c}\,\theta_3 & -\mathrm{s}\,\theta_3 & 0 & \ell_2 \\ \mathrm{s}\,\theta_3 & \mathrm{c}\,\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

with the first matrix expressing the offset $\ell_{\text{off}} = 0{,}7$ from base frame $\mathbb{F}_b$ to zero-frame of the manipulator $\mathbb{F}_0$. The zero-frame is the basis of the manipulator model. By multiplying the matrices as well as pre/post-multiplying constant or time dependent matrices a full kinematic transform can be expressed as [86]

$$ {}^{o}_{e}\mathbf{T}\,(t,\theta,\varpi) = {}^{p}_{o}\mathbf{T}^{\,-1}(t)\,{}^{p}_{b}\mathbf{T}\,(t)\,{}^{b}_{0}\mathbf{T}\left(\prod_{j=1}^{n} {}^{i}_{j}\mathbf{T}\,(\theta_j)\right){}^{n}_{\tau}\mathbf{T}\,{}^{\tau}_{e}\mathbf{T}\,(\varpi,t) \tag{3.2} $$

o	Workbench frame $\mathbb{F}_o$	p	Production cell frame $\mathbb{F}_p$
b	Base frame $\mathbb{F}_b$	$[0,n]$	Manipulator frames $\mathbb{F}_i$
τ	Tool frame $\mathbb{F}_\tau$	e	End-effector frame $\mathbb{F}_e$

For any manipulator configuration the end-effector position can be determined with respect to the production cell coordinate system. The variable ϖ is the angle between the tool frame $\mathbb{F}_\tau$ and the frame attached to the end-effector frame $\mathbb{F}_e$. It is a consequence of small-signal approximation when modeling the flexible behavior of the tool (see section 5.1). It is derived and described in details in section 5.1.

This type of calculation is referred to as *forward kinematics* because the generalized coordinate vector is known in advance [30]. The inverse process is known as *inverse kinematics* and determines the set of possible manipulator configurations for a given end-effector coordinate. This will however not be used in this project. Figure H.1 further shows, how the different parts of the production cell are located, and the figure is repeated in figure 3.5.

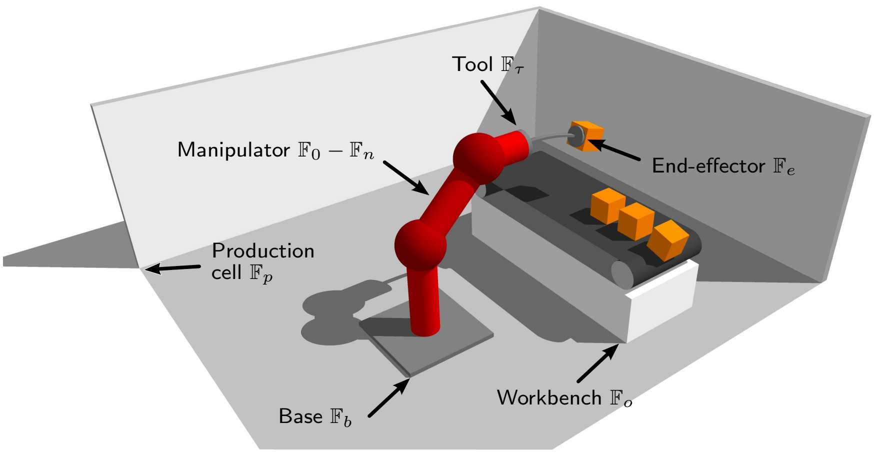


Figure 3.5: *Sketch of frames in production cell (repeated from figure H.1)*

Since the objective of this project does not involve task scheduling of the manipulator and surrounding production, the transformations ${}^{p}_{b}\mathbf{T}\,(t)^{-1}$ (base to workbench) and ${}^{p}_{0}\mathbf{T}\,(t)$ (cell to base) are equivalent to $\mathbf{I}_4$. They can be changed at any time to transform the spatial coordinates representing the motion of the manipulator. For simplicity in the remainder of the report, the applied transformation matrix in given as ${}^{0}_{e}\mathbf{T}\,(t,\theta,\varpi)$, representing motion of the end-effector in base/floor frame coordinates. Based on manipulator specifications from appendix G the constant map ${}^{n}_{\tau}\mathbf{T} = {}^{3}_{\tau}\mathbf{T}$ between last manipulator frame $\mathbb{F}_n$ and tool mount frame $\mathbb{F}_\tau$ is given as

$$
{}^{3}_{\tau}\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & \ell_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}
$$

with $\ell_3 = |{}^{\tau}\nu_n|$ denoting the distance between frame $\mathbb{F}_n$ and $\mathbb{F}_\tau$ origo. After this frame the flexible tool will be mounted. The transforms are all representing a rotation around the local z-axis Z_i and a translation equivalent to the length of the link in question. Before confusing the different terms end-effector, tool frame and tool center point a short description is given explicitly. Also, the relation to other manipulator configurations is denoted. Because this project is used to demonstrate a concept, the hardware configuration may not be similar to other configurations.

Tool mount: Plate at the end of manipulator, link n
Tool mount frame $\mathbb{F}_\tau$: Frame at the end of manipulator without tool
End-effector: Tool mounted on the tool mount frame $\mathbb{F}_\tau$
End-effector frame $\mathbb{F}_e$: Frame at the end of end-effector with origo defining TCP
Tool center point (TCP): Origo of end-effector frame $\mathbb{F}_e$

The kinematic transformation from base to end-effector is now derived though with the transform ${}^{\tau}_{e}\mathbf{T}(\varpi, t)$ being left unknown for now (derived in chapter 4). Deriving of manipulator dynamics is following in the sequel section.

3.4 Manipulator dynamics model

The important transition from a kinematic to a dynamic model is the inclusion of hardware dependent parameters such as inertia, friction etc, that in some manner affects the free motion of the manipulator links. In the previous section the kinematics were derived, making it possible to determine the spatial position of any point on the manipulator for any general coordinate θ. A system of differential equations can subsequently be determined in a regular fashion based on the generalized expression

$$f_M(\theta,t)\ddot{\theta} = f(\theta,\dot{\theta},t) \tag{3.4}$$

with $f_M(\cdot)$ denoting a nonlinear *mass matrix* vector field, which in the rotational case introduces inertia of the mechanics into the dynamic equations. $f(\cdot)$ expresses the system equations depending on time as well as the *generalized coordinate vector* θ and the time derivative $\dot{\theta}$. This includes gravity, frictions, excitation torques $\tau_e(t)$, Coriolis forces and centrifugal forces. Including the terms as separate parts of the equation the manipulator dynamics can be stated as a *joint space model* [86]

$$\mathbf{M}(\theta,t)\ddot{\theta} = \tau(t) - \mathbf{C}(\theta,\dot{\theta}) - \mathbf{F}(\dot{\theta}) - \mathbf{G}(\theta) \tag{3.5}$$

where all friction terms are collapsed into one dissipation term $\mathbf{F}(\dot{\theta})$. All terms are described in details in appendix I. The time dependency of the mass matrix is preserved, since a change in payload will change the mass matrix accordingly. $\mathbf{C}$ is a collection of Coriolis and centrifugal terms, while $\mathbf{G}$ contains terms including the gravitational constant $g = 9{,}82$ m/s^2. A linear version can be constructed from equation (3.5)

$$\mathbf{M}(\theta,t)\ddot{\theta} = \tau(t) - \mathbf{C}(\theta,\dot{\theta})\dot{\theta} - \mathbf{F}\dot{\theta} - \mathbf{G}\theta \tag{3.6}$$

The mass matrix will in both cases be kept linear in the accelerations of joint angles $\ddot{\theta}$. Using one of the above models makes it possible to predict the behavior of an n-DOF manipulator under actuation and dynamic loads. The theory of the model is defined in appendix I, and an explicit model write-out is given in appendix N for the 3-DOF manipulator configuration.

As a result of remark 1 on page 18 the dynamic model is strongly reduced and therefore $\theta \in \mathbb{R}^n$ with $n = 3$. The actuators were modeled by (I.6) and (I.7) in appendix I, and they are repeated below [41]

$$\dot{I}_{a,i} = -\alpha_i I_{a,i} - \beta_i \dot{\theta}_{a,i} + \kappa_i U_{a,i} \tag{3.7}$$

$$\ddot{\theta}_{a,i} = \gamma_i I_{a,i} - \delta_i \dot{\theta}_{a,i} + \tau_{f,i}(t,\theta_{a,i})/J_i \tag{3.8}$$

with the constants being defined as $\alpha_i = R_i/L_i$, $\beta_i = K_{ei}/L_i$, $\gamma_i = K_{mi}/J_i$, $\delta_i = B_i/J_i$ and $\kappa_i = 1/L_i$ being motor specific constants, where R_i is the electrical armature resistance [Ω], L_i the armature inductance [H], J_i the armature inertia [kgm^2], B_i the friction [Nms] and $K_{ei} = K_{mi}$ the electrical and mechanical motor constant [Vs rad], respectively. $\tau_{f,i}(t,\theta_{a,i})$ is denoting the nonlinear friction terms and external torque contributions. Notice how θ has been replaced by θ_a to denote the joint space of the actuators. Two models are now representing basically the same thing, namely the angular accelerations $\ddot{\theta}$ and $\ddot{\theta}_a$. In order to accommodate the fact that two models express the same state (one is a scaled version of the other), the mechanical part of the actuator model is omitted and represented by the joint space model alone. The equations needed to finalize the model are therefore

$$\begin{aligned}
\theta_i &= \frac{1}{N_i}\theta_{a,i} && \text{Drive train gearing}\\
\dot{I}_{a,i} &= -\alpha_i I_{a,i} - \beta_i \dot{\theta}_{a,i} + \kappa_i U_{a,i} && \text{Actuator dynamics}\\
\tau &= \mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta,\dot{\theta}) + \mathbf{NF}(\dot{\theta}) + \mathbf{G}(\theta) && \text{Joint space dynamics}\\
\tau_i &= N_i \gamma_i I_i && \text{Torque from actuator}
\end{aligned}$$

By implicit the generalized coordinate vector θ is the joint angles, whereas θ_a is the actuator angles. The gear ratio $N > 1$ and thus reducing velocity from actuator to link when $\theta = \theta_a/N$. Equation (3.8) from the actuator model has been omitted on behalf of the manipulator model. This already expresses the angular acceleration of a link, while frictions are modeled through the manipulator model. The nonlinear friction part $\tau_{f,i}$ will have to be added manually to the respective entries in $\mathbf{F}$. The only thing to take into consideration is the gear ratio N_i, which is multiplied to torques and frictions (thus decreasing velocity by a factor N_i). The torque produced by the actuator is used as the driving force for the manipulator dynamics. By combining the equations a nonlinear state model can be expressed as

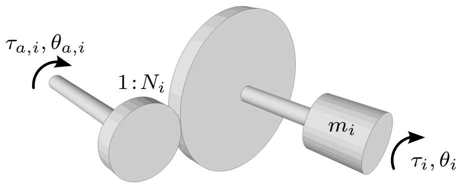
$$\begin{aligned}
\begin{bmatrix} \dot{I}_a \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} &= \overbrace{\begin{bmatrix} -\boldsymbol{\alpha} & \mathbf{0} & -\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} \\ \mathbf{M}^{-1}\mathbf{N}\boldsymbol{\gamma} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} I_a \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\kappa} U \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}^{\text{Linear part}} + \cdots \\
&\qquad \cdots + \overbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{M}^{-1}\left(-\mathbf{C}(\theta,\dot{\theta}) - \mathbf{NF}(\dot{\theta}) - \mathbf{G}(\theta)\right) \end{bmatrix}}^{\text{Nonlinear part}} \qquad (3.9)\\
\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} I_a \\ \theta \\ \dot{\theta} \end{bmatrix}
\end{aligned}$$

The system is a combination of an actuator model without linear friction (linear part) and a manipulator model (nonlinear part including actuator friction). Because of the model reduction described in remark 1 on page 18, the size of the model is given by $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\kappa}, \mathbf{N} \in \mathbb{R}^{n\times n}$ diagonal matrices given by $\boldsymbol{\alpha} = \mathrm{diag}\{\alpha_i\}_{i=1}^n$, $\boldsymbol{\beta} = \mathrm{diag}\{\beta_i\}_{i=1}^n$ and so forth, respectively. Also given are $\mathbf{M}, \mathbf{C} \in \mathbb{R}^{n\times n}$, the vector fields $\mathbf{F}, \mathbf{G} \in \mathbb{R}^{n\times 1}$ and $\mathbf{0}, \mathbf{I} \in \mathbb{R}^{n\times n}$. U_a is a vector of exogenous input signals for each actuator. All model dependent matrices are given explicitly in appendix N.

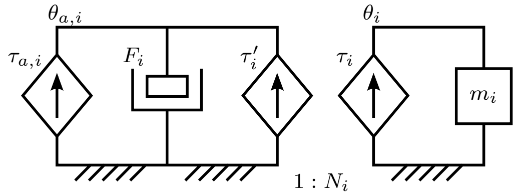
A gear ratio of a factor of 100 is featured by the REIS RV15 [79, 25]. Therefore $\mathbf{N} = 100\mathbf{I}_3$ and the dynamics of the manipulator is shadowed by the friction terms of the model and thus $\boldsymbol{\tau} = \mathbf{NF}(\dot{\theta})$. A bold τ-symbol has been used to distinguish between tool frame index τ and link torque $\boldsymbol{\tau}$. Also, the actuator drivers are featuring voltage inputs to represent a current to the actuators which in turn generates a torque. The explicit actuator model can thus be replaced by a direct torque input, and the model from (3.9) can be simplified to the following form

$$\begin{aligned}\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} & \boldsymbol{I}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ (\mathbf{NF})^{-1}\end{bmatrix}U\\ \begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} &= \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}\end{aligned} \tag{3.10}$$

This is emerging from the relation $\boldsymbol{\tau} = \mathbf{NF}(\dot{\theta})$, which can be differentiated to $u \triangleq \dot{\boldsymbol{\tau}} = \mathbf{NF}(\ddot{\theta})$, which is producing a relation with the joint space acceleration. In order to see the different relations from the above equations, figure 3.6 illustrates the principle.



(a) *Free-body diagram*



(b) *Free-body electrical analog diagram*

Figure 3.6: *Mechanical diagrams of the model in* (3.10) *for a single link*

The actuator side of the diagram satisfies the relation $\tau_{a,i} - \tau_i' = F\dot{\theta}_{a,i}$. Due to the gear ratio between actuator angle and joint angle of $N_i > 0$, then $\theta_{a,i} = N_i\theta_i$ which implies $\tau_i = N_i\tau_i'$. Using the acceleration/torque relation $\tau_i = m_i\ddot{\theta}_i$ a combined relation can be expressed as

$$N_iF_i\dot{\theta}_i = F\dot{\theta}_{a,i} = \tau_{a,i} - \tau_i' = \tau_{a,i} - \frac{\tau_i}{N_i} = \tau_{a,i} - \frac{m_i\ddot{\theta}_i}{N_i} \tag{3.11}$$

which can be rearranged to

$$\tau_{a,i} = \left(F_i\dot{\theta}_i + \frac{m_i\ddot{\theta}_i}{N_i^2}\right)N_i \approx N_iF_i\dot{\theta}_i \tag{3.12}$$

Because the second term is scaled with a factor of N_i^2, it can be canceled out and what remains is the relation $\tau_{a,i} = N_iF_i\dot{\theta}_i$ and thus $u \triangleq \dot{\tau}_{a,i} = N_iF_i\ddot{\theta}_i$. This expresses the angular acceleration of the i-th joint in terms of the input signal u, which is defined as the actuator torque derivative. The manipulator dynamics is therefore described by a friction model alone. This significantly reduces the computational load when compared to a full dynamics model for a 6-DOF manipulator.

Assumption 1 below is used to separate this model from the one derived in the sequel chapter. The flexible beam is assumed to have no impact on the motion of the remaining manipulator hardware. This implies, that the two models are decoupled.

> **Assumption 1:** *Based on practical assessments of the accessible manipulator hardware and the construction of the tool, it is assumed, that the tool is not affecting the motion of the manipulator structure itself.* □

The last part of this chapter involves determining the vertical acceleration of the tool frame $\mathbb{F}_\tau$, which is exciting the flexible tool. This will be used in the flexible model as a direct acceleration term as a function of the joint space and its derivatives.

3.5 From angular to Cartesian acceleration

In order to determine the acceleration of the tool frame $\mathbb{F}_\tau$, an expression of ${}^0\nu_\tau$ (vector from base to origo of tool frame) is differentiated in respect with time. The kinematics is used alone to determine this, as opposed to a_τ (derived in appendix I), which is including gravity. This is omitted for simplicity as the friction terms are dominating the dynamics. Accelerometer measurements in the direction of gravity must therefore be corrected by subtracting $-g$ from the measurement. First step in the acceleration derivation process involves expressing $\mathbf{T}' \triangleq {}^0\nu_\tau$ using the transformations from page 3.3

$$\mathbf{T}' = {}^0_1\mathbf{T}\,{}^1_2\mathbf{T}\,{}^2_3\mathbf{T}\,{}^3_\tau\mathbf{T}\,\mathbf{e}_4 = \begin{bmatrix} \ell_2\,\mathrm{c}_{12} + \ell_1\,\mathrm{c}_1 + \ell_3\,\mathrm{c}_3\,\mathrm{c}_{12} - \ell_3\,\mathrm{s}_3\,\mathrm{s}_{12} \\ 0 \\ \ell_0 - \ell_2\,\mathrm{s}_{12} - \ell_1\,\mathrm{s}_1 - \ell_3\,\mathrm{c}_3\,\mathrm{s}_{12} + \ell_3\,\mathrm{s}_3\,\mathrm{c}_{12} \end{bmatrix}$$

Then the velocity can be determined from

$$\begin{aligned} {}^0\dot{\nu}_\tau &= \mathbf{T}'_\theta(\theta)\dot{\theta}, \qquad \text{given } \mathbf{T}'_\theta(\theta) = \frac{\partial \mathbf{T}'(\theta)}{\partial \theta} \\ &= \begin{bmatrix} -\ell_1\,\mathrm{s}_1 - \ell_2\,\mathrm{s}_{12} - \ell_3\,\mathrm{s}_{123} & -\ell_2\,\mathrm{s}_{12} - \ell_3\,\mathrm{s}_{123} & -\ell_3\,\mathrm{s}_{123} \\ 0 & 0 & 0 \\ -\ell_1\,\mathrm{c}\,\theta_1 + \ell_2\,\mathrm{c}_{12} - \ell_3\,\mathrm{c}_{123} & -\ell_2\,\mathrm{c}_{12} - \ell_3\,\mathrm{c}_{123} & -\ell_3\,\mathrm{c}_{123} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \end{aligned}$$

Lastly, the acceleration can be derived from

$${}^0\ddot{\nu}_{\tau k} = \left\langle \left\langle {}_k\mathbf{T}'_{\theta\theta^\mathsf{T}}(\theta), \dot{\theta} \right\rangle, \dot{\theta} \right\rangle + \left\langle \left\langle {}_k\mathbf{T}'_\theta(\theta), \dot{\theta} \right\rangle, \ddot{\theta} \right\rangle, \qquad \text{given } \mathbf{T}'_{\theta\theta^\mathsf{T}}(\theta) = \frac{\partial \mathbf{T}'_\theta(\theta)}{\partial \theta}$$

with the inside dot products determined from

$${}^0\dot{\nu}_{\tau,k} = \left\langle {}_k\mathbf{T}'_\theta(\theta), \dot{\theta} \right\rangle$$

Pre-index is used as row notation, thus picking out only the k-th row of the adjacent matrix. This yields the acceleration

$$
{}^0\ddot{\nu}_\tau = \begin{bmatrix}
-\ell_1\ddot{\theta}_1\,\mathrm{s}_1 - \ell_1\dot{\theta}_1^2\,\mathrm{c}_1 - \ell_2\dot{\theta}_1^2\,\mathrm{c}_{12} - \ell_2\dot{\theta}_2^2\,\mathrm{c}_{12} - \ell_2\ddot{\theta}_1\,\mathrm{s}_{12} - \ell_2\ddot{\theta}_2\,\mathrm{s}_{12} - \cdots \\
\cdots - \ell_3\dot{\theta}_1^2\,\mathrm{c}_{123} - \ell_3\dot{\theta}_2^2\,\mathrm{c}_{123} - \ell_3\dot{\theta}_3^2\,\mathrm{c}_{123} - \ell_3\ddot{\theta}_1\,\mathrm{s}_{123} - \ell_3\ddot{\theta}_2\,\mathrm{s}_{123} - \ell_3\ddot{\theta}_3\,\mathrm{s}_{123} - \cdots \\
\cdots - 2\ell_2\dot{\theta}_1\dot{\theta}_2\,\mathrm{c}_{12} - 2\ell_3\dot{\theta}_1\dot{\theta}_2\,\mathrm{c}_{123} - 2\ell_3\dot{\theta}_1\dot{\theta}_3\,\mathrm{c}_{123} - 2\ell_3\dot{\theta}_2\dot{\theta}_3\,\mathrm{c}_{123} \\
0 \\
-\ell_1\ddot{\theta}_1\,\mathrm{c}_1 + \ell_1\dot{\theta}_1^2\,\mathrm{s}_1 + \ell_2\dot{\theta}_1^2\,\mathrm{s}_{12} + \ell_2\dot{\theta}_2^2\,\mathrm{s}_{12} - \ell_2\ddot{\theta}_1\,\mathrm{c}_{12} - \ell_2\ddot{\theta}_2\,\mathrm{c}_{12} + \cdots \\
\cdots + \ell_3\dot{\theta}_1^2\,\mathrm{s}_{123} + \ell_3\dot{\theta}_2^2\,\mathrm{s}_{123} + \ell_3\dot{\theta}_3^2\,\mathrm{s}_{123} - \ell_3\ddot{\theta}_1\,\mathrm{c}_{123} - \ell_3\ddot{\theta}_2\,\mathrm{c}_{123} - \ell_3\ddot{\theta}_3\,\mathrm{c}_{123} + \cdots \\
\cdots + 2\ell_2\dot{\theta}_1\dot{\theta}_2\,\mathrm{s}_{12} + 2\ell_3\dot{\theta}_1\dot{\theta}_2\,\mathrm{s}_{123} + 2\ell_3\dot{\theta}_1\dot{\theta}_3\,\mathrm{s}_{123} + 2\ell_3\dot{\theta}_2\dot{\theta}_3\,\mathrm{s}_{123}
\end{bmatrix}
$$

The acceleration of $\mathbb{F}_\tau$ given in Cartesian coordinates ${}^0\ddot{\nu}_\tau(\theta, \dot{\theta}, \ddot{\theta})$ is a function of the joint space and its derivatives. Using assumption 2 below, the small-signal approximation limits the joint space motion and thus $\theta = \text{const}$. This leads to a reduction of the expression, and only two states for each active joint are needed to describe the motion of the manipulator.

Assumption 2: *The control problem of damping the oscillations in the flexible tool requires only a small local section of the joint space being used. This enables small-signal approximation, making $\theta = const$, which alters ${}^0\ddot{\nu}_\tau(\theta, \dot{\theta}, \ddot{\theta})$ to ${}^0\ddot{\nu}_\tau(\dot{\theta}, \ddot{\theta})$ when evaluated at the point of operation.* □

Only the last component of ${}^0\ddot{\nu}_\tau(\theta, \dot{\theta}, \ddot{\theta})$ is needed, and when evaluated at the zero condition (offset is possible later) it becomes

$$
\upsilon(\dot{\theta}, \ddot{\theta}, t) = \begin{bmatrix}
-\ell_1\dot{\theta}_1^2 - \ell_2(\dot{\theta}_1^2 + \dot{\theta}_2^2) - \ell_3(\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2) - \cdots \\
\cdots - 2\dot{\theta}_1\dot{\theta}_2(\ell_2 + \ell_3) - 2\ell_3(\dot{\theta}_1\dot{\theta}_3 + \dot{\theta}_2\dot{\theta}_3) \\
0 \\
-\ell_1\ddot{\theta}_1 - \ell_2(\ddot{\theta}_1 + \ddot{\theta}_2) - \ell_3(\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3)
\end{bmatrix} \tag{3.13}
$$

with $\upsilon(\dot{\theta}, \ddot{\theta}) \triangleq {}^0\ddot{\nu}_\tau(\dot{\theta}, \ddot{\theta})$. The time parameter is provided by the time dependent joint space coordinates. This ends the modeling introduction, which summarized the important elements from appendices H and I. A complete model of the system has been derived and the linear acceleration of the tool frame has been expressed. The acceleration is used to excite the mode shapes of the flexible beam, which is introduced in the sequel chapter.

Chapter 4
Flexible tool model

The scope of this project is concerning how to improve manipulator control when using flexible tools. As introduced in appendix G when introducing the hardware, a flexible tool analog has been constructed for testing with an associated payload. Before the control task can be initiated, a mathematical model of the non-rigid tool dynamics must be known, which is the topic of this chapter. The mass of the payload mass may also be unknown, and it must be estimated through system identification methods, which is the topic of chapter 8.

This chapter begins with a short introduction to the term flexibility and how it can be measured. First section looks at the static case with a constant force applied to a flexible beam. Afterwards, the behavior of the tool is derived when affected by the motion of a non-stationary tool frame $\mathbb{F}_\tau$, which will excite the modes of the flexible tool. The first mode is important, because it features both the highest amplitude of vibration and is the best to counteract by slow electromechanical actuators.

The configuration involves a moving platform - the tool frame $\mathbb{F}_\tau$, and a tool that is rigidly attached. Whenever the manipulator is moving, the free end of the flexible tool is not able to follow the motion exactly due to the internal elasticity properties. Another phenomenon occurs whenever the manipulator is exciting an exact eigen-frequency of the tool, which will cause it to resonate. This can damage the tool if the oscillations become to large. In order to both protect the tool and making the end-effector follow a given trajectory, a controller must avoid exciting resonance frequencies as well as damp oscillations.

Different flexibilities are depicted in figure 4.1 for a flexible beam. Subfigure 4.1a shows twisting around a vertical z-axis, subfigure 4.1b shows bending motion around a y-axis, subfigure 4.1c shows twisting around a z-axis + bending around a y-axis and 4.1d shows bending around both an x- and a y-axis. In this figure the z-axis is going through the beams, and the beams are located on the (x, y)-plane.

In order to develop a model suitable for the purpose of this project, it must be proper defined how it must be used afterwards. Since the manipulator workspace is defined in $\mathbb{E}^2$ (reduced from $\mathbb{E}^3$ due to remark 1 on page 18), the motion of the flexible tool must also be evaluated in this two-dimensional Euclidean space only. With this planar case the oscillations of the tool can only occur around one axis. Therefore, only bending as shown by figure 4.1b is possible and will be the only effect to model for the purpose of this project.

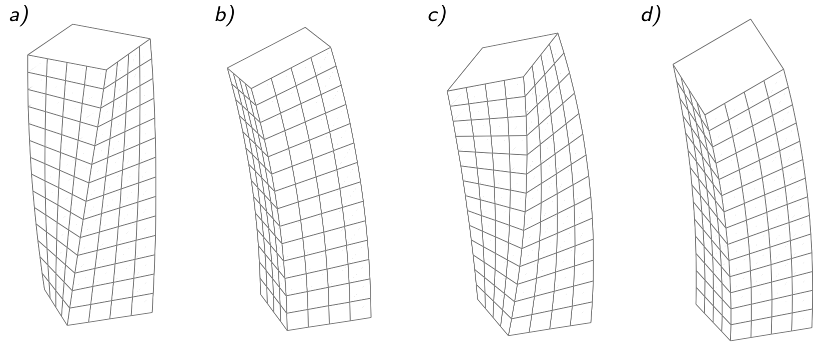


Figure 4.1: *Different types of beam flexibilities in three dimensions (twisting and bending)*

It is assumed, that the flexibility is constrained to only the first four modes of vibration (will be explained in details later in this chapter). This reduces the accuracy of the model but can be fully justified for the most practical cases. The frequency of operation will most likely, at least for this project, be limited to a low range bandwidth and therefore not able to excite higher modes of vibration. All assumptions from above are summarized in assumption 3.

> **Assumption 3:** *Due to the assumed planar manipulator structure (see remark 1 on page 18), bending of the tool is only possible around one axis. The oscillatory bending motion is modeled by the first four modes of vibration in order to limit the complexity of the model and only include modes that can be measured by sensors.* □

The aim of this project is to design a controller that is capable of damping flexible tool oscillations in order to make the end-effector track a given reference with low error. In cases where only the point-to-point tracking is important the controller must damp the oscillations before the end-effector reaches the desired position. Based on the sensor measurements from strain gauges the future behavior of the load can be predicted and thus also counteracted. The following section will introduce the concepts of modeling flexible system dynamics, which will form a basis for deriving a complex time dependent model of the flexible tool later on.

4.1 Basic flexibility model

Before introducing any math to model the flexible tool the notation is described. In respect with the tool frame, the X_τ axis is tangential to the tool center line, and the difference between these two axes defines the deflection denoted by $w(x)$ for the one-dimensional case [74]. Figure 4.2 shows an illustration of a flexible beam together with frame denotations.

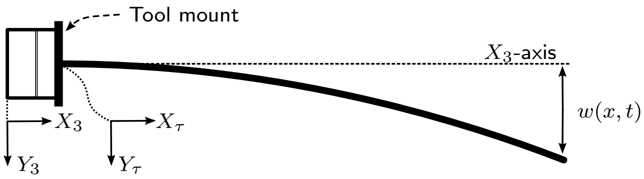


Figure 4.2: *Flexible beam problem with frame denotations*

The bending problem can be described on the basis of the *Timoshenko beam theory*, which introduces the following partial differential equation in time t and longitudinal position x on the beam [18]

$$EI\frac{\partial^4 w(x,t)}{\partial x^4(t)} + a\rho\frac{\partial^2 w(x,t)}{\partial t^2} - \left(I_\rho + \tfrac{a\rho EI}{k_s}\right)\frac{\partial^4 w(x,t)}{\partial x^2(t)\partial t^2} + \frac{a\rho I_\rho}{k_s}\frac{\partial^4 w(x,t)}{\partial t^4} = 0 \qquad (4.1)$$

with $w(x,t)$ describing the deflection of the beam in the two arguments. This equation is only describing the beam deflection in one direction, and another approach will have to be used when the manipulator is operating in $\mathbb{E}^3$. For the scope of this project, only the first 2 terms will be applied and thus also described (unused terms are not described or included in the nomenclature) [74]. The last terms are included to show, that the Timoshenko theory can include more aspects in the bending of a beam, making the model more accurate. A similar beam theory could have been applied in form of the *Euler-Bernoulli beam theory*, which would have been similar to the Timoshenko model in the simplified case [18]. The main difference between the two methods is their basic assumptions on how to model bending/twisting behavior of a beam.

Appendix M features the derivations of models for time independent beam bending phenomenas concerning simple beams. The main results will be repeated at the end of this section. This will form a basis of the theory provided in the remainder of this chapter. The last sections will introduce time and tool frame motion to the model. All models derived in this report respect the following assumption regarding direction of gravity.

Definition 1: *Scalar forces acting in a single point and distributed forces acting on a line are defined in the positive z-direction (opposing gravity direction in the initial manipulator configuration), and deflections will thus be positive when the beam is subjected to a positive force (in the assumed upwards direction)* □

A summary of the models from appendix M are given in table 4.1 for a beam with boundary conditions from a cantilever beam configuration, repeated from appendix M [108]

$w(0) = \dfrac{\mathrm{d}w(0)}{\mathrm{d}x} = 0$	*No deflection or non-tangential derivative at the point of beam attachment $x = 0$*
$\dfrac{\mathrm{d}^2 w(\ell)}{\mathrm{d}x^2} = \dfrac{\mathrm{d}^3 w(\ell)}{\mathrm{d}x^3} = 0$	*No bending moment or shear force at the endpoint of the beam $x = \ell$*

Beam subjected to self-weight $w_b(x) = \frac{m_b g}{24EI\ell}x^4 - \frac{m_b g}{6EI}x^3 + \frac{m_b g \ell}{4EI}x^2$
Beam subjected to tip load $w_l(x) = -\frac{m_l g}{6EI}x^3 + \frac{m_l g \ell}{2EI}x^2$
Beam subjected to both load types $w_c(x) = \frac{m_b g}{24EI\ell}x^4 - \left(\frac{g}{6EI}(m_l + m_b)\right)x^3 + \left(\frac{m_b g \ell}{4EI} + \frac{m_l g \ell}{2EI}\right)x^2$

Table 4.1: *Beam deflection when subjected to different load types (repeated from appendix M)*

Notations are given in appendix M as well as in the nomenclature. These solutions are solved by integrating the differential equation four times and then expressing all unknown constants. The sequel section introduces motion of the tool frame $\mathbb{F}_\tau$ and a tip mass to the simple beam. It will be clear, that the polynomial realizations from table 4.1 is replaced by a sum of trigonometric functions which are more suitable for solving the partial differential equation in (4.1).

4.2 Flexible tool model with non-stationary tool frame

The above theory included a stationary inertial frame causing only the gravity to give rise to a force in the deflection model. This section will include two more elements to the model: time dynamics and a moving tool frame. Whenever the tool frame is in motion, it will affect the beam and causing it to bend. Equation (4.1) will be applied in the simplified case described in the chapter introduction, yielding

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + a\rho\frac{\partial^2 w(x,t)}{\partial t^2} = f(x) \tag{4.2}$$

which includes the time parameter in the deflection function $w(x,t)$ and no torsional components. E is the Young's modulus [N/m^2] and I the second moment of inertia [m^4] and the values are given in appendix G. The product between the variables E and I expresses the *flexural rigidity* of the flexible tool [20]. Furthermore, the force is now determined from the motion of the tool frame alone (constant gravity term neglected) and corresponds directly to Newtons's second law of motion stating that $F = m\ddot{x}$, which in this particular case yields [74]

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + (m_l\delta(x-\ell) + a\rho)\left\{\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2} + \frac{\partial^2 w(x,t)}{\partial t^2}\right\} = 0 \tag{4.3}$$

with the force described per unit length instead. Two new terms have been included in the model and all four right terms are described below

$a\rho\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}$	*Force on beam due to acceleration of tool frame*
$m_l\delta(x-\ell)\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}$	*Force on load due to acceleration of tool frame*
$a\rho\frac{\mathrm{d}^2 w(x,t)}{\mathrm{d}t^2}$	*Force on beam due to acceleration of beam*
$m_l\delta(x-\ell)\frac{\mathrm{d}^2 w(x,t)}{\mathrm{d}t^2}$	*Force on load due to acceleration of beam*

A sketch of the problem is depicted in figure 4.3. Tool frame motion is possible in all directions in the (X_τ-Y_τ)-plane, but only components in the Z_0-direction will be considered as formulated in assumption 4. As seen from the $\mathbb{F}_\tau$ frame itself no motion is happening of the $\mathbb{F}_\tau$ origo. Therefore, as the gravity force is acting perpendicular to the inertial frame $\mathbb{F}_0$, the motion y must be given according to this.

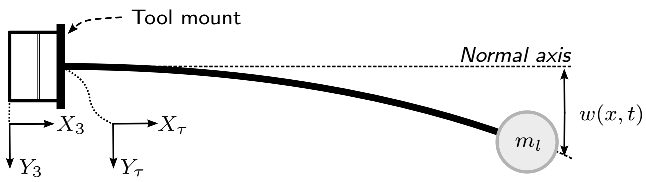


Figure 4.3: *Flexible beam problem with moving base and tip mass*

Some of the initial conditions are similar to the ones used in the models from the previous section [108] with time dependencies added as well. Because of the added tip mass m_l and the tool frame dynamics, the boundary conditions need no longer be zero as suggested by [74]. However, by maintaining the boundaries as they are for the regular cantilever beam, the frequency analysis of the beam will be able to indicate the natural beam frequencies as well as the change in frequency due to the load. The two new boundary conditions treat the vertical motion of the beam, which is assumed in steady state at time $t = 0$, $y(0) = 0$. Assumption 4 forms the basis of the model in equation 4.3. No other disturbances than $y(t)$ are affecting the beam, making the independent variable x independent of time as well.

> **Assumption 4:** *Only tool frame $\mathbb{F}_\tau$ motion as represented in the Z_0-direction (negative direction of gravity) is considered an excitation to the flexible tool model, because motion in the X_0-direction is assumed to have no impact on the deflection of the beam. This is supported by the flexible model only containing deflection components and not stretch components along the length of the beam.* □

As was the case in the modeling of the beam deflection due to the load in equation (M.5), the Dirac delta function is only activating the forces on the load, whenever $x = \ell$. Therefore, it is convenient to introduce a Laplace transformation once again to handle this function. The independent function to solve for is the deflection of the beam $w(x,t)$ when affected by tool frame motion $y(t)$.

Similar to a Fourier transformation of a signal, the motion of the beam elements can be described as an infinite sum of contributions for each mode shape and the corresponding dynamics. In order to enable valid calculations, a specific number of modal frequencies is considered to make up for the response within the entire frequency span. The deflection can thus be evaluated in the following way [74]

$$w(x,t) = \sum_{i=1}^{M} \phi_i(x) q_i(t) \tag{4.4}$$

with $\phi_i(x)q_i(t)$ expressing the shape of the i-th solution to equation (4.3). $q_i(t)$ is the so-called *modal coordinate* for the i-th mode of vibration and describes the motion of the specific mode over time [102]. $\phi_i(x)$ describes on the other hand the *mode shape* of the i-th resonance. M denotes the number of active modes in the approximation of a solution. The function $q_i(x)$ from equation (4.4) is assumed to be describing a second-order damped system. Before proceeding with the model derivation, an important note is stated.

The notation $q(t)$ will be referred to as both the modal coordinate function and the strain. This is because the strain is determined as a linearly scaled version of the modal coordinate. The double meaning of the notation will, however, not interfere with the understanding of the material, because the variable remains undefined (no practical values inserted) until the end of this chapter.

The modal coordinate is determined from [21]

$$\begin{aligned} q_i(t) : \frac{\mathrm{d}q_i^2(t)}{\mathrm{d}t^2} + 2\zeta_i\omega_i \frac{\mathrm{d}q_i(t)}{\mathrm{d}t} + \omega_i^2 q_i(t) = 0 \\ \Rightarrow q_i(t) = \mathrm{e}^{\lambda_r t}(d_1 \cos\lambda_1 t + d_2 \sin\lambda_2 t) \end{aligned} \tag{4.5}$$

where $\lambda_r = \Re\{\Lambda_k\}$ (either k can be used due to complex conjugated solution pair), $\lambda_k = \Im\{\Lambda_k\}$ given the roots to the characterstic equation $\Lambda_k \in \mathbb{C}$ and $d_1, d_2 \in \mathbb{R}$ are arbitrary, but given from initial conditions. The general solution is given from [21] and requires the discriminant of the characteristic equation $\Delta < 0$, which is obtained when $\zeta^2 < 1$ given that $\Delta = 4\omega^2(\zeta^2 - 1)$. The roots are determined from the characteristic equation factorized to $(s - \Lambda_1)(s - \Lambda_2) = 0$ in the Laplace domain.

Since $\zeta \in [0, 1)$ [32] it is assumed that the beam cannot behave in pure exponentially decaying manner, but will oscillate around the zero-reference. By [102] it is described, how the damping theory of a flexible beam is not understood in complete details, and thus damping must be added to ODE solutions only to obtain responses close to actual behavior. The modal damping factor ζ has been estimated by experiment as described in appendix A. The test shows that all damping factors for $M < 2$ are respecting the constraint $\zeta_i \leq 0,3122$, which will decrease the eigenfrequency ω_i by a maximum of 5 %. The origin of the constraint is provided in appendix A. A differential equation on the form

$$q_i(t) : \frac{\mathrm{d}q_i^2(t)}{\mathrm{d}t^2} + \omega_i^2 q_i(t) = 0 \tag{4.6}$$

will therefore be assumed a solution to the PDE [74]. Damping will be added later on. With this defined the assumed deflection is inserted into the model from equation 4.3, yielding

$$EI \sum_{i=1}^{\infty} q_i(t) \frac{\partial^4 \phi_i(x)}{\partial x^4} + (m_l \delta(x-\ell) + a\rho) \left\{ \frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2} + \sum_{i=1}^{\infty} \phi_i(x) \frac{\partial^2 q_i(t)}{\partial t^2} \right\} = 0 \quad (4.7)$$

It can be shown, that the modal shape functions (eigenfunctions) are orthogonal. An important assumption for the Hilbert space approach to apply for this type of problem is orthogonality of the eigenfunctions $\phi_i(x)$. This enables the infinite sum of mode shape functions and modal coordinate functions to be a solution, since the individual solutions can be summed up independent of each other due to the orthogonality property [6]. To prove the orthogonality between two eigenfunctions, the Hilbert space definition must be applied

$$\{\phi_i | H(\phi_i) = \lambda_i \phi_i\}, \quad H(\phi) \triangleq \frac{\partial^4 \phi(x)}{\partial x^4} \quad (4.8)$$

which is expressing the relation between eigenfunctions and eigenvalues of the *eigenvalue problem* stated as

$$\frac{\partial^4 \phi_i(x)}{\partial x^4} = \lambda_i \phi_i \quad (4.9)$$

The procedure of showing orthogonality is given in [11] with boundary conditions from a cantilever beam setup. It requires considering two different eigenvalue problems

$$\frac{\partial^4 \phi_i(x)}{\partial x^4} \phi_i = \lambda_i \phi_i \quad \text{and} \quad \frac{\partial^4 \phi_j(x)}{\partial x^4} \phi_j = \lambda_j \phi_j, \qquad i \neq j$$

and by a mathematical proof, the equations are true when

$$\int_0^{\ell} \phi_i(x) \phi_j(x) \, \mathrm{d}x = 0 \quad (4.10)$$

which is the definition of orthogonality between two real valued functions determined from the definition of an inner product between functions in a closed interval $[x_1, x_2]$ [83]

$$\langle \phi_a(x), \phi_b(x) \rangle = \int_{x_1}^{x_2} \phi_a(x) \phi_b^*(x) \, \mathrm{d}x, \quad \phi_a(x), \phi_b(x) : [x_1, x_2] \mapsto \mathbb{C}$$

$$\boxed{\therefore \langle \phi_a(x), \phi_b(x) \rangle = \int_{x_1}^{x_2} \phi_a(x) \phi_b(x) \, \mathrm{d}x, \quad \phi_a(x), \phi_b(x) : [x_1, x_2] \mapsto \mathbb{R}} \quad (4.11)$$

The last property is possible because real valued functions are just a special case of complex valued functions and thus $\phi_b(x) = \phi_b^*(x)$ when $\phi_b(x) \mapsto \mathbb{R}$. The eigenfunctions are therefore linearly independent, and the sum in (4.4) is applicable [11]. Based on the orthogonality principle the following property is therefore applicable

$$\langle \phi_a, \phi_b \rangle = 0, \quad a \neq b$$

This property can be added as a constraint to the PDE by the following inner product, which is possible when the PDE is on homogeneous form

$$\left\langle \phi_k,\ \frac{EI}{a\rho}\sum_{i=1}^{\infty} q_i(t)\lambda_i^4\phi_i(x) + \left(\frac{m_l}{a\rho}\delta(x-\ell)+1\right)\left\{\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2} + \sum_{i=1}^{\infty}\phi_i(x)\frac{\partial^2 q_i(t)}{\partial t^2}\right\}\right\rangle = 0$$

for a given eigenvalue k. A factor of $(a\rho)^{-1}$ has been multiplied to normalize the dynamics. This property will be exploited later. The differential operator is linear and can be defined as a linear operator $H(\phi)$ as a function of the modal shape function $\phi(x)$. This operator will be defined on the specific Hilbert space described by the inner products, and the operator has the following spectrum

$$\{\phi_i | H(\phi_i) = \lambda_i^4\phi_i\}, \quad H(\phi) \triangleq \frac{\partial^4\phi(x)}{\partial x^4} \tag{4.12}$$

when denoted for every modal shape function i satisfying the PDE. λ_i^4 denotes the eigenvalue for the i-th eigenfunction, and the fourth order is used to simplify the derivation of the dynamic solutions. Each of the terms in the inner product can be explicitly expressed within the closed interval $[0, \ell]$

$$\begin{aligned}
&\left\langle \phi_k,\ \frac{EI}{a\rho}\sum_{i=1}^{\infty} q_i(t)\lambda_i^4\phi_i(x)\right\rangle = \omega_i^2 q_k(t)\\
&\left\langle \phi_k,\ \frac{m_l}{a\rho}\delta(x-\ell)\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}\right\rangle = \frac{m_l}{a\rho}\phi_k(\ell)\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}\\
&\left\langle \phi_k,\ \frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}\right\rangle = \frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2}\int_0^\ell \phi_k(x)\,\mathrm{d}x\\
&\left\langle \phi_k,\ \sum_{i=1}^{\infty}\phi_i(x)\frac{\partial^2 q_i(t)}{\partial t^2}\right\rangle = \frac{\partial^2 q_k(t)}{\partial t^2}\\
&\left\langle \phi_k,\ \frac{m_l}{a\rho}\delta(x-\ell)\sum_{i=1}^{\infty}\phi_i(x)\frac{\partial^2 q_i(t)}{\partial t^2}\right\rangle = \frac{m_l}{a\rho}\sum_{i=1}^{\infty}\frac{\partial^2 q_i(t)}{\partial t^2}\int_0^\ell \phi_k(x)\phi_i(x)\delta(x-\ell)\,\mathrm{d}x\\
&\qquad\qquad = \frac{m_l}{a\rho}\sum_{i=1}^{\infty}\frac{\partial^2 q_i(t)}{\partial t^2}\phi_k(\ell)\phi_i(\ell)
\end{aligned} \tag{4.13}$$

all determined using the definition of an inner product between functions in a closed interval $[x_1, x_2]$ from (4.11). Further, all mode shapes are normalized and thus [78]

$$\int_0^\ell \phi_k(x)\phi_i(x)\,\mathrm{d}x = 1, \quad i = k \tag{4.14}$$

The inner products however, can be simplified, if the mass of the load is moved to the boundary conditions such that [74]

$$\left.\frac{\mathrm{d}^3 w(x)}{\mathrm{d}x^3}\right|_{x=\ell} = \frac{m_l\ell}{EI}\left\{\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2} + \left.\frac{\partial^2 w(x,t)}{\partial t^2}\right|_{x=\ell}\right\} \tag{4.15}$$

Removing the inner products including m_l and inserting all terms back into (4.12) yields the differential equation in $q(t)$

$$\frac{\partial^2 q_k(t)}{\partial t^2} + \omega_i^2 q_k(t) = -\frac{\mathrm{d}^2 y(t)}{\mathrm{d}t^2} \int_0^\ell \phi_k(x)\,\mathrm{d}x \tag{4.16}$$

This new structure is a regular second order ordinary differential equation, because the eigenfunctions are constant for each k (in this case $k = 1$). By knowing the eigenfunctions, the modal coordinate functions $q_k(t)$ [m], which are solutions to the ODE, can be expressed. This will be treated next for the cases with and without tip mass. Both cases are solved for comparison.

4.2.1 Eigenfunctions without tip mass

In order to express the eigenfunctions, the *eigenvalue problem* will have to be solved, which in this case is given from the PDE with no tip mass provided in (4.2) (without external force term) and the modal decomposition from (4.4) for one mode i only (without loss of generality) [58]

$$\frac{1}{\phi_i(x)} \frac{\partial^4 \phi_i(x)}{\partial x^4} = -\frac{a\rho}{EI} \frac{1}{q_i(t)} \frac{\partial^2 q_i(t)}{\partial t^2} = \lambda_i^4 \tag{4.17}$$

with the constant term λ_i^4 being a definition, since the two ODE's are equal. The equations can be written explicitly as eigenvalue problems given by

$$\frac{\partial^4 \phi_i(x)}{\partial x^4} = \lambda_i^4 \phi_i \quad \text{and} \quad \frac{\partial^2 q_i(t)}{\partial t^2} + \omega_i^2 q_i = 0 \tag{4.18}$$

with the last equation being a consequence of no damping in the beam [94] (added explicitly later). This relates the constants λ_i^4 and ω_i^2 by

$$\lambda_i = \sqrt[4]{\tfrac{a\rho\omega_i^2}{EI}} \tag{4.19}$$

The fourth order on the eigenvalue λ_i^4 is introduced to achieve simple roots to the polynomial $s^4 - \lambda_i^4$, and the first eigenvalue problem of (4.18) is solved in appendix J. A *frequency equation* on the form

$$\boxed{\cos \lambda_i \ell \cosh \lambda_i \ell = -1} \tag{4.20}$$

was found as the solution to the eigenvalue problem. It is then possible to determine the eigenvalues λ_i from equation (4.20). The equation is transparent, and thus it cannot be solved analytically. A table of the first $M = 4$ eigenvalues is therefore shown in table 4.2.

	$\lambda_i \ell$	λ_i [m^{-1}]	ω_i [rad/s]	f_i [Hz]
Mode 1	1,8751	4,7232	160,8404	25,5986
Mode 2	4,6941	11,8239	1007,9781	160,4247
Mode 3	7,8548	19,7854	2822,3877	449,1970
Mode 4	10,9955	27,6965	5530,6525	880,2307

Table 4.2: *First four eigenvalues to* (4.20) *on* Ξ*-system without tip mass*

Using the numerically derived eigenvalues, the unknown boundary conditions $\phi_i^{(2)}(0)$ and $\phi_i^{(3)}(0)$ can then be determined using the procedure of [78], which involves equation (J.16) to express a relation between the unknowns W_2 and W_4

$$W_4 = -W_2 \frac{\cosh \lambda_i \ell + \cos \lambda_i \ell}{\sinh \lambda_i \ell + \sin \lambda_i \ell} \tag{4.21}$$

which can be substituted into the remainder of the general solution given in (J.15) yielding

$$\phi_i(x) = W_2 \left\{ (\cosh \lambda_i x - \cos \lambda_i x) - \frac{\cosh \lambda_i \ell + \cos \lambda_i \ell}{\sinh \lambda_i \ell + \sin \lambda_i \ell} (\sinh \lambda_i x - \sin \lambda_i x) \right\} \tag{4.22}$$

with W_2 given as

$$W_2 = \frac{\phi_i^{(2)}(0)}{2\lambda_i} \triangleq 1 \tag{4.23}$$

being a scale factor of the mode shape function, which is defined as unity [18]. Scaling of the eigenfunctions $q_i(t)$ will provide the necessary scaling of the PDE solution. The modes can be illustrated using (4.22) with an arbitrary beam length, see figure 4.4

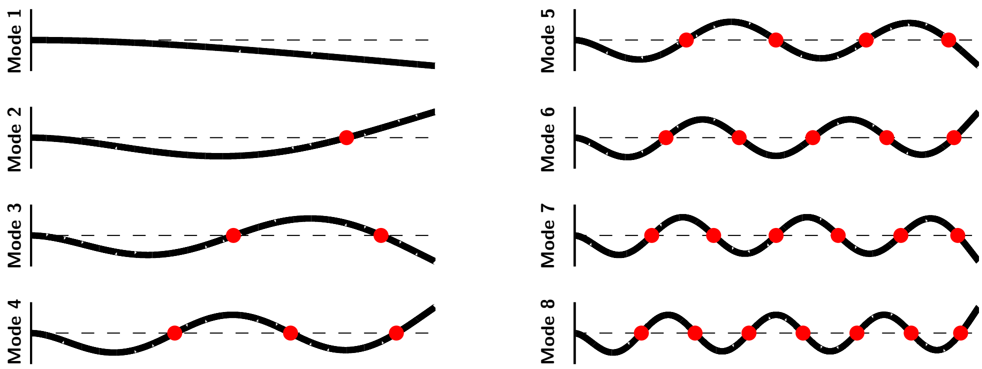


Figure 4.4: *Illustration of first 8 mode shapes (inspired by fig. 11.6 in [78])*

The dots indicating intersections between the normal axis and the mode shape function are also dead spots of zero dynamics. No strain gauges are therefore to be placed at these locations.

4.2.2 Eigenfunctions with tip mass

The mode solution derived above is not including the mass of the load. In order to include this, the frequency equation will have to be a function of the mass ratio between the beam and the load. This requires the new boundary condition from (4.15) without the external force term $\ddot{y}(t)$. The condition can be reformulated in the following steps starting with the basic proposed condition without the $\ddot{y}$ term [18]

$$
\begin{aligned}
\left.\frac{\mathrm{d}^3 w(x)}{\mathrm{d}x^3}\right|_{x=\ell} &= \frac{m_l \ell}{EI} \left.\frac{\partial^2 w(x,t)}{\partial t^2}\right|_{x=\ell} \quad \Downarrow \\
q(t)\left.\frac{\mathrm{d}^3 \phi(x)}{\mathrm{d}x^3}\right|_{x=\ell} &= \frac{m_l \ell \phi(\ell)}{EI} \frac{\partial^2 q(t)}{\partial t^2} = -\frac{m_l \ell \omega_i^2}{EI} \phi(\ell) q(t)
\end{aligned}
$$

using the eigenvalue problem from (4.18). Removing the time dependency in $q(t)$ results in a new boundary condition on the form as proposed by [18]

$$
\left.\frac{\partial^3 \phi_i(x)}{\partial x^3}\right|_{x=\ell} + \lambda_i^4 \frac{m_l \ell}{a\rho} \phi_i(\ell) = 0 \tag{4.24}
$$

which will substitute the previous condition given by

$$
\frac{\mathrm{d}^3 w(\ell)}{\mathrm{d}x^3} = 0 \tag{4.25}
$$

Substituting the known terms (J.17) and (J.15) from the eigenvalue derivation in appendix J into (4.24) yields together with (J.16) two equations

$$
\begin{aligned}
& W_2(\cosh \lambda_i \ell + \cos \lambda_i \ell) + W_4(\sinh \lambda_i \ell + \sin \lambda_i \ell) = 0 \\
& \lambda_i^3 W_4(\cosh \lambda_i \ell + \cos \lambda_i \ell) + \lambda_i^3 W_2(\sinh \lambda_i \ell - \sin \lambda_i \ell) + \cdots \\
& \qquad \cdots + \lambda_i^4 \frac{m_l \ell}{a\rho} \left\{ W_2(\cosh \lambda_i \ell - \cos \lambda_i \ell) + W_4(\sinh \lambda_i \ell - \sin \lambda_i \ell) \right\} = 0
\end{aligned}
$$

A solution can be derived using a similar approach as for the case without tip mass. However, the two equations are linear in the constants W_2 and W_4 and can be solved using a determinant on the form

$$
\begin{vmatrix}
\cosh \lambda_i \ell + \cos \lambda_i \ell & \sinh \lambda_i \ell + \sin \lambda_i \ell \\
\begin{matrix} \sinh \lambda_i \ell - \sin \lambda_i \ell + \cdots \\ \cdots + \lambda_i \frac{m_l \ell}{a\rho} \cosh \lambda_i \ell - \lambda_i \frac{m_l \ell}{a\rho} \cos \lambda_i \ell \end{matrix} & \begin{matrix} \cosh \lambda_i \ell + \cos \lambda_i \ell + \cdots \\ \cdots + \lambda_i \frac{m_l \ell}{a\rho} \sinh \lambda_i \ell - \lambda_i \frac{m_l \ell}{a\rho} \sin \lambda_i \ell \end{matrix}
\end{vmatrix} = 0
$$

This yields a new frequency equation [18]

$$
\boxed{\underbrace{1 + \cos \lambda_i \ell \cosh \lambda_i \ell}_{\text{Without tip mass}} + \underbrace{\frac{m_l \lambda_i}{a\rho} (\sinh \lambda_i \ell \cos \lambda_i \ell - \cosh \lambda_i \ell \sin \lambda_i \ell)}_{\text{New term due to tip mass}} = 0} \tag{4.26}
$$

The equation is composed by a term for the unloaded case and a term relating the mass of tip load and beam. As suggested by both [18] and [74] the relation is commonly concatenated into the mass ratio

$$
r \triangleq \frac{m_l}{a\rho\ell} = \frac{m_l}{m_b} \tag{4.27}
$$

which transforms the frequency equation into the final form

$$1 + \cos\lambda_i\ell\cosh\lambda_i\ell + r\lambda_i\ell(\sinh\lambda_i\ell\cos\lambda_i\ell - \cosh\lambda_i\ell\sin\lambda_i\ell) = 0 \tag{4.28}$$

Tables 4.3 and 4.4 shows the first 4 eigenvalues and eigenfrequencies ($M = 4$) for the Ξ-system with a tip mass of $m_l = 0,351$ kg and $m_l = 0,522$ kg, respectively.

	$\lambda_i\ell$	λ_i [m^{-1}]	ω_i [rad/s]	f_i [Hz]
Mode 1	0,7994	2,0982	31,7399	5,0516
Mode 2	3,9432	10,3496	772,2805	122,9123
Mode 3	7,0784	18,5785	2488,5575	396,0662
Mode 4	10,2170	26,8163	5184,7075	825,1718

Table 4.3: *First four eigenvalues to* (4.28) *on* Ξ*-system with tip mass* $m_{l1} = 0{,}351$ *kg (*$r = 7{,}1$*)*

	$\lambda_i\ell$	λ_i [m^{-1}]	ω_i [rad/s]	f_i [Hz]
Mode 1	0,7259	1,9052	26,1716	4,1653
Mode 2	3,9478	10,3354	770,1668	122,5758
Mode 3	7,0752	18,5701	2486,3079	395,7082
Mode 4	10,2148	26,8105	5182,4749s	824,8165

Table 4.4: *First four eigenvalues to* (4.28) *on* Ξ*-system with tip mass* $m_{l2} = 0{,}522$ *kg (*$r = 10{,}6$*)*

The values from table 4.3 will be used when solving the PDE, see remark 2 on page 42. Notice how the first eigenfrequencies are decreasing due to the load mass. The first mode is reduced from 25,5986 Hz to 4,1653 Hz. Next step is to determine the modal coordinate function $q_i(t)$ describing the time dynamics.

> **Remark 2** (tip mass)**:** *From practical considerations only the small tip mass of* $m_l = 0{,}351$ *kg will be used in the remainder of the report. The flexible beam cannot withstand the load from the larger tip mass. Furthermore, only a single mode of vibration will be used for the final model, because the second mode of vibration was found insignificant through the experiment described in appendix A. Assumption 3 is therefore omitted and* $M = 1$ *is used instead.*□

Modal coordinate functions $q_i(t)$

The partial differential equation describing the flexible behavior of a simple beam without tip load was given from (4.2) as

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + a\rho\frac{\partial^2 w(x,t)}{\partial t^2} = f(x)$$

which resulted in the frequency equation from (4.20)

$$\cos \lambda_i \ell \cosh \lambda_i = -1$$

Using this information to determine the set $\{\lambda_i\}_{i=1}^{\infty}$, a solution can be given on the form from (4.4)

$$w(x,t) = \sum_{i=1}^{\infty} \phi_i(x) q_i(t)$$

Using the method of inner products from (4.13), a time dynamic equation can be determined in a similar way, yielding the form

$$\begin{aligned}\left\langle \phi_k,\ \frac{EI}{a\rho} \sum_{i=1}^{\infty} q_i(t)\lambda_i^4 \phi_i(x) + \sum_{i=1}^{\infty} \phi_i(x) \frac{\partial^2 q_i(t)}{\partial t^2} - f(t) \right\rangle = 0 \ \Downarrow \\ \frac{\partial^2 q_k(t)}{\partial t^2} = -\omega_k^2 q_k(t) + f(t) \int_0^{\ell} \phi_k(x)\, \mathrm{d}x\end{aligned}$$

by assuming that $f(t)$ is only dependent on time and with the knowledge from appendix A, a damping ratio can be added

$$\frac{\partial^2 q_k(t)}{\partial t^2} = -\omega_k^2 q_k(t) - 2\zeta_k \omega_k \frac{\partial q_k(t)}{\partial t} \tag{4.29}$$

which completes the time dynamics derivation. Given a determinable set of functions $\{\phi_k\}_{i=1}^{\infty}$ and $\{q_k\}_{i=1}^{\infty}$ the deflection $w(x,t)$ can be determined for any spatial location at any time instance. A state space representation can be given for M of the ODE's by [9]

$$\begin{aligned}\begin{bmatrix} \dot{\bar{q}} \\ \ddot{\bar{q}} \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_M & \mathbf{I}_M \\ -\boldsymbol{\omega}^2 & -2\boldsymbol{\zeta}\boldsymbol{\omega} \end{bmatrix} \begin{bmatrix} \bar{q} \\ \dot{\bar{q}} \end{bmatrix} + \begin{bmatrix} \bar{0} \\ \int_0^{\ell} \bar{\phi}(x)\, \mathrm{d}x \end{bmatrix} f(t) \\ w(\ell,t) &= \begin{bmatrix} \bar{\phi}(x)|_{x=\ell}^{\mathsf{T}} & \bar{0}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \bar{q} \\ \dot{\bar{q}} \end{bmatrix}\end{aligned}$$

given

$$\bar{q} = \begin{bmatrix} q_1 & \cdots & q_M \end{bmatrix}^{\mathsf{T}}, \quad \bar{\phi} = \begin{bmatrix} \phi_1 & \cdots & \phi_M \end{bmatrix}^{\mathsf{T}}, \quad \boldsymbol{\omega} = \mathrm{diag}\{\omega_i\}_{i=1}^{M}, \quad \boldsymbol{\zeta} = \mathrm{diag}\{\zeta_i\}_{i=1}^{M}$$

This structure is possible because the eigenmodes are orthogonal to each other, resulting in M decoupled ODE's describing the dynamics. The forcing function is time dependent and constant in spatial location. Other forcing functions can be used, which will, however, increase the complexity of the state space description.

The partial differential equation from (4.3) includes the tip mass m_l. Since only the first eigenmode is needed by the control system (see remark 2 on page 42), a solution will be approximated using the general solution from (4.29) (time dynamics) and (4.26) (eigenfrequencies). A state space representation will therefore be on the form

$$\begin{aligned}
\begin{bmatrix} \dot{q}_1 \\ \ddot{q}_1 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_1^2 & -2\zeta_1\omega_1 \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \int_0^\ell \phi_1(x)\,\mathrm{d}x \end{bmatrix} f(t) \\
w(\ell,t) &= \begin{bmatrix} \phi_1(\ell) & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix}
\end{aligned} \tag{4.30}$$

This model is both controllable and observable, because the rank of the controllability matrixand the observability matrixequals the rank of the system matrix. If the partial differential equation was to be solved for M modes, the dynamics for each mode will depend on $M-1$ other modes. A total of M coupled differential differential equations will then describe the dynamics of the system. The problem is not only difficult to solve explicitly for M modes, but does also requires the exact relationship between modal shapes $\phi_k(x)$ as well as the shape of the excitation signal. A finite element solution would be preferable in this case as introduced in appendix K.

Another approach can be to use the frequency equation from (4.28) to estimate the eigenfrequencies of the loaded beam, and then approximate the dynamics using M decoupled ODE's on the form from (4.29). Each mode is then described by a second order system with a tip mass dependent eigenfrequency. A model structure including an arbitrary number of flexible modes is a generalization of the model in (4.30) can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{[M\times M]} & \mathbf{I}^{[M\times M]} \\ -\boldsymbol{\omega}_l^2 & -2\boldsymbol{\zeta}\boldsymbol{\omega}_l \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{Q} \end{bmatrix} u$$

with $q, \dot{q} \in \mathbb{R}^M$ as well as $\boldsymbol{\omega}_l = \mathrm{diag}(\omega_{l,1}, \ldots, \omega_{l,M}) \in \mathbb{R}^{M\times M}$ and $\boldsymbol{\zeta} = \mathrm{diag}(\zeta_1, \ldots, \zeta_M) \in \mathbb{R}^{M\times M}$ containing eigenfrequency and modal damping for each mode of the tool with a tip load. $\mathbf{Q}$ is a constant matrix used to scale the excitation signal appropriately. The matrix includes the scaling from input signal to joint acceleration given from (3.10) as $(NF)^{-1}$.

This is, however, not a concern in this case, where only one mode is used. By Laplace transforming the model from (4.30) the following transfer function is describing the mode dynamics of the first eigenmode

$$\boxed{q_1(s) = \frac{\phi_1(\ell)\int_0^\ell \phi_1(x)\,\mathrm{d}x}{s^2 + 2\zeta_1\omega_1 s + \omega_1^2}} \tag{4.31}$$

Using the estimated model parameters $\omega_1 = 26{,}39$ rad/s and $\zeta_1 = 0,0023$ from appendix A and a unit input gain, the following transfer function can be expressed

$$q_1(s) = \frac{1}{s^2 + 0{,}1214s + 696{,}3993} \tag{4.32}$$

Figure 4.5 shows the bode plot of the transfer function with the resonance peak clearly showing at 26,39 rad/s

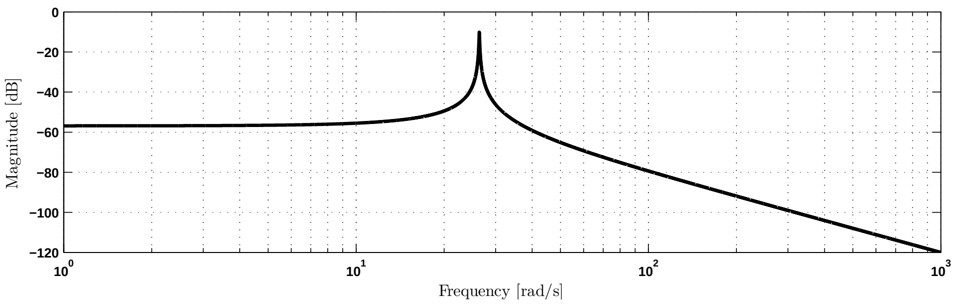
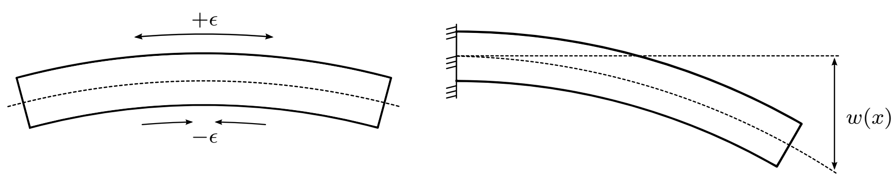


Figure 4.5: *Bode plot of transfer function in* (4.32)

Expressions of both the mode shape $\phi_1(x)$ and the time dynamics $q_1(t)$ for the first mode of vibration have been derived. Using a simple version of the general Timoshenko beam equation, an eigenvalue problem was solved to express the eigenfrequencies of the beam with and without payload. Since the eigenfrequency of the first mode of vibration is the only eigenfrequency that can be excited by the manipulator, see appendix E, there is only need for one mode in the final model. Before ending this chapter, figure 4.6 summarizes the two concepts of strain and deflection.



(a) *Strain* **(b)** *Deflection*

Figure 4.6: *Graphical representation of strain and deflection*

Both concepts will be used in the sequel, and table 4.5 lists how they are determined (single and M-mode case) and their relations [73].

Strain/deflection	Single case	M-mode case
Strain	$\epsilon_1(x,t) = -\frac{h}{2}\frac{\mathrm{d}^2\phi_1(x)}{\mathrm{d}x^2}q_1(t)$	$\epsilon(x,t) = -\frac{h}{2}\sum_{i=1}^{M}\frac{\mathrm{d}^2\phi_i(x)}{\mathrm{d}x^2}q_i(t)$
Deflection	$w_1(x,t) = \phi_1(x)q_1(t)$	$w(x,t) = \sum_{i=1}^{M}\phi_i(x)q_i(t)$

Table 4.5: *Expressions for strain and deflection*

The negative sign of the strain equations is used because the strain is determined for the upper strain gauge pair. Removing the sign will provide the strain of the lower strain gauge pair instead. As shown by the equations the only difference between strain and deflection is a mode shape dependent scale factor. Therefore, the above modeling of the flexible tool in the variable q makes it possible to determine either strain or deflection by scaling the state trajectory differently. The manipulator kinematics/dynamics and the flexible dynamics will be linked together as well as summarized in the sequel chapter. This will result in a model suitable for the control of the manipulator while damping the unwanted oscillations in the tool.

Chapter 5

Model summary

Two separate models have been derived in the previous two chapters: one describing the dynamics of the manipulator itself and one describing the behavior of the flexible tool to random tool frame excitations. Before initiating the controller design in the following part, the two models must be concatenated into one single model. The controller must be able to damp the oscillations of the tool as well as position/orientate the end-effector. The control algorithm must be supplied with the following elements

Flexible tool strain
A frame $\mathbb{F}_e$ is dedicated to the end-effector to describe orientation and position. In order to control the position/orientation of this frame in reference with the inertial frame, the strain $\epsilon_1(\ell^{\mathsf{str}})$ is needed as a model output. The deflection $w_1(\ell, t)$ and corresponding space derivative $w_1'(\ell)$ at the end of the beam can be derived from the strain and used to determine the position/orientation of $\mathbb{F}_e$. A general model using M modes will need a strain on the form $\epsilon(\ell^{\mathsf{str}})$ in stead of just $\epsilon_1(\ell^{\mathsf{str}})$.

Manipulator joint angle
While damping the end-effector oscillations, the manipulator joints must follow a reference trajectory based on given requirements. It is therefore necessary to output the joint angle θ_3 and the joint velocity $\dot{\theta}_3$ of the active joint. A general model will require the joint space θ and the joint velocity space $\dot{\theta}$ to be given. Higher derivatives must be estimated from $\dot{\theta}$ if needed to further constrain the manipulator motion.

Based on the listed requirements to the model structure for controller design, the general structure can be formed. The tool dynamics described by a single eigenmode was expressed in (4.30) and is repeated below

$$\begin{aligned}\begin{bmatrix}\dot{q}_1\\ \ddot{q}_1\end{bmatrix} &= \begin{bmatrix}0 & 1\\ -\omega_1^2 & -2\zeta_1\omega_1\end{bmatrix}\begin{bmatrix}q_1\\ \dot{q}_1\end{bmatrix} + \begin{bmatrix}0\\ (NF)^{-1}\int_0^\ell \phi_1(x)\,\mathrm{d}x\end{bmatrix}u(t)\\ w(\ell,t) &= \begin{bmatrix}\phi_1(\ell) & 0\end{bmatrix}\begin{bmatrix}q_1\\ \dot{q}_1\end{bmatrix}\end{aligned} \tag{5.1}$$

with the scaling factor $(NF)^{-1}$ from (3.10) used to convert between applied input signal $u(t)$ and joint acceleration $\ddot{\theta}_3$. A reduced manipulator model was derived in (3.10) and was given by

$$\begin{aligned}\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} &= \begin{bmatrix}\mathbf{0} & \boldsymbol{I}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ (\mathbf{NF})^{-1}\end{bmatrix}U\\ \begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} &= \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix}\end{aligned} \tag{5.2}$$

Combining model (5.1) and (5.2) yields the following form when using only θ_3 and $\dot{\theta}_3$ yields the following structure

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}q_1\\ \dot{q}_1\\ \theta_3\\ \dot{\theta}_3\end{bmatrix} = \begin{bmatrix}\dot{q}_1\\ -2\zeta_1\omega_1\dot{q}_1 - \omega_1^2 q_1 + \ell_3(NF)^{-1}\left(\int_0^\ell \phi_1(x)\,\mathrm{d}x\right)u(t)\\ \dot{\theta}_3\\ (NF)^{-1}u(t)\end{bmatrix} \tag{5.3}$$

and substituting U with $u(t)$ in the single case. A simplification is by defining the parameters

$$K_1 = \ell_3(NF)^{-1}\left(\int_0^\ell \phi_1(x)\,\mathrm{d}x\right) \quad \text{and} \quad K_2 = (NF)^{-1} \tag{5.4}$$

The model consists of two decoupled models controlled by the same input signal. A single controller must therefore be designed, which is capable of damping the tool oscillations, while keeping the third manipulator link in a fixed position. This issue is addressed in the controller design procedure in chapter 9. The assumption that only the third axis is used to damp tool oscillations is summarized in assumption 5 below.

Assumption 5: *The tool oscillations are first to be counteracted, when the end-effector is close to its destination. It is therefore assumed, that the third and last axis of the manipulator is sufficient to provide that control. Further, it is the axis with the largest acceleration and closest to the end-effector, which is desirable in this type of control situation. Including more links for the task complicates matters unnecessarily in this case.* □

Furthermore, the nonlinear friction term has been omitted, which is described in assumption 6.

Assumption 6: *The nonlinear friction term derived in appendix I.4 will not be used in the complete model. Even though the friction model has a level of complexity suitable for fitting, it can also be a limiting factor. The system identification process that must be applied to fit the model must be very accurate, and the parameter estimates cannot make large fluctuations. This will completely change the friction dynamics, and thus also the control. It is assumed, that the only friction is linear due to the high gearing and fast acceleration of the manipulator.* □

Since only the actuator input in controllable, the explicit form of υ (only third component due to assumption 4, defined as the scalar version υ) must be substituted from (3.13) given as

$$\upsilon = -\ell_1\ddot{\theta}_1 - \ell_2(\ddot{\theta}_1 + \ddot{\theta}_2) - \ell_3(\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3) \tag{5.5}$$

with only the $\ddot{\theta}_3$ term used due to assumption 5. An ℓ_3 has therefore been added to the input gain of the mode dynamics part, and it will be part of the constant K_1. In order to apply the model in (5.3) for control purposes it must have a number of selected outputs as described in the beginning of the chapter. The outputs must be strain $\epsilon_1(\ell^{\mathsf{str}})$ and manipulator joint angle θ_3. Because the joint angle θ_3 is a direct state of the model in (5.3) it does not need to be converted. The strain on the other hand is a scale of the state q_1 describing the modal coordinate of the first eigenmode. A state space representation of the final model can be expressed as [73]

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 q_1 & -2\zeta_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \\ 0 \\ K_2 \end{bmatrix} u(t) \\
\begin{bmatrix} \epsilon_1(\ell^{\mathsf{str}}) \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} &= \begin{bmatrix} \left.\frac{\mathrm{d}^2\phi_1(x)}{\mathrm{d}x^2}\right|_{x=\ell^{\mathsf{str}}} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix}
\end{aligned} \tag{5.6}$$

with $\phi_1''(\ell^{\mathsf{str}})$ being a conversion factor between strain measurements in [V] and actual strain [-]. This model will be used throughout the controller design, and provides the necessary measures for damping tool oscillations using only the third axis of the manipulator. An important note is stated before continuing.

Because the model has been reduced to include only one manipulator link and a single mode of vibration, the notation of strain q_1 may be stated as q alone and the joint angle θ_3 stated as θ. Both notations may be used in the remainder of the report.

Before ending the modeling part the final transformation from strain to the position/orientation of the end-effector frame $\mathbf{F}_e$ and the spatial position of the TCP is derived in the next section.

5.1 Transformation from tool mount to end-effector

Besides the rigid transformation matrices ${}^{i}_{j}\mathbf{T}$ to describe the relationship between orientation and position of adjacent manipulator links, a transformation must be able to include the deflection of the beam. The transformation in question is ${}^{\tau}_{e}\mathbf{T}$, from tool mount to end-effector, and consists similarly of two elements: a rotation and a translation. The deflection $w(x,t)$ can be used to describe the change in orientation by spatial differentiation, which is derived from the strain $\epsilon(\ell^{\mathsf{str}})$ in the following way

$$w(x,t) = \frac{\epsilon(\ell^{\mathsf{str}}) + 2{,}403}{121{,}1} \tag{5.7}$$

with $v_o^{\mathsf{str}} \sim \epsilon(\ell^{\mathsf{str}})$. The relation has been derived from appendix F, which measured the relation between strain and tip deflection. Because the original relation from appendix F is measured in mm, the relation from above has been scaled by a factor of 0,001. First, two unit normal vectors must be defined in $\mathbb{R}^4$ as

$${}^{\tau}\hat{N}_e = {}^{\tau}_{e}\mathbf{T}\,{}^{\tau}\hat{N}_{\tau}, \quad {}^{\tau}\hat{N}_{\tau} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.8}$$

They are normals to the (Y_τ, Z_τ)-plane and (Y_e, Z_e)-plane, respectively, see figure 5.1. To make a connection between the beam equations, then $x \sim X_\tau$ and $w \sim Y_\tau$.

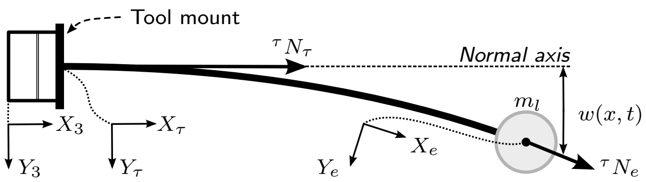


Figure 5.1: *Normal vectors used for flexible transformation* ${}^{\tau}_{e}\mathbf{T}$

Using a *direction cosine* matrix, the rotation between the vectors can be obtained. Since the vectors can be described in two dimensions alone, the rotation matrix is given by the elementary rotation matrix from (H.1) expressing a rotation around the Z_τ-axis

$${}^{\tau}_{e}\mathbf{R}\,(\varpi, t) = \begin{bmatrix} \cos\varpi(t) & -\sin\varpi(t) & 0 \\ \sin\varpi(t) & \cos\varpi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \varpi(t) = \operatorname{atan} \left.\frac{\partial w(x,t)}{\partial x}\right|_{x=\ell} \tag{5.9}$$

with ϖ denoting the angle of rotation around the Z_τ-axis at the x_ℓ location. The spatial derivative $w'(x,t)$ is determined from the PDE solution in (4.4). Due to the small-signal approximation described in assumption 5, the point of evaluation $x = \ell$ is approximated as the distance to the end point of the beam in the X_τ-direction. To complete the transformation the translation between the frames must be added. This yields the following homogeneous transformation matrix.

$$
{}^{\tau}_{e}\mathbf{T}\left(\varpi,t\right)=\begin{bmatrix}\cos\varpi(t) & -\sin\varpi(t) & 0 & \ell \\ \sin\varpi(t) & \cos\varpi(t) & 0 & w(\ell,t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1\end{bmatrix} \tag{5.10}
$$

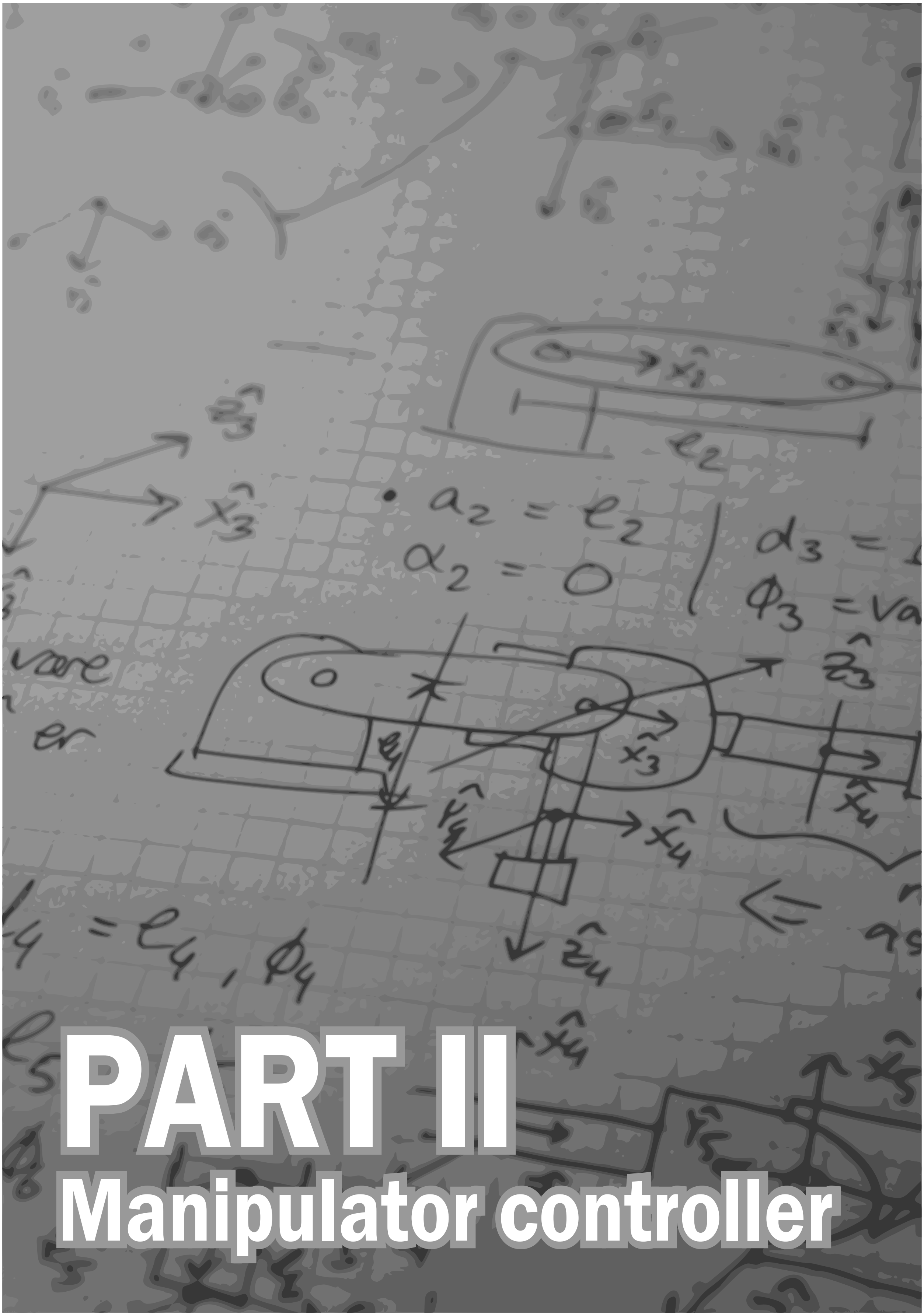
Time dependency of the mapping has been added to include modal coordinate function $q_1(t)$ influence. Based on the transformation from the production cell frame $\mathbb{F}_p$ to the tool mount frame $\mathbb{F}_\tau$ given from equation (3.2), a transformation from base frame $\mathbb{F}_0$ to the end-effector frame $\mathbb{F}_e$ can be described as

$$
{}^{0}_{e}\mathbf{T}\left(\theta,\varpi,t\right)={}^{0}_{n}\mathbf{T}\left(\theta\right){}^{n}_{\tau}\mathbf{T}\,{}^{\tau}_{e}\mathbf{T}\left(\varpi,t\right) \tag{5.11}
$$

This transform is contributing with most of the dynamics model, because it features both manipulator kinematics/dynamics as well as the dynamics of the flexible tool. This transformation can also be applied to determine the instantaneous TCP with respect to the base frame $\mathbb{F}_0$ by

$$
{}^{0}p_{\mathsf{tcp}}(t)={}^{0}_{e}\mathbf{T}\left(\theta,\varpi,t\right)\begin{bmatrix}0\\0\\0\\1\end{bmatrix} \tag{5.12}
$$

because the TCP is defined as origo of the frame $\mathbb{F}_e$. It is therefore possible to determine the position of the end-effector frame with respect to the base frame $\mathbb{F}_0$ as a function of the beam strain. The TCP is then determined using this transformation. This ends the modeling part, and the sequel part is dedicated to the design of a controller capable of damping tool oscillations.



PART II
Manipulator controller

Chapter 6

Controller introduction

The main goal of this project is to damp oscillations in a flexible tool mounted on a manipulator. Oscillations are undesirable in tasks involving precise positioning of the tool center point, but they are naturally emerging when using less rigid tools or when the payload increases. Consequently, a controller must be adaptable to alternating payloads and tool interchanges. A flexible tool analog has been constructed for empirical experiments using an industrial manipulator. The tool is fitted with strain gauges, that are capable of measuring the strain in a certain point on the flexible beam. If the strain measurements differ from zero, the tool is bending in a certain direction, which is interpreted by the controller as an error signal.

By adding the strain measurements to the closed-loop, the oscillations can be damped. If the controller is additionally supported with a model of the flexible tool, the oscillations can be more efficiently counteracted. The previous part was dedicated to modeling the flexible behavior in the theoretical case, with the entire tool configuration known in advance. However, the tool dynamics may be unknown in practical manipulator setups and must be automatically estimated by means of system identification. This will be treated in chapter 8.

Because the general manipulator model is highly nonlinear it is difficult to find a single equilibrium point for linearization. Considering a *gain scheduling* approach is also unwanted for manipulators of several degrees of freedom. The state space will have to be divided into several subspaces for precise control.

An approach which is both nonlinear and capable of performing robustly when subjected to uncertainties in model parameters is the *sliding mode control* (SMC) [38]. The SMC will be designed with a view to the frequency content of the control signals to avoid supplying additional oscillations while trying to damp them. The controller will be designed on the basis of the model from the last part given in (5.6). From a Lyapunov based design approach the controller gain can be selected to maintain stability $\forall t > 0$ and for all uncertainties.

Model uncertainties are caused by system noise, which may be added to the system loop at various locations. Both measurements and model states can be corrupted with noise, which implies less accurate control of the system. If the boundaries of the different noise components are known (or estimates of), a robustness can be achieved. However, a robust controller decreases performance in turns of pace, because the remaining bandwidth is used to reject disturbances. Therefore, positive and negative consequences must be weighted up against each other and compared with given requirements for the control system. Different filtering techniques will moreover be applied to reduce measurement noise and thus improve the quality of model state estimates.

The controller is adaptive in the sense, that the internal model is adjusted with new estimates of the resonance frequency of the flexible tool. This is a consequent of a parameter estimation loop, that estimates the system model based on measurements. Before initiating the design of an SMC, the requirements of the controller must be mentioned and are therefore repeated from section 2.4

- The improved controller must perform equal or better than a controller based on a rigid manipulator model regarding oscillation settling time. This corresponds to a termination of the controller after the manipulator joints have reached their reference values
- Tool oscillations must be damped to allow fast TCP positioning. A practical example (given by Ole Madsen, see personal profile in acknowledgements) has a 10-12 s task time and a settling time of $t_{\mathsf{set}} < 5$ s is selected on the basis of this time interval
- Tool tip deflection must be below the level of the path accuracy of $\pm 0{,}1$ mm from the default manipulator controller [80]
- The controller structure must be based on nonlinear control theory

The listed requirements must be measurable. An experiment will be conducted with a controller without oscillation damping. This will provide a set of nominal data. Differences between the nominal response and the improved response are evaluated on the basis of the acceptance test in chapter 10. Before designing the controller, the technique of sensor fusion is described in chapter 7. By applying a Kalman filter to the measurements the model estimates can be improved over the model estimates alone. This can be achieved by supporting the system model with sensor measurements. Next, system identification methods are described in chapter 8. This makes it possible to control the system even though some parameters are not known in advance.

Chapter 9 derives the sliding mode controller based on the model from the previous part supported by Kalman filtered estimates and estimated parameters. An important part of the controller design is to ensure, that the implemented controller performs as intended. This is also treated in the controller chapter. The specifications of the system in question, consisting of a 3-DOF industrial manipulator and a flexible tool, are summarized in table 6.1 with data from section G.1 and G.4. Certain selections have been made to filter out unnecessary data like the large tip mass, that was concluded to heavy for the current hardware configuration, see remark 2 on page 42.

Manipulator	Drive train gear ratio [25, 79]	1:100
	Link length, axis 1 [79]	0,600 m
	Link length, axis 2 [79]	0,684 m
	Link length, axis 3 (measured)	0,100 m
Interface	Analog output (NI 9263) limitations [67]	$\pm$10 V
	Digital input (NI 9411) differential limitations [70]	$\pm$24 V
	Analog input (NI 9201) single-ended limitations [69]	$\pm$10 V
	Analog input (NI 9201) resolution [69]	12 bit
Flexible tool	Tool length	0,381 m
	Tip mass	0,351 kg
Sensors	Accelerometer range [4]	$\pm$3,6 g
	Gyroscope range [90]	$\pm$1200 deg/s
	Encoder line count [79]	4000/rev
	Effective line count (due to gear ratio) [25]	400000/rev
	Strain gauge range [27]	$\pm(2-4)$ %

Table 6.1: *Known system specifications (inspiration and terms from table 1.1 in [96])*

Minor adjustments can be necessary to show different aspects of the control system, but they will be described if needed. The list contains all known information, but in order to design a suitable controller, a number of additional parameters will be needed. This is e.g. the link inertias, drive train frictions and motor parameters. The sequel will shortly describe the controller design process involving estimation of these unknown parameters. The design process of the controller is affected by the fact, that the actual model parameters are unknown in advance. In order to design a useful controller with verifiable performance, the following workflow is used

- Use of test signals to estimate system parameters and model uncertainties offline
- Design automatic parameter estimation algorithm, that is able to work online and estimate the same parameters as the offline process
- Use the model parameters to design a controller with specific performance
- Test the controller on hardware with and without online parameter estimation

By following the proposed workflow, it is ensured, that the parameter estimation process is capable of fitting correct parameters to the model structure. The controller must be designed on the basis of the nominal system, but robustness to model uncertainties makes the controller able to perform even if the system differs from the nominal system. Next chapter describes the sensor fusion technique.

Chapter 7

Sensor information fusion

Information from the various sensors is used to improve the accuracy of the feedback loop. Sensor noise, model uncertainties and model shortage can be of less significance when estimating the trend of the signals (filtering the signals) before they are used for control. A Kalman filter is useful for doing just that and requires prior knowledge of the sensor noise distributions. The noise variances for each sensor has been given in appendix G.2. Before describing the details around the filter the objective of the filtering process must be outlined. An illustrative representation of the sensor fusion process is shown in figure 7.1.

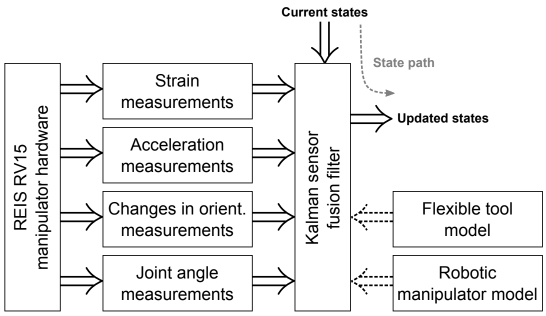


Figure 7.1: *Overview of sensor fusion process*

Acceleration, change in orientation and joint angles, are measured by accelerometers, gyroscopes and rotary encoders, respectively. The joint sensors are providing measurements of the interconnected angle between consecutive links, whereas the accelerometers and gyroscopes measures from the $\mathbb{F}_\tau$ frame. The current system states are filtered by a Kalman filter, that includes the sensors measurements to estimate the updated states. This will over time improve the accuracy of the estimate[1], because the states are not only simulated, but also adapted to actual measurements. Notice also, that the Kalman filter gains knowledge from the manipulator model as well as the flexible tool model.

[1]An improved accuracy by using multiple sensors over a single one depends of course on the uncertainty of the sensor itself and must be statistically estimated in each case

This chapter will first include a description of the linear Kalman filter, which is the basic tool for fusing model estimates and sensor measurements together. After that a number of sections are dedicated to show how the filter is running. This includes e.g. the common problem of using acceleration measurements when there is no acceleration state for comparison.

7.1 Linear Kalman filter

From appendix G the different sensors were introduced, and their outputs were expressed by stochastic models. This gives the following sensor noise covariance matrix $R \in \mathbb{R}^{13\times 13}$ for the Kalman filter algorithm

$$R = \begin{bmatrix} \sigma^{\mathsf{acc}}\mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma^{\mathsf{gyro}}\mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma^{\mathsf{ja}}\mathbf{I}_6 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^{\mathsf{str}} \end{bmatrix}^2 \tag{7.1}$$

with the zero matrices of appropriate sizes. The reason for the diagonal structure is due to the assumption of sensor independence. A total of 13 sensor signals can be acquired, but due to the model reduction (see remark 1 on page 18 and assumption 5 on page 48) a total of only 4 sensor signals are necessary to be processed by the filter. The encoder is sampled by an NI-9411 input module and the remaining are sampled by an NI-9201 A/D sampler. Both card types are described in appendix G. After the reduction the sensor noise covariance matrix can be written in compact form as

$$R = \mathrm{diag}\{\sigma^{\mathsf{acc}}, \sigma^{\mathsf{gyro}}, \sigma^{\mathsf{ja}}, \sigma^{\mathsf{str}}\}^2 \tag{7.2}$$

First step in the Kalman filtering is the *a priori* estimation of the model states one step ahead using a discretized version of the continuous model structure [43]

$$\begin{aligned} \hat{x}^-_{k+1} &= \mathcal{A}_k \hat{x}^+_k + \mathcal{B}_k u_k \\ \hat{z}_{k+1} &= \mathcal{C}_k \hat{x}^-_{k+1} \end{aligned} \tag{7.3}$$

Notice how the system matrices are indexed as well because they are updated with new parameters estimates based on system identification, see chapter 8. The continuous model is given in equation (5.6) at the end of chapter 3. The error covariance matrix is also estimated, which is the covariance of the state estimate errors

$$P^-_{k+1} \triangleq \mathbb{E}[\tilde{x}^-_{k+1}(\tilde{x}^-_{k+1})^\intercal] = \mathcal{A}_k P^+_k \mathcal{A}^\intercal_k + Q_k \tag{7.4}$$

with Q_k expressing the covariance matrix of the process noise at the k-th sample. In this case, an assumption is in order to address the propagation of the sensor noise covariance matrix. Q is estimated through simulations.

> **Assumption 7:** *By means of the process being unaffected by wear and mechanical vibrations, the propagation of the process noise covariance matrix Q can be simplified by the relational definition $Q_{k+1} \triangleq Q_k$* □

The state estimate error is defined as the difference between the estimated state and the actual state, $\tilde{x}_k^- \triangleq x_k^- - x_k$, where x_k by means of equation noise cannot be given. Next step in the Kalman filtering process is to use the a priori estimates to achieve an a posteriori estimate of the actual state [43]

$$K_{k+1} = P_{k+1}^- \mathcal{C}_{k+1}^\intercal (\mathcal{C}_{k+1} P_{k+1}^- \mathcal{C}_{k+1}^\intercal + R_{k+1})^{-1} \tag{7.5}$$
$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + K_{k+1}(z_{k+1} - \hat{z}_{k+1}) \tag{7.6}$$
$$P_{k+1}^+ = (I - K_{k+1}\mathcal{C}_{k+1})P_{k+1}^- \tag{7.7}$$

where R is the covariance matrix of the *sensor noise*. A similar assumption to assumption 7 can be given for the sensor noise R_k.

Assumption 8: *By means of the sensor noise being assumed unaffected by wear and mechanical vibrations, the propagation of the sensor noise covariance matrix R can be simplified by the relational definition $R_{k+1} \triangleq R_k$.* □

A set of initial conditions must be given to start the iterative filtering process, and they can be given as the following

$$P_0 = R, \quad x_0 = \bar{0} \tag{7.8}$$

Using the $x_0 = \bar{0}$ condition is only applicable when the manipulator is reset at every startup. Whenever $x_0 \neq \bar{0}$ the state vector is measured by the joint angle sensors[2].

The different variables supported by a Kalman filter and how measurements are related to them are described in the sequel section. All variables are used to improve the accuracy of the strain estimate. This enables a more precise state estimation, which will improve the overall accuracy of the controller.

7.2 Relating measurements and model outputs

A number of sensors provide measurements for the control system. Some of the sensor read-outs are directly comparable to model states, whereas others must be converted in order to represent the same as some model state. The reason follows from (7.6), where the difference between an estimated model output and a measurement of that particular output is used to update the estimate model state vector. Four different measurements performed by sensors introduced in appendix G.2 will be used, and their implementation in the sensor fusion process will be explicitly given in the following. A posteriori state vector update from (7.6) can then be stated as

$$\hat{x}_{k+1}^+ = \hat{x}_{k+1}^- + K_{k+1}\tilde{z}_{k+1} \tag{7.9}$$

with $\tilde{z}_{k+1} = z_{k+1} - \hat{z}_{k+1}$ expressing the difference between measurement z_{k+1} and estimated model output $\hat{z}_{k+1}$. In cases where the measurement must be converted to

[2]Absolute rotary encoders are useful in this context, as they will provide an exact joint angle even after power has been removed. This is not an option on the REIS RV15, and the manipulator will have to reset/synchronize on every startup

fit this form, a difference will be given as $\tilde{z}_{k+1} = f(z_{k+1}) - \hat{z}_{k+1}$ with $f(\cdot)$ expressing a mapping function. This notation will be used throughout the section without individually denoting each mapping function with a distinctive index.

7.2.1 Joint angles

Joint angles are defined as the angle between two consecutive links. The joint angles are included directly as model states within the dynamics model of the system and are measured directly by rotary encoders at each joint. Denoting the measurements by z^{ja}, the difference $\tilde{z}^{\mathsf{ja}}_{k+1}$ used in (7.9) is given as

$$\tilde{z}^{\mathsf{ja}}_{k+1} = z^{\mathsf{ja}}_{k+1} - \hat{\theta}_{k+1} \tag{7.10}$$

Next sensor to describe is the gyroscope.

7.2.2 Change in orientation

A gyroscope is applied to measure the change in orientation of the frame $\mathbb{F}_\tau$. The orientation of that particular frame is depending on three model states $\dot{\theta}_1$ through $\dot{\theta}_3$. Therefore

$$\dot{\theta}_\tau = \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \tag{7.11}$$

However, since the two lower axes are not applied, the gyroscope is measuring $\dot{\theta}_3$ directly. The difference $\tilde{z}^{\mathsf{gyro}}_{k+1}$ is therefore on the form

$$\tilde{z}^{\mathsf{gyro}}_{k+1} = z^{\mathsf{gyro}}_{k+1} - \dot{\theta}_{3,k+1} \tag{7.12}$$

Two states have now been improved using measurements. Next sensor is measuring the strain of the flexible tool.

7.2.3 Tool strain

Similar with the joint angle the strain is represented by a model state. The measurement can therefore be applied directly without transforming it. The measurement is $\tilde{z}^{\mathsf{str}}_{k+1}$ and the different equation is given as

$$\tilde{z}^{\mathsf{str}}_{k+1} = z^{\mathsf{str}}_{k+1} - \hat{q}_{k+1} \tag{7.13}$$

The next one cannot be made directly comparable to a model state without a transformation.

7.2.4 Acceleration

This last variable for the Kalman filter is not a direct state in the dynamic manipulator model and will thus have to be added as a "sensor state". Two ways are considered to include the sensor state within the Kalman filter. Firstly, a CA-model (*constant acceleration*) is using the jerk (third derivative of position) in the following way [56]

$$\ddot{\theta}_{k+1} = \ddot{\theta}_k + T_s \dddot{\theta}_k, \quad \dddot{\theta} \sim \mathcal{N}(0, \dddot{\theta}^2_{\max}) \tag{7.14}$$

The acceleration is thus assumed constant with some fluctuation modeled as a normal random process. However, determining the variance $\dddot{\theta}^2_{\max}$ is not straightforward, since no actual model is available. An optimal guess is therefore the only method and only through simulations can the results be revealed. This method, however, is conflicting with the fact, that there exists (even though unknown) a continuous model of the jerk, which in this case is set equal to a random white process. To overcome this non-trivial method, a second approach can be taken. Before this method can be exploited, the model must be discretized, expressing the model on the form [66]

$$\begin{aligned} x_{k+1} &= (\mathbf{I} + \mathbf{A}T_s)x_k + \mathbf{B}T_s u_k \\ y_k &= \mathbf{C}x_k + \mathbf{D}u_k \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} x_{k+1} &= \mathcal{A}_k x_k + \mathcal{B}_k u_k \\ y_k &= \mathcal{C}_k x_k + \mathcal{D}_k u_k \end{aligned}$$

with T_s being the sample time of the discrete time model. The last expression is similar to the one in (7.3) though with a different state vector notation specially for the Kalman filter theory. With this model a fictitious state $a_i(k) = \dot{\theta}_i(k-1)$ can be added, yielding the following structure (state vector written explicitly)

$$\begin{aligned} \left[\begin{array}{c} \theta_{3,k+1} \\ \dot{\theta}_{3,k+1} \\ \hline a_{k+1} \end{array}\right] &= \left[\begin{array}{cc|c} \multicolumn{2}{c|}{\mathcal{A}_k} & 0 \\ \hline 0 & 1 & 0 \end{array}\right] \left[\begin{array}{c} \theta_{3,k} \\ \dot{\theta}_{3,k} \\ \hline a_k \end{array}\right] + \left[\begin{array}{c} \mathcal{B}_k \\ \hline 0 \end{array}\right] u_k \\ z^{\mathsf{acc}}_{k+1} &= \frac{1}{T_s} \left[\begin{array}{cc|c} 0 & 1 & -1 \end{array}\right] \left[\begin{array}{c} \theta_{3,k+1} \\ \dot{\theta}_{3,k+1} \\ \hline a_{k+1} \end{array}\right] \end{aligned} \tag{7.15}$$

where z^{acc} represents the acceleration estimate. A simpler notation can be used to show what happens

$$\hat{\ddot{\theta}}_{3,k+1} = \frac{\dot{\theta}_{3,k+1} - a_{k+1}}{T_s} = \frac{\dot{\theta}_{3,k+1} - \dot{\theta}_{3,k}}{T_s} \tag{7.16}$$

It is now possible to express a difference for the Kalman filter as

$$\tilde{z}^{\mathsf{acc}}_{k+1} = z^{\mathsf{acc}}_{k+1} - f(\dot{\theta}_{3,k+1}, a_{k+1}) = z^{\mathsf{acc}}_{k+1} - \hat{\ddot{\theta}}_{3,k+1} \tag{7.17}$$

with z^{acc}_{k+1} denoting the measurements from the accelerometer. The accelerometers are working in Cartesian coordinates, and the measurements will therefore have to be converted to polar coordinates. However, assumption 9 expresses how the small-signal approximation from assumption 5 simplifies the case.

> **Assumption 9:** *Because the oscillation damping procedure is only to operate around a certain operation point defining the point of destination, the horizontal accelerometer is not needed. Furthermore, the vertical acceleration can be approximated as linear, and a conversion to polar coordinates is not necessary.* □

Using the above four difference equations a complete model can be included in the Kalman filter. Four different sensors are used to gather information from the system, and will be used to improve the accuracy of the estimated model outputs. Table 7.1 summarizes the difference equations derived in the previous section, when using only the third axis of the manipulator.

Difference equations from (7.9)	
Joint angles	$\tilde{z}^{\mathsf{ja}}_{k+1} = z^{\mathsf{ja}}_{k+1} - \hat{\theta}_{3,k+1}$
Change in orientation	$\tilde{z}^{\mathsf{gyro}}_{k+1} = z^{\mathsf{gyro}}_{k+1} - \hat{\dot{\theta}}_{3,k+1}$
Flexible tool strain	$\tilde{z}^{\mathsf{str}}_{k+1} = z^{\mathsf{str}}_{k+1} - \hat{q}_{1,k+1}$
Acceleration	$\tilde{z}^{\mathsf{acc}}_{k+1} = z^{\mathsf{acc}}_{k+1} - \hat{\ddot{\theta}}_{3,k+1}$

Table 7.1: *Summary of difference equations from section 7.2*

A model used only for testing must be constructed to fit the model from (7.3) as well as including the estimation of acceleration. The model is constructed from (5.6), which yields the following matrices

$$\mathbf{A}_s = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\omega^2 & -2\zeta\omega & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_s = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{C}_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{T_s} & -\frac{1}{T_s} \end{bmatrix} \tag{7.18}$$

which will provide the estimated model output vector

$$\hat{z}_{k+1} = \begin{bmatrix} \hat{q}_1 & \hat{\theta}_3 & \hat{\dot{\theta}}_3 & \hat{\ddot{\theta}}_3 \end{bmatrix}^{\mathsf{T}}_{k+1} \tag{7.19}$$

The matrix notations $\mathbf{A}_s$, $\mathbf{B}_s$ and $\mathbf{C}_s$ have been used to denote the continuous model used by the Kalman filter. This model and the measurement vector is used together with the Kalman equations (7.3) through (7.8). Before the model can be applied, it must be discretized. The covariance matrices Q and R must be tuned in order to achieve satisfactory performance of the filter. Even though the model estimates can be improved using measurements, there is a trade-off between trusting the model or trusting the measurements. The measurements will give the exact response of the system but will be subject to sensor noise. The theoretical model on the other hand will provide noise free estimates but the model may be incorrect. The best possible estimate is obtained by measuring the resonance frequency experimentally and then trusting the model more than the measurements. Then the measurements will can be used to perform the small corrections needed to match the exact resonance frequency at all times. The filter is tested in appendix D and evaluated in the acceptance test in chapter 10. A short summary will end this chapter before the system identification processes used to estimate the resonance frequency are discussed in the sequel chapter.

7.3 Summary

In the above, a linear Kalman filter was discussed and tested on a simulation of the actual system. A linear Kalman filter uses measurements from sensors to update the model state estimates from a theoretical model. The adaption is performed on the basis of a filter gain matrix K, which is generated from a system model matrix and a covariance matrix. The covariance matrix is a measure of the model estimate variances, and tells how deviating the estimates may be from the actual ones.

Each sensor output is compared to a corresponding model state, and the model estimates are updated accordingly. The method is not the answer to everything, and it must be tuned properly to benefit from the filter. However, since all the noise distributions of both process and sensors may not be known in advance, which they barely are, the tuning must be performed empirically. From there, the performance of the filter must be manually evaluated in each case. In this case with a significant resonance peak in the frequency response, the model must be accurate, if the measurements are subject to a high level of noise. An incorrect resonance frequency will make the controller inefficient, and perhaps making the oscillation damping task impossible. Therefore, the resonance peak must be accurately determined for each tool, which is the topic of the next chapter involving system identification.

Chapter 8

System identification

The objective of this project is to damp oscillatory motions of a flexible tool mounted onto a manipulator structure. This ensures a faster settling time and thus a faster positioning of the end-effector. Dynamics of the tool will naturally decay in time, but in order to achieve a faster decay the dynamics must be damped using a controller with knowledge of the system behavior to different control signal excitations.

Physical models were derived for both the manipulator itself and for the tool in the modeling part. The manipulator model requires system constants, that are not given in advance, and they must be estimated before the model is used for control purposes. In the case of the tool model all parameters are given in advance (see appendix G.4), but they may vary in the practical setup, and the natural response will change accordingly. This calls for a parameter estimation[1] procedure. All available sensor measurements will be used together with the corresponding input signals to estimate the parameters. The measurements needed (available) to estimate parameters for different parts of the system are listed below

Actuator parameters: Joint angles
Manipulator parameters: Acceleration, change in orientation
Tool parameters: Strain of tool

The reason why no sensors are mounted on the load itself e.g. accelerometer/gyroscope is because no additional sensors are to be mounted on tools operated by the manipulator. Strain measurements on the tool beam simulate strain measurements from the tool mount itself, which is sufficient information to estimate the behavior of the tool if a general model is known in advance. All measurements have been "improved" using sensor fusion techniques as described in the previous chapter.

An important factor of an autonomous system is the ability to gather information and process that into useful control signals. The signals will be designed to follow given requirements or simply to keep the system running no matter what disturbances are affecting the process. System identification techniques estimate a parameter vector

[1]Parameter estimation is a special case of system identification, whereas system identification in general can be used to identify model structure and model size. The terms may be interchanged within the chapter, since they are closely related

with system parameters currently describing the system in the best way, when fitting them to a gray-box model. This type of model contains a structure and variable parameters that can be estimated from measurements. The general model forms considered in the sequel will be either on linear or nonlinear state-space form with noise terms. The linear form is given by

$$\mathcal{S}: \begin{cases} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}u(t) + d_s(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}u(t) + d_m(t) \end{cases} \tag{8.1}$$

with d_s and d_m representing uncertainties of the system and measurements, respectively. A more generalized nonlinear version of the model, as given in [54], is expressed in the following way

$$\mathcal{S}: \begin{cases} \dot{x}(t) &= f(t, x(t), u(t), d_s(t) \mid \vartheta) \\ y(t) &= g(t, x(t), u(t) \mid \vartheta) + d_m(t) \end{cases} \tag{8.2}$$

where $\vartheta \in \mathbb{R}^n$ is the *system parameter vector* of system $\mathcal{S}$. Further conditions are $x, d_s \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $y, d_m \in \mathbb{R}^{n_y}$ with n_x, n_u and n_y representing the number of states, inputs and outputs, respectively. The parameter vector must be updated while the system is running and will thus have to rely on an algorithm capable of running in real-time. A parameter estimation algorithm relies on data from the system, and the more data the more accurate estimation of parameters (if the data is containing information, only unique data vectors). Different identification types will be applied for different purposes. Two major categories will be investigated for offline and online computation. This is outlined by the following list

Online computation: Real-time update of the system model parameter vector is necessary to achieve the best possible systems control. An *extended Kalman filter* (EKF) will be used to update the parameters based on a nonlinear model structure, which is known in advance. This gray-box model will be defined with variable parameters modeled as states in the EKF model.

Offline computation: When verifying the results from the EKF algorithm, an offline parameter estimation process will be applied to empirically gathered data. The offline algorithm is a *prediction error method* (PEM), which is an optimal algorithm in the sense of minimizing a cost function. Due to the computational time inequality between samples it is not suitable for real-time use on slower processing platforms.

Both types of computation requires data and a model to perform, which will be described in details in the sequel sections. Firstly, all denoted methods, EKF and PEM, will be outlined, and later on the methods for data acquisition will be given. This involves how to gather data for offline parameter estimation, that contains the best possible information and how to initiate the online parameter estimation process, when the most of the model parameters are initially unknown. Further, the *subspace method* N4SID is used offline to investigate whether or not the constructed model structure is containing all necessary information to predict the response of the system. Even though estimating a linear model the significant dynamics will be estimated and will be compared with the results from both nonlinear methods. For reasons that will become clear later in the chapter a replacement of the EKF is necessary. In this case the *relay tuning* method has been selected. More on that in section 8.4. First method to investigate is the EKF.

8.1 Extended Kalman filter (EKF)

When using a Kalman filter the majority of tasks involves improving the estimate of a model state using external measurements. Using knowledge on both the uncertainty of the measurements and the system model, a state prediction can be achieved which in the nominal case, converges to the actual state after some time. The same principle however, can be used to estimate model parameters when treating the parameters as virtual states of the model with associated uncertainty model given by [50]

$$\vartheta_{k+1} = \vartheta_k + \xi_k \tag{8.3}$$

with $\xi_k \sim \mathcal{N}(0, \sigma^2)$. The additional noise component is preventing ϑ_k to be independent of progressing time. Since no "real" model of the parameters is providable, the Gaussian noise component will due, which is always used when constructing Kalman filters. The linear Kalman was introduced in chapter 7, but the more general EKF method applicable to nonlinear systems, will be described using the basic structure [50] (the generalized state vector x will be used to maintain recognizability with Kalman filter theory)

$$\begin{aligned} \hat{x}_{k+1}^- &= f_k(\hat{x}_k^+, u_k) \\ \hat{z}_{k+1} &= g_k(\hat{x}_{k+1}^-) \end{aligned}$$

It is now only necessary to determine the Jacobians of the nonlinear fields f and g and evaluate them at the current point of operation, thus

$$\mathbf{J}_{f,k} = \left.\frac{\partial f_k(x,u)}{\partial x}\right|_{\substack{x=\hat{x}_k^+ \\ u=u_k}} \qquad\qquad \mathbf{J}_{h,k} = \left.\frac{\partial h_k(x)}{\partial x}\right|_{x=\hat{x}_k^+}$$

The theory from chapter 7 can be applied to complete the last prediction step [43, 50]

$$P_{k+1}^- = \mathbf{J}_{f,k} P_k^+ \mathbf{J}_{f,k}^\intercal + Q_k$$

and the update steps

$$\begin{aligned} K_{k+1} &= P_{k+1}^- \mathbf{J}_{h,k+1}^\intercal (\mathbf{J}_{h,k+1} P_{k+1}^- \mathbf{J}_{h,k+1}^\intercal + R_{k+1})^{-1} \\ \hat{x}_{k+1}^+ &= \hat{x}_{k+1}^- + K_{k+1}(z_{k+1} - \hat{z}_{k+1}) \\ P_{k+1}^+ &= (I - K_{k+1}\mathbf{J}_{h,k+1}) P_{k+1}^- \end{aligned}$$

The variation over the linear version is the evaluation of the nonlinear fields at an operation point and the direct use of the nonlinear term $\hat{z}_{k+1}$ in the a posteriori prediction of $\hat{x}^+_{k+1}$. Using the model structure from chapter 3 and including virtual states to represent parameters, the filter structure for parameter estimation becomes [50]

$$\begin{bmatrix}\hat{x}_{k+1}\\ \hat{\vartheta}_{k+1}\end{bmatrix} = \begin{bmatrix} f_k(\hat{x}_k, d_{s,k}, u_k) \\ \hat{\vartheta}_k + \xi_k \end{bmatrix}$$
$$\hat{z}_{k+1} = g_k(\hat{x}_{k+1}) + d_{m,k}$$

based on a discrete version of the general nonlinear system description from (8.2). The discretization is performed using an Euler method [97]

$$\hat{x}_k = f_k(\hat{x}_k, d_{s,k}, u_k) \approx \hat{x}_{k-1} + T_s f(\hat{x}_k, d_{s,k}, u_k) \tag{8.4}$$

with $f(\cdot)$ denoting the continuous version of $f_k(\cdot)$. The system in question is the first part of (5.6) expressing the dynamics of the first mode of the flexible tool with the field (index 1 omitted for simplicity)

$$f = \begin{bmatrix} \dot{q} \\ -\omega_l^2 q - 2\zeta_l \omega_l \dot{q} + \ell_3 (NF)^{-1} u \end{bmatrix} \tag{8.5}$$

which can be simplified to

$$f(x,u) = \begin{bmatrix} \dot{q} \\ a_1 q + a_2 \dot{q} + a_3 u \end{bmatrix}, \qquad \text{with } x = \begin{bmatrix} q & \dot{q} & a_1 & a_2 & a_3 \end{bmatrix}^\top \tag{8.6}$$

The algorithm and the model for parameter estimation has been introduced, and the selection of Q and R matrices is covered in the sequel subsection. The matrices show to have great influence on the resulting estimates.

8.1.1 Selection of matrices Q and R

Two covariance matrices that appear in the Kalman equations must be selected prior to applying the filter. The matrices are given as the process covariance matrix Q and the sensor covariance matrix R. Both matrices cannot be derived analytically, and the process noise is difficult to even measure. They will thus have to be based on trial and error estimation. The first entry of matrix R can be measured from sampling the noise on the strain gauges and scaling the result, but it has been chosen to try different constellations and use the best one. In order to see what different choices of Q means for the estimation results, an example is shown in figure 8.1 for a second order system with $\omega = 2$ and $\zeta = 0{,}5$. The main choice of Q is $Q = \mathrm{diag}(10, 10, 100, 100)$, which is multiplied by a factor in the set $\{0{,}1 \;\; 1 \;\; 10 \;\; 100\}$ to see the effects of different Q. The initial guesses are selected as $2\zeta\omega = 3$ and $\omega^2 = 10$ with a simulated response given a noise term with $w_{\mathsf{noise}} \sim \mathcal{N}(0, W_{\mathsf{noise}})$, $W_{\mathsf{noise}} = \mathrm{diag}(1\text{E}{-4}, 1\text{E}{-2})$. Meanwhile, the selection of R for the test is $R = \mathrm{diag}(1, 10)$.

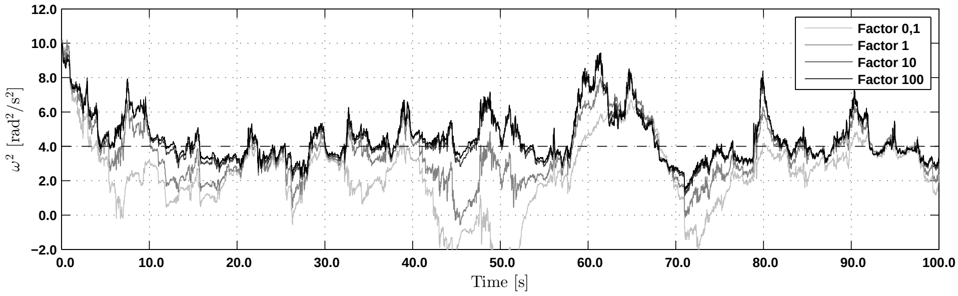


Figure 8.1: *EKF estimation of ω^2 for different Q*

Even though the initial guess of ω^2 is 2,5 times the original value, the EKF is able to converge to $\omega^2 = 4$, however, with larger fluctuations. Some of the Q selections show faster convergence but larger fluctuations, although with no fixed pattern. This shows one of the disadvantages of the EKF, namely that convergence is not guaranteed. The selection of Q cannot therefore not be formalized. The same is in evidence with the selection of R. Smoothing out the estimations using an MA-filter of order 1000 is shown in figure 8.2.

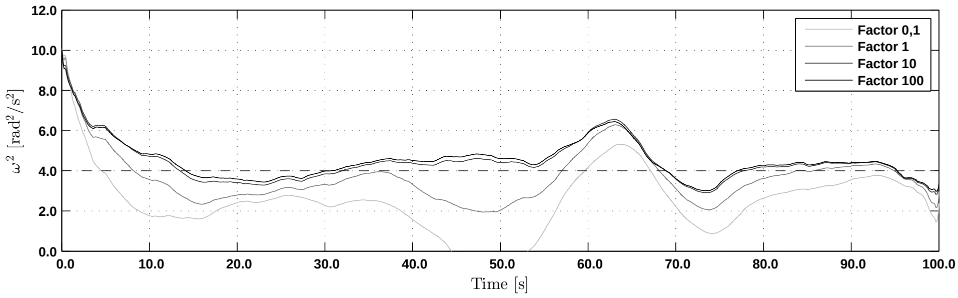


Figure 8.2: *Smoothing of EKF estimation of ω^2 for different Q*

After smoothing out the estimation results, the parameter is more applicable to the purpose of oscillation damping. However, the performance of the controller is highly depending on the accuracy of the resonance peak. The EKF method will be applied to the real system in appendix B, and a conclusion will be given based on the estimation results in chapter 10. This will conclude whether or not another method must be applied for online parameter estimation.

The second parameter to estimate is the damping factor ζ, which is also a derivative of the natural frequency ω. This makes the estimate fluctuating even further, since ω is part of the denominator. The estimation is given in figure 8.3 for different selection of Q.

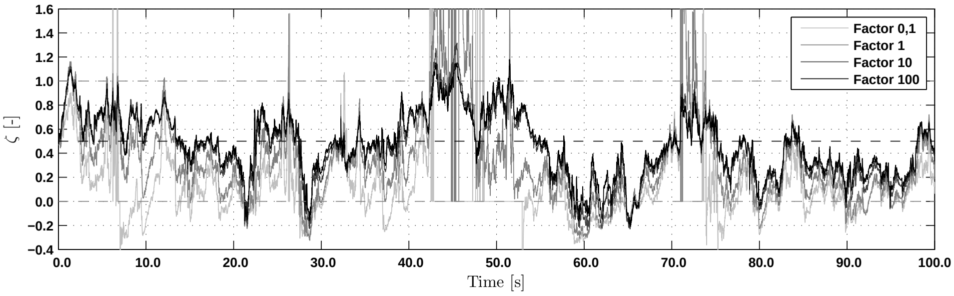


Figure 8.3: *EKF estimation of ζ for different Q*

Because this parameter is a derivative of another estimated parameter, the estimation is noisy. Using the filtered ω-estimates and smoothing the estimate of ζ yields the response in figure 8.4.

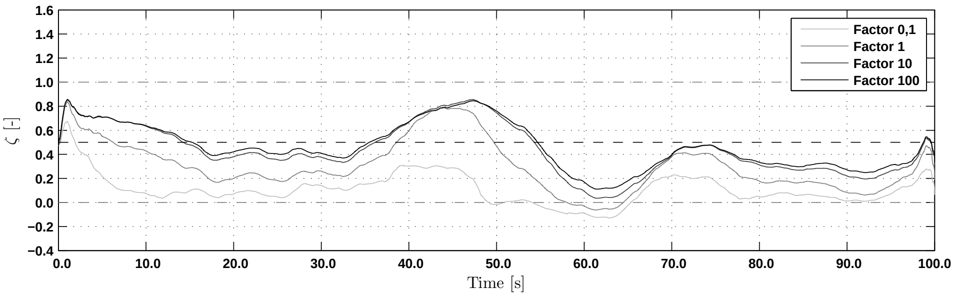


Figure 8.4: *Smoothing of EKF estimation of ζ for different Q*

Even though using filters, the estimate is varying and even beyond the domain of the definition $0 \leq \zeta \leq 1$. The damping factor has a great influence on the settling time of the tool oscillation and must be estimated properly. As with the ω-parameter, the use of online EKF will be evaluated after testing the algorithm on hardware. Possible reasons for non-accurate estimations are listed below

- High peaks: Due to division by estimates of ω close to zero
- Complex valued: Due to division by $\sqrt{\omega^2}$ estimate
- Noise: Due to large variances of the process and sensor models

These elements can only be bypassed, if noise is damped as much as possible, but the disadvantage will be slow convergence. The parameter estimation properties of the EKF is shown in appendix B and evaluated in chapter 10. Next, the prediction error method is described in short. This is applied offline to investigate the performance of the EKF by using data sampled from practical experiments. If the performance of the methods coincide (within some boundary), the EKF is applicable for this specific application.

8.2 Prediction error method (PEM)

Unlike the extended Kalman filter, the prediction error method is optimal in the sense of minimizing a specific performance function. The EKF cannot be guaranteed optimal due to the nonlinearities. Using the PEM enables a more precise estimate of the model parameters, but it also takes longer to perform the computation. The basics of the PEM will be explained in short, but an explicit algorithm will no be derived, because it has to run only in offline mode - unlike the EKF. Commercial software has been created to handle the estimation process, and this will be applied to do the offline computation. MATLAB System Identification Toolbox [93] with the `pem.m` function will be used to process the exact same data as the EKF. The basic optimization problem is given by a cost function [63, 53]

$$\hat{\vartheta} = \arg\min_{\vartheta} \left\{ \sum_{t=k+1}^{N+k} y(t) - g(t, \hat{x}_k, u_k|\vartheta) \right\} \tag{8.7}$$

By minimizing the argument on the basis of the selected parameter vector ϑ, an estimate $\hat{\vartheta}$ can be provided. The procedure of using the MATLAB function `pem.m` requires a nominal guess of the system [93], which will be a trial and error procedure in case of failed estimation (unrealistic parameter estimates). With the EKF procedure each parameter is modeled by a stochastic model. The PEM method in MATLAB requires that each parameter is defined as a variable, making the algorithm able to test different parameter vectors until the cost function is minimized within the limits of computation time, maximum number of iterations, cost function progress etc. The PEM is tested in appendix B and evaluated in the acceptance test in chapter 10.

The two significant parameter estimation methods have been explained, and what remains is the subspace method and the relay tuning. The subspace method is used to verify the dynamics of the model and check whether or not it contains enough information to represent the system. Relay tuning is used as a substitute for the EKF, which is described in details in section 8.4. These last two methods are also tested in appendix B and evaluated in the acceptance test in chapter 10.

8.3 Subspace identification method (N4SID)

A *subspace approach* is used to generate a complete linear model on the basis of input/output relations alone. No prior model structure knowledge is necessary, and the method will be used to generate a system for comparison with the ones used for EKF and PEM. The latter methods both require the basic system structure in advance (gray-box model), but if this is not correct, the parameters will not provide extra information no matter how accurate they can be fitted.

The N4SID relies on solving a set of LS-problems [53]. The theory is based on *N4SID* (*subspace state space*), which is able to estimate a free parameter system model by

$$\hat{\boldsymbol{\vartheta}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left(\sum_{t=0}^{N-1} \begin{bmatrix} x_{t+1}x_t^\intercal & x_{t+1}u_t^\intercal \\ y_t x_t^\intercal & y_t u_t^\intercal \end{bmatrix} \right) \left(\sum_{t=0}^{N-1} \begin{bmatrix} x_t x_t^\intercal & x_t u_t^\intercal \\ u_t x_t^\intercal & u_t u_t^\intercal \end{bmatrix} \right)^{-1} \tag{8.8}$$

from a set of data series on the form

$$\begin{aligned} \mathbf{X}(t|t,\ldots,t-N+1) &= \begin{bmatrix} x_t & x_{t-1} & \cdots & x_{t-N+1} \end{bmatrix} \\ \mathbf{U}(t|t,\ldots,t-N+1) &= \begin{bmatrix} u_t & u_{t-1} & \cdots & u_{t-N+1} \end{bmatrix} \\ \mathbf{Y}(t|t,\ldots,t-N+1) &= \begin{bmatrix} y_t & y_{t-1} & \cdots & y_{t-N+1} \end{bmatrix} \end{aligned}$$

with the system parameter vector being expanded to a parameter matrix including the entire system $\boldsymbol{\vartheta} \in \mathbb{R}^{(n_A+n_y)\times(n_A+n_u)}$ where n_A is the size of $\mathbf{A}$. The size of $N \in \mathbb{R}$ is limited from below to achieve an overdetermined system for the least-square problem to be solvable. No structure of the system can be given as a constraint when using N4SID over PEM/EKF. This is because the N4SID approach uses a free parametrization of the system, whereas PEM/EKF is able to estimate specific parameters due to their recursive construction. Advantages over the PEM includes the fixed execution time, since the PEM is optimizing the result by means of a squared error function. The N4SID methods is thus only optimal in a least-squares sense, and the necessity of numerous data samples is crucial. EKF is a recursive method, that is working on a nonlinear system model; unlike the N4SID.

Through small experiments in the design phase it was clear, that the method is not providing stable results, and that it requires data close to the actual model. Otherwise, the results are either unstable or does not match the physics involved. It has therefore been decided to omit this method to check for model structure. Instead the remaining methods will be used together with a gray-box model.

A last method is considered next - the relay tuning method. This is different from the ones above, because it works as a controller. Therefore, it cannot be operating in real-time, but it is able to make estimates within a few samples. It can therefore be executed whenever the manipulator is waiting.

8.4 Relay tuning (RT)

Before describing the method of relay tuning, it is important to mention, that this section requires reading the describing function method introduction in section 9.2 in the next chapter, since the theory is used through this section.

As a result of the poor estimate of the flexible tool resonance frequency when using EKF (see appendix B), another method must be considered to replace it. The method of *relay tuning* for parameter estimation is a new candidate [8]. This involves using a relay controller, and after some time the closed-loop will reach a limit cycle and oscillate with a constant frequency. This frequency, and the corresponding amplitude, defines an intersection between the flexible tool dynamics (related to the frequency) and the describing function of the relay (related to the amplitude). It is then possible to calculate backwards to the resonance frequency of the flexible tool. Figure 8.5 shows the closed-loop configuration with relay.

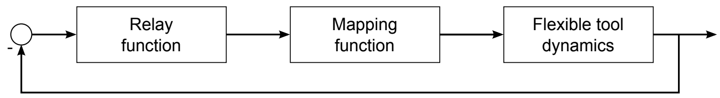


Figure 8.5: *Relay tuning closed-loop for resonance estimation (inspired by fig. 2 in [8])*

According to [8], the mapping function in the middle will have to be included when considering underdamped systems, which is the case for this project. If the function is excluded, the resulting limit cycle oscillation will not reflect the dynamics of the flexible tool. The mapping function is given as a sinusoid, and the initial frequency is arbitrarily selected, which may be the closed-loop frequency without the mapping function inserted [59].

The gains of both the transfer function, the relay and the mapping function are not considered important for the purpose of finding the resonance frequency. The method can also be applied to determine the exact transfer function for the system, but in this case, the gains are not important. However, they must be selected in such a way, that the practical system can be excited by the input signal.

After a number of iterations, the resulting limit cycle oscillation frequency is used as the new sinusoidal frequency. The phase time from $t = 0$ to some rising edge of the relay is also added to the mapping function. Similarly to numerical methods for solving equations, the procedure is selected to terminate, when the change in estimate is below a certain value. An example of the procedure is shown in table 8.1 on the assumed dynamics given by

$$G(s) = \frac{1}{s^2 + 2 \cdot 0,07 \cdot 4s + 4^2} \tag{8.9}$$

with 0,07 and 4 rad/s defining the damping and eigenfrequency, respectively. The results will be evaluated shortly. In order to test the system the initial guesses 0,0594 and 4,96 rad/s are used. The simulation setup listed in table 8.2.

	$k = 0$	$k = 1$	$k = 2$	$k = 3$
ω_{lc} [rad/s]	5,8670	2,9530	0,1534	-
A_{lc} [-]	0,0659	0,0507	0,0247	-
t_{shift} [s]	-	25,2800	52,3500	48,9300
ω_{res} [rad/s]	3,7743	2,1000	0,1317	-

Table 8.1: *Results from relay tuning method for estimation of limit cycle oscillation frequency*

Simulation time	100 s
Simulation frequency	100 Hz
Relay hysteresis h	0,01
Relay gain g	1

Table 8.2: *Simulation setup for resonance estimation using relay method*

The notation of relay hysteresis h clashes with the height of the flexible beam, but the height is not applied explicitly within the report. Instead it is implicitly given through the strain/deflection relation from appendix F. An FFT was used to determine the dominating frequency of the output signal. Figure 8.6 shows how the output is converging in frequency and gain. The direct FFT amplitude was used to estimate the gain, which is possible since the system converges within the first 10 seconds, but runs for an additional 90 seconds. No scaling is needed since the chosen relay has unity gain. Otherwise, the gain must be determined as the ratio between relay and plant amplitude. The time shift t_{shift} is determined from the 25th rising edge of the relay output, after which the outputs has converged.

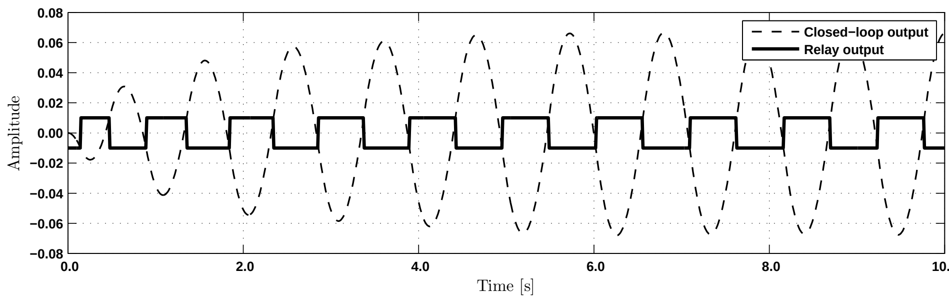


Figure 8.6: *Output from relay tuning closed-loop*

As seen from the plot, the self-oscillation frequency is achieved after the first couple of relay switches. The resonance frequency, or natural frequency, is calculated on the basis of the phase angle of both the relay and the plant. Given that the plant is a second order system on the form from (8.9), the phase angle can be expressed as

$$\angle G(\mathbf{j}\omega) = \text{atan}\,\frac{\Im\{G(\mathbf{j}\omega)\}}{\Re\{G(\mathbf{j}\omega)\}} \tag{8.10}$$

Using the technique of partial fraction expansion on $G(s)$ and evaluating the frequency content of the result, makes it possible to derive the expression

$$\begin{aligned} G(\mathbf{j}\omega) &= \frac{1}{(r_1 - r_2)(\mathbf{j}\omega - r_1)} + \frac{1}{(r_2 - r_1)(\mathbf{j}\omega - r_2)} \\ &= \frac{-r_1 - \mathbf{j}\omega}{\sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}(\omega^2 + r_1^2)} + \frac{-r_2 - \mathbf{j}\omega}{-\sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}(\omega^2 + r_2^2)} \\ &= \frac{1}{-\sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}(\omega^2 + r_1^2)} \left(\frac{-r_1 - \mathbf{j}\omega}{\omega^2 + r_1^2} - \frac{-r_2 - \mathbf{j}\omega}{\omega^2 + r_2^2} \right) \end{aligned} \tag{8.11}$$

with r_1 and r_2 being the roots to the characteristic polynomial $s^2 + 2\zeta\omega_n s + \omega_n^2$

$$r_1 = \frac{1}{2}\left(-2\zeta\omega_n + \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}\right) \quad \text{and} \quad r_2 = \frac{1}{2}\left(-2\zeta\omega_n - \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}\right)$$

The intersection point is given as the solution to $G(\mathbf{j}\omega)N(A) + 1 = 0$ (Barkhausen criterion from [76]), where $N(A)$ is the describing function of the relay with hysteresis given as [91]

$$N(A) = \frac{4g}{A\pi}\sqrt{1 - h^2A^{-2}} - \mathbf{j}\frac{4gh}{A^2\pi} \tag{8.12}$$

with g denoting the gain of the relay and h the size of the hysteresis. This method can only be applied, if the relay includes a hysteresis. Otherwise, the describing function will be on the real axis only, and no intersection can be estimated between the relay and the plant dynamics. Based on the output from the closed-loop, a frequency ω_{lc} and a gain A_{lc} is measured, and the values are replacing ω and A, respectively, in equations (8.11) and (8.12). The only unknown is the natural frequency ω_n, which is solved numerically by equating the phase angle of (8.11) and the phase angle of the negative inverse of (8.12)

$$\angle G(\mathbf{j}\omega_{\text{lc}}) = \angle\frac{-1}{N(A_{\text{lc}})} \tag{8.13}$$

Even though the method has been proved to improve estimates, this is not the case for this configuration, and the regular method without mapping function will be applied instead. The resulting frequency estimate from table 8.1 is converging towards zero. Using the first results ($k = 0$) of the above simulation, where the mapping was not used, a resulting limit cycle oscillation frequency of 5,867 rad/s and a gain of 0,0659 was observed from the FFT. This results in a resonance frequency of $\omega_{\text{res}} = 3{,}7743$ rad/s, when applying the above theory. This yields a difference of -5,64 % when compared to the expected value of 4 rad/s. A greater variance in the estimates is expected when using actual measurements. The frequency was determined using figure 8.7, which shows, that the constant phase of the relay (only for constant hysteresis and constant A_{lc}) intersects the phase angle of the flexible tool dynamics (determined for ω_{lc}) at -176,2676 degrees corresponding to a resonance of 3,7743 rad/s.

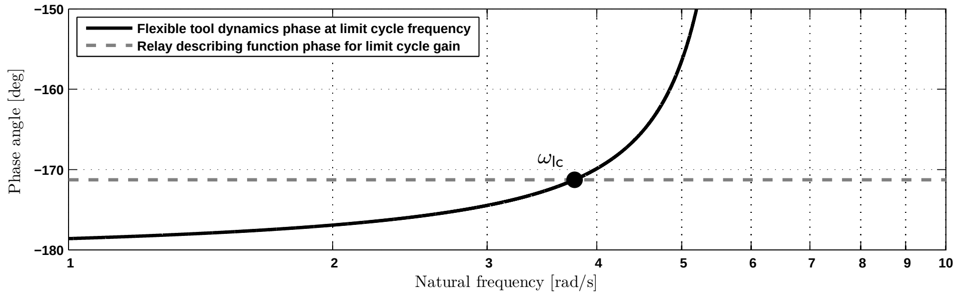


Figure 8.7: *Intersection between phase angles*

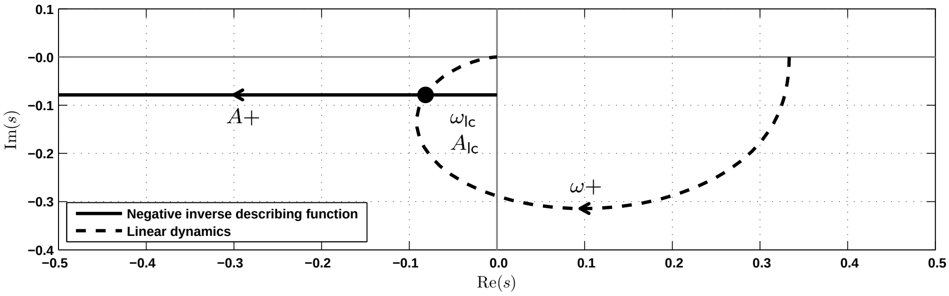


Figure 8.8: *Modified Nyquist plot (inspired by fig. 4 in [85])*

Figure 8.8 shows how the two dynamic functions intersect in the complex plane. Before ending this section, the convergence time and corresponding accuracy must be considered. The simulations above are performed using 100 seconds of data sampled at 100 Hz, but that is too long time if the manipulator is operating with different sized objects. It takes 33 samples at 100 Hz before the estimate is within the ±10 % band of the actual frequency and 44 samples to reach the ±5 % band of the correct frequency. However, it may be different in reality, and the number of samples may be larger to guarantee a steady estimate. Figure 8.9 shows the resonance frequency estimates as a function of the number of samples used.

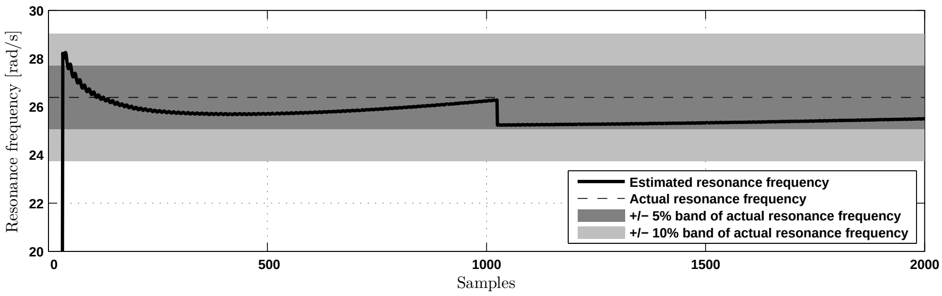


Figure 8.9: *Estimated limit cycle frequency as a function of sample size*

The first couple of estimates are not reliable, since the limit cycle is not fully reached yet, and the frequency is converging towards the correct one. Therefore, an experiment must determine the optimal sample size for this configuration, see appendix B. The size can then be increased to include variance and tools with lower dominating resonances. Because the system is underdamped, there will be a steady state error. In order to remove this, a mapping function is needed as introduced in the beginning of this section.

Through simulations though the mapping function converged the frequency estimates to zero. The relay tuning method without the mapping function was therefore applied in the practical experiment in appendix B. In order to sample useful data through experiments the test signals must be constructed properly. This is treated in details in appendix L, and the methods will be used to design control signals for the tests in appendix B. All methods needed for identifying the system have been explained in the above, and they will be summarized in the last section below.

8.5 Summary

A number of system identification methods for parameter estimation has been introduced in the above. The EKF is intended for online estimation and the PEM for offline estimation. A subspace method has been introduced as well, which is used to verify the model dynamics by estimating a complete model on input/output relations alone. Because the EKF showed to provide fluctuating parameter estimates, it has been substituted with the relay tuning method. This method is, however, not able to run online like the EKF. It must apply control signals to the system, but has a fast convergence time. Therefore, it can still be applied for realtime applications.

All methods can be applied to the system while the tool is mounted on the manipulator. Separate tests of the tool are therefore unnecessary. This ends the system identification chapter, and the model estimates will be used when designing a controller in the sequel chapter.

Chapter 9

Sliding mode controller

A model structure has been derived in the modeling part and a number of system identification methods has been introduced in the previous chapter. Both the model and the system identification algorithms will form the foundation for the controller design, which will be explained in this chapter. For the sake of simplicity it is assumed, that the parameters are not *time-varying*, but merely *unknown* and are therefore to be estimated. Further considerations must be taken if the model is essentially allowed to be hybrid. This part can be omitted by assuming that the model is only updated in between manipulator operations. Figure 9.1 shows how the assumption must be interpreted.

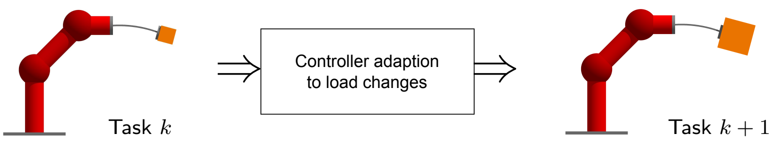


Figure 9.1: *Adaption of controller to changing load between tasks*

A *sliding mode controller* (SMC) has been selected to counteract the oscillating behavior of the flexible tool. The sliding mode controller is a nonlinear controller type, that is also robust to system model uncertainties [100]. It is therefore applicable in cases, where the systems are difficult to model (inadequate model) or the states suffers from model uncertainties. The term *sliding mode* denotes a trajectory that remains on a defined surface for all future time. This concept will be clarified throughout the controller chapter.

The SMC is in a category of systems called *variable structure systems* (VSS), that changes structure as a function of specific system states [33]. In this case, the controller gain changes as a function of the trajectory coordinates in the phase plane. A generalized VSS closed-loop is shown in figure 9.2 with the controller changing as a function of the system states in order to keep the trajectory pointing in direction of the switching surface (direction of gradient) at all time. The terms positive and negative are expressing which of the subspaces $s > 0$ and $s < 0$ the trajectory is currently moving in.

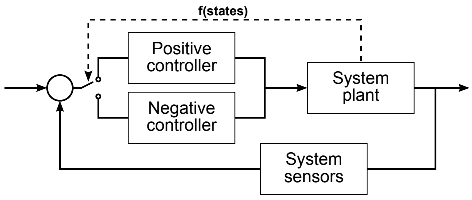


Figure 9.2: *Generalized VSS closed-loop (inspired by fig. 4 in [33])*

In practice, the trajectory cannot move along the surface due to effects like quantization, disturbances and round-off errors when solving the system equations numerically. This produces a trajectory, that crosses the *manifold* between the two subspaces defined by $s > 0$ and $s < 0$. The term switching manifold will be used interchangeably with the term *switching surface* throughout the chapter. Discontinuous control signals are used to guide the trajectory towards the switching surface again, when the trajectory is present in one of the two subspaces. By selecting the control signal properly it will make the trajectory converge to an equilibrium on the surface and therefore compose an asymptotically stable system. The manifold is defined in the following way [38, 87]

$$s(q) \triangleq \tilde{q} \left(\frac{\mathrm{d}}{\mathrm{d}t} + K \right)^{k-1} \tag{9.1}$$

with K denoting a gain that will ensure convergence to the surface according to given requirements and k denoting the order of the system to control. This manifold is applicable to multivariate cases, but only the single variable case will be considered and interpreted for this project. The variable q has been used in the modeling chapter to represent a scaled version of the flexible tool strain. It can be expressed by the single mode model in (4.30) from chapter 5, which is given in a slightly altered version here as

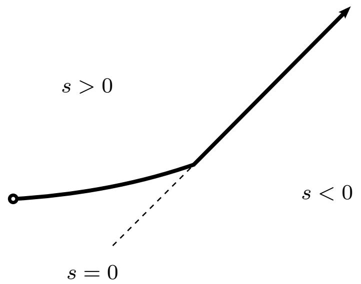
$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \end{bmatrix} u(t) \tag{9.2}$$

with the variables q_1, ω_1 and ζ_1 interchanged with q, ω and ζ, respectively, for simplicity. The constant K_1 is scaling the input signal and is determined through system identification in appendix B together with the remaining model parameters. From (9.2) the order $k = 2$ can be identified, and the surface is therefore given as

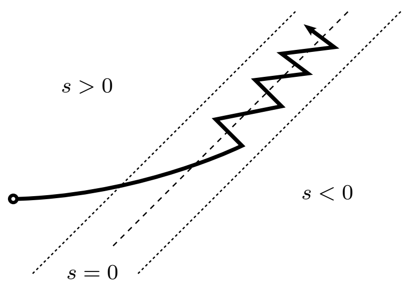
$$s(q) = \dot{\tilde{q}} + K\tilde{q} = \dot{q} - \dot{q}_{\mathsf{ref}} + K(q - q_{\mathsf{ref}}) \tag{9.3}$$

when including a reference trajectory q_{ref}. This provides the definition of a *tracking error* $\tilde{q}$ between the current state q and a reference q_{ref} given as $\tilde{q} = q - q_{\mathsf{ref}}$. Stability can be guaranteed in the general case if $s = \bar{0}$ is stable. Here s is given as a vector, since higher-order control can involve several switching surfaces, which will not be considered for this project. In this case the s-function is a scalar function, and the stability condition is therefore given as $s = 0$.

Two stable trajectories are illustrated in figure 9.3 with the dashed lines defining the bounds of the *chattering* that occurs when the trajectory is switching between the two subspaces in the presence of control delays. The phenomenon is explained in details in section 9.2. In the continuous theoretical case, chattering is not a concern, but when applied in discrete time steps, the problem emerges. Figure 9.3a shows the theoretical phase plane trajectory. The trajectory initiates from an initial position in the $(q, \dot{q})$-plane, and when the switching surface is reached, the trajectory remains there for $t \to \infty$. The practical case in figure 9.3b is slightly different, because the manifold is being crossed infinitely many times as $t \to \infty$, because the values of q and $\dot{q}$ are quantized. However, the trajectory is stable and remains within given bounds.



(a) *Theoretical*



(b) *Practical*

Figure 9.3: *Sliding surface interpretations*

Before initiating the controller design, the model from (9.2) must be converted into the vector field form

$$f(q) = \dot{x} \triangleq \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ \omega^2 q + K_1 u \end{bmatrix} \tag{9.4}$$

which is the common model structure for designing a sliding mode controller. The first part of the controller design process will be considering a model without friction part described by the differential equation

$$\ddot{q} + \omega^2 q = u \tag{9.5}$$

with unit gain input signal. The friction term $2\zeta\omega\dot{q}$ will be added later on. In order to separate the different parts of the discontinuous control, the $(q, \dot{q})$-plane is divided by a plane/surface. Using the definition in (9.3) without the external reference q_{ref}, because the vibrations must be regulated to zero, defines the surface

$$s = \dot{q} + Kq \tag{9.6}$$

By tuning the gain K, magnitude and direction of the vector field gradients can be selected to achieve the required trajectory behavior. The sliding mode controller drives the state trajectory $x(t)$ to the sliding surface $s = 0$. The notation $x(t)$ is used to denote the trajectory given by states q and $\dot{q}$. When it reaches the surface, it changes behavior and moves along the surface by $\dot{q} = -Kq$ given from the surface definition in (9.6). This yields an asymptotically stable solution

$$q(t) = q(t_r)\mathrm{e}^{-K(t-t_r)} \tag{9.7}$$

causing the q state to converge to zero as $t \to \infty$ [100]. t_r is the *reaching time* determined as the time between initial state $q(0)$ and the state $q(t_r)$, when reaching the surface. A control signal must be constructed to ensure a trajectory that is driven towards the surface independent of the initial conditions.

The division between subspaces is a common problem when dealing with *hybrid systems*. When a particle is traveling on the surface (an *invariant set* in which it remain for $t \to \infty$) between the two subspaces, no field is defined. A field is only given for the two subspaces, and the *Filippov solution* is therefore used on the set given by the condition $s = 0$. Using a discontinuous control allows for controlling the gradient in each subspace. All gradients are pointing towards the surface (if designed correctly), and the solution to the *differential inclusion* $\dot{x} \in f(x, u)$ is pointing towards the origo $(q, \dot{q}) = (0, 0)$ [60]. The solution is given as a *convex combination* of the fields on either side of the surface $s = 0$ expressed as $f_+(x, u)$ and $f_-(x, u)$. The variable x will be used as placeholder for the model states q and $\dot{q}$, and is not be be confused with the longitudinal position on the flexible beam. A convex combination between the fields is defined as

$$f_{\mathrm{cc}}(\alpha_{\mathrm{cc}}) = \alpha_{\mathrm{cc}} f_+(x, u) + (1 - \alpha_{\mathrm{cc}}) f_-(x, u), \quad \alpha_{\mathrm{cc}} \in (0, 1) \tag{9.8}$$

with the *convex hull* $\mathrm{co}(\cdot)$ defining the set of convex combinations between the fields that are candidates to form a solution to the differential inclusion [60]

$$x \in \mathrm{co}\{f_+(x, u), f_-(x, u)\} = \{f_{\mathrm{cc}}(\alpha) | \alpha_{\mathrm{cc}} \in (0, 1)\} \tag{9.9}$$

The instantaneous solution to the problem is tangential to the sliding surface and pointing towards the origin. To pick out the solution from the set of admissible solutions to the differential inclusion $x \in \{f_{\mathrm{cc}}(\alpha) | \alpha_{\mathrm{cc}} \in (0, 1)\}$, the surface $s = 0$ yields also $\dot{s} = 0$ and thus [57]

$$\begin{aligned} \dot{s} = \nabla_s^{\intercal}(x)\dot{x} = \nabla_s^{\intercal}(x) f_{\mathrm{cc}}(\alpha_{\mathrm{cc}}) = 0 \ \Downarrow \\ \alpha_{\mathrm{cc}} = \frac{\nabla_s^{\intercal}(x) f_-(x, u)}{\nabla_s^{\intercal}(x)(f_-(x, u) - f_+(x, u))} \end{aligned} \tag{9.10}$$

with one specific α_{cc} selecting the solution tangential to the s-function. In order to achieve the dynamics $q(t) = q(t_r)\mathrm{e}^{-K(t-t_r)}$ on the sliding surface, the ideal vector field must be on the form $\ddot{q} = -K\dot{q}$. Introducing a control signal to the dynamics makes it possible to express two f-functions using a discontinuous control law $u = U_0 \,\mathrm{sgn}(s)$ [100, 82]. The dynamics $f(x, u) = -\omega^2 q + u$ from (9.5) is therefore converted into the form

$$f(x, u) = \begin{cases} f_+(x, u) = -K\dot{q} - U_0 & \text{for } s > 0 \\ f_-(x, u) = -K\dot{q} + U_0 & \text{for } s < 0 \end{cases} \tag{9.11}$$

with U_0 applying a constant acceleration, which will drive the trajectory towards the surface. This yields an α_{cc} on the form

$$\alpha = \frac{\begin{bmatrix} K & 1 \end{bmatrix} \begin{bmatrix} \dot{q} \\ -K\dot{q} + U_0 \end{bmatrix}}{\begin{bmatrix} K & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -2U_0 \end{bmatrix}} = \frac{U_0}{2U_0} = \frac{1}{2} \tag{9.12}$$

by applying (9.10). The convex combination is therefore given as half of each field. In practice the Filippov solution is not derived, since the discontinuous control will automatically create a state trajectory similar to the Filippov solution. As suggested above, a discontinuous control law can be introduced to change the direction of the vector fields in each subspace. A conditional example of a control law can be stated as the following

$$u = \begin{cases} u^+ = q^+ + \omega^2 q & \text{for } s > 0,\ q^+ > K \\ u^- = q^- + \omega^2 q & \text{for } s < 0,\ q^- < K \end{cases} \tag{9.13}$$

which is assuming that instantaneous changes is acceleration is possible. This is of course not the case in practice, but due to the assumption of local control (see assumption 10), the velocity of the tool frame $\mathbb{F}_\tau$ must be kept as low as possible to prevent initiating tool oscillations.

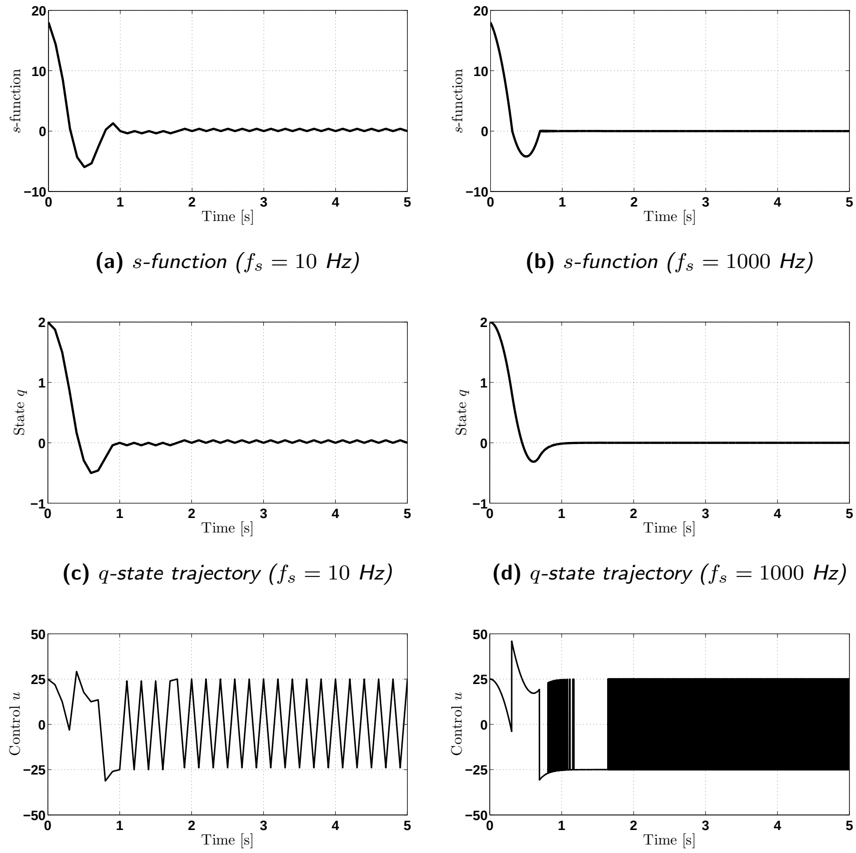
> **Assumption 10:** *Only local control (small-signal approximation) is needed to stabilize the flexible tool dynamics, and by assuming only friction based manipulator dynamics, a direct coupling to the input excitation signal $u(t)$ approximates instantaneous angular joint acceleration $\upsilon(t)$.* $\square$

The conditions $q^+ > K$ and $q^- < K$ are used to guide the trajectory towards the region of attraction given as the surface $s = 0$. Not respecting these conditions will divert the trajectory from the surface, which is causing instability. By applying the control from (9.13) to the model in (9.5) the accelerations on either side of $s = 0$ becomes $\ddot{q} = q^+$ (using u^+) and $\ddot{q} = q^-$ (using u^-), respectively. This is exactly the behavior that is desired, as it reflects the way the acceleration is supplied by the actuator drive electronics. The control law in (9.13) will be used in the sequel, because the $\omega^2 q$ term cancels out the negative counterpart from the model in (9.5), and the only remainder is the direct acceleration term. The control law can be altered to the form

$$u = \omega^2 q - U_0 \operatorname{sgn}(s) \tag{9.14}$$

without violating the existing inequalities $q^+ > K$ and $q^- < K$. In this construction, s is used as argument to a *signum* function, which allows for two different gradients on either side of the surface. The difference between the two cases (9.13) and (9.14) is the sign function of the controller gain U_0, which simplifies future calculations, and makes it possible to describe the control law in a single equation.

Simulating the system from (9.5) with the control law from (9.14) with arbitrary model parameter ($\omega = 5$), controller gains ($K = 9$ and $U_0 = 25$) and initial conditions ($(q, \dot{q}) = (2, 0)$), see block diagram in figure 9.5, yields the asymptotically stable surface function behavior shown in figure 9.4 for two different simulation frequencies. Notice that since $U_0 > K$, the control law from (9.13) is still fulfilled. If the controller stabilizes the plant dynamics, any initial solution will make the trajectory converge to the surface. This is possible because the term $\omega^2 q$ in the control law removes all known system dynamics. Robustness to model uncertainties is attained using the discontinuous control law. Non-conservative systems will make the trajectory converge to the origin of the phase plane with $(q, \dot{q}) = (0, 0)$.



(a) *s-function* ($f_s = 10$ *Hz*) **(b)** *s-function* ($f_s = 1000$ *Hz*)

(c) *q-state trajectory* ($f_s = 10$ *Hz*) **(d)** *q-state trajectory* ($f_s = 1000$ *Hz*)

(e) *Control signal* u ($f_s = 10$ *Hz*) **(f)** *Control signal* u ($f_s = 1000$ *Hz*)

Figure 9.4: *Simulation of system in figure 9.5 with controller from* (9.14)

Using a larger sample frequency limits the fluctuations of the s-function, because the control action can be updated more often. The high frequency control action, however, comes with a price, since the actuators wear out faster and the noise levels are increased. This will be further considered in section 9.2. A smoother s-function also smooths out the state trajectories. The simulation using $f_s = 1000$ Hz shows close behavior to the case where $T_s \to 0$ s ($f_s \to \infty$ Hz), which is referred to as *ideal*

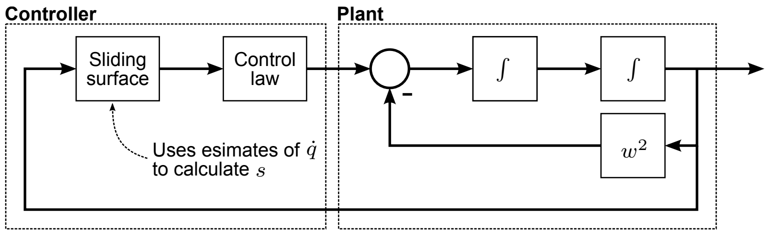
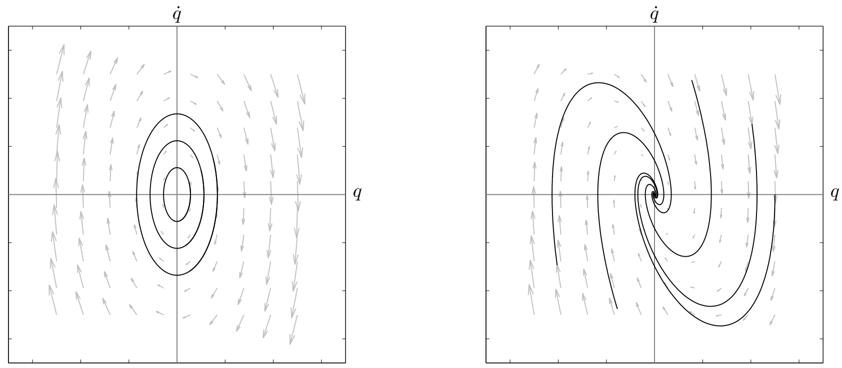


Figure 9.5: *Sliding mode controller for $\ddot{q} + \omega^2 q = u$ with control law from* (9.14)

sliding mode [99]. All other cases, and especially practical cases, can be denoted as *real sliding modes*. The gains in the system from the diagram in figure 9.5 are fitted manually to achieve a stable behavior that also shows chattering. However, a method must be applied to determine the gains, that will stabilize the system and yield the required performance.

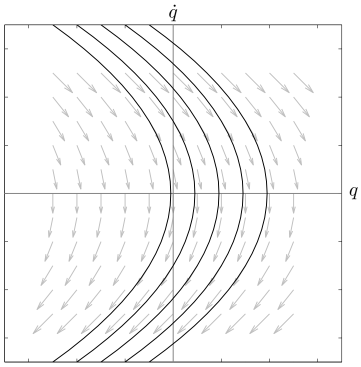
Firstly, the natural response (unforced) is considered for the system in (9.5) with and without the natural damping term $2\zeta\omega$ added. Phase portraits are given in figure 9.6 simulated with arbitrary but positive damping and frequency parameters ($\omega = 2$ rad/s and $\zeta = 0{,}5$). *The axes are not given any numbers, since the plots are only intended to show the relative shape of the trajectories for different initial conditions.* Both figures show stable behavior and figure 9.6b shows an asymptotically stable behavior due to the added friction.



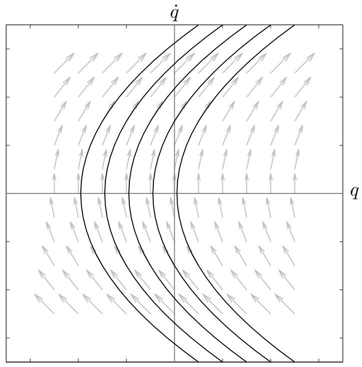
(a) *Without friction* **(b)** *With friction*

Figure 9.6: *Phase portraits of simple systems $\ddot{q} + \omega^2 q = 0$ and $\ddot{q} + 2\zeta\omega\dot{q} + \omega^2 q = 0$, respectively*

When adding control signals to the system using (9.14), two different responses can be achieved depending on the sign of s [100]. Phase portraits are given in figure 9.7 for both the $s > 0$ ($\ddot{q} = -U_0$) and the $s < 0$ ($\ddot{q} = +U_0$) case simulated with $U_0 = 5$ and $K = 1$. The plots show unstable phase portraits. By applying the definition from (9.6), a surface $s = 0$ can be located in the phase plane with the form $\dot{q} = -Kq$. By alternating between the two controls of the discontinuous control law, a globally stable behavior can be achieved, see figure 9.8.



(a) $\ddot{q} = -U_0,\ s > 0$



(b) $\ddot{q} = +U_0,\ s < 0$

Figure 9.7: *Phase portraits of vector fields f_+ (left) and f_- (right)*

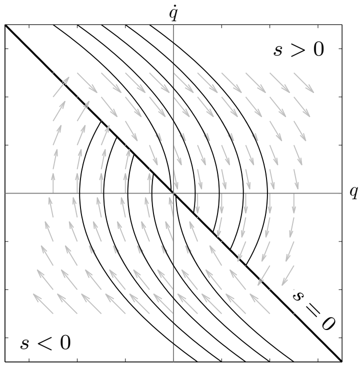


Figure 9.8: *Phase portrait of system from* (9.5) *controlled with law from* (9.14)

By switching between the two controller configurations, all trajectories, independent of initial conditions, will reach the sliding surface $s = 0$ and remain there as $t \to \infty$. By tuning the controller parameters, the shape of the trajectories and the controller performance can be adjusted. The specific U_0 will be selected to achieve certain properties of the controller, which is treated in section 9.3 regarding controller performance. 6 phase plots based on a simulation with different constants U_0 and K are shown in figure 9.9.

Different initial conditions has been used of the system shown in figure 9.5 for the simulation. As U_0 increases, the reaching phase is more direct towards the surface, and an increase in K further reduces the reaching time.

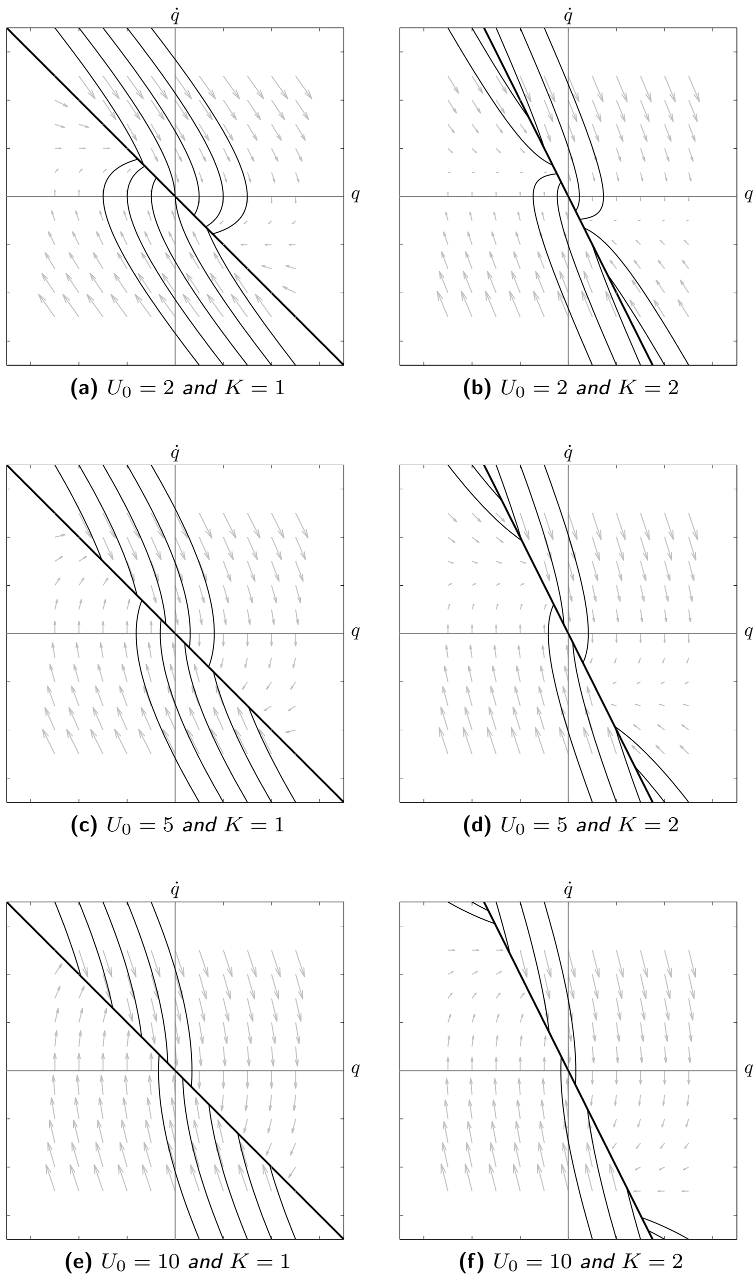


Figure 9.9: *Phase portraits of* (9.5) *controlled with law from* (9.16) *with different parameters*

The basic theory for designing a sliding mode controller has been provided, and the sequel section applies the methods to improve the control law in order to achieve a given performance. Later, the chattering issue will be addressed, which is introduced because of the limited bandwidth of the actuators and the controller. Complete disturbance rejection is possible when using an infinite gain, but in the practical case, there will be a trade-off between the disturbance bandwidth and the amount of chattering. The frequency content of the control signal is also affected, which cannot contain any resonance frequencies of the flexible tool.

9.1 Controller tuning

From the set of all possible control signals[1] $\mathcal{S}_u$ a subset will yield stability, which will be referred to as the *set of admissible control signals* $\mathcal{S}_s$. Moreover, a subset of these, $\mathcal{S}_p$, will yield a performance within the given controller requirements. The problem of elimination is therefore based on the set $\mathcal{S}_p \subset \mathcal{S}_s \subset \mathcal{S}_u$. A way to satisfy stability for the SMC is to apply the *sliding mode attractiveness condition*, which is related to the Lyapunov stability theory [81]

$$\dot{V} = \dot{s}s = \nabla_s^{\intercal}(x)\dot{x}s < 0 \tag{9.15}$$

This can be related to a quadratic Lyapunov candidate function given on the form $V(x) = \frac{1}{2}s^2$ with a derivative given as $\dot{V}(x) = \dot{s}s$ [82]. In order to satisfy this condition, the signs of s and $\dot{s}$ must be unequal. Expanding the inequality from (9.15) with the condition and substituting the field from (9.5) yields

$$\begin{aligned}
\nabla_s^{\intercal}(x)\dot{x}s &= \begin{bmatrix} K & 1 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} s \\
&= (\ddot{q} + K\dot{q})s = (\ddot{q} + K\dot{q})(\dot{q} + Kq) \\
&= \ddot{q}\dot{q} + \ddot{q}Kq + K\dot{q}^2 + K^2\dot{q}q \qquad \text{substitute (9.4)} \\
&= (\omega^2 q + u)(\dot{q} + Kq) + K\dot{q}^2 + K^2\dot{q}q \\
&= K^2\dot{q}q + K\dot{q}^2 + \omega^2 Kq^2 + uKq + \omega^2\dot{q}q + u\dot{q} < 0
\end{aligned}$$

A discontinuous control $u = -U_0 \operatorname{sgn}(s) + \omega^2 q$ is considered from (9.14), which can be substituted into the inequality from above as well as applying the definition given by $|s| = s \operatorname{sgn}(s)$ [82]

$$\begin{aligned}
K^2\dot{q}q + K\dot{q}^2 + \omega^2 Kq^2 + (\omega^2 q - U_0 \operatorname{sgn}(s))Kq + \omega^2\dot{q}q + (\omega^2 q - U_0 \operatorname{sgn}(s))\dot{q} < 0 \\
(K^2 + 2\omega^2)\dot{q}q + K\dot{q}^2 + 2\omega^2 Kq^2 - U_0|s| < 0
\end{aligned}$$

This kind of Lyapunov derivative indicates, that the chosen Lyapunov candidate may not be perfect, since it introduces a number of non-quadratic terms in q and $\dot{q}$. However, in this case the input signal can be the reason and can be altered to yield a

[1] A *control signal* is defined as a signal function derived on the basis of a specific *control law* structure with certain variable parameters (time-invariant)

simpler $\dot{V}$. This can be achieved by selecting $\dot{s} = U_0 \operatorname{sgn}(s)$, which can be related to a control signal on the form (suggested by [82])

$$u(t) = \underbrace{-\hat{f}(x,t) - K\dot{q}}_{\text{Removes } \dot{s} \text{ from } \dot{V}} \quad - \underbrace{U_0 \operatorname{sgn}(s)}_{\text{Actual control}} \tag{9.16}$$

because

$$\dot{s} = \ddot{q} + K\dot{q} = \hat{f}(x,t) + u(t) + K\dot{q} \tag{9.17}$$

with $\hat{f}(x,t)$ denoting the nominal system dynamics without uncertainties. It can be noticed, that the control signal appears in the first derivative of s, indicating a *relative degree* of $k-1$ (equal to one in this case). If the relative degree is more than one, a *higher-order SMC* must be applied, where the *twisting controller* is one type [57, 71]. The twisting controller uses both the s-function and the $\dot{s}$-function to generate a suitable control signal. Since $s = 0$ is a requirement to obtain stability and to track the desired reference (in this case regulate all states to zero), the condition $\dot{s} = 0$ is also a requirement. The first two terms of $u(t)$ are denoted as the *equivalent control* part $\hat{u}(t)$ of $u(t)$ and can be related to the control signal to obtain the Filippov solution [87]. Using again the system in figure 9.5 with $K = U_0 = 10$ (the original control law from (9.14) must be applied to show the point), the control signal and the corresponding equivalent control signal is shown in figure 9.10.

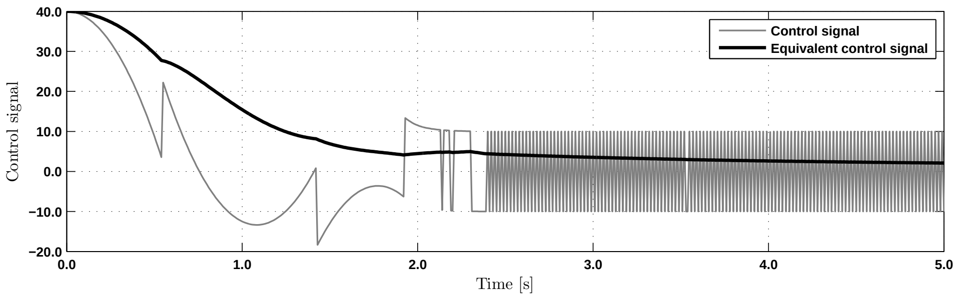


Figure 9.10: *Equivalent control signal for system in figure 9.5 (inspired by fig. 20 in [99])*

The mean of the control action is decreasing, because the control signal remains fluctuating around the time axis when in sliding mode. Therefore $\dot{V} = -U_0 \operatorname{sgn}(s) \leq 0$ for $U_0 \geq 0$ yields a stable control system. $K > 0$ is also required to achieve a decaying behavior $q(t) = q(t_r)\mathrm{e}^{-K(t-t_r)}$ on the sliding surface. All calculations are also performed on a nominal system $\hat{f}(x,t)$ without uncertainties and with a natural frequency of $\omega = 5$ rad/s.

Before determining the actual values of U_0 and K, the topic of chattering is considered in the following section, which is influencing the disturbance rejecting properties of the control system. However, reducing chatter also smooths the response. In the remainder of the control chapter, the nominal dynamics $\hat{f}(x,t)$ will be given as $\hat{f}(x,t) = -2\zeta\omega\dot{q} - \omega^2 q$ including the damping term $2\zeta\omega\dot{q}$.

9.2 Chattering issues

When using the discontinuous control signal proposed in (9.16), the bandwidth of the control signal is theoretically infinite. This is caused by the instant change in signal amplitude, which is advantageous to control nonlinear undermodeled systems with parameter uncertainties. Also, since the control function is amplifying the input s to the output $U_0 \operatorname{sgn} s$, the *equivalent gain* $K_{\text{eq}} \to \infty$. The *equivalent control signal* u_{eq} is the continuous output of the control block, and the gain $\frac{u_{\text{eq}}}{s} \to \infty$ as well [99]. *Invariance* to system disturbances is also possible using a controller with infinite gain [98]. Dynamics in every frequency range can be met in theory, but for this particular project it is a disadvantage, because it may excite resonances of the flexible tool.

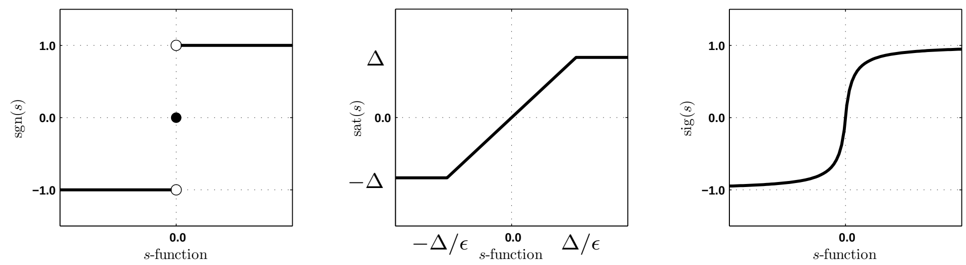
Chattering occurs when the control law is discontinuous or discrete. The discrete case is necessary for implementing the controller on hardware. Defining a chattering boundary layer in the $(q, \dot{q})$-plane called the *quasi-sliding mode band* (QSMB) and given as $\{x| - \Delta_Q < s < \Delta_Q\}$, relaxes the $s = 0$ condition [42, 87]. A band of $\pm\Delta_Q$ will then be placed around $s = 0$. By defining a limit of the chatter amplitude, the controller can be designed to fulfill this condition. Instead of introducing a boundary layer around the surface $s = 0$, a continuous control law can be applied. This will decrease the disturbance rejection properties and limit the bandwidth of the control signal. However, the smoother control signal rejects the chattering and increases the lifespan of the actuators.

The objective of the project is to damp flexible modes appearing in the manipulator tool, and a controller with infinite gain will undoubtedly excite every mode in the dynamics as well as introducing non-smooth control signals. To overcome this problem, the discontinuity must be substituted with a continuous or continuously differentiable function like e.g. sat(s) or sig(s), saturation and sigmoid function, respectively. The two signals are described in the following way [71, 87]

$$u(x) = \hat{u}(x) - U_0 \operatorname{sat}(s, \epsilon_s) = \begin{cases} \hat{u} - U_0 \frac{s}{\epsilon_s} & \text{for } |s| < \frac{\Delta_s}{\epsilon_s} \\ \hat{u} - U_0 \operatorname{sgn}(s) & \text{otherwise} \end{cases} \tag{9.18}$$

$$u(x) = \hat{u}(x) - U_0 \operatorname{sig}(s, d_s) = \hat{u} - \frac{s}{|s| + d_s} \tag{9.19}$$

with $\hat{u}$ denoting the *equivalent control*, which is determined from an ideal system description $\hat{f}(x, t)$. The Δ_s-parameter is only applied when using the sat(s) function, whereas the d_s-parameter is used by the sig(s) function to smooth out the control signal as well. The ϵ_s-parameter is a saturation gain, which for the purpose of this project will be set to unity $\epsilon_s = 1$. Figure 9.11 shows the discontinuous control from (9.16) as well as the newly proposed *boundary layer method* (saturation function) and *sigmoid function method* to remove chattering [71].



(a) *Discontinuous* $\mathrm{sgn}(s)$ **(b)** *Continuous* $\mathrm{sat}(s)$ **(c)** *Continuous* $\mathrm{sig}(s)$

Figure 9.11: *Discontinuous and continuous control functions*

The new methods are continuous and the sigmoid function is further continuously differentiable, which allows for a smoother control signal. Remark 3 states an important fact about continuous control functions.

Remark 3 (null space)**:** *When applying signal shapes different from the signum function, the sliding mode is no longer ideal [19]. As introduced earlier, a manifold* $s = 0$ *also implies* $\dot{s} = 0$ *in the ideal case. This can no longer be guaranteed when using other functions, and therefore* $s\dot{s} \leq 0$ *is only possible under certain conditions. However, the sliding mode will still be present, but the trajectory must be expected to overshoot the manifold at first before entering the expected sliding mode. No proof of the phenomenon will be given (see [19]) and will not be considered in the following. This remark is provided to inform about the trade-off between chatter suppression and controller performance.* □

In order to select one of the methods, the potential of each of the new methods must be evaluated. The concept of *describing function analysis* (DF-analysis) will be exploited to evaluate critical frequencies in the control signal, that may excite harmonics in the tool dynamics. All resonances must be avoided within the control signal spectrum. *Notations in this context are not to be confused with former notations, and the variables will be restored after the analysis. A separate nomenclature is given on page XX for this short analysis.* A basic illustration of the concept of describing function analysis is given in figure 9.12.

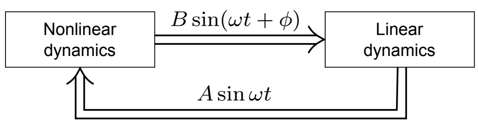


Figure 9.12: *Concept of describing function analysis*

It seems from the figure, that the only output from a nonlinear system is a sinusoid, but this is not the case. The nonlinear dynamics in this case will include a $\mathrm{sat}(s)$ or a $\mathrm{sig}(s)$ function, which is generalized by the two transfer functions $N(A, \omega)$ (change in amplitude) and $\varphi(A,\omega)$ (change in phase). Similar, the linear dynamics is described by $G(B,\omega)$ and $\psi(B,\omega)$. The response $B\sin(\omega t+\phi)$ of the nonlinear part is an approximation of the nonlinear dynamics using the first harmonic, because it is generally the most significant component of the entire frequency response. In order for a *limit cycle* to be stable, the following sufficient conditions must be satisfied

$$G(N(A,\omega),\omega) = A \quad \text{and} \quad \varphi(A,\omega)+\psi(B,\omega) = 0 \tag{9.20}$$

meaning that both amplitude and phase must be unchanged between periods. Stability in this context is given if the trajectory is moving along a *limit cycle* in the phase plane. In that case, the limit cycle acts as an *attractor* to the trajectory, making the closed-loop system critically stable in the case of no friction. When the response from the linear dynamics is fed to the nonlinear block in figure 9.12, the shape of the output is depending on the amplitude of the response signal. A small signal, which is below the saturation limit, will remain unchanged, whereas an increase in amplitude will make the output more distorted [2]. The nonlinear description must be in the form of a describing function, which for the saturation function is given as [76, 85]

$$N(A) = \begin{cases} \frac{1}{\epsilon_s} & \text{for } A \le \Delta_s \\ \frac{2}{\pi\epsilon_s}\left(\operatorname{asin}\frac{\Delta_s}{A} + \frac{\Delta_s}{A}\sqrt{1-\left(\frac{\Delta_s}{A}\right)^2}\right) & \text{for } A > \Delta_s \end{cases} \tag{9.21}$$

where A is the gain of the input sinusoidal given from $A\sin\omega t$. The parameter Δ_s is the saturation limits and $1/\epsilon_s$ is the gain of the saturation function (linear part of the function), see (9.18). Notice how the frequency dependence has been removed from the describing function approximation (not a general property). A plot of the function is given in figure 9.13 as a function of the input gain A. Remaining parameters are selected to be $\Delta_s = 1$ and $\epsilon_s = 1$ for that particular plot. The mathematical theory behind the transformation of the sat()-function is not given, but the general transform that must be applied is expressed as [16]

$$N(A) = \frac{\omega}{A\pi}\int_0^{\frac{2\pi}{\omega}} u(t)\sin\omega t \,\mathrm{d}t + \mathbf{j}\frac{\omega}{A\pi}\int_0^{\frac{2\pi}{\omega}} u(t)\cos\omega t \,\mathrm{d}t \tag{9.22}$$

with $u(t)$ denoting the response of the nonlinear control block. A similar process can be applied to the sig(s) function, which yields the describing function [71]

$$N(A) = \frac{2}{A\pi}\left(2-\frac{d_s\pi}{A}\right) + \frac{4\left(\frac{d_s}{A}\right)^2}{A\pi\sqrt{\left(\frac{d_s}{A}\right)^2-1}}\left(\frac{\pi}{2} - \operatorname{atan}\frac{1}{\sqrt{\left(\frac{d_s}{A}\right)^2-1}}\right) \tag{9.23}$$

For comparison, the describing function of sig(s) is shown together with sat(s) and sgn(s) in figure 9.13. The values used are $\Delta_s = 1$ (absolute limits), $d_s = 1$ (shape of sig(s)) and $\epsilon_s = 1$.

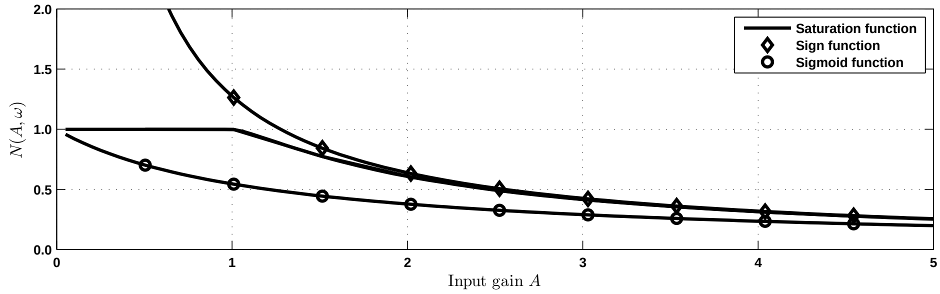


Figure 9.13: *Comparison of describing functions for* $\mathrm{sat}(s)$, $\mathrm{sgn}(s)$ *and* $\mathrm{sig}(s)$

The plots are generated using the MATLAB command `frestimate.m`, which is able to estimate the output amplitude of the nonlinear system part when excited by a sinusoidal input signal with amplitude A. The reason why regular frequency analysis techniques does not apply to nonlinear systems is the possibility for the frequency characteristics to depend on both frequency and amplitude of the input signal. A linear filter excited by a sinusoidal signal will still output a sinusoidal with the possibility of amplification and/or change in phase. A nonlinear filter can on the other hand alter the signal input in different ways, which is why the filter must be approximated to apply frequency analysis techniques. Both $\mathrm{sat}(s)$ and $\mathrm{sig}(s)$ were proposed as candidates for removing chattering around $s = 0$. Compared with the discontinuous $\mathrm{sgn}(s)$-function, the two functions are both continuous as well as defined for all positive gains A. The *Barkhausen criterion* given as [76]

$$G(\mathbf{j}\omega)N(A) + 1 = 0 \tag{9.24}$$

can be used to evaluate both functions in a closed-loop with the system. The frequency ω and the gain A can be determined for both the linear and the nonlinear model part. Figure 9.14 shows a *modified Nyquist plot* for the describing functions of $\mathrm{sat}(s)$ and $\mathrm{sig}(s)$, which turns out to be equal when considering the entire range of A.

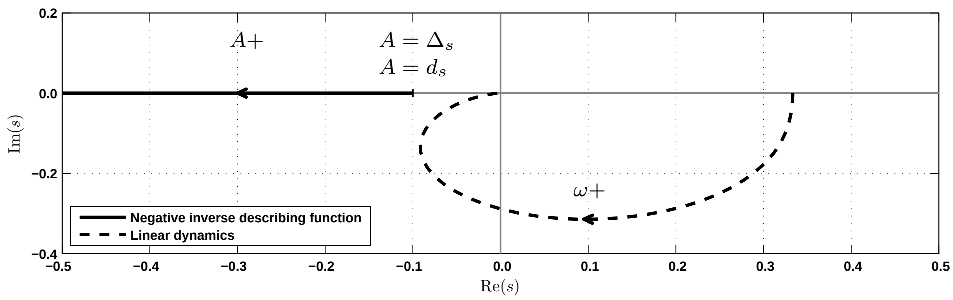


Figure 9.14: *Modified Nyquist plot of* (9.24) *(inspired by fig. 4 in [85])*

The difference is the specific location of the describing function in the Nyquist plot for a given A. Both the describing function $N(A)$ and the linear dynamics $G(\mathbf{j}\omega)$ from the Barkhausen criterion are shown in the figure. Notice how the Δ_s, d_s-parameters determine the distances from origo to the starting points of the responses.

If the two graphs intersect, the specific points will provide information about frequency and gain of *self-sustained oscillations*, that defines a limit cycle [85]. In this case the two functions are not intersecting, indicating that the s-signal does not experience chattering, because the system trajectory is not attracted to a limit cycle. The derivations above are based on a purely theoretical case, and due to discretization when the controller is implemented on hardware the s-signal will eventually fluctuate around $s = 0$. This frequency, however, will be a function of the sampling frequency, which is then chosen in a range above the first resonance frequency. The describing function of the signum function has the limit

$$\lim_{A \to 0} N_{\mathsf{sgn}}(A) \to \infty \tag{9.25}$$

causing the inverse function $-1/N_{\mathsf{sgn}}(A)$ to reach (0,0) in the complex plane. An intersection occurs between the two functions in the modified Nyquist plot at the point $(A, \omega) = (0, \infty)$, which corresponds to an ideal sliding mode [85]. There is, however, no analytical way to determine if chattering will occur or not because the point is undefined.

Based on the fact that the saturation and the sigmoid functions are both capable of rejecting the chattering effects, though with a trade-off in closed-loop performance, they are both applicable. However, for the sequel of the controller design process, the saturation function will be used. Partly because the saturation describing function is less complex, and because the system to be controlled may be assumed undermodeled, since only the first mode shape has been included in the dynamics (the saturation provides a better performance). The final sliding mode controller will therefore consist of a saturation function with the s-function as the argument.

Instead of solving the equations in (9.20) analytically, the modified Nyquist plot can be applied, and the solution can be derived graphically. It has been established, that chattering can be removed in the theoretical case by using a saturation function in the control law. When implementing the controller, the sampling frequency must be selected in such a way, that it does not coincide with the eigenfrequencies of the flexible tool.

Even though the DF-method is an approximation it can be used to estimate the response of a nonlinear closed-loop in the frequency domain [17]. According to [87], the describing function method may prove to be failing in cases with linear dynamics resonances. The results must therefore be compared with measurements to confirm the correctness of the estimations. If the response must be more accurate, *Tsypkin's method* can be applied [17], because it does not approximate the dynamics before analysis. The theory behind this method is not within the scope of this project, and therefore the method will not be further described. The describing function methods will be applied for this project. Before the controller can be applied in a control loop, a number of controller parameters must be selected, which is the topic of the sequel section regarding controller performance.

9.3 Controller performance

When considering controller performance, this project requires the settling time of the strain response state $q(t)$ to be reduced when compared to a system without oscillation damping. The exact requirement as given in section 2.4 is $t_{\mathsf{set}} < 5$ seconds. Other performance measures can be used, but in this case, requirements are only stated for initial location and destination. The behavior in between points is not a concern as long as the tool is not brought to extreme positions that may damage it. According to assumption 10 only local control is needed when the tool is near the destination.

Imagine the response from a controller built on the assumption of a rigid link between the tool frame $\mathbb{F}_\tau$ and the end-effector frame $\mathbb{F}_e$, even though flexible. The flexible link will oscillate after the end-effector has been located at the assumed correct spatial position, causing a failed operation. Including the flexible behavior of the link into the controller allows the strain of the link to be included as a model state, which can then be controlled. The state will be regulated to zero when taking account for the gravity offset.

Because the flexible tool may experience *deformation*, an algorithm must be constantly estimating the deformation of the tool to prevent ineffective control action. The term deformation is used to describe the behavior of a flexible beam not recovering to its initial shape after being bended due to applied force. A constant deflection will then be present in the strain measurement signal. This is, however, not within the scope of this thesis, but is necessary if the controller is to function for longer periods of time. An illustrative strain response is given in figure 9.15 when moving the end-effector from one steady-state position to another. The figure is also part of the technical requirements in section 2.4.

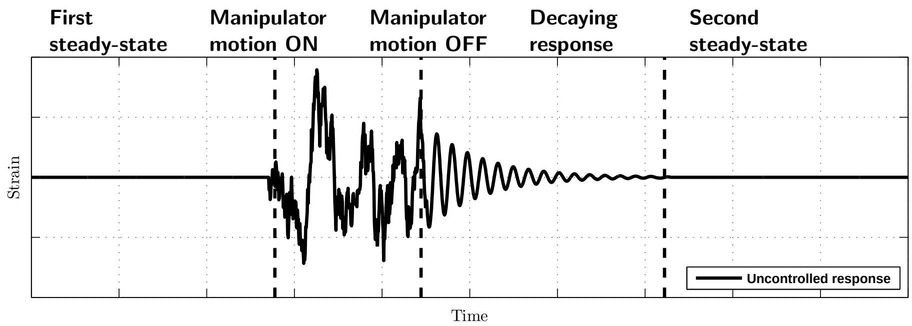


Figure 9.15: *Arbitrary strain response when moving end-effector*

The strain response between manipulator on and off mode is not to be controlled, but only the oscillation after the manipulator has stopped moving. By applying a controller to the system, the response can e.g. look as illustrated in figure 9.16.

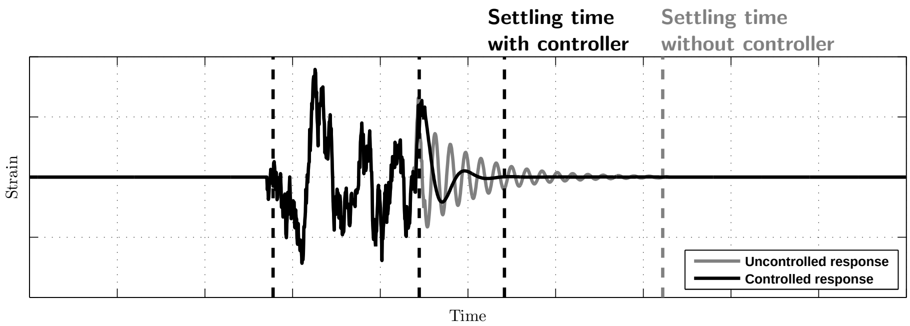


Figure 9.16: *Arbitrary strain response when moving end-effector + damping controller*

The oscillations are damped and the settling time is improved. When determining the performance factor of the controller, the settling time t_{set} is the measure. Even though the term settling time originally is given as the time from a step function is applied until the response is less than a certain percentage of the step reference, the following analog definition is used instead.

Definition 2: *The settling time for a system controlled with a sliding mode controller is given from two different time intervals t_r and t_s based on two types of trajectory motions: reaching phase and exponential convergence phase (sliding mode) [87], respectively.* □

Using definition 2, a settling time equivalent can be determined. This is bounded from above unlike the exact value determined for linear systems. The reaching phase time t_{r} is depending on the initial conditions of the trajectory, and is bounded by [87]

$$t_r \leq \frac{|s(x,0)|}{U_0} = \frac{|\dot{q}(0) + Kq(0)|}{U_0} \tag{9.26}$$

where $s(x,t)$ denotes the time dependent surface function, which can be expressed as

$$s(x,t) = s(q,\dot{q},t) = \dot{q}(t) + Kq(t) \tag{9.27}$$

when remembering that x was used as state vector containing both q and $\dot{q}$. The second time interval is determined from the sliding mode phase, which is constructed to generate the $q(t) = q(t_r)\mathrm{e}^{-K(t-t_r)}$ response. A decay time t_s in sliding mode can therefore be determined as

$$t_s = -\frac{\ln(0,05/q(t_r))}{K} \tag{9.28}$$

if assuming steady state when $|q(t)| \leq 0,05q(t_r)$. The settling time can therefore be guaranteed to be bounded in the following way

$$t_{\mathsf{set}} \leq t_r + t_s = \frac{|\dot{q}(0) + Kq(0)|}{U_0} - \frac{\ln(0,05/q(t_r))}{K} \tag{9.29}$$

which must respect $t_{\mathsf{set}} < t_{\mathsf{nom}}$ to show a damping effect. t_{nom} is the nominal settling time without controller. Table 9.2 at the end of this section lists the all-important parameters, that are to be either selected based on intuition or measured by empirical experiments. The parameters of the table are used for the final project evaluation, which will make sure that the specification requirements are met for this project.

Based on the eigenfrequency measurements from appendix A, the eigenfrequency of the tool with the small tip mass ($m_l = 351$ g) is measured to be 4,20 Hz with a damping ratio of 0,0023. This corresponds to a settling time of $t_{\mathsf{nom}} = 64{,}4055$ seconds, which must be reduced to $t_{\mathsf{set}} < 5$ seconds. With $U_0 = 10$ selected on the basis of hardware limitations (see table 9.2), the only unknown is the gain K, which can be determined from (9.29) as

$$\begin{aligned}
&U_0 K t_{\mathsf{set}} + \ln(0,05/q(t_r))U_0 < |\dot{q}(0) + Kq(0)|K \Downarrow \text{ (assuming } \dot{q}(t_r) = 0,\ q(t_r) \geq 0)\\
&- q(0)K^2 + U_0 t_{\mathsf{set}} K + \ln(0,05/q(t_r))U_0 < 0 \Downarrow\\
&- q(0)K^2 + 322,027K - 10\ln(0,05/q(t_r)) < 0 \qquad (9.30)
\end{aligned}$$

with $q(0)$ and $q(t_r)$ given by definition or from direct measurement when the SMC is activated. By assuming that $\dot{q}(t_r) = 0$, the controller is activated when the strain derivative changes sign or formally when $\dot{q}(t_r) \neq 0$. This value can though also be measured on SMC activation. The theoretical value of t_{set} can be calculated if the point $q(t_r)$ is known. Unfortunately, $q(t_r)$ is difficult to determine without simulation, and an upper limit will be used instead. This will be explained in definition 3.

> **Definition 3:** *In order to determine the theoretical settling time, the initial condition for the sliding mode $q(t_r)$ must be used to determine the time interval t_s. By assumption, the controller will ensure that the state trajectory $q(t)$ is strictly decreasing and therefore, by definition, $q(0) > q(t_r)$ when assuming that $t_r > 0$.* □

A direct substitution of $q(t_r)$ with the initial condition for the reaching phase $q(0)$ is therefore possible, and the inequality of (9.30) will still hold. Figure 9.17 provides a graphical interpretation of the quadratic inequality (9.30) given that $q(0) = 1$ and $U_0 = 10$. The gray area shows the region of feasible K, whereas the dashed gray line indicates the upper limit of the settling time of 5 seconds.

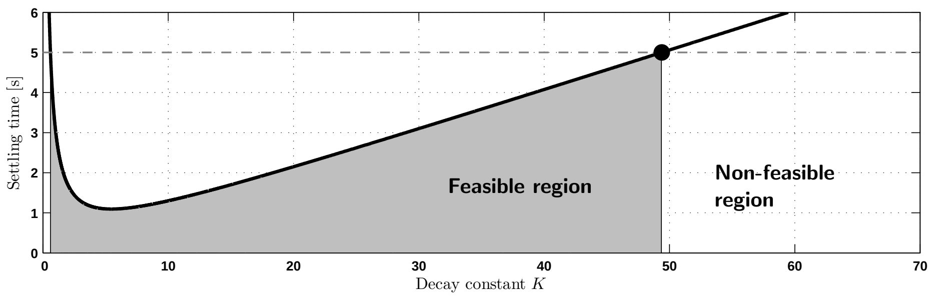


Figure 9.17: *Graphical interpretation of (9.30) with $q(0) = 1$ and $U_0 = 10$*

The settling time is below the requirement when $0{,}6065 < K < 49{,}3935$. The dot in the figure is one solution to (9.30), and the solution can be explicitly expressed as

$$K_{\text{sol}} = \frac{U_0 t_{\text{set}} \mp \sqrt{U_0^2 t_{\text{set}}^2 + 4q(0)\ln(0,05/q(t_r))U_0}}{2q(0)}$$

To satisfy the inequality, the selected K must respect the limits

$$\min(K_{\text{sol}}) < K < \max(K_{\text{sol}}) \tag{9.31}$$

By expressing the global minimum of (9.29) using the assumptions from the derivation of (9.30), a minimal settling time can be achieved. The derivative of (9.29) is given as (and equated to zero)

$$\frac{\mathrm{d}}{\mathrm{d}K}\left(\frac{Kq(0)}{U_0} - \frac{\ln(0,05/q(t_r))}{K}\right) = \frac{q(0)}{U_0} + \frac{\ln(0,05/q(t_r))}{K^2} = 0$$

It is therefore necessary to select a K that satisfies

$$K = \sqrt{-\frac{U_0\ln(0,05/q(t_r))}{q(0)}} \tag{9.32}$$

which will give the following settling time when substituted into (9.29)

$$t_{\mathsf{set}} \leq \sqrt{-\frac{U_0\ln(0,05/q(t_r))}{q(0)}}\frac{q(0)}{U_0} - \ln\left(\frac{0,05}{q(t_r)}\right)\left[-\frac{U_0\ln(0,05/q(t_r))}{q(0)}\right]^{-1/2} \tag{9.33}$$

This will be considered the theoretical selection of K, and can be added to table 9.2 listing all necessary parameters for the sliding mode controller. A number of simulations have been made with different controllers (simulation running at 100 Hz), and the results are given in table 9.1 on page 9.1.

A saturation limit of U_0 has been selected because the maximum control signal amplitude is 10 V, which is limited by hardware. The differing from the theoretical value can be explained by the fact, that (9.32) defines an upper limit of t_{set}, and is only based on the discontinuous control law. All values are plotted on their respective functions in figures 9.18 (*with* overall control signal saturation) and 9.19 (*without* overall control signal saturation). A close-up of figure 9.19 is given in figure 9.20

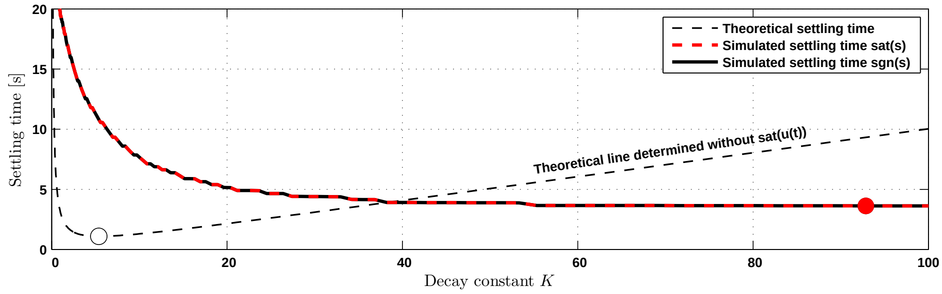


Figure 9.18: *t_{set} from simulation with different K with* $\mathrm{sat}(u(t))$

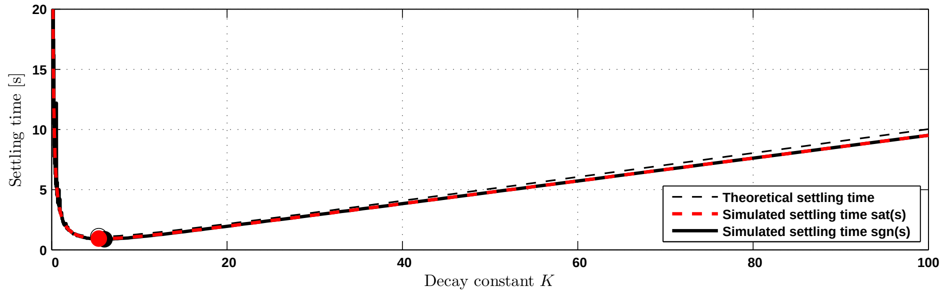


Figure 9.19: *t_{set} from simulation with different K without* $\mathrm{sat}(u(t))$

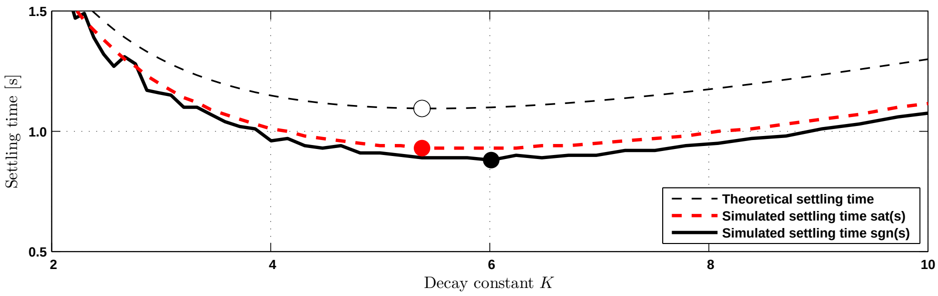


Figure 9.20: *t_{set} from simulation with different K without* $\mathrm{sat}(u(t))$ *(close-up)*

Whenever a saturation function is applied to the control law, the controller performance is reduced, and the settling time is increased when compared with the discontinuous control law. Figure 9.21 compares the state trajectory for both open-loop and closed-loop. Both controller types are simulated, and overall saturation has been added to resemble the practical configuration.

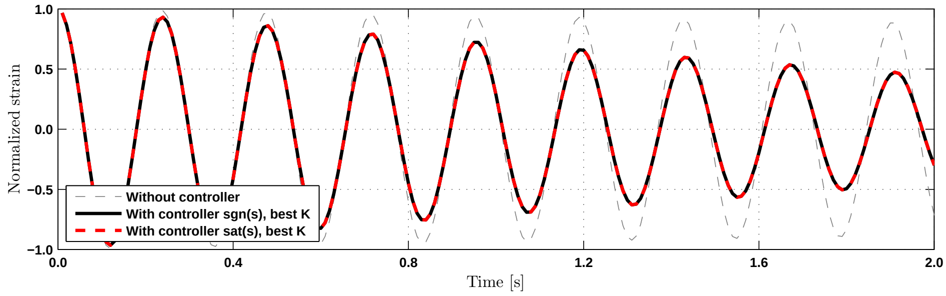


Figure 9.21: *Control system state trajectory from simulation with best K*

The trajectories are close to each other, but the s-function is smooth in the case with saturation, as shown in figure 9.22 and 9.23 (close-up of chatter).

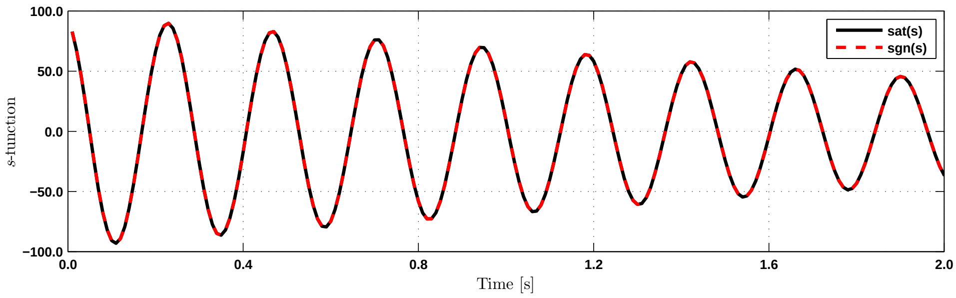


Figure 9.22: *s-function from simulation with best K*

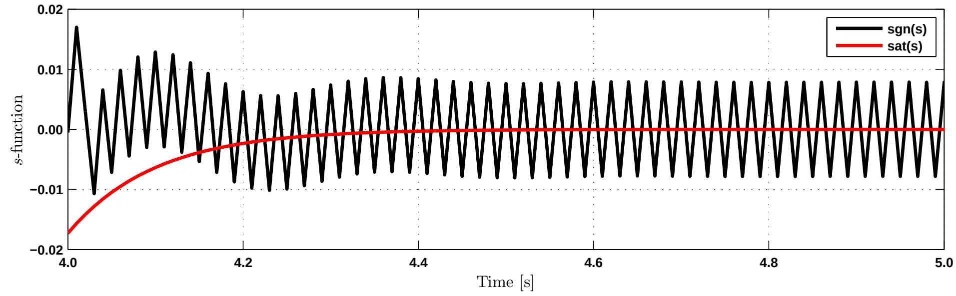


Figure 9.23: *s-function from simulation with best K zoomed in on chattering*

Since the simulation/controller is running at 100 Hz, the chattering of the discontinuous control will be of half that frequency, namely 50 Hz. Because the chatter phenomenon can be omitted when using the sat(s)-function in the control law, this is chosen over the discontinuous control law. The above simulations are executed at the same frequency as the controller. Because the simulation does not update in between control samples, some dynamics may be left out of the simulation.

With overall control signal saturation (discontinuous control $\mathrm{sgn}(s)$)	$u = \mathrm{sat}\left[\hat{u} - U_0\,\mathrm{sgn}(s), U_0\right]$ $K = 92{,}869$ $t_{\mathsf{set}} = 3{,}62$ s
With overall control signal saturation continuous control $\mathrm{sat}(s)$)	$u = \mathrm{sat}\left[\hat{u} - U_0\,\mathrm{sat}(s, \Delta_s), U_0\right]$ $K = 92{,}869$ $t_{\mathsf{set}} = 3{,}62$ s
Without overall control signal saturation (discontinuous control $\mathrm{sgn}(s)$)	$u = \hat{u} - U_0\,\mathrm{sgn}(s)$ $K = 6{,}013$ $t_{\mathsf{set}} = 0{,}88$ s
Without overall control signal saturation (continuous control $\mathrm{sat}(s)$)	$u = \hat{u} - U_0\,\mathrm{sat}(s, \Delta_s)$ $K = 5{,}381$ $t_{\mathsf{set}} = 0{,}93$ s
Settling time without controller	$t_{\mathsf{set}} = 64{,}41$ s
Theoretical t_{set} with (9.13) controller	$K = 5{,}38$ $t_{\mathsf{set}} \leq 1{,}10$ s

Table 9.1: *Optimal K-values and corresponding settling times based on simulations*

To investigate whether or not the different frequencies may influence the chatter frequency, figure 9.24 shows the chatter from figure 9.23 as well as the chatter from a 10 kHz simulation (maintaining 100 Hz controller frequency).

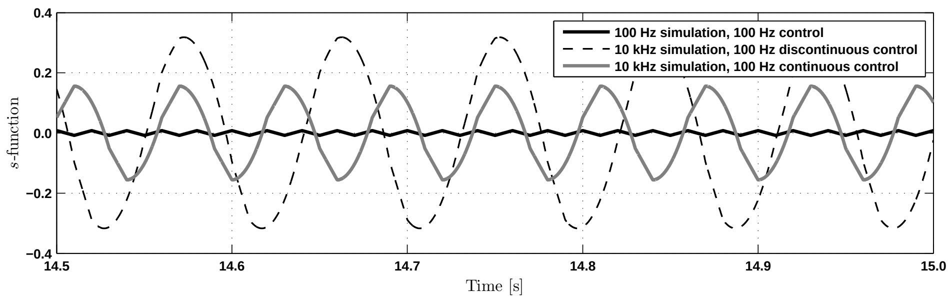


Figure 9.24: *s-function from simulations using different frequencies*

The new chatter frequency is 11,14 Hz, when using the discontinuous control law and shows sinusoidal behavior because it is sampled at a higher frequency. During the nonlinear frequency analysis, no frequency was revealed at 11,14 Hz. This shows some of the downsides of using describing function analysis. Limit cycles may appear in reality (in this case found from simulation) even though they are not determined from the analysis [87]. Using the saturation control function $\mathrm{sat}(s)$ will in theory remove the chattering issues, but when using a discrete version of the controller it reappears. The controller update frequency is affecting the frequency of the chattering, but when using saturation, no explicit coupling has been determined.

When applying the saturation control law (which has been selected as the final method), a distorted sinusoidal signal with a significant frequency component of 16,63 Hz appears. This is close to the 4,20 Hz resonance frequency of the flexible tool, but is assumed to have no significant effect on the controller performance. The resonance peak has a gain of -10,1 dB whereas the gain at the 16,63 Hz chatter frequency is -80,19 dB. The low damping of the system makes the resonance very dominating, and it can be difficult to excite this particular harmonic with other frequencies.

It is clear from the chatter frequency analysis, that even though the DF-analysis indicates a chatter-free s-function, the introduction of sampling will make it appear anyway. Therefore, simulations and practical experiments must reveal the true nature of the control system before making conclusions. Based on the simulations from above the final control law is given from (9.18) with the saturation function $\text{sat}(s)$ and the overall saturation. All controller types introduce chatter, but the magnitude is lower for the saturation controller. This is, however, based on the assumption that the lower chatter frequency is not too close to the resonance frequency. The controller can be expressed as

$$\boxed{\begin{aligned} u &= \text{sat}\left[\hat{u} - U_0\,\text{sat}(s,\Delta_s), U_0\right] \\ &= \text{sat}\left[2\zeta\omega\dot{q} + \omega^2 q - K\dot{q} - U_0\,\text{sat}(s,\Delta_s), U_0\right] \end{aligned}} \tag{9.34}$$

Table 9.2 lists the necessary values for constructing the final controller based on theoretical values. The empirical values have been filled in from experiments in appendix D for comparison.

	Equation	Theoretical	Empirical
Saturation limit Δ_s	-	-	1 (selected)
Decay constant K	$\sqrt{-\frac{U_0 \ln(0{,}05/q(0))}{q(0)}}$	92,869	10
Input gain U_0	-	10	10

Table 9.2: *Selected parameters for the sliding mode controller*

The input gain is selected on the basis of the maximum servo amplifier amplitude, when assuming unity gain for the remainder of the system. Selection of the saturation limit $\Delta_s = 1$ is based upon the measure a maximum measurable strain of $\sim \pm 9$ V. Using the voltage representation was clarified in section G.2 describing the sensors. From practical experiments, a strain measurement of 1 V has been found useful. However, according to figure 9.24, the chatter amplitude is bounded by the band $\sim \pm 0{,}2$ V, which is 5 times narrower than the selected Δ_s-band.

A discrete version of the SMC (DSMC [40]) will be the result of the implementation process. The QSMB, defining the discrete sliding surface band, can be bounded in the discrete case in the following two ways [42, 99]

$$|s_k| < T_s U_0 \triangleq \Delta \qquad\qquad |s_k| \approx T_s U_0 \triangleq \Delta \tag{9.35}$$

with s_k denoting the k-th sample of the discrete sliding surface description. The first one is derived from applying the method from [42] to the definition of the discrete *reaching law* in this project, while the other is a discrete version of the one from [99]. Both cases are amplified by U_0. This is a similar bound, when using the discontinuous control, with the chattering generated from discretization in this case. The width of the boundary layer is therefore determined from the sample period.

When $T_s \to 0$, the sliding surface $s_k = 0$ will appear, and the bound $\Delta \to 0$. Even though a control law featuring the saturation function will remove the chattering issue in the ideal case, the discretization will naturally introduce switching across the surface. By selecting a high sample frequency will limit the bound of the chattering, and thus the accuracy of the controller performance. When using saturation functions, however, the limit must be as high as possible to avoid unexpected frequency components, which the DF-analysis is not able to detect.

The process of making a DSMC is not given explicitly, because the LabVIEW environment is based on a graphical representation of the controller. The diagram is therefore automatically compiled into executable code, which is running on the cRIO controller unit. Figure 9.25 shows the structure of the controller, which must be implemented in LabVIEW. Rhe SMC-block contains the control law from (9.34). The block is not only requiering the s-function input, but uses ω^2, $2\zeta\omega$ and $K\dot{q}$ as well. This ends the controller design part, and the following part evaluates the performance of the entire control system.

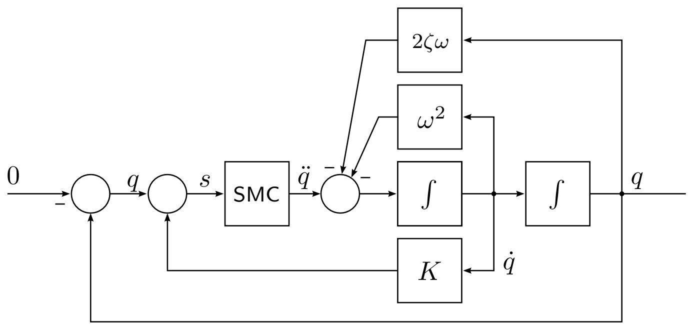
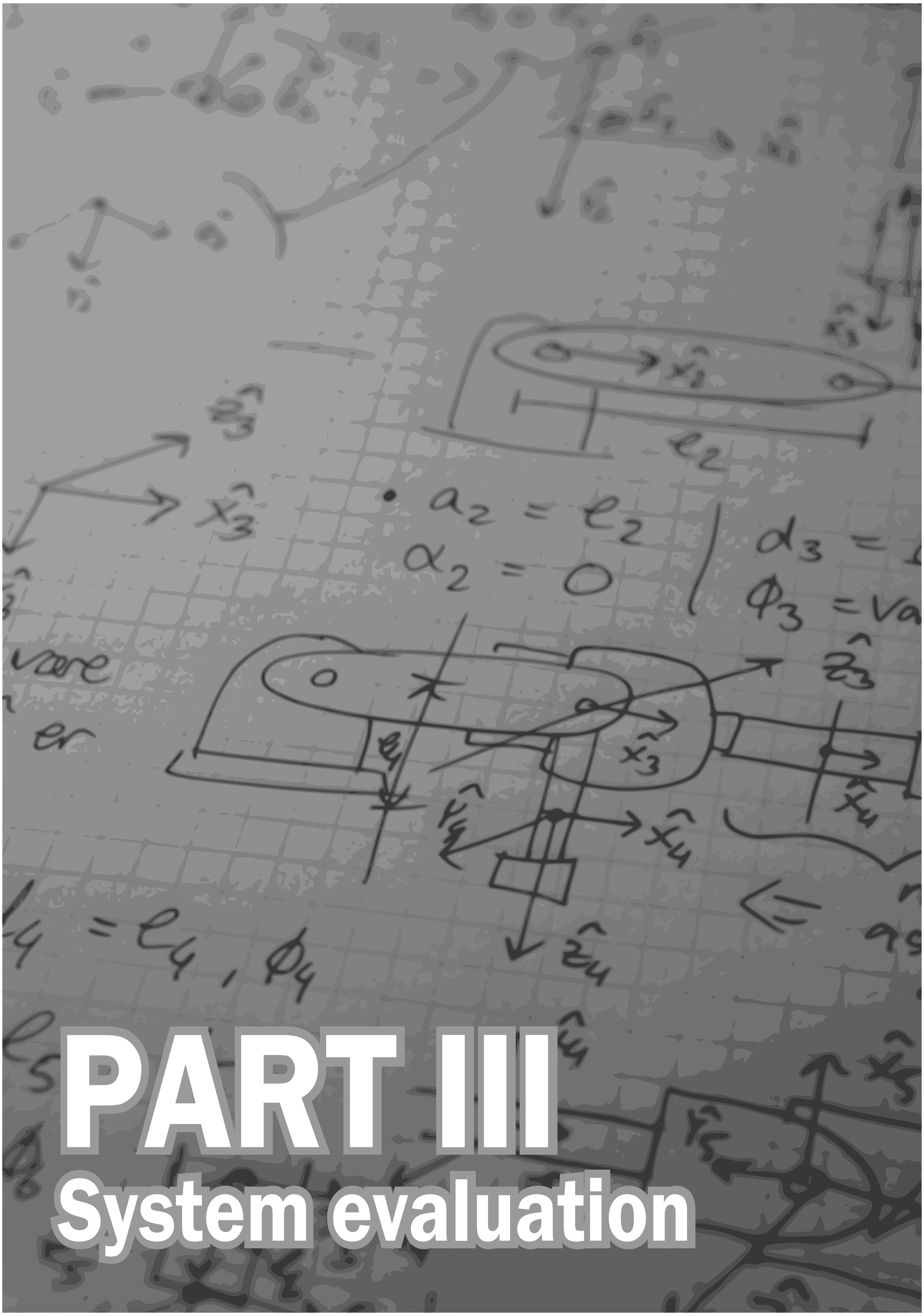


Figure 9.25: *Sliding mode controller diagram*

PART III
System evaluation



Chapter 10
Acceptance test

This project is composed from a number of different parts each requiring different methods and techniques. Each part must be individually evaluated to assess, if they fulfill the project requirements from the project description in chapter 2. Afterwards, an overall evaluation will estimate the capabilities of the entire product and measure its performance. One question must be answered, and that is whether or not the product is capable of improving the accuracy of robotic manipulator end-effector control while reducing operating time. The exact definition of the project objective was formulated in the project description in chapter 2 and is repeated below. An answer must be given in order to successfully complete this project.

> *How to improve end-effector position/orientation control of industrial robotic manipulators in terms of accuracy and operating time when handling flexible tools?*

Based on this question, a number of requirements were outlined in the project description as well, which will be verified at the end of this chapter. As part of the overall validation of the project, the requirements must be respected by the applied methods and verified through experiments.

A number of methods and requirements were listed in the project description, and were selected in order to solve the problem of improving manipulator control. System identification, sensor fusion and prediction are amongst the methods, and model complexity, controller performance, kinematics and controller structure are elements of the requirements. An acceptance test is given in the sequel, which is designed to test all requirements from section 2.4 and verify the use of required methods. The performance of the different methods will be analyzed afterwards on the basis of the results from the acceptance test, and the requirements will be verified at the end.

10.1 Acceptance test description

A final acceptance test will verify the performance of the product as a complete unit, and a subsequent requirement verification will investigate whether or not the requirements from section 2.4 are respected. The test is based on the question about improving manipulator control, and the methods outlined throughout the report will be applied. Listed below are the three major elements forming the final product as was described in chapter 2.

- System identification (relay tuning)
- Sensor fusion (linear Kalman filter)
- Control (sliding mode controller)

The different elements form an autonomous system. It is capable of adapting automatically to changes in tool and/or load configuration and update the controller to damp tool oscillations under these conditions. Figure 10.1 shows how the different elements are used and when.

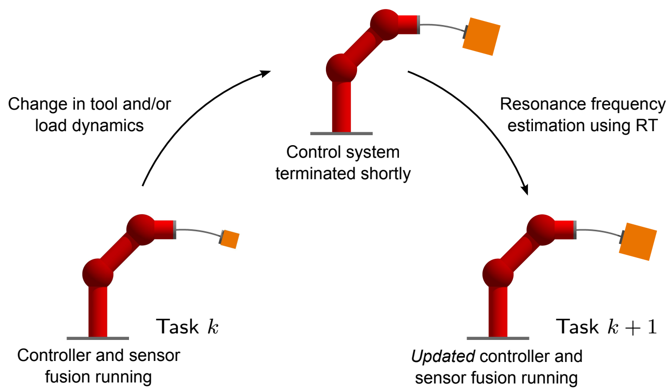


Figure 10.1: *Operation of final product with configuration change between tasks*

The controller and the sensor fusion process are running until the tool and/or load is changed. Then the relay tuning method is used to estimate the resonances of the new configuration, and the control system is started again afterwards with this new information. A short delay will be present between the tasks, because the RT-method is required to work in its own closed-loop configuration. The actual delay of the tuning is not part of the requirements for this project, since the throughput of the production has not been included in the scheduling.

Because the relay tuning method was performing better than the extended Kalman filter for the purpose of estimating the first resonance frequency of a flexible tool, the identification method does not need to be tested in parallel with the controller. The experiments already carried out in appendix B show the capabilities of the method, and a reference will be given to those at the end of this chapter.

Third element on the list is the sensor fusion method used to improve the model estimates. However, a significant amount of measurement noise was detected on the IMU sensor channels, and the method is therefore not going to improve the estimates by using all sensors. Only the strain and the joint angle measurements will be used. Due to the LabVIEW software being unable to execute regular programming languages, the Kalman filter is not implemented onto the cRIO. This will make the actual results differ from a case with the filter implemented. Instead the filter is tested on its own using measurement data from the acceptance test.

This brings it down to the controller, which can now be tested on its own using a constant model. The performance of the controller is therefore decoupled from the remaining methods, making it easier to evaluate different parts of the product. If all methods are applied in parallel, their performance cannot be measured individually. The controller will be using the model from (5.6) and the parameters measured in appendix A. The controller performance is measured by activating the control algorithm after the tool has been brought into oscillation, and the settling time and accuracy is measured. Appendix F describes the experiment in details, and contains the sensor fusion evaluation as well. Next section will summarize the acceptance test results from experiments in appendix B and F.

10.2 Acceptance test results

The final product is based upon three different elements listed in the beginning: system identification, sensor fusion and controller. Each of the elements has been tested and will be evaluated in the following three subsections. Before proceeding, two important remarks must be stated.

> **Remark 4** (strain gauge deformation)**:** *The flexible tool was deliberately constructed to have low frequency eigenfrequency components to emphasize the damping effect of the designed controller. This unfortunately showed to be a disadvantage, when applying some of the methods methods for system identification. The tool was deflected beyond the capabilities of both beam and strain gauges, especially when oscillating at the resonance frequency. This caused the tool and gauges to permanently deform. A few tools were constructed, but since the tests require the control signals with given amplitude and frequency content the deformation was unavoidable. A control signal limit below* U_0 *must therefore be given to avoid deforming the tool during the tests.* □

Figure 10.2 shows how the strain gauges becomes permanently deformed due to a combination of overload and excitation at the resonance frequency.

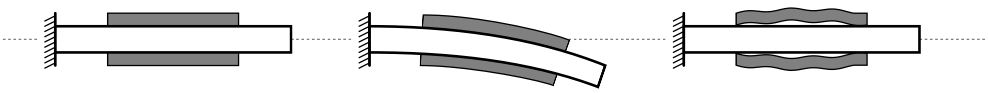


Figure 10.2: *State of strain gauge in nominal, overloaded and deformed configuration*

The point of overload is given from the maximum strain of the strain gauges in a given configuration regarding height of beam and location of the gauges on the beam. Even though the figure shows equal deformation of both gauges, the strain gauges on one side are likely to be stretched while the others are compressed. This causes the sensor output to saturate, resulting in a constant either positive or negative offset of the output signal. The sensitivity is further increased on one side and decreased on the opposite. Another issue is described in remark 5 on page 112.

> **Remark 5** (important note on equivalent control)**:** *It was discovered, that the control law from* (9.34) *was not applied in full during the simulations and the later controller implementation. The term* $-K\dot{q}$ *was not included in the equivalent control* $\hat{u}$ *and therefore not used in the remainder of this controller chapter. Some properties derived in the beginning of section 9.1 are therefore not obtained, but the controller is still of the sliding mode controller type though with a different performance than the one given in the sequel. The impact of the mistake will be evaluated in the conclusion in chapter 11.* □

Despite the two issues, the acceptance test was conducted. The first element to evaluate is the system identification.

10.2.1 System identification

First element on the list is *system identification.* A crucial part of a control system is knowledge about the system to control. The more accurate the model, the better the performance of the control system. For this particular project the system is divided into two parts: *manipulator model* and *flexible tool model.* The first model expresses the dynamics of the manipulator and has been assumed stationary in this case. Second model part is the tool model describing the strain/deflection of the tool.

Different system identification methods have been applied to solve the problem of identifying the missing parameters of the hardware configuration. All methods are based on the gray-box model approach with the model structure known in advance. Several methods were tested to see, which of the methods are best suited for solving different problems in the control system. The methods and their assumed advantages are listed as

Subspace method (N4SID): Estimation of model structure before PEM/EKF
Prediction error method (PEM): Offline estimation of parameters for full model
Extended Kalman filter (EKF): Online estimation of parameters for full model
Relay tuning method (RT): Offline estimation of parameters for tool model

All method are described in full in chapter 8 and tested by experiments in appendix B. The first method was described in the attempt to apply it for estimation of model structure before any other method was tried. However, as described in section 8.3 the method requires very accurate data in order to provide a useful model. It was therefore omitted, and a general model structure derived manually was applied instead. In order to estimate the parameters of the entire system before designing the control system, the prediction error method was investigated.

The full system model from (5.6) was used by the PEM algorithm together with empirical data from experiments. An immediate problem occurred when estimating the input gain of the sub-model describing the manipulator dynamics. The estimate was far from the measurements, but because the two sub-models (tool and manipulator) are decoupled it can be solved manually. This resulted in goodness of fits of around 80 %. When validating the results, however, the goodness measure was 76,66 % and 38,08 % for the strain and joint angle state, respectively. One reason for the low 38,08 % is caused by the randomized drifting of the manipulator link.

From the data alone the models appear decoupled, but in practice it seems that the tool is affecting the joint angle. Furthermore, a control signal of zero does not always correspond with a steady joint angle. Because these effects are not included by the model, the results are not perfect.

Because the PEM is solving an optimization problem, it is not suitable for realtime processing. It does not work within strict time intervals, and the results may not converge if the model is nonlinear. Another method will therefore have to be investigated for online estimating of system parameters. The extended Kalman filter was selected and applied the same data as the PEM. Because the method is working on input/output relations it is applicable for realtime control systems. The data is processed by the filter and the model is updated. The same model applied by the controller. This constructs an autonomous system, which is adapting to changes in the hardware configuration.

Unfortunately, when testing the filter it was clear that the accuracy and convergence time was not suited for systems with rapid changes in e.g. load or tool. The estimation time was slow and the parameters were fluctuating. Furthermore, the filter is nonlinear and convergence cannot be guaranteed. If the filter is to be used, it must be tuned to the specific application and the convergence time requirements must be relaxed. However, since divergence problems may occur the results must be supervised to avoid low performance.

As an attempt to compensate for the low performance of the EKF used for online estimation of parameters, the relay tuning method was investigated. Similar with the PEM it cannot be running along side with the control system. Also, the method requires to work in closed-loop on the system to function. Furthermore, it is only able to estimate the resonance frequency of the flexible tool. In this case the method is beneficial because the most important parameter is the resonance frequency. All other parameters must be estimated using e.g. PEM.

Because the method must be used in closed-loop, it must be applied every time the tool/load configuration changes or when the manipulator is waiting to complete the next task. From the experiments in appendix B, a convergence time of around 5 seconds was achieved depending on the control signal. An accuracy of around 8-11 % was achieved when compared to the measurements from appendix A. Based on the tested method, the PEM and the RT are preferrable for this type of system. The PEM can be used to estimate an overall model from a gray-box model structure, and the RT method will accurately estimate the resonance frequency alone to achieve better controller performance. The sensor fusion will be evaluated next.

10.2.2 Sensor fusion

The system identification procedures were all evaluated in the sequel subsection. They are used to adapt the model to the actual system to increase performance of the control system. Sensor fusion is also applied to improve the model estimates using direct measurements. Several sensors were applied to the system as described in section G.2. The sensors are listed below.

Strain gauges: Measure flexible tool strain q
Accelerometers (part of IMU): Measure linear acceleration of tool frame $\mathbb{F}_\tau$
Gyroscopes (part of IMU): Measure angular velocity of tool frame $\mathbb{F}_\tau$
Rotary encoders: Measure manipulator joint angles θ

By combining the information from the sensors and relating that information to model states, the states can be improved. A linear Kalman filter is applied to update the model estimates as a function of the difference between pure estimates and measurements.

Chapter 7 described the theory behind the filter and how it must be applied. Two matrices are crucial for the performance of the filter. They will have to be selected on the basis of process and sensor noise. The process noise is often unknown and cannot be measured. It is therefore necessary to tune the covariance matrix Q to achieve the expected performance from the filter.

The sensor noise is represented by the covariance matrix R and can be measured. In this case it is assumed that all sensors are decoupled, and the covariance matrix is therefore given as a diagonal matrix, because the sensor responses are uncorrelated. A noise variance was estimated for every sensor in section G.2, but they were not applied. It is difficult to achieve expected performance if only the process noise can be tuned. Furthermore, the sensors showed to be affected by noise magnitudes higher than the theoretical levels.

The origin of the noise was undiscovered, and the final results are therefore subject to increased noise variances. Appendix C describe the noise measurements on each sensor channel, and the noise levels increased by a factor of around 20 on most channels, when the power to the actuator drivers was turned on. Bad cable shielding and grounding may be the cause, but no further investigation has been conducted.

Only the IMU channels were too noisy to be used, and therefore only tool strain and manipulator joint angles was applied. The actual Kalman filter is not implemented onto hardware, but it has been tested using measurements from the controller performance experiment. Appendix D describe the experiment in details. Similar with the extended Kalman filter, the matrices Q and R have a high impact on the filter performance. The best matrices were found from trial and error.

The error between joint angle estimates and measurements is below 0,006 rad for the tested signal. A different behavior is seen for the strain estimates, which are converging to the measurements in around 3 seconds. By tuning the matrices even more, a better performance may be achieved. The Kalman filter will also filter sensor noise, because the measurements are not applied directly by are added together by the iterative process.

Even though the filter was not running in realtime it was shown that the filter is able to suppress the noise overlay of the measurements and improve the accuracy of the model estimates. Before implementing the filter in an autonomous system it must be tuned to converge faster, and the noise of the IMU must be limited to apply even more sensors. The accuracy of the model can be increased by relating different sensor types to the same model state. The last element to evaluate is the controller, which is given next.

10.2.3 Controller

The system identification and the sensor fusion methods has both been evaluated above. They are applied to improve the model estimates by adapting the model to the actual system through measurements. System identification is used to derive model parameters based on a set of measurement data whereas sensor fusion finds a compromise between measurements and estimates. In order to damp the oscillations of the flexible tool mounted at the end of the manipulator, it is necessary to apply a controller.

It was decided to design a sliding mode controller to achieve the damping effect. The controller type is robust to model uncertainties and is also working on nonlinear models. However, through model delimitations, the complete system model for this project turned out linear. The controller design is described in chapter 9. Basically, the control signal is designed to cancel out the natural response of the system and replace it with a new one. For this to work in practice the model must be very close to the actual one.

The experiment described in appendix D has been conducted to measure the performance of the controller. 5 different controllers and a single case without controller were tested. The uncontrolled case was used for comparison with the controlled cases. Of the 5 first cases, performance could only be measured on the first one. The remaining controllers did not damp the oscillations within the time frame or not at all.

Because only one test was successful, the remaining evaluation will be based on this alone. Several other experiments must be conducted to completely verify the performance of that particular controller configuration. The uncontrolled experiment showed a settling time of 50,85 seconds, whereas the controlled case managed to damp the oscillation within 3,85 seconds. This is an improvement of 92,43 %. However, small oscillations remain in the response which set a tip deflection tolerance level of 2,79 mm with a standard deviation of 0,49 mm.

Plotting a phase portrait of the strain trajectory shows an interesting fact. It was expected, that sliding mode would occur as soon as the trajectory reached the switching surface $s = 0$, but that does not seem to be the case. Even though the controller significantly increases the damping of the system it does not behave as a sliding mode controller. Two reasons can be explaining this behavior. First of all the use of a continuous control function (the one apart from the equivalent control $\hat{u}$) limits the performance because the control action is clipped at a certain level of magnitude. The term $-K\dot{q}$ was also forgotten within the controller algorithm. Simulation have shown, that the missing term makes the trajectory cross the switching surface a number of times before going into sliding mode.

Secondly, the overall control action has been limited as well to protect the tool and the hardware. Practical experiments showed, that a high amplitude control signal can damage the tool or the manipulator in case of error. Several tools were also constructed for the experiments due to this problem. Limiting the control action decreases the overall controller performance and the "softens" the gradient field. In this way the trajectory is not forced towards the equilibrium fast enough, and the trajectory is moving around the equilibrium point for a longer time.

Taking every part into consideration, the control system performs as required in section 2.4. Despite the tuning issues of the EKF, noise on IMU channels and unreached sliding mode due to hardware limitations and a missing controller term, the control system is capable of damping unwanted tool oscillations at the resonance frequency when tuned properly. A thorough conclusion to the project is given in the sequel chapter. The following section evaluates the methods and requirements from the project description in chapter 2.

10.3 Requirement evaluation

The project description was initiated with a list of methods necessary to form an autonomous system around the project formulation (question on page 4), and the methods were

- Load dynamics must be estimated from analyzing the response changes of the manipulator when compared to nominal response without a load
- Measurements from several sensors must be used together to improve the estimation of end-effector response when both affected and unaffected by a load
- Oscillations of flexible parts must be reduced by a controller

All the methods have been described throughout the report, and they have also been tested in the acceptance test. The methods are collected within the fields of system identification, sensor fusion and control, respectively. A full evaluation of each of the methods was given in section 10.2. Before the requirements were listed, a project delimitation was performed. This reduced the scope of the project.

It was stated that pick-up position, destination and process time window could have been provided for the controller, and it would automatically find the optimal control path. However, the trajectory of the manipulator was omitted, and only requirements to time can be given. This time is however fixed for the final controller.

After the project delimitation in section 2.3, a number of technical requirements requirements were listed for both manipulator model and manipulator control. The following requirements were given for the manipulator model

- The model must include a certain amount of complexity to allow a parameter estimation algorithm to fit system states, outputs and control signals to the model structure to provide the control algorithm with the best conditions for estimating the manipulator behavior and respect given controller requirements

- Generalized kinematics must be derived to allow for arbitrary manipulator location within a production cell as well as arbitrary workbench coordinates relative to a local coordinate system
- Dynamics of the flexible tool must be modeled in order to determine the TCP trajectory from strain measurements performed on the flexible tool

The complexity of the model has been reduced for both the manipulator and the tool dynamics. Only friction is remaining within the manipulator model due to the high gear ratio. The flexible tool is only described by one eigenmode because the frequency of the second mode of vibration is beyond the bandwidth of the manipulator link. An experiment, see appendix A, showed that the amplitude of the second eigenmode is much smaller than the first one, and it will not be a problem. So even after the model reduction, the model is complex enough to describe the behavior of the system. However, it does not include the gravity effect, that causes the joint angle to drift.

A generalized kinematic model was derived in chapter 3. Each part of the production cell has an attached frame, and the kinematic relations makes it possible to measure the position and orientation between frames. Additional frames can be added, if the production cell contains more elements than depicted in figure 3.5. Last requirement to the model is fulfilled by measuring the strain/deflection relation in appendix F. It is then possible to apply the dynamic model and the relation to determine the spatial position of the TCP. Besides the modeling requirements, a list of requirements was also stated for the manipulator control, which are given as

- The improved controller must perform equal or better than a controller based on a rigid manipulator model regarding oscillation settling time. This corresponds to a termination of the controller after the manipulator joints have reached their reference values
- Tool oscillations must be damped to allow fast TCP positioning. A practical example (given by Ole Madsen, see personal profile in acknowledgements) has a 10-12 s task time and a settling time of $t_{\mathsf{set}} < 5$ s is selected on the basis of this time interval
- Tool tip deflection must be below the level of the path accuracy of $\pm$0,1 mm from the default manipulator controller [80]
- The controller structure must be based on nonlinear control theory

If the hardware configuration is assumed rigid even though the tool is flexible, the TCP will oscillate after the controller has been terminated. This oscillation settling time has been measured to be 50,85 seconds for the specific test configuration. After applying the controller with knowledge about the flexible tool the, oscillation is reduced to 3,85 seconds. A significant improvement is therefore obtained over the initial case.

A requirement for the settling time was stated at $t_{\mathsf{set}} = 5$ which was respected for that particular case. With a 10-12 s task time given, $\sim$ 4 s used for damping and $\sim$ 3 s used for relay tuning, only around 3-5 s is left for the manipulator motion.

Because no practical production has been considered, the 3-5 s time interval must be evaluated for each specific production. Different tools and manipulators will provide different results as well.

An accuracy of ±2,79 mm with ±0,49 mm deviation was measured from the experiment described in appendix D. This is much more than the original path accuracy of ±0,1 mm. The requirement cannot be fulfilled because the controller cannot damp the oscillation further without introducing new oscillations. However, the last requirement is fulfilled because sliding mode control theory was applied to solve the control problem. An overall conclusion of the project is provided in the following chapter, which will also summarize the requirements.

Chapter 11

Conclusion

The scope of this thesis is to develop and document an autonomous control system capable of improving accuracy and operating time of an industrial robotic manipulator working with flexible tools. Several elements have been included and described in details throughout the report, and a final evaluation of the performance was given in chapter 10. This conclusion will summarize the work from the entire project period and state whether of not the project has respected the requirements listed in section 2.4. Because the project deals with a number of different topics, the conclusion will be divided into a number of sections with the following content

- System modeling
- System identification
- Sensor fusion
- Control system
- Overall conclusion
- Future work
- Project applications

Autonomy is introduced by applying system identification algorithms to estimate the load in stead of being given the informations manually every time. The first four topics are describing the benefits, results and problems from the analysis and design phase of the project. Afterwards, an overall conclusion will compare each of the technical requirements with the outcome of the project work. Furthermore, the overall question from the project description will be answered conclusively. Lastly, ideas for future work on this project will be outlined, and two cases will be formulated, in which the product from this project will be applied. This will relate the project to practical applications.

A control system is developed around the REIS RV15 industrial manipulator. Using a National Instruments cRIO controller system, the original control algorithm has been bypassed to allow other controller structures to be tested. A number of sensors including accelerometers, gyroscopes, strain gauges and rotary encoder, are sampled by the cRIO to close the feedback loop. In order to test the damping performance of the final controller, a flexible tool was constructed. The tool consists of a flexible beam (deliberately designed with high flexibility around one axis) and a tip mass. First topic of the conclusion is the system modeling.

11.1 System modeling

One of the basic elements in a control system is a model describing the dynamics of the plant. In this case the model must describe the dynamics of the REIS RV15 industrial manipulator. The modeling part was initiated with a description of manipulator kinematics, which can be used to determine position and orientation in Euclidean space. Homogeneous transformation matrices were derived for each link of the manipulator, and the position of the tool center point (TCP) was determined as a function of the joint angle space θ.

With the kinematics sorted out, dynamics were added to the model. A recursive Newtonian approach was applied to determine the torque acting on each joint as a function of the links below. Using the full 6-DOF manipulator configuration returned an unacceptable large system model. Instead of trying to simplify or approximate the model equations, a remark was stated about applying only the swivel axes 1, 2 and 5 (see remark 6 on page 152). This reduced the size significantly, and a model describing the manipulator dynamics was given in a standard joint space model formulation. The removal of three degrees of freedom limits the overall usage of the control system. Only tools with bending around one axis can be modeled. In order to apply oscillation damping to arbitrary tool configurations with arbitrary flexibility, the model must include all 6 axes to describe motion in $\mathbb{E}^3$.

A second step of simplification was introduced due to the high drive train gearing of a factor of 100. This resulted in a friction model alone, and the model was reduced to a direct input model. The input signal is a direct expression of the torque on the joint. This is not actually possible, but due to the high gearing it is assumed to be approximately the same. The final manipulator model is a simple approximation of the actual behavior, and will not resemble the manipulator dynamics perfectly. However, for the purpose of control, the model provides a basic structure, and the selected controller structure is able to work with inadequate models.

Last part of the complete model is a sub-model of the flexible tool dynamics, which is derived on the basis of general beam theory. The resulting mode shape of a bending beam can be approximated as an infinite sum of products $\phi_i(x)q_i(t)$ with $\phi_i(x)$ denoting the mode shape function (describing the shape of the i-th eigenmode) and the modal dynamics $q_i(t)$ (describing the time dynamics of the i-th eigenmode). The eigenfrequencies have been determined from a frequency equation, which is a solution to an eigenvalue problem. They depend on both the specifications of the tool and the mass of the load. A lumped mass has been assumed, meaning that the inertia of the flexible beam is the most important part. In a real application, however, the inertia must be included to increase accuracy to arbitrary tool configurations.

An experiment was carried out in appendix A to measure the specifications of the flexible tool used in this project. The first two eigenfrequencies were measured to 4,20 Hz and 81,01 Hz. Because the second eigenfrequency is beyond the capabilities of the manipulator, only the first eigenmode is included in the model.

The dynamics of the flexible tool were described by a second order system with a damping ζ_1 and a natural eigenfrequency ω_1. During the modeling part, a number of constants were needed to scale the resulting dynamics to resemble actual behavior.

However, the constants are collected in one input gain constant, which will have to be estimated using system identification procedures. A lot of work can be dedicated to express each of the constants, but they will have to be tuned to fit the actual system anyway. The low damping of the system and the low-frequency eigenmode frequency is accurately resembled by the second order system. The system identification procedures are summarized and evaluated in the sequel section.

11.2 System identification

System identification is applied for two reasons in this case: to estimate the unknown parameters describing the manipulator dynamics and determine the dynamics of the flexible tool. Because none of the technical documents describing the REIS RV15 includes all parameters necessary to fully complete the manipulator model, system identification is applied. Originally the parameters were supposed to be updated on the run, but the manipulator is operated in such a limited range of time that the parameters are assumed to remain constant. These parameters are therefore only estimated once, and the system identification methods need not to run in realtime.

For pure offline estimation of model parameters, the prediction error method was appled, and works by minimizing a performance index - in this case the error between the measurements and the model estimates. Based on measurements from the actual system when excited by a random control signal, the performance of the method was tested. By comparison with the measurements from appendix A, the eigenfrequency of the tool was accurately estimated. The damping, however, was off by a factor 5, and the input gains have no reference for comparison. Model simplicity may be a reason for the only ~ 80 % goodness of fit. Using a less-reduced model and introducing more than only 6 sinusoids in the input signal may increase performance. A subspace method was also tested to estimate the significant pole location of the system dynamics, but the method was providing useful and unstable estimates.

The flexible tool on the other hand is supposed to be arbitrary including the load. This means that the control system must be able to automatically identify the changes in tool and/or load dynamics. First attempt involved an extended Kalman filter, which is able to run in parallel with the existing control system, and simultaneously update the model. However, the method provided fluctuating estimation results when tested on a set of data sampled from the system.

Since the resonance must be estimated very accurate, it was decided to find another approach. The relay tuning method was the next choice, which can be used on the hardware as well as the EKF. Unfortunately, this method cannot be executed in the same was as the EKF. Where the EKF can update the parameters on the basis of input/output-relations, the relay tuning method must be able to excite the system itself. Then there is no actual online parameter estimation on the system. However, the method converges much faster than the PEM, but will have to be executed between tasks.

All in all, the choice of system identification method relies on the specific manipulator operation. If the manipulator is handling several identical loads, each new configuration can be undergoing a system identification process to estimate the new dynamics. Based on the measurements from this report, the PEM and the RT are possible candidates. If the dynamics are changing for every task, a method like the EKF will have to be applied, but supported by a comprehensive tool model and more sensors to increase performance. Security algorithms will have to check for instability of the filter, because a nonlinear filter cannot be guaranteed to remain stable over the entire domain of usage. If the convergence time of the EKF can be improved and ensured to be faster than the completion time for a single task, the method can be applied.

In a practical application, however, the diversity between tools and loads may be limited. Each combination between tool and load can then be estimated once, and a specific controller configuration can be ascribed each of the combinations. This method is only applicable in situations where the tool/load configuration is identical for each task. Every deviation from the pre-measurements will degrade the performance of the controller.

Because the model was simplified two times, the complexity requirement has not been met for the manipulator part of the model. Only friction elements are included in the final model, which is only one of several parameters describing the dynamics of the manipulator links. However, it was argued in the modeling part, that due to the gear ratio of 100, no dynamics will be observed on the joint angle response. The response will be a direct double integration of the applied input signal, and the actual dynamics need not to be included.

Several system identification methods have been evaluated and tested. It is difficult to select a final method, because the overall requirements to the production influence the choice. For the purpose of this project, the PEM and the RT method were superior over the EKF and N4SID, and provided useful estimates for the controller. The controller will be evaluated after the sensor fusion technique described next.

11.3 Sensor fusion

Before the controller can be fully evaluated the sensor fusion technique is considered. The controller applies the dynamic model structure with parameters estimated by system identification methods described in the modeling part and chapter 8. Sensor fusion has been added to make the autonomous system robust, and to further improve the accuracy of the model estimates. The technique is based upon a linear Kalman filter, which is fusing model estimates and measurements to achieve better estimates. It behaves different from the system identification methods, because it does not alter the mode but only the estimates.

The technique was not tested in parallel with the controller, but was tested on its own using measurement data from the controller performance experiment in appendix D. By tuning the matrices Q and R correctly, a filter response can be achieved, which removes noise from the measurements. The joint angle response is within 0,006 rad of the model estimate throughout the experiment from appendix D.

Strain measurement and strain estimates are, however, not immediately coinciding.

The filter response converges to the strain measurements in around 3 seconds, which must be taken into consideration for realtime applications. By tuning the matrices Q and R, a better response can be achieved. Because the filter was not implemented onto hardware during the controller performance test, it is not possible to see how much the overall control system performance is affected by the filtered estimates. The filter is able to remove noise from measurements, but it cannot be concluded whether or not the overall performance is affected by the slow convergence time. Up next is the evaluation of the controller.

11.4 Control system

In order to improve the accuracy and operating time for a manipulator with a flexible tool, a controller must be applied. The sliding mode controller is of the nonlinear controller type, and was selected over a range of other possible candidates. It was considered the best choice in this case due to several things. It is robust to model uncertainties and will perform even though the system model is inadequate.

The controller is based on a model of the flexible tool dynamics. An experiment (see appendix A) showed that the second mode of vibration is 81,01 Hz and thus only one mode of vibration is used to model the behavior. A frequency of 81,01 Hz is too fast to compensate with the manipulator, which was concluded in appendix E. This results in a pure linear model, and linear controllers are therefore also candidates for this control system. However, the analysis has been based on the sliding mode controller type to control a general nonlinear flexible system described by M modes of vibration.

A measure of the controller performance is the ability to damp an oscillation as fast as possible and is defined by the settling time. An expression of the upper bounded settling time as a function of the controller gain was determined in chapter 9. This is based on the ideal case with infinite control bandwidth using a discontinuous control law. Practical configurations were also considered through simulations, and when using a continuous control law, the performance was degraded by a factor of ~ 4. However, the theoretical settling times were still below the requirement of 5 seconds, but shows the behavior of a non-ideal controller configuration.

Chattering was considered in section 9.2 and is considered to be the main issue regarding controller performance. If the frequency of the chatter signal is coinciding with the resonance frequency of the tool, additional and unnecessary control effort will have to be applied. A continuous control law using a saturation function was proved to remove chattering in the theoretical case. However, when applying the controller in a closed-loop of a given sample frequency, a new frequency component appeared. No explicit relation has been expressed to describe the correlation between sample frequency and chatter frequency, but a frequency of ~ 16 Hz was discovered through simulation. This is considered to be harmless, because the frequency is far from the 4,20 Hz resonance measured in appendix A.

During experiments it was discovered, that the tool drifted from a horizontal position to a vertical position even though the joint actuator was excited with a zero-mean control signal. This problem was evaluated in section 11.1 to originate from the lack of gravity effect in the model and possibly an error in the drivers. Model simplifications have neglected the gravitational pull on the lumped tip mass, which seems to be important anyway. Even with the high gear ratio of the manipulator joints, the gears can still be turned by the torque contribution from the tool.

Using a sliding mode controller to track two references requires the use of higher order manifolds. This was shortly introduced in 9.1, but was not considered for this project based on assumption 1, saying that the manipulator was not affected by the flexible tool. By introducing more constraints to the control problem, the performance is degraded. A trade-off will appear between damping the tool oscillations and tracking the tool mount frame $\mathbb{F}_\tau$ to a given reference, because both subsystems are controlled by the same control signal.

A test of the controller performance was conducted, and the results are given in chapter D. Different controller configurations were tested, and the settling time was measured and compared with the uncontrolled case. Only one configuration was able to damp the tool oscillation below the 5 % amplitude level required by the definition of settling time given in section 9.3. This uses the sliding surface $s = \dot{q} + Kq$ and limits the control signal to ± 1 V. The limits are used as precautionary measures to avoid damaging tool and/or manipulator during test.

Because only one of the proposed controllers has measurable performance, this will be used for the final evaluation. A 92,43 % improvement was discovered over the uncontrolled case with a settling time of 3,85 seconds, which is below the 5 seconds requirement given in section 2.4. However, the tool oscillations are only damped to a certain level. An oscillation measured by a strain of $\pm 0{,}3376$ V is present for the remaining ~ 6 seconds of the experiment. This corresponds to a tip deflection of $\pm 2{,}79$ mm with a deviation of $\pm 0, 49$ mm by applying the strain/deflection relationship measured in appendix F.

Depending on the application, it must be evaluated whether or not this is acceptable. In this case, however, an initial oscillation corresponding to a tip deflection of 61,77 mm (measured strain 7,48 V) is reduced to 2,79 mm in 3,85 seconds. Therefore, even for tools with high flexibility, the controller can damp the oscillation by a factor of ~ 22 in under 4 seconds.

When investigating the phase portrait of the measurements, a sliding mode is not present. The trajectory is converging towards the equilibrium faster than the uncontrolled case, but an actual sliding mode behavior is not measurable. Due to a hardware limited control signal and the application of a continuous control law the control law is not able to activate the sliding mode behavior. A missing term $-K\dot{q}$ in the controller implementation can also be part of the cause. It is necessary to excite the actuator with even higher control actions to achieve the expected behavior, but due to security measures, this is not possible. However, even though sliding mode is not present, the tool oscillations are damped significantly. Several tests must be conducted though to definitely conclude that the controller performs as measured from the single case experiment. The main four parts of the final product have all been evaluated, and an overall conclusion of the project is given next.

11.5 Overall conclusion

The final product was built around three parts: *system identification* from chapter 8, *sensor fusion* from chapter 7 and *sliding mode controller* from chapter 9. All parts use the dynamic model derived in the modeling part, which proved to describe the system behavior satisfactory as described in appendix B. The overall model is composed from a single mode of vibration and a single manipulator link. It was decided not to use the full hardware setup in order to focus more on the actual control problem using only a single actuator to damp the oscillations of the flexible tool.

Several system identification methods were evaluated to estimate the dynamics of the system. It was considered to use the subspace method N4SID from section 8.3 to verify the overall model structure and the location of the system poles before deriving the model. However, the method proved to be very unstable and did require very accurate measurements to perform satisfactory.

The prediction error method from section 8.2 proved instead to be useful for identifying the entire system model in an offline configuration. This is used to derive a model before designing the control system to achieve some knowledge of the performance. An extended Kalman filter described in section 8.1, was investigated to update system parameters on the run. It proved, however, to provide fluctuating estimations, and convergence cannot be guaranteed because it uses a nonlinear model. They are both unwanted for this system, because especially the resonance frequency must be precisely estimated to achieve a damping effect and not the opposite.

An alternative to the extended Kalman filter was given in the shape of the relay tuning method from section 8.4. This was tested in appendix B together with the other methods. Different from the other methods it is only able to estimate the first eigenfrequency of the tool, which it did with 8-11 % accuracy when compared with the measurements from appendix A. From the system identification experiments it was clear, that the prediction error method and the relay tuning are the best candidates. Disadvantages of the methods is the offline use of the methods and the closed-loop requirement for the relay tuning. However, the PEM is not constraint by any timing requirements because it is used in the design phase, and the relay tuning converges fast enough to be used when the system is in operation.

Second part of the product is the sensor fusion, which is used to combine model estimates and sensor measurements. It was described in chapter 7 and tested on measurement data in appendix D. The filter was able to filter measurement noise and converge to the measurements after around 3 seconds. Performance can be increased by further tuning of the matrices Q and R expressing the process and sensor noise covariance matrices, respectively. Even though the filter was not running on hardware during the controller performance experiment in chapter D, it was verified to function. It is, however, not possible to evaluate the performance of the filter when it is combined with the controller.

Third and last part of the product is the sliding mode controller described in chapter 9. The controller removes the original dynamics of the manipulator and replaces it by a new one. A more accurate model will make the controller perform better. The original sliding mode controller uses a discontinuous control law, but for the purpose of this project it has been replaced by a continuous control law instead. This has been done to remove chattering, at least for the theoretical case, but the trade-off is a degradation of performance. When the controller, however, is executed with a given sample time, the chatter reappears in a different shape. Unfortunately the chatter frequency is closer to the resonance frequency but not coinciding.

The controller was implemented onto hardware and tested in appendix D. Only one controller configuration provided measurable controller performance. This uses the switching surface of the form $s = \dot{q} + 10q$ and a control signal limitation of ± 1 V. The tool was first excited by a sinusoidal signal with resonance frequency. Then the controller was activated, and the oscillations were damped by 95 % after 3,85 seconds. For the uncontrolled case the oscillations reached the 5 % level efter 50,85 seconds. An improvement has therefore been achieved, but since only one case provided useful data it cannot be concluded whether or not a better performance can be achieved.

Furthermore, by investigating the phase portrait of the measurements it is clear, that the trajectory does not enter sliding mode. When the trajectory reaches the sliding surface it is not affected to move faster towards the equilibrium point. A missing $-K\dot{q}$ term in the controller implementation diagram may be causing this error. Simulations have shown, that the lack of this term makes the trajectory circle cross the switching surface a couple of times before reaching sliding mode.

The controller, however not in the optimal configuration, adds more damping to the closed-loop control system. Due to the saturated control action and the limited bandwidth of the manipulator, measured in appendix E, the sliding mode cannot be reached. Despite this, the controller is able to reduce the settling time of the tool oscillations by a factor of around 22.

From the acceptance test is was measured how the task time is consumed. Using an initial $\sim$3 seconds to estimate the dynamics using relay tuning, a couple of seconds to grasp an item and move the manipulator to/from the final destination as well as 3,85 seconds to damp the tool oscillations makes the completion time close to the 10-12 seconds estimated task time. However, the 3,85 seconds is based on a single configuration of the tool and amplitude of the oscillation, whereas the remaining times are estimates. No actual production was considered, and it is therefore not possible to conclude on the overall production performance, but only on the single case tested by experiment.

A conclusion can be given on the delimitated case from this project. System identification methods were able to estimate the model parameters, a linear Kalman filter was capable of removing measurement noise and provide more accurate model estimates when converged and the applied controller damped the oscillation of the flexible tool below the 5 seconds requirement. Combining all parts makes an autonomous system capable of improving control of flexible end-effectors in terms of accuracy and settling time if the different parts are manually adapted to given requirements. After tuning the covariance matrices and the sliding surface description the system will automatically adapt to changes in the tool/load configuration.

The methods and requirements from chapter 2 were all evaluated in section 10.3. Based on the methods applied throughout the report and the results from the acceptance test, all but one requirement were fulfilled. Only the accuracy requirement of $\pm$0,1 mm was not fulfilled, because the practical experiment yielded an accuracy of

$\pm 2{,}79$ mm. This is caused by the fact, that the controller introduces new oscillations when trying to reduce oscillations with a small amplitude. The question from the problem formulation describing the problem to solve was given in chapter 2 as

> *How to improve end-effector position/orientation control of industrial robotic manipulators in terms of accuracy and operating time when handling flexible tools?*

This is accomplished by making an accurate model of the flexible tool behavior. The settling time can be improved by damping the eigenmode oscillations using a controller that applies a model of the tool and thus reducing operating time. Accuracy can be improved if the manipulator bandwidth is much higher than the first resonance frequency of the flexible tool. This ends the overall evaluation of the project, and future work is described in the sequel section.

11.6 Future work

A remark on future work is given, which will improve the features of the product described in the report. Other controller types features different advantages, that may be exploited for future development on this kind of system. Repetitive control is capable of both controlling the manipulator to a given position while damping the eigenmodes of the tool using a second controller. When using SMC for that purpose, a higher order SMC must be applied, which is more complex. Another strength of repetitive control is the natural improvement of the controller over time when subjected to repetitive control tasks.

Mass production of identical parts is an example of repetitive tasks, which can benefit from this control technique. It is not the intention to mention all advantages/disadvantages of other controller types over the SMC, but merely inform that other possibilities exist. Each individual case must be evaluated before selecting a specific controller type. If the SMC is kept, it must be able to change the K-value on its own. The current system must be manually provided a K-value, which is derived through simulations. This remains unchanged even though the tool/load configuration changes. This is theoretically possible because the system dynamics is replaced with new dynamics. However, in practical setups the change in tool/load also changes the requirements to the controller dynamics, and the K-value must be changed accordingly.

Based on the experiment from appendix B.1, the joint angle showed a drifting behavior even when excited with a zero-mean control signal. This may be caused by gravity and the kinetic energy of the flexible tool, causing the tool to slowly drift towards a 90 degree angle. It was considered in the beginning of the chapter, that the sliding mode controller could introduce a drift in the joint angle, but the phenomenon was spotted even without the controlled in action. In order to compensate for the drifting a second controller must be added to the control loop.

Figure 11.1 shows the control system with the sliding mode controller and a state feedback controller. The sliding mode controller regulates the flexible tool strain q to q_{ref} (in this case zero) and the joint angle θ to the reference θ_{ref}. A *separation in control* must be introduced because the two controllers must excite the same actuator to accomplish two different control goals. Advantageously, the sliding mode controller is working around 16 Hz in sliding mode according to the simulation in figure 9.24. The joint angle controller does not need to work at a frequency higher than 1 Hz due to the slow drift of around 0,05 rad/s ($\sim$ 1 rad in 20 seconds, see figure B.3a). The last part of the conclusion describes the practical applications for the product developed in this thesis.

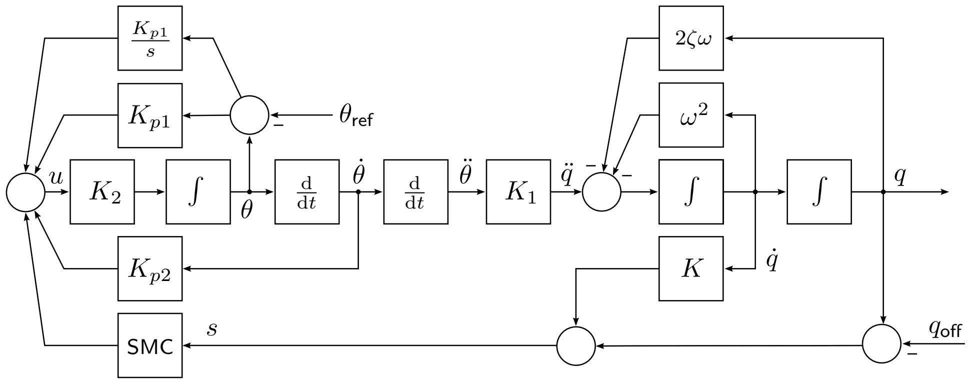


Figure 11.1: *Control system with sliding mode controller and state feedback controller*

11.7 Project applications

This project covers a range of different techniques for improving the control of a flexible tool mounted onto a manipulator. Different productions have different requirements to accuracy, speed and reliability, but including them all will expand the scope of this project too far. Therefore, the project has been working on a single case with one flexible tool configuration lifting a constant load. In order to show the versatility of the final product, two project applications are described in the following, in which the product can be applied. The first case is an extension of the example used throughout the report, whereas the second case involves robotics for surgery.

11.7.1 Intelligent production cells

Whenever a new production cell is installed or the tasks are changed, each manipulator must be re-programmed/adapted to these new tasks. It may involve changing the trajectory of the TCP or the velocity profile due to a change in the payload. Consider a case where the TCP trajectory must be identical when handling two objects of varying mass. If the payload is within the limits of the manipulator capabilities and the construction is assumed rigid, the trajectories will be identical. When using flexible robotics or flexible tools, however, the problem changes.

As an extension of the case considered throughout the report, it is assumed that the production cell is equipped with vision systems. Besides ensuring quality and safety, the vision system can also estimate the dynamics of the items to lift. Figure 11.2 shows a manipulator assisted by a vision system to make decisions on how to move.

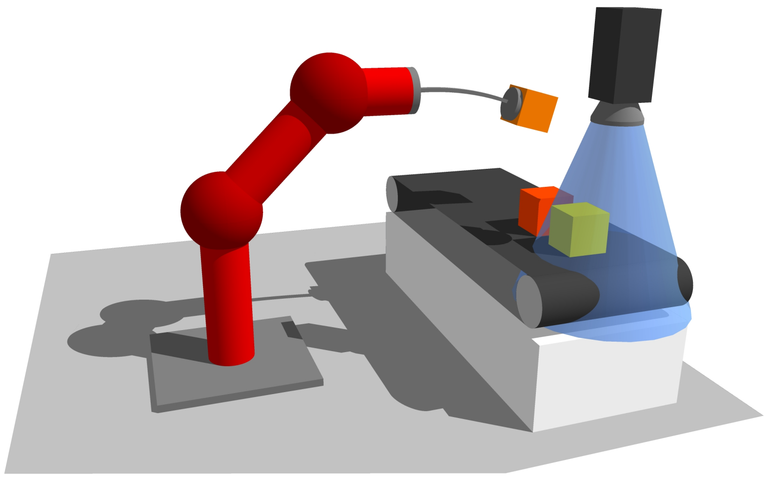


Figure 11.2: *Manipulator assisted by vision system for decision making*

Information about weight and structure can be fused together with information from system identification procedures. The vision system will provide an immediate guess of the load dynamics before the manipulator is activated. System identification, however, will occupy the operation before the load has been identified. This case shows the flexibility of this product, because the different parts can be supported by additional blocks to further improve the performance for the specific production.

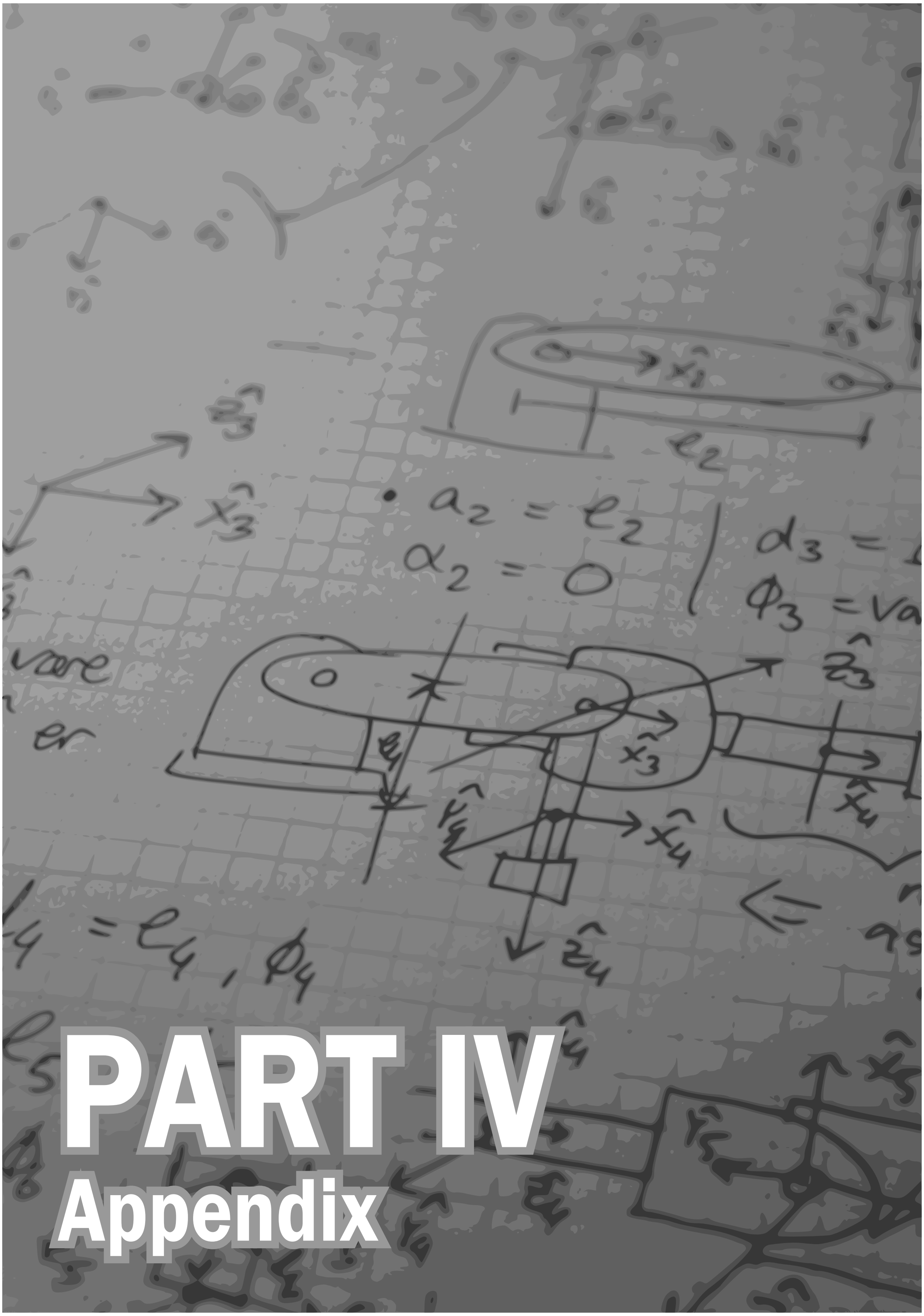
11.7.2 Accurate surgery robots

The medical sector is beginning to benefit from robotics to perform different tasks like the industrial sector [84]. Robotics can be either guided by a trained surgeon or automated for less complicated tasks. The end-effector will be a tool specifically designed to perform required surgery tasks and must be kept steady during the entire operation. Requirements can also be added to constrain the tool in Euclidean space during operation. Maximizing the reachability and minimizing the size of the tool will naturally introduce flexibility.

With flexible tools, a larger control action must be added in order to maintain track of the TCP and via points. By introducing multiple sensors and multiple controllable degrees of freedom, the path complexity can be increased and thus maximizing the possibility for the TCP to reach destination in an optimal way. Optimality in this respect may ascribe trajectory length and maneuverability more value than pace.

It may also be necessary to estimate the current environment along the way, which may contribute different external disturbances. The dynamic response of the system will consequently change, and the controller must be informed to adapt the actuation to the given situation. An online system identification must be applied to quickly adapt the controller to a new environment.

PART IV
Appendix



Appendix A

Journal 1: Flexible beam specifications

The flexible tool in this project serves to emulate a tool, which is able to bend the mounting bracket between tool and manipulator. A flexible beam is simulating the bracket, which has been modeled through chapter 4. This journal will verify two important parameters of the mathematical construction and the sufficient number of modes M to describe the system dynamics. First parameter is the modal damping of the beam ζ_i for the i-th mode, which is not included directly in the model derivation. It will have to be added afterwards if it respects given assumptions, making sure that the new damped response is still a solution to the partial differential equation expressing the flexible tool behavior. A second parameter is the modal frequency ω_i, which is the eigenfrequency for the i-th mode dynamics. Both tests will be conducted on the flexible tool with or without tip mass m_l.

Both tests can be performed by exciting the beam with a certain signal. The decaying motion of the measured response can be related to the modal damping constant ζ_i, and the spectrum of the response verifies the locations of the modal frequencies ω_i.

From a spectrum of the strain measurements, the different eigenfrequencies can be estimated from significant peaks. Whenever the relative factor between first and k-th peak drops below 5 %, then $M = k - 1$ modes will sufficiently describe the full system dynamics. Before presenting the results from the experiments, the theory behind the test procedure will be explained.

A.1 Theoretical foundation

When modeling the flexible dynamics, damping is not part of the derivation. It is however assumed from the analysis, that a damping term can be added afterwards without affecting the credibility of the solution. This is because the damping factor is considered to be small $< 0{,}1$ based on empirical hardware impressions. An experiment will, however, tell the exact value of ζ_i. The expected shape of the strain measurements is given as a sum of exponentially decaying sinusoid on the form

$$\epsilon(t) = \epsilon_{\text{off}} + \sum_{i=1}^{M} \epsilon_{0,i} \mathrm{e}^{-\zeta_i t} \sin \omega_i t \tag{A.1}$$

with $\epsilon_{0,i}$ denoting a scale of the i-th eigenmode component and ϵ_{off} an offset. A curve fitting procedure will respond with a set of M damping factors, that can be added to the dynamics. From a practical perspective, a maximum 5 % drop in eigenfrequency due to damping has been found sufficient, which corresponds to

$$\zeta \leq \sqrt{1 - 0{,}95^2} = 0{,}3122 \tag{A.2}$$

which is derived from the relation between damped and undamped eigenfrequency $\omega_{\text{damped}} = \omega_{\text{undamped}} \sqrt{1 - \zeta^2}$ [78]. As long as $\zeta \leq 0{,}3122$, the eigenfrequency is assumed to be unchanged. The relation is illustrated in figure A.1

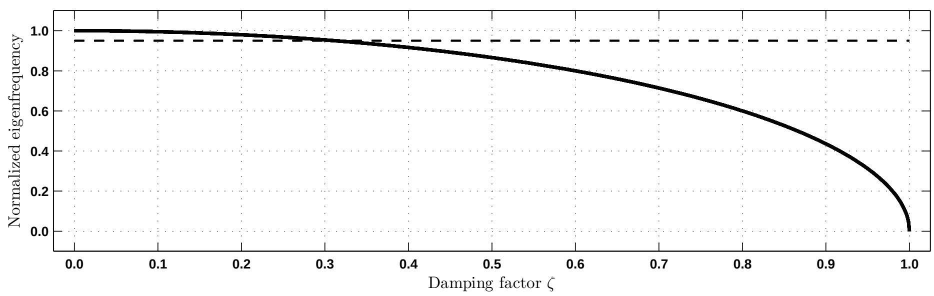


Figure A.1: *Relation between damped and undamped eigenfrequency as a function of damping* ζ

Since the scaling of the eigenfunction is not affecting the eigenfrequencies, the strain measurements can be used to measure the eigenfrequencies directly and compared with the theoretical values.

A.2 Experimental procedure

Two different parameters are to be estimate through the experiments described in this journal: the beam damping parameter ζ for each measurable mode and their corresponding eigenfrequencies. Each experiment will be conducted for three different hardware configurations as listed below

- Tool without tip mass
- Tool with small tip mass ($m_l = 351$ g)
- Tool with large tip mass ($m_l = 522$ g)

The damping parameter, which is important for the end-effector settling time, is only investigated for the first mode in each hardware configuration, since the damping is assumed large for $i > 1$. Their significance will thus be limited, but this will be considered when the data is available. The eigenfrequencies for each mode must also be estimated. If the eigenfrequencies for $i > 1$ are must smaller than the first eigenfrequency, they will have little impact on the end-effector position. Further, the high frequencies cannot be excited by the manipulator due to limited actuator bandwidth. Procedures for estimating the damping ratio and the eigenfrequencies are described in table A.1, and an illustration is shown in figure A.2.

Experimental procedure	
Damping ratio ζ	The tool is mounted to a solid table (table mass $\gg$ tip mass) to enable oscillatory motion. When the data logging software is running, the tool is deflected to a maximum position (without deforming the tool permanently). The tool is released and the logging stops, when the strain measurements read below 50 μm/m. All three hardware configurations are tested in this way, and the exponential decay can be estimated from the acquired data sets.
Eigenmode frequencies ω	Similar to the above description, the tool is mounted to a table, but in this experiment, only the two hardware configurations involving a tip mass are considered. A nylon hammer is used to make an impact on the tip mass (procedure used in [6]). This simulates an impulse, which will excite all nonlinear tool dynamics including all mode frequencies. Calculating the spectrum for the acquired data sets makes it possible to determine each mode frequency. The first mode frequency, however, is determined from the data gathered in the first experiment, because it only involves that specific frequency. This provides are more accurate estimate.

Table A.1: *Experimental procedure for journal 1 experiment*

No exact scaling of the signals is considered in this experiment, only the intermediate distance between frequencies in the spectrum as well as the decaying properties of the time series. The tip mass is considered a lumped mass located immediately where the load is mounted. Therefore, the beam length is slightly longer when no tip mass is added, which will be calibrated before performing any calculations. The equipment used for the experiments are listed in table A.2

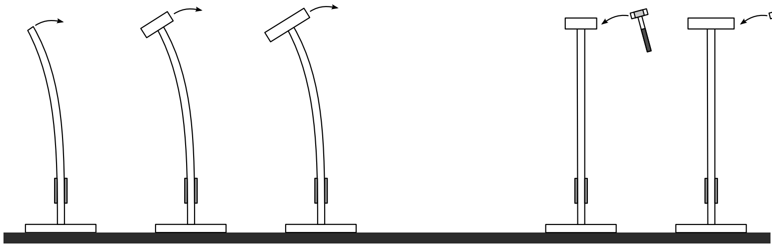


Figure A.2: *Experiment setup for first (left) and second (right) sub-experiment*

CatmanEasy data acquisition software
Spider8 data acquisition unit (AAU 30836)
Flexible tool with two interchangeable tip masses (351 g and 522 g)
Nylon hammer

Table A.2: *Equipment for journal 1 experiment*

Results will be presented for both experiments in the sequel section. Firstly, the mode eigenfrequencies are derived using FFT on the sampled strain measurements. Secondly, the damping ratio will be estimated from the exponential decay in the strain response. Different sampling frequencies have been used for each experiment, due to the difference in settling time and the limited sample memory. The sample frequencies are given in table A.3 below.

	$m_l = 0$ g	$m_l = 351$ g	$m_l = 522$ g
Determine mode frequencies	-	9600 Hz	9600 Hz
Determine damping	1200 Hz	400 Hz	400 Hz

Table A.3: *Sample frequencies for experiments in journal 1*

A.3 Results

Using the above procedures, a number of experiments have been conducted. The second experimental procedure from the above section has been used to collect data for estimation of mode frequencies. The data is presented in figures A.3 and A.4.

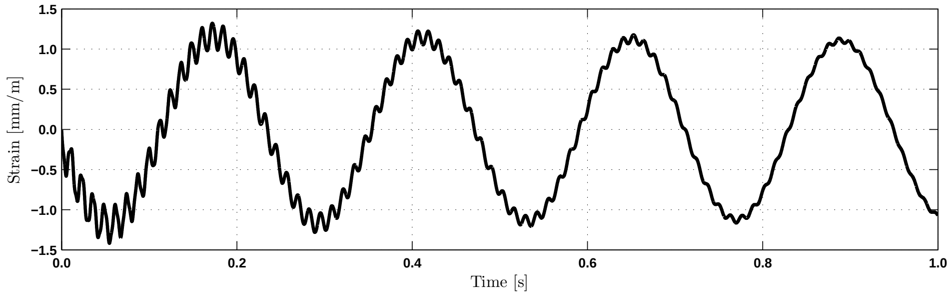


Figure A.3: *Strain measurements from impact experiment with small tip load* $m_l = 351$ *g*

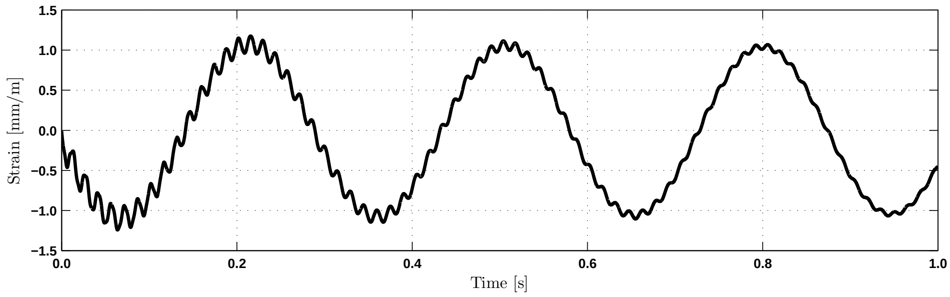


Figure A.4: *Strain measurements from impact experiment with large tip load* $m_l = 522$ *g*

Based on a spectrum of each data set, shown in figures A.5 (data from first experiment) and A.6 (data from second experiment), the individual eigenfrequencies can be determined. The varying amplitude is a consequence of the different damping ratios, and will not be physically interpreted.

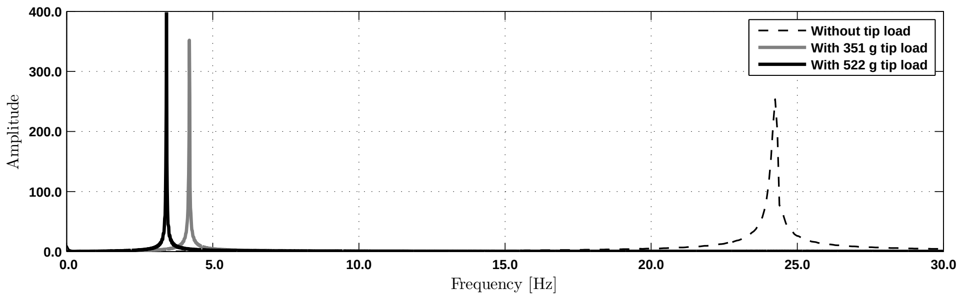


Figure A.5: *Spectrum of strain measurements from figure A.7 through A.9*

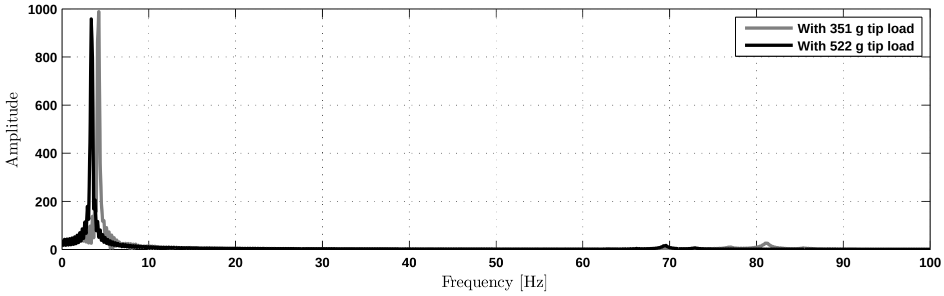


Figure A.6: *Spectrum of strain measurements from figure A.3 and A.4*

See the estimation of first and second mode resonances in tables A.4 and A.5. A comparison with the theoretical values from tables 4.2, 4.3 and 4.4 in chapter 4 is also given in the tables.

	Theoretical	Empirical	Error
Without tip load	25,60 Hz (160,85 rad/s)	24,24 Hz (152,30 rad/s)	-5,31 %
With 351 g tip load	5,05 Hz (31,73 rad/s)	4,20 Hz (26,39 rad/s)	-16,83 %
With 522 g tip load	4,17 Hz (26,20 rad/s)	3,42 Hz (21,49 rad/s)	-17,99 %

Table A.4: *Comparison between theoretical and experimental values of* ω_1

	Theoretical	Empirical	Error
With 351 g tip load	122,91 Hz (772,27 rad/s)	81,01 Hz (509,00 rad/s)	-34,09 %
With 522 g tip load	122,58 Hz (770,19 rad/s)	69,58 Hz (437,18 rad/s)	-43,24 %

Table A.5: *Comparison between theoretical and experimental values of* ω_2

The first set of data is used to determine the damping ratio ζ for the first mode in each configuration, and the data is presented in figure A.7 (without load), figure A.8 (with 351 g load) and figure A.9 (with 522 g load).

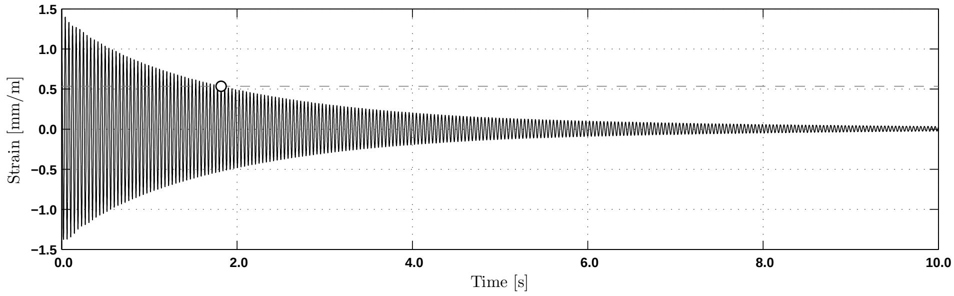


Figure A.7: *Strain measurements without tip load*

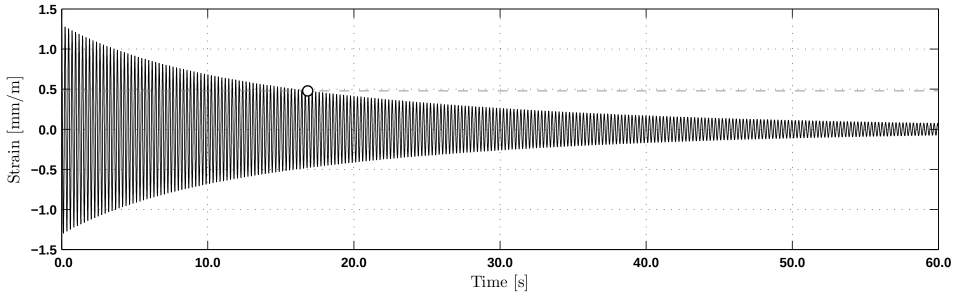


Figure A.8: *Strain measurements with small tip load* $m_l = 351$ *g*

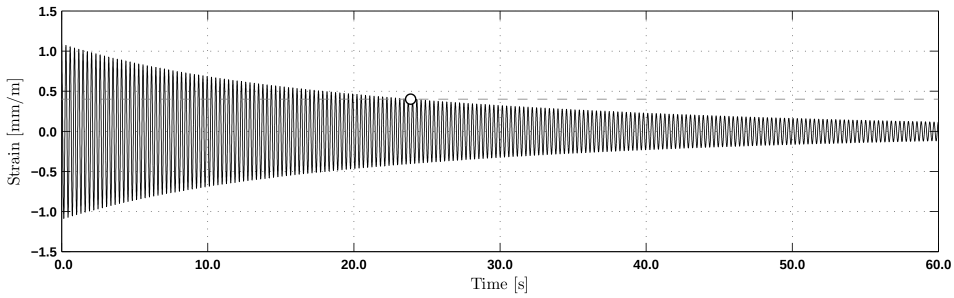


Figure A.9: *Strain measurements with large tip load* $m_l = 522$ *g*

A point is used to indicate the location, where 0,368 of the initial amplitude is remaining, which is used to determine the rate of decay $\sigma_1 = \zeta_1\omega_1$ [41]. 60 seconds of data is shown from the experiments involving tip masses and only 15 second for the one without. This is due to the large difference between the damping ratios between the configurations, which will decrease with increasing load. The damping ratios for each experiment are given as the inverse time constant, and they are shown in table A.6.

	$\sigma_1 = \zeta_1 \omega_1$	ζ_1	$\tau_1 = 1/\zeta_1$	ω_1
Without tip load	0,5498 s^{-1}	0,0036	1,8188 s	152,30 rad/s
With 351 g tip load	0,0594 s^{-1}	0,0023	16,8350 s	26,39 rad/s
With 522 g tip load	0,0419 s^{-1}	0,0019	23,8663 s	21,49 rad/s

Table A.6: *Estimations of ζ_1 from experiment in journal 1*

A difference arises in damping ratios, when the tip mass is added and increasing. This is also a consequence of the decreasing eigenfrequency. The damping ratio has only been estimated for the first eigenmode, since this is the most significant. Furthermore, it is difficult to estimate higher mode damping ratios using this method. Each eigenmode must be excited individually to achieve accuracy.

A.4 Conclusion

Based on the data from the experiments, the mode damping was determined for all three configurations. It can be further concluded, that the damping is below the boundary of 0,3122, which was specified in the theoretical section of this journal. Therefore, the theoretical eigenvalues for each mode are still assumed to be part of the model solution, since the damping was not part of the derivation of the solution. It was also shown, that the addition of additional tip mass decreased the damping of the tool as well as decreased the eigenfrequencies. Due to lowpass actuator characteristics, a lower eigenfrequency increases the controllability of that particular eigenmode. However, the decrease in natural damping of the mode makes the control loop more sensitive to external disturbances, that may affect the damping effect of the control system.

When comparing the measured and the estimated eigenmode frequencies, there is a significant difference. This can be partly explained due to the lack of damping in the solution, which will naturally decrease the frequencies. However, the damping is low in the cases involving a tip mass. It is therefore not possible to ascribe the frequency deviation to the damping ratio alone. Undermodeling and approximations must be considered as possible sources to the difference in frequency. The tip mass is further considered a lumped mass, but has in practice an inertia different from zero. In a practical point of view, the deviation is not crucial, since eigenfrequencies will be adapted to the real system using system identification methods. Also, the upper modes will not be excited, because the system bandwidth is low.

Based on the spectrum of the data, it was given, that only the first mode has significant impact on the end-effector position. The amplitude of the second mode is a factor of more than 35 smaller than the first, and the approximation $M = 1$ is therefore sufficient for this project.

Appendix B
Journal 2: System identification

As described in chapter 6 about controller design workflow, the unknown model parameters must be estimated offline to guide the controller design. The controller itself must be able to find the parameters online, but in order to verify the estimation process, the parameters must be known in advance. It is therefore necessary to make an accurate offline estimation of the system, which is then used for comparison with estimates from the online process. However, the parameters describing the flexible tool have been accurately measured in appendix A, and they will be applied for comparison instead.

Appendix L described how to generate test signals, that are guaranteed to excite the system within specific frequency ranges as well as allowing the manipulator to move. Movements in the low-frequency range are ideal to estimate the friction parameters of the model, whereas signals with frequency content around the expected frequency of the first eigenfrequency will excite the resonance of the flexible tool. A number of tests will be conducted to gather data sets that contain as much information as possible.

The parameters will be estimated using the different methods, PEM and EKF, described in chapter 8 based on the collected data. Part of the data will be used for validation purposes. The relay tuning method will also be tested, but this requires no test data to function, since it works as a controller. System identification based on PEM and EKF will be treated in sections B.1 and B.2, respectively, whereas the relay tuning will be treated in section B.3. A 100 Ω (gain $\sim$ 1003) gain resistor has been used for the strain gauge amplifier during PEM/EKF experiments and 50 Ω (gain $\sim$ 499) during the RT experiment.

B.1 Method 1: PEM

The setup used for this experiment is the manipulator and tool with associated cRIO control hardware. A random test signal is generated for each actuator on the basis of (L.2) from appendix B, which is repeated here

$$u_i = \sum_{k=1}^{n_k} \alpha'_{i,k}\omega'_{i,k}\cos(\omega'_{i,k}t)$$

All constants of the signal must be selected properly to gain the most information from the estimation process. No method of selection will be given here, since several constants will be tested until suitable motion is captured from the manipulator. The selection of the constants is also following the ranges given in (L.3). A total of 8 tests will be conducted with the constants being interchanged between each experiment. Each excitation is granted 20 seconds of run time and a total of 2000[1] data points will be gathered from each sub-experiment per channel, which is listing strain, acceleration, angular velocity and joint angle. The division of data is 50 % (from each sub-experiment) for estimation and validation. A PEM algorithm will be configured with an absolute tolerance and a relative tolerance of 0,0001. This is assumed to provide significant estimates.

The constants for generating the test signals were automatically generated within the given bounds from (L.3) and listed in table B.9 on page 158 (unused coefficients are shown in gray). A LabVIEW diagram has been constructed to generate the needed test signals. Figure B.1 shows one out of 8 cases of the diagram used for one axis.

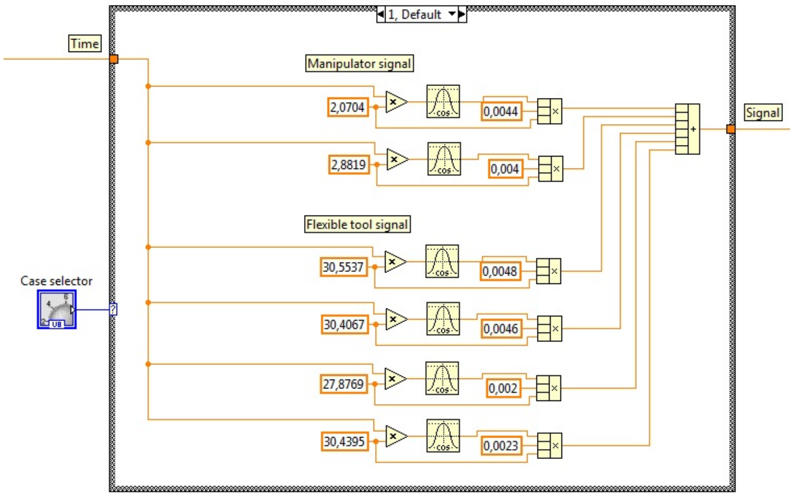


Figure B.1: *LabVIEW signal generation diagram for system identification*

All measured channels are used to estimate different parts of the model. The joint angles are used to estimate the joint frictions, and the strain is used to estimate tool damping and resonance frequency of the first mode of vibration.

All collected data will first be applied for system identification using the PEM, which will provide a basis for the EKF in section B.2. The model parameters are

[1]The number of data points is actually 2001 because the channels are also sampled for $t = 0$

then estimated, and the performance of the methods is compared to one another in the conclusion at the end of the chapter.

B.1.1 Results using PEM

When applying the prediction error method, the measurements are fitted to a nonlinear gray-box model (known structure, unknown parameters). The structure of the gray-box model is the model from (5.6) given as

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 q_1 & -2\zeta_1\omega_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \\ 0 \\ K_2 \end{bmatrix} u(t) \\
\begin{bmatrix} q_1 \\ \theta_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} q_1 \\ \dot{q}_1 \\ \theta_3 \\ \dot{\theta}_3 \end{bmatrix}
\end{aligned} \tag{B.1}$$

with an slightly changed output matrix. Only the joint angle θ_3 and the strain equivalent q_1 is needed. This model can be applied directly for parameter estimation of ω, ζ, K_1 and K_2 (notice that the index has been omitted). The latter two parameters are denoting the gain from the input signal to the first and second part of the model, respectively. Table B.1 lists the PEM-algorithm settings in MATLAB.

Absolute tolerance	0,0001
Relative tolerance	0,0001
Initial states	$q(0) = 0,\ \dot{q} = 0$
Initial parameters	$\omega = 26{,}39,\ \zeta = 0{,}0023,\ K_1 = 100$

Table B.1: *Configuration of PEM algorithm in journal 2*

A problem, however, occurred when fitting the model to the joint angle measurements. This part of the model is a second-order integrator, but the PEM-algorithm kept fitting a straight line. Because the model in (B.1) *consists of two decoupled models, the manipulator part is fitted by integrating the joint angle measurements twice and then scale them by a given K_2 until the error is minimized.*

The resulting parameters are listed in table B.2, but for reasons presented in remark 4 on page 111 only two sub-experiments were performed on the third axis alone. Therefore, the model is only containing θ_3 instead of θ_1 to θ_3.

The offsets $q_{1,\mathsf{off}}$ and $\theta_{3,\mathsf{off}}$ are used to reset the states before initiating a new experiment with the tool being manually located in a horizontal position. NRMSE_1 and NRMSE_2 are *normalized root mean square errors* provided by MATLAB as a measure of the goodness of fit for the first three parameters and the K_2-parameter, respectively. The shorting *inc.* is denoting increments of the joint encoder.

Figure B.2 shows the goodness of fit measure NRMSE as a function of the gain K_2. The maximum points of the curves provide the K_2-parameters resulting in the best fit to the measurement data.

PEM results	Data set 1	Data set 2
ω_1	26,7165 rad/s (4,2521 Hz)	26,7119 rad/s (4,2513 Hz)
ζ_1	0,0119	0,0117
K_1	74,4233	72,2914
K_2	0,6934	0,8874
$q_{1,\mathsf{off}}$	-1,4897 V	-1,9214 V
$\theta_{3,\mathsf{off}}$	-0,9482 rad (-60363 inc.)	-1,4207 rad (-90447 inc.)
NRMSE_1	79,38 %	79,41 %
NRMSE_2	87,10 %	80,89 %

Table B.2: *Estimated parameters and settings using PEM in journal 2*

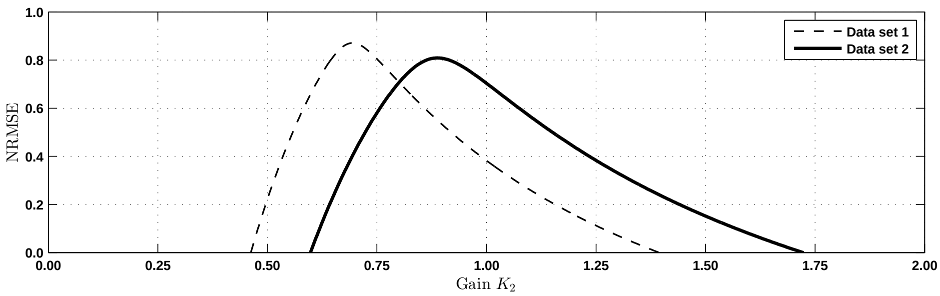


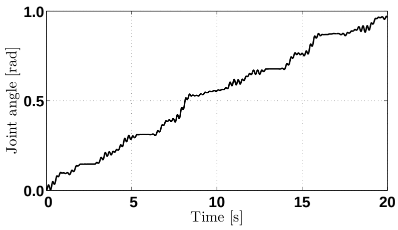
Figure B.2: *NRMSE as a function of the gain K_2*

Because only two data sets have been sampled, it has been chosen to use the entire first data set as validation data. A model is fitted using the second data set, and the parameters from that process are selected as the final estimation. Table B.3 shows the goodness of the fitting for the validation process.

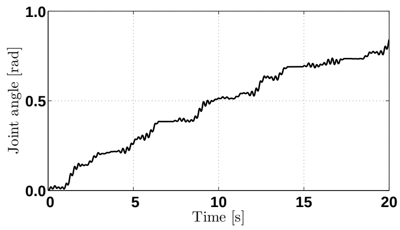
PEM validation	Strain	Joint angle
Data set 1 (validation)	76,66 %	38,08 %

Table B.3: *Goodness of PEM estimation in journal 2*

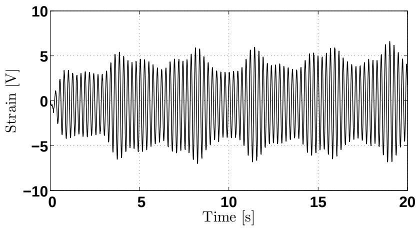
Figures B.3a, B.3c and B.3e show the joint angle measurements, strain measurements and corresponding control signal from first sub-experiment, respectively. Second sub-experiment is shown in figures B.3b, B.3d and B.3f. All data sequences are sampled at 100 Hz and the experiments are running for 20 seconds.



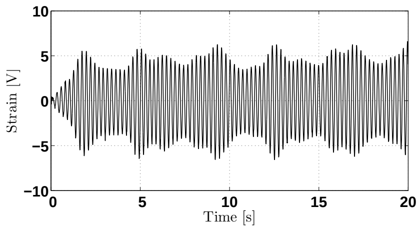
(a) *Joint angle measurement, case 1*



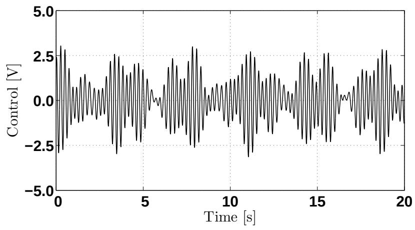
(b) *Joint angle measurement, case 2*



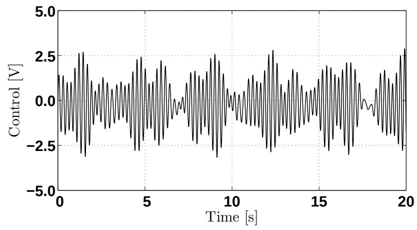
(c) *Strain measurement, case 1*



(d) *Strain measurement, case 2*



(e) *Control signal, case 1*

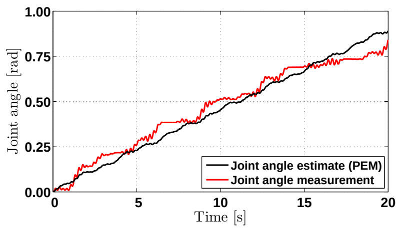


(f) *Control signal, case 2*

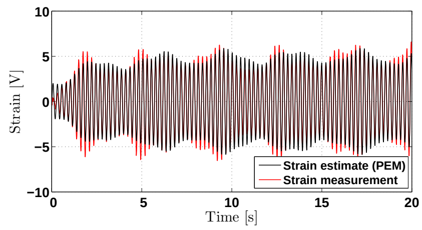
Figure B.3: *Joint angles and strain measurements from experiments in journal 2*

Results from the fitting process are shown in figure B.4. The goodness of fit is high when looking at the eigenmode oscillations in figure B.4b and B.4d using both estimation and validation data. However, the amplitudes are are not following the exact envelope of the measurements. This may be caused by an inadequate model because the model is basically an approximation that solves the model equations mathematically.

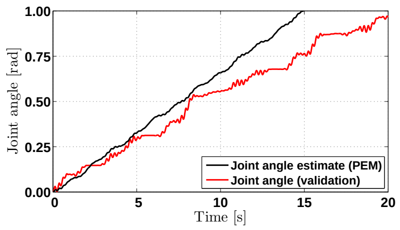
The joint angle estimation is more problematic. Due to an unexpected and significant gravitational effect on the third manipulator link, the joint angle measurements are drifting towards the downward position. No dynamics have been included in the manipulator model, because the robot is not operated at high velocities.



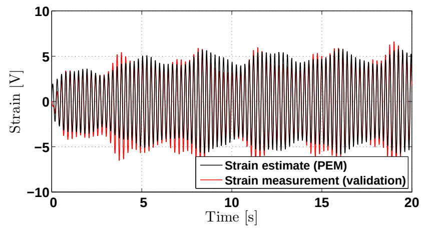
(a) *Joint angle estimation*



(b) *Strain estimation*

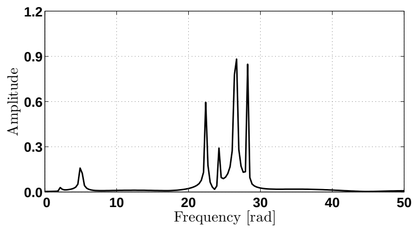


(c) *Joint angle validation*

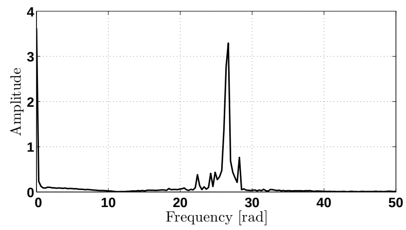


(d) *Strain validation*

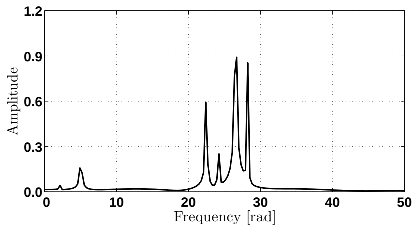
Figure B.4: *Comparison between estimation data and validation data*



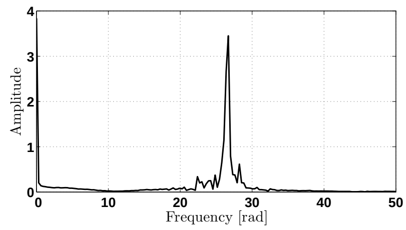
(a) *Control signal, case 1*



(b) *Strain signal, case 1*



(c) *Control signal, case 2*



(d) *Strain signal, case 2*

Figure B.5: *Control signal and strain measurements frequency responses*

A couple of frequency responses have been generated on the basis of both the control signals and the strain measurements. The responses are given in figure B.5. From appendix A the flexible tool eigenfrequency of the first mode was measured to 26,39 rad/s (4,20 Hz). Both frequency responses in figure B.5b and B.5d show a significant resonance peak at 26,69 rad/s (4,25 Hz), which is very close to the 4,20 Hz measurement. The resonance has even been excited by data set 1 with the closest frequency of 27,8760 rad/s (4,43 Hz). A final evaluation of the prediction error method is given in the conclusion at the end of this chapter.

Next up is the EKF, which will be working on the same data as the PEM. This will show the potential of the EKF as a method for online system identification. A final comparison is given between the methods at the end of the chapter.

B.2 Method 2: EKF

To verify the accuracy of an extended Kalman filter when compared with the prediction error method and relay tuning results, the method is tested using the measurements from the first experiment. Because the EKF is iterative, the parameter estimates are also changing for each iteration. In order to compare the results with the results from using PEM, the last estimate after 20 seconds of estimation will be used. An augmented version of the system from (B.1) must be used, and is expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x \\ x_p \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{I}_4 \end{bmatrix}\begin{bmatrix} x \\ x_p \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \bar{0}_4 \end{bmatrix} u(t)$$
$$\begin{bmatrix} q_1 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}\begin{bmatrix} x \\ x_p \end{bmatrix}$$

with $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ denoting the original system matrices from (C.1). x is expressing the original state vector for x_p the parameter vector augmentation. The Jacobians are then given as

$$\mathbf{J}_f = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\omega_1^2 & -2\zeta_1\omega_1 & 0 & 0 & -2\zeta_1\dot{q}_1 - 2\omega_1 q_1 & -2\omega_1\dot{q}_1 & u & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & u \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$
$$\mathbf{J}_g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The Euler discretization method proved instable to make a discrete time model, and a zero-order hold was automatically derived by MATLAB instead. A more thorough description of the theory behind the filter is given in section 8.1 and will not be presented here. Instead the results will be given in the sequel.

B.2.1 Results using EKF

Based on the two data set introduced in section B.1 and shown in figure B.3, the EKF has been applied to estimate the model parameters. In a practical case, the updated model will be used by the controller, and the estimates will be closer to the actual states. Table B.4 shows the estimation results using the EKF. Figures B.6 and B.7 show the estimation trends for all entries of the augmented state vector.

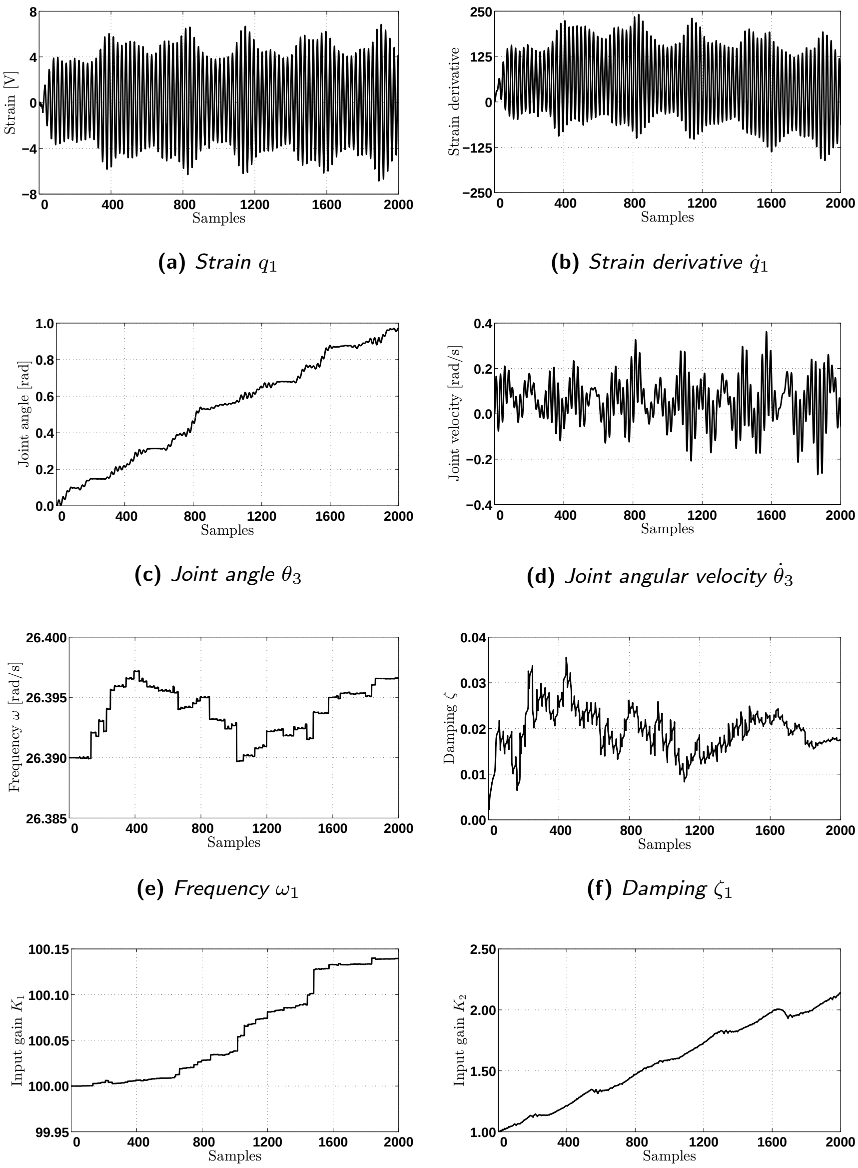
EKF results	Data set 1	Data set 2
ω_1	26,3963 rad/s (4,2011 Hz)	26,3938 rad/s (4,2007 Hz)
ζ_1	0,0170	0,0141
K_1	100,1383	100,0003
K_2	2,0589	2,2031
$q_{1,\mathrm{off}}$	-1,4897 V	-1,9214 V
$\theta_{3,\mathrm{off}}$	-0,9482 rad (-60363 inc.)	-1,4207 rad (-90447 inc.)

Table B.4: *Estimated parameters and settings using EKF*

Measurement offsets are similar to the ones from table B.4. The results are not unique, since the selection of Q, R and x_0 affects the outcome. For this particular estimation process, the filter was configured as listed in table B.5. A final evaluation of the filter and a comparison with the PEM/RT methods will be given in the conclusion at the end of the journal.

Process covariance matrix Q	$\mathrm{diag}\{1 \quad 10 \quad 10 \quad 100 \quad 10 \quad 10000 \quad 1000 \quad 1000\}$
Sensor covariance matrix R	$\mathrm{diag}\{1 \quad 10\}$
Initial states x_0	$q_1(0) = 0,\ \dot{q}_1 = 0,\ \theta_3 = 0,\ \dot{\theta}_3 = 0$
Initial parameters	$\omega_1 = 26{,}39,\ \zeta_1 = 0{,}0023,\ K_1 = 100,\ K_2 = 1$
Augmented state vector	$\begin{bmatrix} x^\intercal & x_p^\intercal \end{bmatrix}^\intercal = \begin{bmatrix} q_1 & \dot{q}_1 & \theta_3 & \dot{\theta}_3 & \omega_1 & \zeta_1 & K_1 & K_2 \end{bmatrix}$

Table B.5: *Configuration of EKF in journal 2*



(a) *Strain* q_1

(b) *Strain derivative* $\dot{q}_1$

(c) *Joint angle* θ_3

(d) *Joint angular velocity* $\dot{\theta}_3$

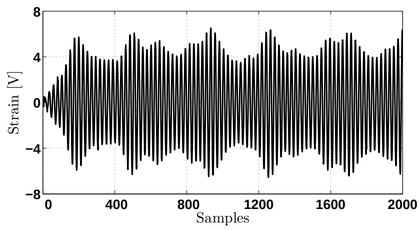
(e) *Frequency* ω_1

(f) *Damping* ζ_1

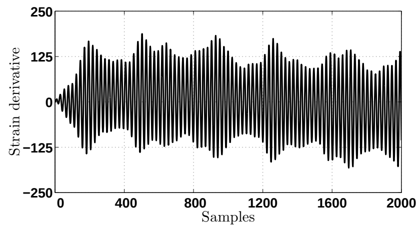
(g) *Input gain* K_1

(h) *Input gain* K_2

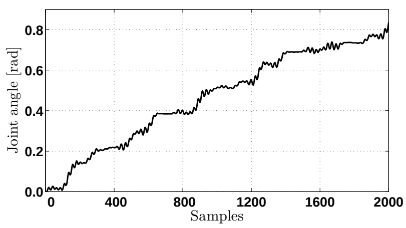
Figure B.6: *EKF estimates for data set 1*



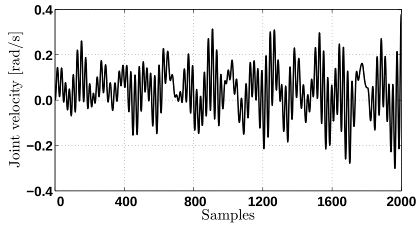
(a) *Strain* q_1



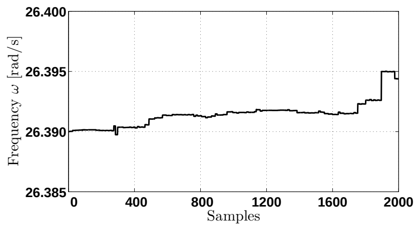
(b) *Strain derivative* $\dot{q}_1$



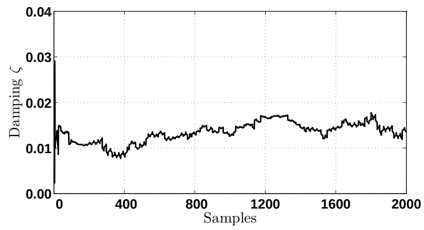
(c) *Joint angle* θ_3



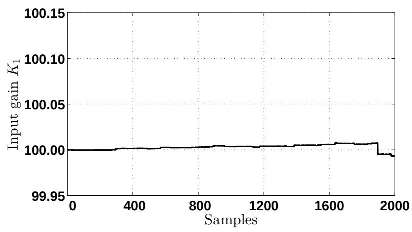
(d) *Joint angular velocity* $\dot{\theta}_3$



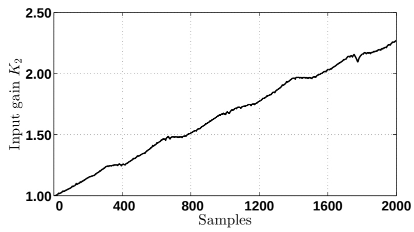
(e) *Frequency* ω_1



(f) *Damping* ζ_1



(g) *Input gain* K_1



(h) *Input gain* K_2

Figure B.7: *EKF estimates for data set 2*

B.3 Method 3: RT

As a substitution for the extended Kalman filter, the relay tuning can be applied. The method is not able to identify a system in general, but will estimate a single point on the Nyquist plot corresponding to a limit cycle oscillation frequency. The resulting frequency can then be used to estimate the resonance frequency, because the model structure is already assumed. A thorough introduction to the relay tuning process is given in chapter 8.4. Figure B.8 shows the closed-loop used for the relay tuning process. A gain block is used to amplify the relay signal to a given amplitude, which in this experiment will be 1 V or 5 V (found empirically).

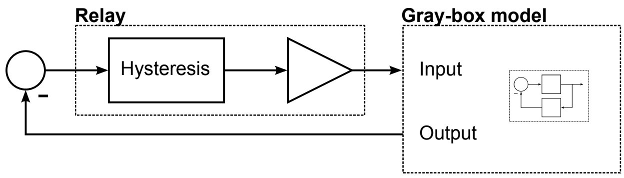


Figure B.8: *Relay tuning block diagram*

The system will act as a gray-boxmodel, with the structure known and the parameters unknown. By relating the input and output signals to each other, the parameters can be estimated. A piece of code, see figure B.9, makes up for the relay with hysteresis, and the gain is chosen in such a way, that the resonance amplitude will not damage the tool. In this case 1 V and 5 V will be selected from a maximum of 10 V.

```
if y == b
  if x < -h
    y = -b
  else
    y = b
  end
elseif y == -b
  if x > h
    y = b
  else
    y = -b
  end
end
```

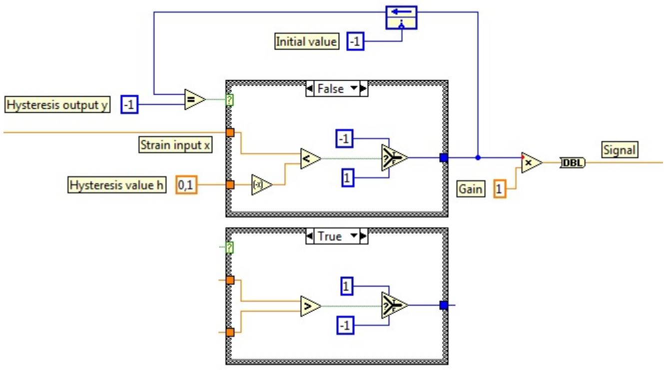


Figure B.9: *Relay code*

Figure B.10: *LabVIEW implementation of relay with hysteresis*

The code uses x as the input to the relay, y as the previous relay output, h as the hysteresis width and b as the gain. None of the parameters must be confused with formerly used variables in the report, and the parameters are not noted in the nomenclature. In order to use the code in LabVIEW, a graphical interpretation of

the code must be generated as shown in figure B.10. Both the TRUE case and the FALSE case are shown in the same figure.
The relay with hysteresis from figure B.10 has been used to control the system, and the results are given in the sequel section.

B.3.1 Results using RT

A relay with hysteresis has been programmed in LabVIEW and is included in a closed-loop together with the manipulator. Before proceeding with the results from the experiments, a remark on the configuration of the servo amplifier controlling the third axis actuator is presented in remark 6 below.

> **Remark 6** (servo amplifier configuration)**:** *The relay tuning experiments are carried out using the built-in velocity control configuration of the servo amplifiers. To avoid damaging the tool during experiment, the direct acceleration configuration is omitted. An initial test revealed, that the manipulator axis is accelerated to an unacceptable high angular velocity before the strain measurement changes sign. Using either of the configurations, the relay tuning automatically tunes the control signal to a frequency similar to the eigenfrequency of the tool.* □

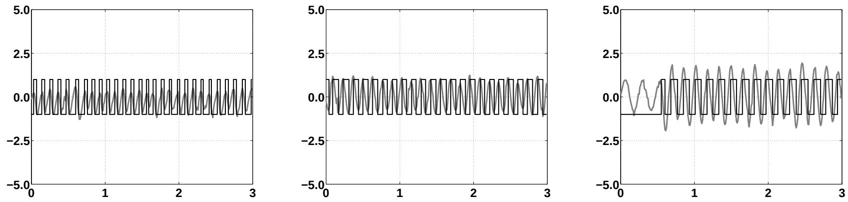
Two different gains and three different hysteresis settings were tested during the experiments. Tables B.7 and B.8 as well as figures B.12 and B.13 on the following pages summarizes the eigenfrequency estimation results. The term *last* from the tables refers to the last estimation in the trends. As the number of samples increases so does the precision of the estimate. The last sample in this case is therefore based on 20 seconds of data material.

As shown in figure 8.6 from section 8.4, the control signal and the strain measurement will eventually synchronize and oscillate with a frequency similar to the resonance frequency of the flexible tool subject to conversion to achieve exact frequency. The plots in figure B.12 show the estimation trend when more data is available. Common to the trends is the fast converging to the assumed correct resonance frequency of 4,20 Hz no matter the gain and hysteresis. After some additional samples, the estimates become more uncertain.

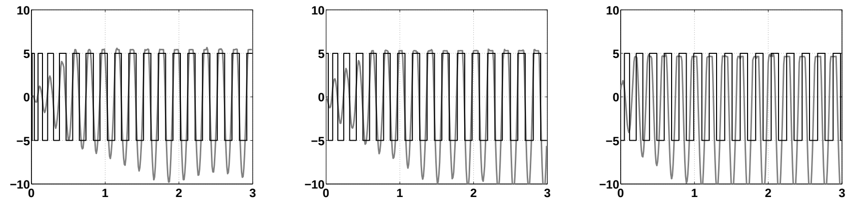
A frequency response is calculated on the basis of the output signal, and the most significant frequency component is used as the resonance estimate (before conversion). This frequency and associated amplitude correspond to a point on the system locus, and the natural frequency of the system can be estimated from this coordinate using the method described in section 8.4. An example of the resolution and trend of the frequency response of a strain signal is shown in figure B.14.

All trends in figure B.12 converge to a constant estimate after at least 708 samples (7,08 seconds) for the trend in figure B.12a. However, by using a gain of 5 will ensure a faster convergence with no significant steady state error when compared to using a unit gain. The error is within 8-11 % of the 4,20 Hz measurement in appendix A.

Figure B.13 shows the trend of the frequency of the relay output. Because of the dominating resonance peak in the linear dynamics of the flexible tool, the relay output will be very close to the actual resonance frequency of the tool. The first 4 seconds of the relay outputs and strain measurements for each experiment is shown in figure B.11.



(a) *Gain = 1 and h =0,1* **(b)** *Gain = 1 and h =0,5* **(c)** *Gain = 1 and h =1,0*



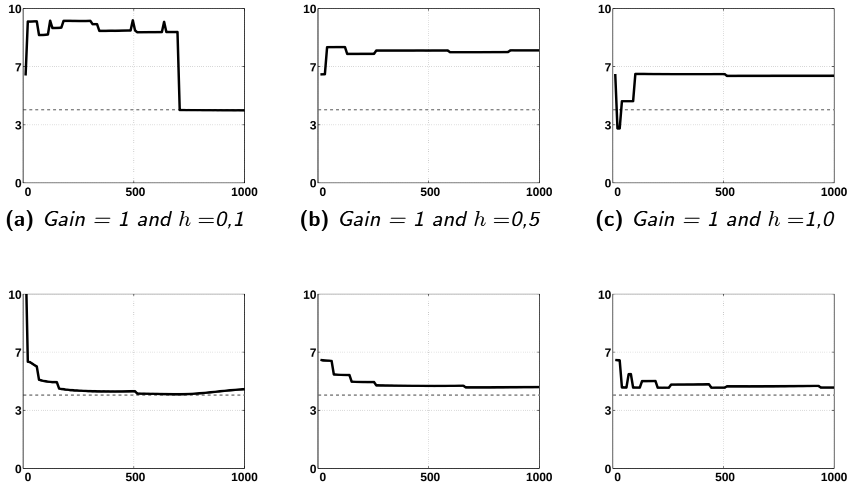
(d) *Gain = 5 and h =0,1* **(e)** *Gain = 5 and h =0,5* **(f)** *Gain = 5 and h =1,0*

Figure B.11: *Relay tuning control signal progress (axes are given in time [s] and amplitude [V])*

Each strain measurement has been biased to obtain zero strain for $t = 0$. The offsets are given in table B.6. A positive strain is measured by the strain gauges due to gravity acting on the tip load. It is assumed, that the margin of the strain gauges remains unchanged even though the gauges are offset to one direction.

Offset	Gain = 1	Gain = 5
$h = 0{,}1$	-4,39062 V	-2,88477 V
$h = 0{,}5$	-4,03076 V	-3,00293 V
$h = 1{,}0$	-4,11328 V	-3,65576 V

Table B.6: *Strain measurement offsets*



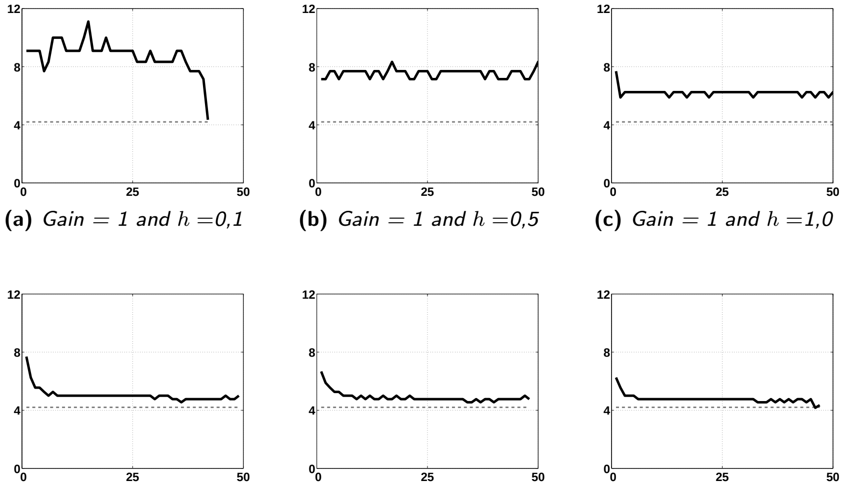
(a) *Gain = 1 and h =0,1* (b) *Gain = 1 and h =0,5* (c) *Gain = 1 and h =1,0*

(d) *Gain = 5 and h =0,1* (e) *Gain = 5 and h =0,5* (f) *Gain = 5 and h =1,0*

Figure B.12: *Eigenfrequency estimations progress (axes given in samples and frequency [Hz])*

Eigenfrequency	Min	Max	Last	Error
Gain = 1				
$h = 0{,}1$	4,1632 Hz	9,3287 Hz	4,1632 Hz	-0,8752 %
$h = 0{,}5$	6,2244 Hz	7,7923 Hz	7,6032 Hz	81,0284 %
$h = 1{,}0$	3,1264 Hz	6,2512 Hz	6,1472 Hz	46,3610 %
Gain = 5				
$h = 0{,}1$	4,2468 Hz	12,3574 Hz	4,5399 Hz	8,0918 %
$h = 0{,}5$	4,6423 Hz	6,2228 Hz	4,6612 Hz	10,9807 %
$h = 1{,}0$	4,6249 Hz	6,2149 Hz	4,6344 Hz	10,3429 %

Table B.7: *Estimated eigenfrequencies using relay tuning method*



(a) *Gain = 1 and h =0,1* (b) *Gain = 1 and h =0,5* (c) *Gain = 1 and h =1,0*

(d) *Gain = 5 and h =0,1* (e) *Gain = 5 and h =0,5* (f) *Gain = 5 and h =1,0*

Figure B.13: *Eigenfrequency estimations progress (axes given in oscillations and frequency [Hz])*

Eigenfrequency	Min	Max	Last	Error
Gain = 1				
$h = 0{,}1$	4,3478 Hz	11,1111 Hz	4,3478 Hz	3,5197 %
$h = 0{,}5$	7,1429 Hz	8,3333 Hz	7,6923 Hz	83,1502 %
$h = 1{,}0$	5,8824 Hz	7,6923 Hz	6,2500 Hz	48,8095 %
Gain = 5				
$h = 0{,}1$	4,5455 Hz	7,6923 Hz	5,0000 Hz	19,0476 %
$h = 0{,}5$	4,5455 Hz	6,6667 Hz	4,7649 Hz	13,3787 %
$h = 1{,}0$	4,1667 Hz	6,2500 Hz	4,3478 Hz	3,5197 %

Table B.8: *Estimated eigenfrequencies using relay tuning method (direct estimation)*

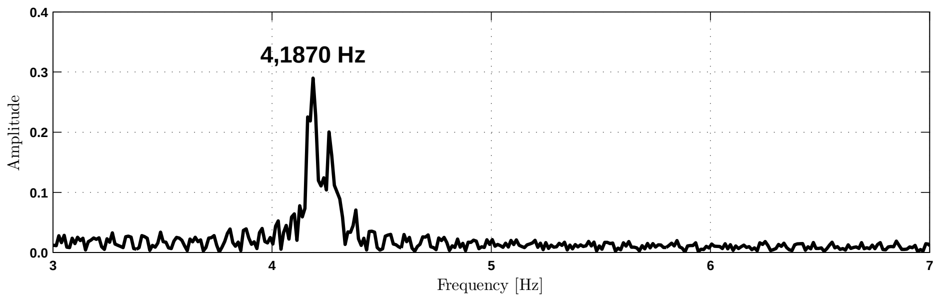


Figure B.14: *Frequency response for gain = 1 and $h = 0{,}2$ (determined using all data)*

Because the resonance peak is more dominating than the remaining frequency components, the peak can be detected by searching the highest amplitude over the entire frequency range. This ends the relay tuning method for system identification, and the method will be evaluated and compared with the EKF/PEM methods in the sequel section.

B.4 Conclusion

In order to verify the applicability and accuracy of the different parameter estimation methods presented in the latter three sections, the methods will be compared based on the three criteria *accuracy*, *stability* and *convergence time*. The first criterion is based on the flexible tool parameters measured in appendix A and how close the estimates are to those values. Second criterion is concerning how the estimation fluctuates over time, and last criterion considers the convergence time of the estimation processes to whether or not the methods are useful in a practical setup. In the following three subsections the methods PEM, EKF and RT are evaluated.

B.4.1 Prediction error method (PEM)

The method is accurate in terms of estimating ω_1 because the resonance is very dominating. Even though the dynamics are not excited with the resonance frequency, the primary frequency component of the frequency response is still the most dominating. This evaluation is only based on highly underdamped systems like in this case, because systems with higher damping may behave differently (the resonance peak is not that dominating). The damping ratio ζ_1 is, however, a factor of 10 higher that the measurement from appendix A and the amplitudes are not following the envelope of the measurements. The input gains can be fitted to follow the overall trend, but not the fast responses. Based on the application in this case, the accuracy of the resonance is the most important, and an overall ~ 80 % goodness of fit is satisfactory.

Because the method works on a large data sample, a cost function can be minimized to provide the best possible estimates. In case the estimates become too small/large compared to the physics involved, the PEM configuration can be adjusted. Convergence time is not a concern in this case, because the method is intended for offline usage and therefore not built for running in real-time.

B.4.2 Extended Kalman filter (EKF)

Compared with the PEM, the EKF is intended for online applications. The filter has been applied to the same data as the PEM, but the results are not considered reliable. One initial guess can result in parameters very different from another guess. Also, the covariance matrices Q and R are very important for the outcome of the filter. Stability of a nonlinear Kalman filter cannot be guaranteed, which is important for online filters. If the filter suddenly becomes instable, the flexible tool may be excited with a resonance frequency. This decreases accuracy and settling time of the TCP, and the tool may get damaged if the oscillations become too large.

Because the filter must be executed in parallel with the realtime system, it must be fast to estimate parameters. Unfortunately, since the filter may not converge at all, the parameter estimates may change all the time. Based on this experiment, the EKF is not preferable for this control system. The resonance frequency cannot diverge from the actual value in order to achieve satisfactory results and respect the requirements.

B.4.3 Relay tuning (RT)

As an alternative to the online EKF, the relay method has been considered. The method is fast enough to work as a semi-online estimator but also reliable to function offline. When working in semi-online mode, the method is required to apply control action to the manipulator actuator. However, because the method is fast, this is not a problem. The method is accurate because it excites the resonance, which is very dominant in this case. A relay with hysteresis has been used to control the actuator, and based on the describing function method, the relay output switching frequency is converted into the natural frequency of the linear dynamics. Even though the DF-method has some undesirable properties for underdamped systems, it provides accurate estimates in this experiment. The estimates are within 8-11 % of the 4,20 Hz measurement from appendix A when using the best configuration (gain = 5).

Based on the three different experiments documented in this journal, the relay tuning methods is preferable for several reasons. Compared with the PEM, the method can be running almost in realtime and provide estimates of the resonance frequency. The PEM is more general, and is capable of estimating all model parameters. However, the offline property is not preferable in cases with arbitrary loads. In comparison with the online EKF, the relay tuning will be slower, because is cannot work in parallel with the remaining control system. The stability and accuracy issues of the EKF are degrading performance, and the RT method is therefore the best candidate to estimate the resonance frequency. If other parameters of the model needs to be estimated, then the PEM must be applied.

Test no.	#01	#02	#03	#04	#05	#06	#07	#08
Manipulator link $i = 1$								
$\alpha'_{1,1}$	0,0050	0,0039	0,0049	0,0023	0,0037	0,0050	0,0037	0,0043
$\alpha'_{1,2}$	0,0017	0,0042	0,0043	0,0041	0,0033	0,0017	0,0046	0,0020
$\alpha'_{1,3}$	0,0014	0,0019	0,0029	0,0045	0,0012	0,0021	0,0018	0,0019
$\alpha'_{1,4}$	0,0022	0,0024	0,0033	0,0035	0,0024	0,0045	0,0040	0,0035
$\alpha'_{1,5}$	0,0031	0,0035	0,0030	0,0045	0,0035	0,0016	0,0036	0,0023
$\alpha'_{1,6}$	0,0038	0,0033	0,0029	0,0051	0,0038	0,0040	0,0014	0,0022
$\omega'_{1,1}$	2,7955	4,1474	3,7093	6,0025	2,9535	3,6541	1,6110	3,1856
$\omega'_{1,2}$	5,7689	2,9597	5,2009	3,7654	4,6171	3,1195	2,2372	4,7669
$\omega'_{1,3}$	27,7091	22,5417	29,7926	29,0530	24,2378	26,8923	31,0146	26,8139
$\omega'_{1,4}$	23,3722	26,3847	21,4085	22,4928	23,3904	29,0426	21,1891	22,3395
$\omega'_{1,5}$	23,5934	22,6929	30,1954	21,4166	28,2899	22,0476	24,9866	31,4931
$\omega'_{1,6}$	24,3260	31,0967	30,7956	27,5405	22,1979	27,8457	23,7464	29,7323
Manipulator link $i = 2$								
$\alpha'_{2,1}$	0,0027	0,0052	0,0023	0,0031	0,0027	0,0014	0,0044	0,0049
$\alpha'_{2,2}$	0,0037	0,0045	0,0014	0,0015	0,0032	0,0051	0,0042	0,0034
$\alpha'_{2,3}$	0,0033	0,0029	0,0038	0,0042	0,0046	0,0013	0,0013	0,0042
$\alpha'_{2,4}$	0,0020	0,0016	0,0037	0,0030	0,0044	0,0014	0,0015	0,0021
$\alpha'_{2,5}$	0,0017	0,0044	0,0048	0,0039	0,0043	0,0017	0,0045	0,0049
$\alpha'_{2,6}$	0,0034	0,0050	0,0038	0,0018	0,0039	0,0014	0,0041	0,0043
$\omega'_{2,1}$	5,6298	4,6767	5,7241	4,5544	1,9656	2,8814	4,3119	1,0280
$\omega'_{2,2}$	5,6869	2,6765	2,5222	5,6005	1,6493	1,3797	5,3250	0,8573
$\omega'_{2,3}$	27,1029	30,6335	24,7874	30,5579	25,9965	25,2825	21,1149	31,1799
$\omega'_{2,4}$	21,1522	29,3829	31,1961	23,1082	21,1222	26,6253	22,2349	27,9328
$\omega'_{2,5}$	23,6962	28,4648	31,0243	21,2074	26,5472	23,7594	23,6407	23,7330
$\omega'_{2,6}$	29,8266	30,4682	23,8211	30,8114	31,6343	28,6002	25,1973	29,8733
Manipulator link $i = 3$								
$\alpha'_{3,1}$	0,0044	0,0016	0,0014	0,0021	0,0011	0,0051	0,0022	0,0048
$\alpha'_{3,2}$	0,0040	0,0043	0,0041	0,0042	0,0021	0,0036	0,0045	0,0044
$\alpha'_{3,3}$	0,0048	0,0011	0,0018	0,0025	0,0013	0,0023	0,0036	0,0031
$\alpha'_{3,4}$	0,0046	0,0029	0,0011	0,0014	0,0051	0,0025	0,0015	0,0014
$\alpha'_{3,5}$	0,0020	0,0028	0,0015	0,0022	0,0041	0,0013	0,0045	0,0030
$\alpha'_{3,6}$	0,0023	0,0048	0,0023	0,0023	0,0050	0,0036	0,0049	0,0043
$\omega'_{3,1}$	2,0704	2,2158	0,9214	1,7978	1,7313	4,4197	3,4484	1,9927
$\omega'_{3,2}$	2,8819	5,0470	1,9215	3,7829	3,5820	3,1120	5,2560	3,0289
$\omega'_{3,3}$	30,5537	24,1688	22,8582	29,9978	30,8780	25,3156	27,8713	23,3167
$\omega'_{3,4}$	30,4067	28,2584	28,2925	22,7333	30,6814	26,2884	22,2388	22,2559
$\omega'_{3,5}$	27,8769	22,4733	28,2915	27,4075	26,6223	25,8358	24,7653	27,5120
$\omega'_{3,6}$	30,4395	26,5478	30,4599	28,8549	24,9007	31,5894	25,5559	26,2737

Table B.9: *Parameters used for experimental test signals in journal 2*

Appendix C

Journal 3: IMU response

The IMU is used to measure the acceleration and change in orientation of the tool frame $\mathbb{F}_\tau$. Before using the measurements, the response is compared to the model. Two things must be achieved from this journal. Firstly, the noise distributions of the IMU outputs are measured, which are used in the covariance matrix for the extended Kalman filter. Secondly, when the model was derived in chapter 3, the acceleration and change in orientation was explicitly determined. IMU measurements and the approximated model response to identical excitation signals will be compared. If they are significant, the model can be used to represent the reality. The remainder of this journal will be divided into two parts - one for each of the two topics to examine. Similarly, two result sections will be present.

C.1 Noise distributions

A Kalman filter requires knowledge of the sensor variance in order to determine the probability of fit correctness. When integrating e.g. position state using noisy sensors, the variance increases with time, because the distribution flattens out. It is therefore preferable to have accurate estimates of the sensor variances prior to applying the filter. The IMU consists of three accelerometers (x, y and z-direction) and three gyroscopes (yaw, pitch and roll). Both sensor types provide zero-rate outputs when the sensor is in steady-state. The sensor is therefore located on a steady table, and the zero-rate signals are sampled for 1 ms. Afterwards, the distribution of the noise floor can then be estimated using a histogram. A list of the necessary equipment for the test is given in table C.1.

GW Instek GDS-1062A 60 MHz digital oscilloscope
SparkFun inertial measurement unit (IMU) board
Caltek PSM3/2A power supply

Table C.1: *Equipment used for noise estimation in journal 3*

Notice that the gyroscope is capable of providing amplified outputs with a factor of 1 or 4. Both settings are measured in the experiment. The digital storage oscilloscope is used to sample the noise signals, and the oscilloscope setup provides the best results when setup as given in table C.2.

Sample count	2 millions
Vertical scale	100 mV/div
Horizontal scale	100 μs/div
Time	1 ms

Table C.2: *Oscilloscope setup for experiment in journal 3*

A set of data has been sampled for each of the 9 IMU output channels in steady-state, and the data is processed in the sequel section. When the next experiment has been performed, a combined conclusion of the experiments will be given.

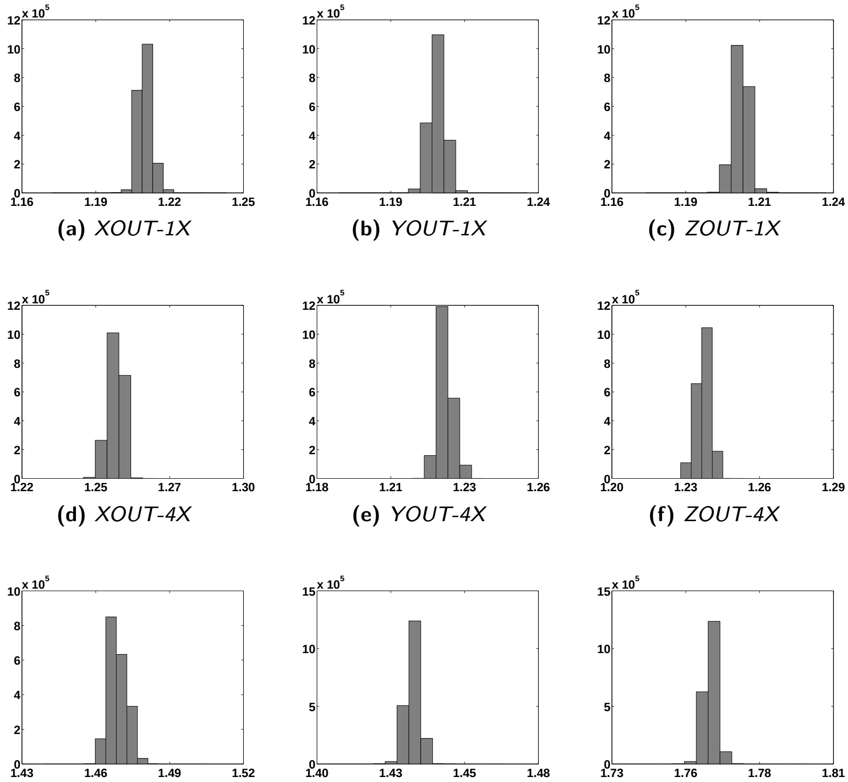
C.1.1 Results

The oscilloscope has sampled the zero-rate signals from all 9 outputs (three accelerometers and three gyroscopes with two amplification modes). Table C.3 shows the resulting sample means and sample standard deviations.

IMU channel	Sample mean	Sample std.
XOUT-1X	1,2110 V	0,0029 V
YOUT-1X	1,2037 V	0,0031 V
ZOUT-1X	1,2052 V	0,0029 V
AXOUT	1,4705 V	0,0038 V
AYOUT	1,4353 V	0,0027 V
AZOUT	1,7669 V	0,0025 V
XOUT-4X	1,2529 V	0,0027 V
YOUT-4X	1,2252 V	0,0027 V
ZOUT-4X	1,2386 V	0,0029 V

Table C.3: *Results from IMU zero-rate measurements in journal 3*

As the results show in table C.3, the sample standard deviations are around 2,9 mV, which is close to one LSB ($\sim$2,4 mV) of the ADC card NI-9201 used to sample the sensors for the controller [69]. The noise is of little concern in this case when considering the sensors alone. A sample distribution is shown graphically in figure C.1 on the adjacent page for each of the IMU channels. The sample variances (squared sample std.) will be included in the covariance matrix R for application in the Kalman filter, see chapter 7. Next up is part 2 regarding model approximation verification.



(a) *XOUT-1X* **(b)** *YOUT-1X* **(c)** *ZOUT-1X*

(d) *XOUT-4X* **(e)** *YOUT-4X* **(f)** *ZOUT-4X*

(g) *AXOUT* **(h)** *AYOUT* **(i)** *AZOUT*

Figure C.1: *IMU zero-rate output noise histograms (first axis is output voltage [V])*

C.2 Verification of model approximations

In order to verify the model approximations from the modeling chapter, a set of data must be collected from the IMU. By exciting the three axes of the manipulator, the IMU will provide read-outs of the corresponding acceleration and change in orientation of the tool frame $\mathbb{F}_\tau$. From the manipulator modeling, the angular velocity (change in orientation) was expressed as

$$\omega_j = {}^j\omega_i + \dot{\theta}_j \mathbf{e}_3 = {}^j_i\mathbf{R}\,\omega_i + \begin{bmatrix}0 & 0 & \dot{\theta}_j\end{bmatrix}^\intercal \tag{C.1}$$

from equation H.9, which corresponds to

$$\omega_0 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \tag{C.2}$$

when expressed in reference to the inertial frame $\mathbb{F}_0$. Similarly, the acceleration was expressed in (3.13) as

$$\upsilon(\dot{\theta},\ddot{\theta},t) = \begin{bmatrix} -\ell_1\dot{\theta}_1^2 - \ell_2(\dot{\theta}_1^2+\dot{\theta}_2^2) - \ell_3(\dot{\theta}_1^2+\dot{\theta}_2^2+\dot{\theta}_3^2) - \cdots \\ \cdots - 2\dot{\theta}_1\dot{\theta}_2(\ell_2+\ell_3) - 2\ell_3(\dot{\theta}_1\dot{\theta}_3+\dot{\theta}_2\dot{\theta}_3) \\ 0 \\ -\ell_1\ddot{\theta}_1 - \ell_2(\ddot{\theta}_1+\ddot{\theta}_2) - \ell_3(\ddot{\theta}_1+\ddot{\theta}_2+\ddot{\theta}_3) \end{bmatrix} \tag{C.3}$$

when keeping the joint angle in a small neighborhood of the operation point $\theta = \bar{0}$. The last entry and the last axis 3 is the only parts used for control, and therefore, the IMU response can be compared in the following way

$$r_{\mathsf{gyro}} \sim \dot{\theta}_3 \quad \text{and} \quad r_{\mathsf{acc}} \sim -\ell_3\ddot{\theta}_3 \tag{C.4}$$

with r_{gyro} and r_{acc} representing the y-axis[1] gyroscope and accelerometer, respectively. Due to the small signal approximation, only the vertical accelerometer is needed. The measurements are performed on the system, when data for the extended Kalman filter is collected. The experimental procedure can therefore be read from appendix B on page 141. Before proceeding with the interpretation of measurement data, a note on sensor noise must be stated. For unexplainable reasons the cRIO analog-to-digital converter card is picking up unacceptable noise levels when the servo amplifiers are powered on. The origin of the problem cannot be determined, and therefore, a number of supporting analysis are given to support the interpretation of the measurements. Table C.4 shows the sample mean $\hat{\mu}$ and sample standard deviation $\hat{\sigma}$ of all 5 sensor channels in both hardware modes (drivers ON and OFF).

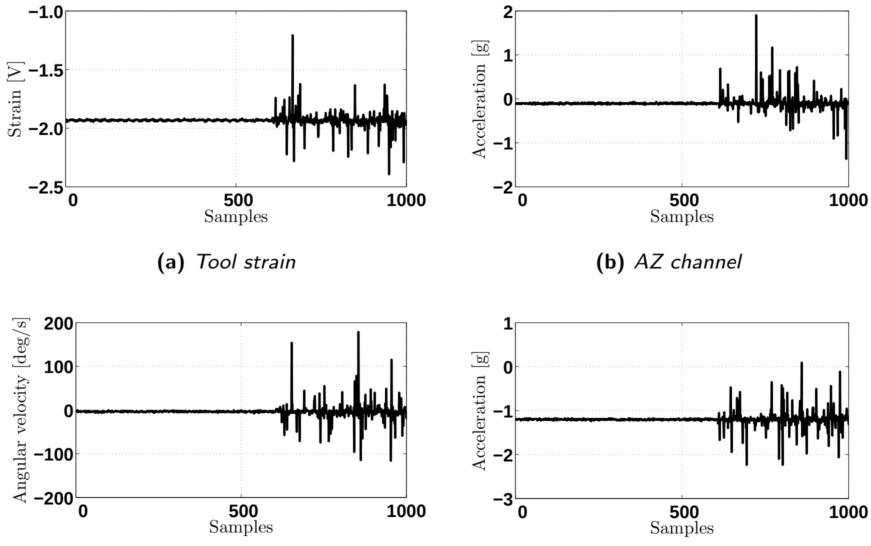
	Driver OFF	Driver ON	Std. ratio
Strain gauges	$\hat{\mu} = -1{,}9325$ V $\hat{\sigma} = 5{,}4499\text{E}{-3}$ V	$\hat{\mu} = -1{,}9343$ V $\hat{\sigma} = 7{,}9172\text{E}{-2}$ V	14,5273
IMU (AZ-output)	$\hat{\mu} = -1{,}0604\text{E}{-1}$ g $\hat{\sigma} = 1{,}2727\text{E}{-2}$ g	$\hat{\mu} = -9{,}0463\text{E}{-2}$ g $\hat{\sigma} = 2{,}1106\text{E}{-1}$ g	16,5840
IMU (AY-output)	$\hat{\mu} = -1{,}2021$ g $\hat{\sigma} = 1{,}2649\text{E}{-2}$ g	$\hat{\mu} = -1{,}1987$ g $\hat{\sigma} = 2{,}1146\text{E}{-1}$ g	16,7169
IMU (GY4-output)	$\hat{\mu} = -3{,}3119$ deg/s $\hat{\sigma} = 1{,}1157$ deg/s	$\hat{\mu} = -3{,}8874$ deg/s $\hat{\sigma} = 2{,}2367\text{E}1$ deg/s	20,0465
Encoder	$\hat{\mu} = 1{,}4208$ rad $\hat{\sigma} \sim 0$ rad	$\hat{\mu} = 1{,}4208$ rad $\hat{\sigma} = 4{,}4733\text{E}{-5}$ rad	-

Table C.4: *Noise levels from 5 sensor channels*

[1]The y-axis on the IMU sensors are not necessarily coinciding with the axis denotations of the manipulator configuration

IMU channels are denoted as AZ (X_0-axis acceleration), AY (Z_0-axis acceleration) and GY4 (Y_0-axis angular velocity, 4x amplification). The only sensor almost unaffected by the switch between ON and OFF mode is the joint angle sensor, which is introducing a negligible sample variance of $\hat{\sigma}^2 = 2{,}0011\text{E}{-9}$. This is also the only digital sensor in the system. Column three lists the sample standard deviation ratio between the two modes, and the values show, that noise levels are increased by as much as 20 times.

The noise levels will have the biggest impact on the IMU channels, where the levels are 5,86 % (AZ channel), 5,87 % (AY channel) and 1,86 % (GY4 channel) of the sensor range. However, since the entire range is not used for the experiments, the signal to noise ratio decreases and corrupts the measurements. Since the problem cannot be localized, the measurements will be analyzed anyway, but with the noise levels in mind. Figure C.3 shows the change in steady-state output response from 4 of the sensors (encoders omitted) during mode switch (OFF to ON).

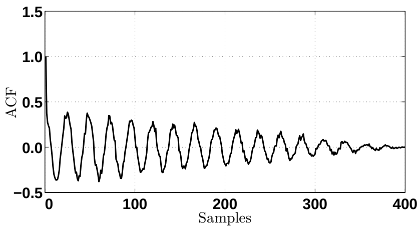


(a) *Tool strain* **(b)** *AZ channel*

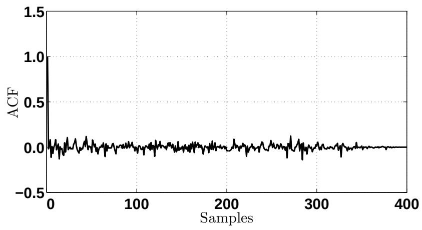
(c) *GY4 channel* **(d)** *AY channel*

Figure C.2: *Steady-state sensor output in hardware OFF and ON mode*

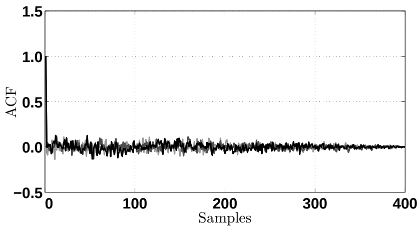
Another important factor to analyze is the noise distribution. Figure C.2 shows the autocorrelation function ACF for both strain measurement and all IMU measurements in a single plot. All ACF's, besides the one from figure C.3a, are respecting 95 % confidence levels, and the samples are therefore assumed to be uncorrelated (white noise) with a 95 % probability.



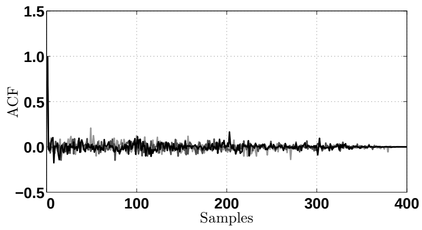
(a) *Strain, driver power OFF*



(b) *Strain, driver power ON*



(c) *IMU, driver power OFF*



(d) *IMU, driver power ON*

Figure C.3: *Sensor noise autocorrelation for strain and IMU measurements*

The analysis is using the signal snippets shown in figure C.2 with the first 400 samples defining the OFF mode and the last 400 samples the ON mode. After introducing the noise levels of the measurement channels the IMU measurements are presented next.

C.2.1 Results

The importance of this test is to verify the model approximations for angular acceleration and angular velocity using accelerometers and gyroscopes, respectively. One measurement is needed, which has been saved from the experiment described in appendix B. This is the joint angle θ of the third manipulator joint as shown in figure C.4 from case 2. A derivative $\dot{\theta}$ is given in figure C.5.

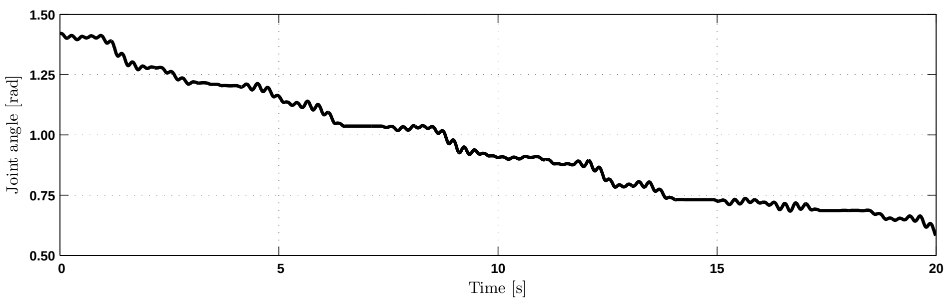


Figure C.4: *Joint angle measurements*

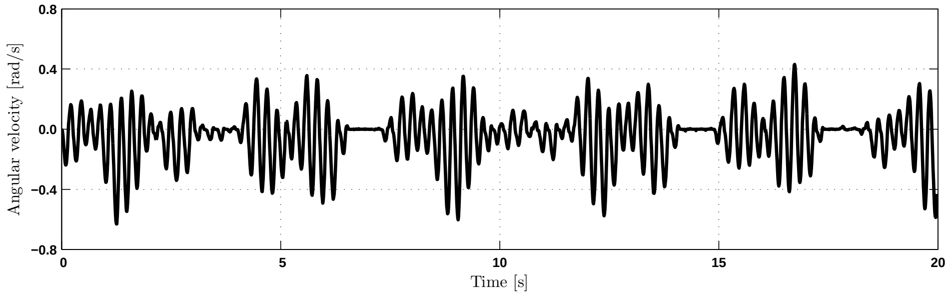


Figure C.5: *Angular velocity estimate (joint angle derivative)*

The data need not to be filtered before differentiation, because the joint angle sensors are not subject to noise (or at least negligible, see table C.4). As a results of the first equation in (C.4), then $r_{\mathsf{gyro}} \sim \dot{\theta}_3$ and the joint angle derivative from figure C.5 must be similar with the gyroscope measurements. Figure C.6 shows the gyroscope measurements from the same experiment that provided the joint angle data.

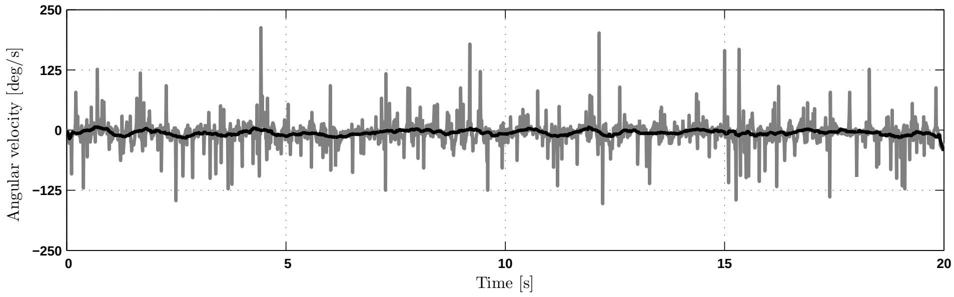


Figure C.6: *Gyroscope measurements*

Due to the GY4 channel noise, the measurements have been filtered by a 50 order MA-filter corresponding to an averaging time of 0,5 s. A combined plot between the joint angle derivative and the filtered gyroscope measurement is given in figure C.7.

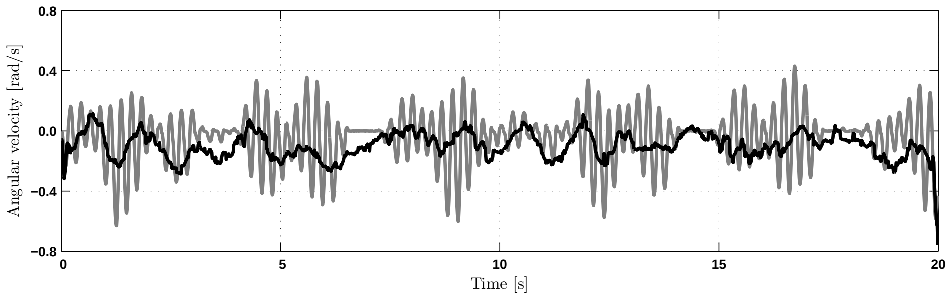


Figure C.7: *Joint angle derivative and filtered gyroscope measurements*

As indicated from the plot, the data are not similar in either trend nor amplitude. The amplitude can be a consequence of the MA-filter, but the trend is due to other elements. Noise levels are assumed to be the significant source of error in this case. Since even the joint angle derivative is smooth, the manipulator joint is not excited with a noisy control signal. Otherwise, the low pass characteristic actuators will filter out high frequency components. A final conclusion to the measurements is given in the conclusion at the end of this appendix.

Besides the gyroscope, the IMU also contains three accelerometers. Similar with the gyroscope measurements, they can be compared with a model approximation from (C.4). In this case, it is $r_{\mathsf{acc}} \sim -\ell_3\ddot{\theta}_3$. Figure C.8 shows the acceleration measurements from both the AY and the AZ-channel.

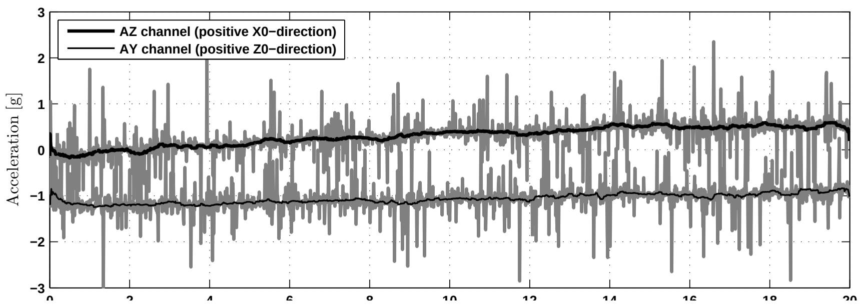


Figure C.8: *Accelerometer measurements*

As was the case with the gyroscope, the data must be filtered to show the trend. The AY-channel is corresponding to the positive Z_0-direction in the inertial frame $\mathbb{F}_0$ and measures the gravitational constant of around 1 g in the initial position, see figure C.8. The AZ-channel on the other hand is located perpendicular to the other accelerometer and measures therefore around 0 g in the initial position. Due to the noise levels, the exact acceleration levels cannot be verified.

By combining the second derivative of the joint angle measurements and the acceleration data, the trends should be similar for the approximation to be valid. Figure C.9 shows the angular acceleration of the joint angle measurements from figure C.4.

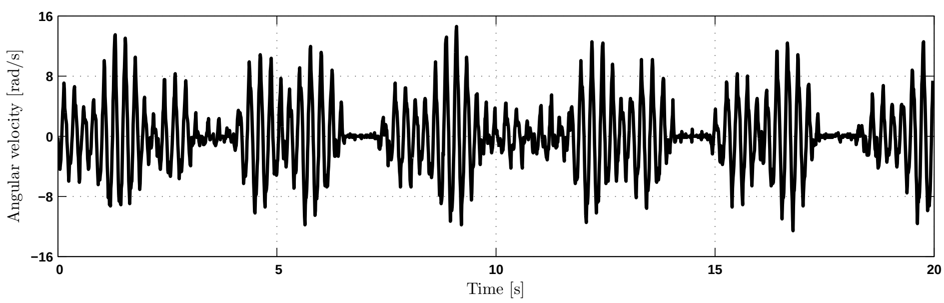


Figure C.9: *Angular acceleration estimation (joint angle second derivative)*

Because of the noise levels of the acceleration measurements they cannot be compared with the angular acceleration. The results will be useless with that much noise. Instead, the two accelerometer channels are compared relative to each other. They show a similar trend, which they must when the IMU is rotated. Due to the noise levels on the channels it is not able to compare the model approximation with the data to verify the similarities. A final conclusion is given next.

C.3 Conclusion

Two elements was tested in this journal: the IMU output distribution in steady-state and the verification of model approximation using IMU measurements. The distribution of the outputs was measured, and the variances were estimated. This can be used for Kalman filter tuning in chapter 7. A low signal-to-noise ratio was measured on the IMU channels, and the comparison between model approximations and the measurements was not possible. The approximations were thus not verified.

Appendix D

Journal 4: Controller performance

The main part of the control system is the sliding mode controller explained in chapter 9. An experiment must be designed to measure the performance of the controller, and the results must be analyzed. This will be explained in details in this journal. The controller is using strain measurements to determine the size of the applied control signal through the relation from (9.34) given as

$$u = \text{sat}\left[2\zeta\omega\dot{q} + \omega^2 q - K\dot{q} - U_0\,\text{sat}(s, \Delta_s), U_0\right] \tag{D.1}$$

with $s = \dot{q} + Kq$ describing the dynamics of the state trajectory in the phase plane. Performance is measured by the settling time of the strain state. Accuracy will also be considered. However, as described in remark 5 on page 112, the term $-K\dot{q}$ was not implemented in the controller diagram i LabVIEW, and the experimental data are therefore based on the control law

$$u = \text{sat}\left[2\zeta\omega\dot{q} + \omega^2 q - U_0\,\text{sat}(s, \Delta_s), U_0\right] \tag{D.2}$$

This is still a sliding mode controller, but the performance will be slightly reduced because the trajectory is not reaching the sliding mode as fast as it would have using (D.1). The effects will be evaluated in the acceptance test in chapter 10. The experimental procedure will be described next.

D.1 Experimental procedure

This experiment must be designed to show the performance of the sliding mode controller. The tool must be oscillating before activating the controller in order to see the damping effect. An excitation signal will first make the tool oscillate at the resonance frequency, and subsequently the controller is applied to damp the oscillation. The exact procedure is listed below

- Reset the system to initial configuration ($\theta_i = 0,\ i \in [1, 6]$)
- Measure the strain offset (different from $q = 0$) and reset q-state

- Introduce a sinusoidal signal $u(t) = 1{,}5\sin(26{,}39t)$ to make an oscillation
- Terminate the sinusoidal signal after 5 seconds and activate the controller
- Run the controller to 10 seconds while sampling the strain response

Notice how all 6 degrees of freedom are used for this experiment. This is a necessary procedure to ensure, that the third axis is moving within the (X_0, Z_0)-plane as described by theory in chapter 3. Five different configurations are tested with different sliding surfaces and limitation on the control signal.

- Case 1: Sliding surface $s = \dot{q} + 10q$ and control action saturation of ± 1 V
- Case 2: Sliding surface $s = \dot{q} + 10q$ and control action saturation of ± 2 V
- Case 3: Sliding surface $s = \dot{q} + 92{,}869q$ and control action saturation of ± 1 V
- Case 4: Sliding surface $s = \dot{q} + 92{,}869q$ and control action saturation of ± 2 V
- Case 5: Sliding surface $s = \dot{q} + 300q$ and control action saturation of ± 2 V
- Case 6: Control case with controller $u = 0$

The slope of $K = 92{,}869$ was determined to provide the fastest settling time in section 9.3. However, due to hardware limitations identified from practical experiments the control action is selected to a maximum of 2 V. This will not provide the same results as the simulations, but the test will show performance at different control signal saturations and K-values. The results from the experiment will be given in the sequel section, and the performance will be compared with the uncontrolled case. Beside the controller itself the sensor fusion technique is also verified with the measurement data. This consists of a linear Kalman filter, which will use a combination of measurements and a model to improve the quality of the model estimates used by the controller. Despite the fact that no Kalman filter was applied in parallel with the experiment, the filter is tested anyway to measure its performance to practical measurements instead of simulations.

D.2 Controller results

The experimental data from the 6 experiments is shown in figures D.7 through D.11 from page 177. L denotes the control signal limitation for each specific test case. For this experiment only $L = 1$ and $L = 2$ is applied. Subplot **a** depicts the strain measurements, and the excitation phase shows how the strain is reaching a steady oscillation before the controller is activated. Even though the timing requirements were given from the experimental procedure in section D.1 if was not possible to achieve an exact timing of the control system in practice. However, the manipulator was given time enough to both introduce an oscillation to the tool as well as running the controller for several seconds. Some of the experiments were terminated before 15 seconds had passed, because the joint was moving outside the secure range.

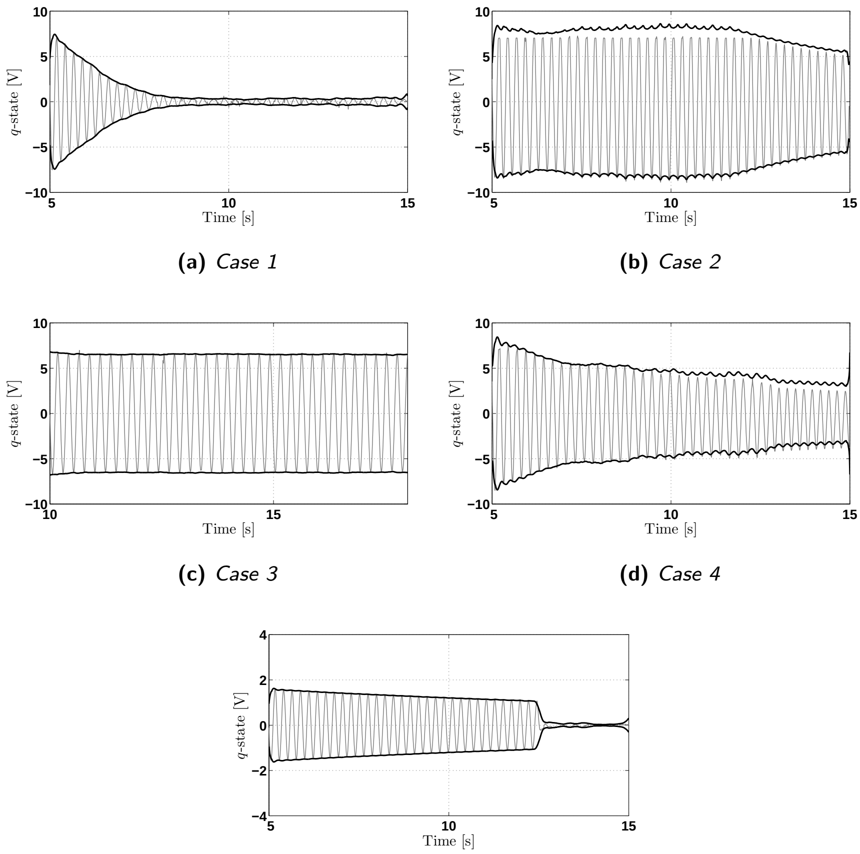
Subplot **b** shows the strain derivative, which is estimated by discrete differentiation. Combining the data from subplots **a** and **b** provides the s-function in subplot **c**. The shape resembles the one of $\dot{q}$ from subplot **b** because $|q| \ll |\dot{q}|$. Subplot **d** is a plot of the joint angle, which is drifting in all cases. The drifting is a two-part problem caused by gravity and offset error in the controller hardware. A gravitational pull on the tip mass is generated, when the tool is oscillating, which is generating a torque on the manipulator link. Furthermore, it has been noticed that without the tool mounted on the manipulator, the links can drift. The last four subplots **e** to **h** show different parts of the control signal. They are structure in the following way

$$\underbrace{u = \mathrm{sat}\left[\underbrace{\overbrace{\omega^2 q + 2\zeta\omega\dot{q}}^{\textsf{Subplot h}} - \overbrace{U_0\,\mathrm{sat}(s, \Delta_s)}^{\textsf{Subplot f}}}_{\textsf{Subplot g}},\ U_0\right]}_{\textsf{Subplot e}} \tag{D.3}$$

This approach has been used to show the consequences of using the saturation function in the control law. Disturbance rejecting properties are degraded, and the settling time is increased on this behalf. In order to analyze the performance of each controller, the envelope of the strain measurements must be estimated to find the settling time. The accuracy will be given by the amplitude of the oscillation of the last second of data. Figure D.1 shows the strain response with the controller activated and the envelope superimposed onto the data.

The envelope is constructed using the MATLAB command `hilbert.m` and an MA-filter of order 20 constructed using `smooth.m` (theory behind the functions will not be given). A maximum amplitude level is sampled for $q(t_1)$ and $q(t_2)$ with t_1 denoting the time of controller activated and t_2 the time when only 5% of the amplitude remains, respespectively. Consequently, only the case 1 response can be used to esitmate an actual settling time. The remaining responses does not reach 5 % of their initial conditions, and they can be excluded as admissible controller candidates. This is based purely on the given curcumstances, where the joint angle is drifting.

Based solely on the first case two points can be sampled: $q(t_1) = 7{,}4800$ V and $q(t_2) = 0{,}3706$ for $t_1 = 5,00$ s and $@t_2 = 8{,}85$ s, respectively. This gives a settling time $t_{\mathsf{set}} = 3{,}85$ s. A consequence of the added MA-filter is the a peak at the last sample. This does not resemble the actual behavior of the strain response, and the last 20 samples are therefore removed. Exactly 20 samples have been removed, because the last peak of the strain response is located 20 samples before the signal ends. This ensures, that the accuracy measure is not accidentally improved.



(a) *Case 1*

(b) *Case 2*

(c) *Case 3*

(d) *Case 4*

(e) *Case 5*

Figure D.1: *Strain measurements with envelope for controller performance analysis*

Figure D.2 shows the last 6,15 seconds after the 5 % level. The dashed lines indicate the mean of the envelope of $\hat{\mu}_q = 0{,}3376$ V. This corresponds to a deflection at the tip of 2,79 mm (using the relation measured in appendix F). A sample standard deviation of $\hat{\sigma} = 0{,}0598$ V corresponds too a deflection deviation of 0,49 mm. In order to compare the measured settling time of $t_{\mathsf{set}} = 3{,}85$ s with the uncontrolled case, the strain response from case 6 is shown in figure D.3.

This provides a settling time of $t_{\mathsf{set}} = 50{,}85$ s going from $q(t_1) = 8{,}413$ V to $q(t_2) = 0{,}4204$ in 5085 samples. An improvement of 92,43 % is thus obtained when applying the controller from case 1. Another way to investigate performance is to construct a phase portrait of the strain response. Figure D.12 on page 172 shows a phase plot for all 6 cases from the first 5 seconds after the controller has been activated. The sliding surface is shown as a line in the plots. Only the portrait in figure D.12a converges to zero within the 5 second window of the plots. The remaining either stays in an approximative limit cycle (subplot **b**, **c** and **e**) while others are converging very

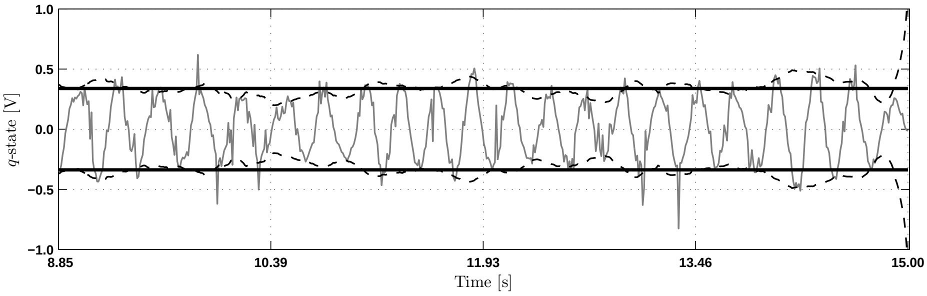


Figure D.2: *Accuracy measurement of case 1 controller*

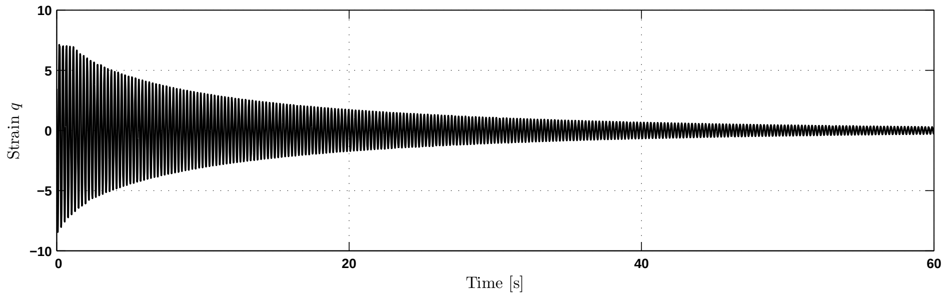
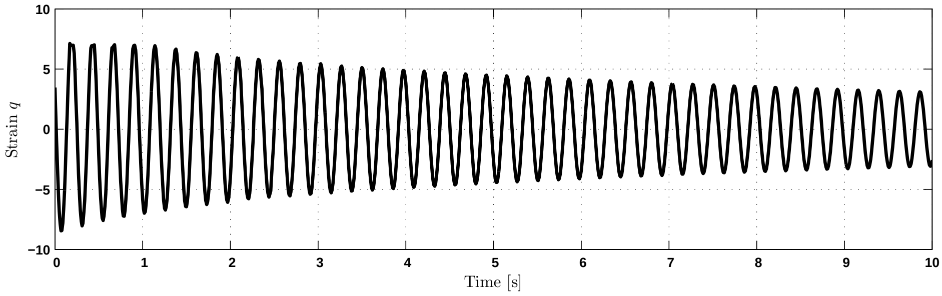


Figure D.3: *Strain measurements from case 6 (control experiment)*

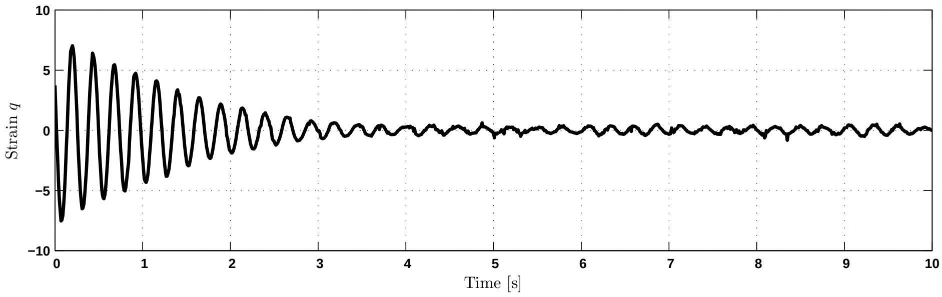
slowly (subplot **d**). The uncontrolled case is shown in figure D.12f for comparison. All trajectories have been filtered using a 10th order MA-filter. Otherwise, the noise levels makes it difficult to interpret the portraits.

Another piece of information visible from the phase plots is the lack of sliding mode. It is expected from theory, that the trajectory must remain on the sliding surface after it first reaches it. Even though the controller seems to have an effect (especially case 1) the behavior is not sliding mode behavior. In order to compare the results between case 1 and case 6 figure D.4 shows the first 10 seconds of strain measurements from both cases after the controller has been activated (not for the uncontrolled case 6). A frequency response estimation can be estimated from the measurements as well as from the control signal, see figure D.5. Some samples in the sampled control signal are ∞ and will be set to zero for the FFT algorithm to function properly. The strain responses in figure D.5b show a slight difference in resonance frequency. From case 1 (controlled case) the peak is located at ω = 4,10 Hz and at ω = 4,25 Hz for the uncontrolled case (close to the 4,20 Hz measurement from appendix A).

Even though there is only a slight difference between the frequencies, the controlled case have several other frequency components that influences the oscillation. The control signal in figure D.5 shows frequency components around the tool, which must be avoided. However, it may be advantageous if the signals are π rad out of phase with the tool oscillation. This was not considered to be an option in chapter 9.



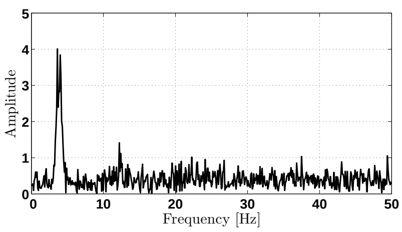
(a) *Controlled experiment, case 1*



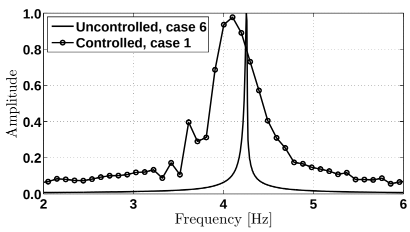
(b) *Unontrolled experiment, case 6*

Figure D.4: *Strain response comparison between case 1 (controlled) and case 6 (uncontrolled)*

Figure D.6 shows how the oscillation is around $\pi/2$ rad out of phase with the control signal in the first periods. After some time it is difficult to use the phase terminology, because the control signal is not a harmonic signal like the strain. A conclusion of the measurements and the results is provided in section D.4.



(a) *Saturated control signal (only case 1)*



(b) *Strain responses (both cases)*

Figure D.5: *Frequency response of control signal and strain responses from cases 1 and 6*

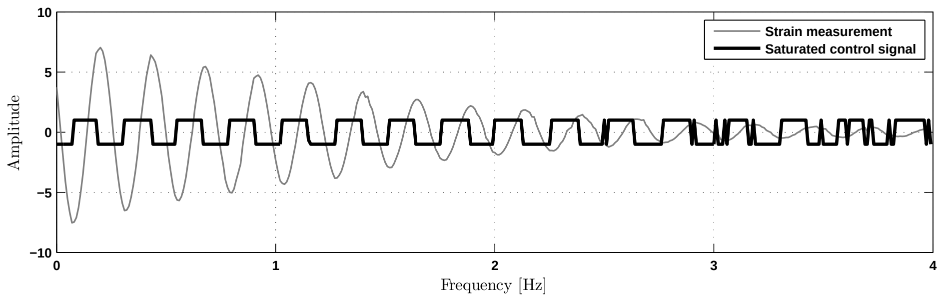


Figure D.6: *Measurements showing that the strain is not in phase with the control signal*

D.3 Sensor fusion results

In order to test the linear Kalman filter from chapter 7, the strain and joint angle measurements from above are filtered by the Kalman filter. A manual tuning of the matrices Q and R has been made, and they are given as

$$Q = \operatorname{diag}\begin{bmatrix}1 & 10 & 10 & 100\end{bmatrix}, \qquad R = \operatorname{diag}\begin{bmatrix}1 & 100\end{bmatrix} \tag{D.4}$$

and initial values given from the measurement directly. Figure D.13 on page 183 shows the measurements, estimates from the Kalman filter and the corresponding error between them both. It is clear from figure D.13e that the strain estimate takes some time to line up with the measurements, whereas the joint angle is off by no more than 0,006 rad. The Kalman filter therefore seems to perform as it should, and will be able to filter out erroneous sensor measurements. Both the controller performance and the Kalman filter is evaluated in the sequel section.

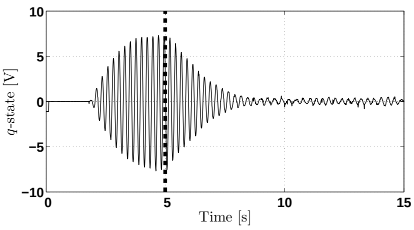
D.4 Conclusion

A total of 6 cases were tested in this journal. They included different sliding surfaces and controller limits. A new hardware limitation was introduced because the original control signal limit of $U_0 = 10$ V showed to unmanageable in case of error. Through simulations in chapter 9 the sliding surface $s = \dot{q} + 92,869q$ was determined as the best possible. This was derived on the basis of the original hardware limitations. The surface was tested anyway, but the low saturation limits resulted in unmeasurable performance. Another source of error is the fact that the simulation was based on initial conditions different from the practical ones.

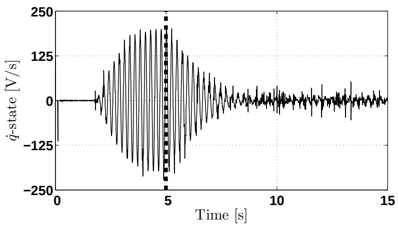
From the phase plots on figure D.12 it was shown, that the strain trajectory did not behave as expected. No sliding mode was shown from the plots, since the trajectory does not change direction when reaching the switching surface. The first cause can be the limited control action. If the control signal is not high enough, the vector field gradient are not pointing sharply towards the switching surface. Analog to the theory of a second order system with friction, the control field will increase the damping ratio. Also, the missing $-K\dot{q}$ term decreases the performance, and may cause the trajectory to circle around the equilibrium point instead of reaching sliding mode. This will in turn make the trajectory converge faster towards the point of equilibrium.

The data are not supported statistically, since only one test could be used, but due to hardware difficulties it was not possible to gain any more data. No definite conclusion can therefore be given on the controller performance rather than from the single useful experiment. Based solely on the case 1 experiment, even though sliding mode was not reached, the controller showed a damping effect on the oscillation and decreased the settling time by 92,43 %. Consequencly, the continuously applied control action introcudes a lower limit of the strain response corresponding to a tip deflection of 2,79 mm with a standard deviation of 0,49 mm.

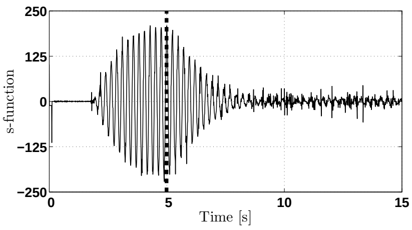
The linear Kalman filter for sensor information fusion was also tested on the measurement data from the first experiments. It was able to converge to the strain measurements within around 3 seconds and almost immediately for the joint angle measurements. In order to work in parallel with the controller it will have to converge faster in order to improve the quality of the model estimates.



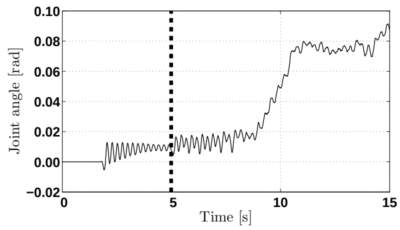
(a) *q-state*



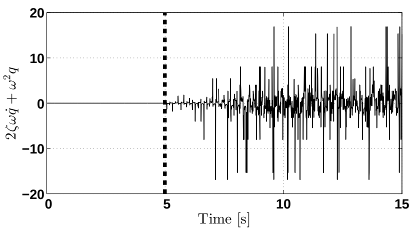
(b) *$\dot{q}$-state*



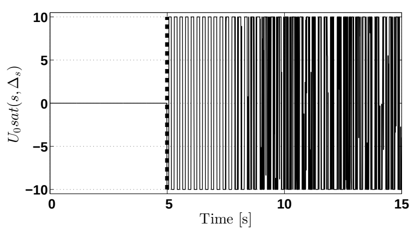
(c) *s-function*



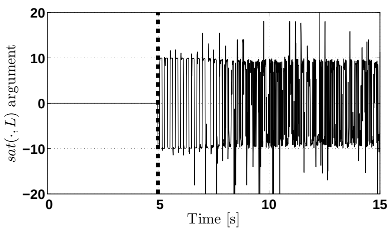
(d) *Joint angle measurements*



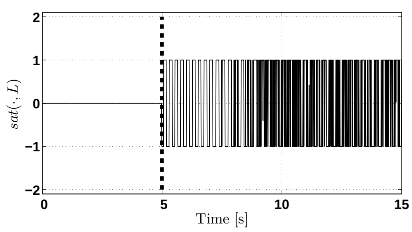
(e) $2\zeta\omega\dot{q} + \omega^2 q$



(f) $U_0\,\mathrm{sat}(s, \Delta_s)$

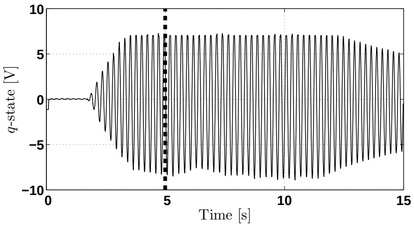


(g) *Unsaturated control action*

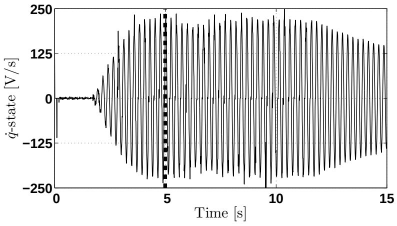


(h) *Control action*

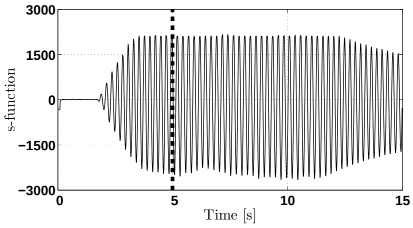
Figure D.7: *Case 1 measurements and controller functions (gain = 1 and $K = 10$)*



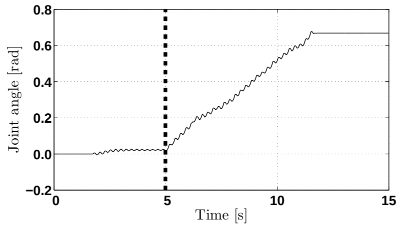
(a) *q-state*



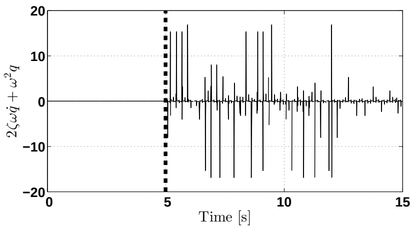
(b) *$\dot{q}$-state*



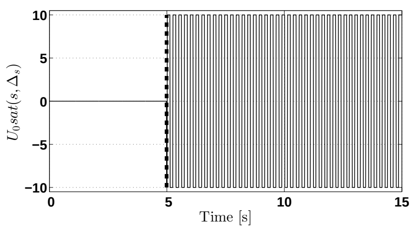
(c) *s-function*



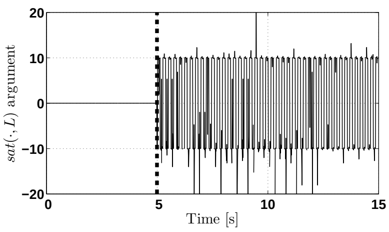
(d) *Joint angle measurements*



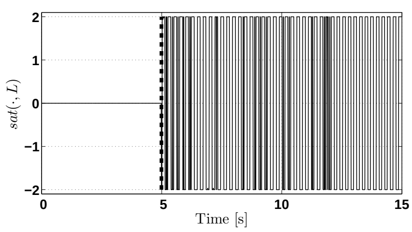
(e) $2\zeta\omega\dot{q} + \omega^2 q$



(f) $U_0\,\mathrm{sat}(s, \Delta_s)$

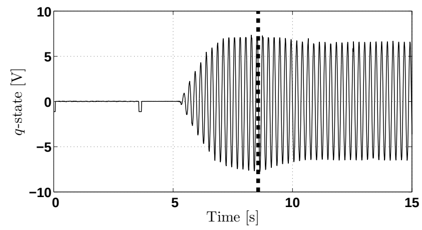


(g) *Unsaturated control action*

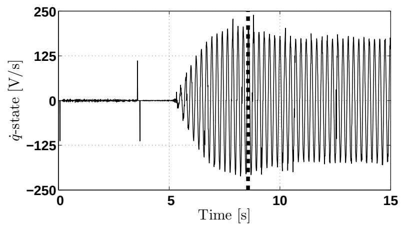


(h) *Control action*

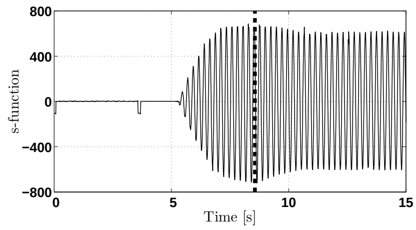
Figure D.8: *Case 2 measurements and controller functions (gain = 2 and $K = 10$)*



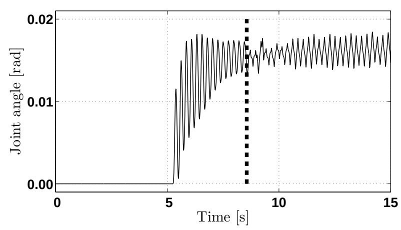
(a) *q-state*



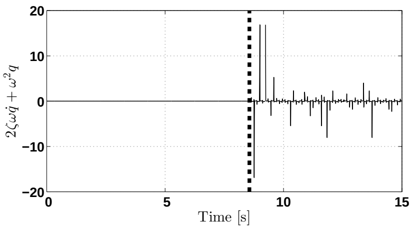
(b) *$\dot{q}$-state*



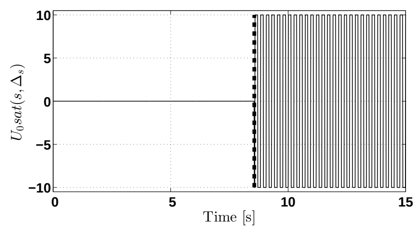
(c) *s-function*



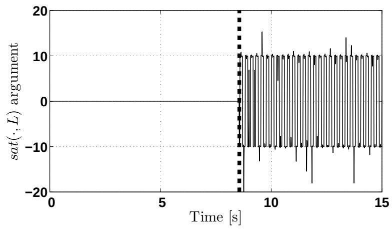
(d) *Joint angle measurements*



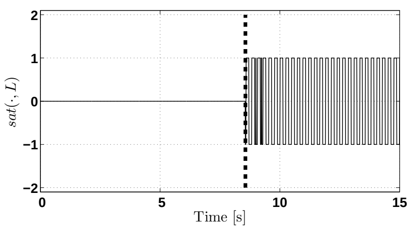
(e) $2\zeta\omega\dot{q} + \omega^2 q$



(f) $U_0\,\mathrm{sat}(s, \Delta_s)$

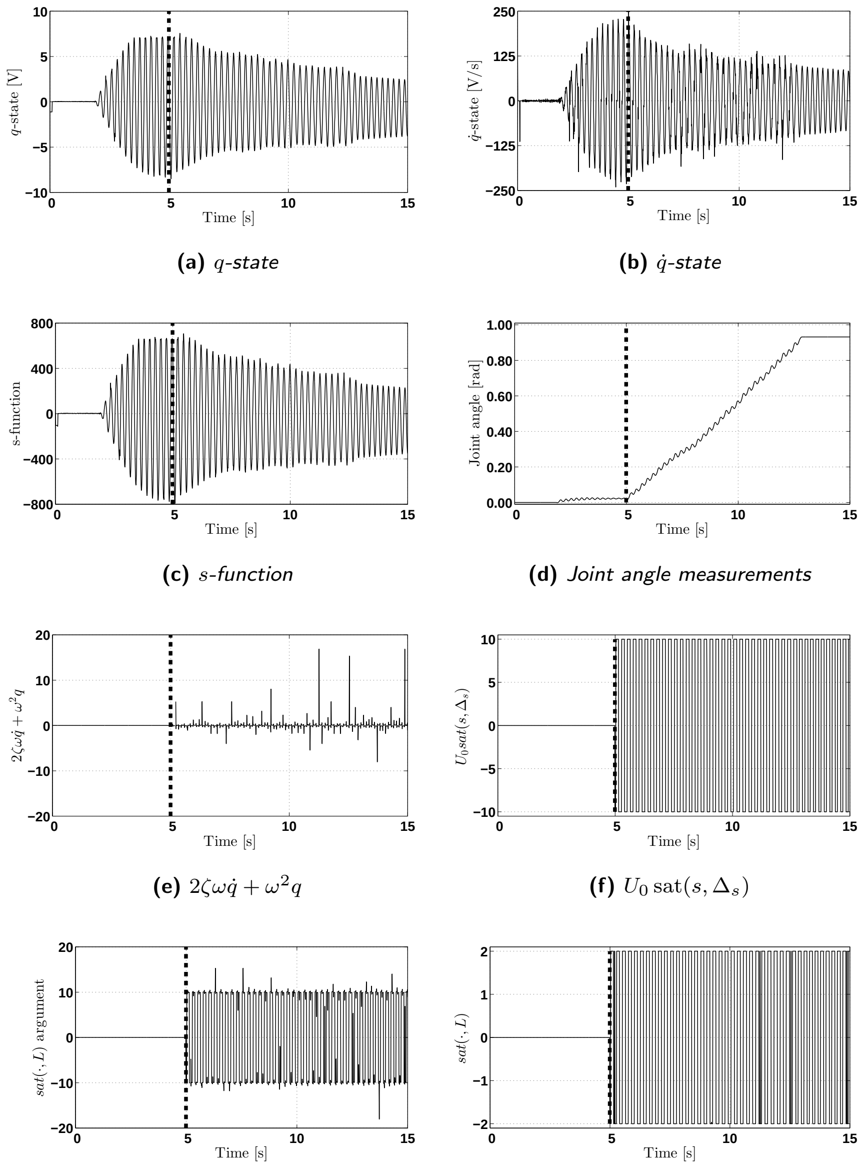


(g) *Unsaturated control action*



(h) *Control action*

Figure D.9: *Case 3 measurements and controller functions (gain = 1 and $K = 92{,}869$)*



(a) *q-state*

(b) *$\dot{q}$-state*

(c) *s-function*

(d) *Joint angle measurements*

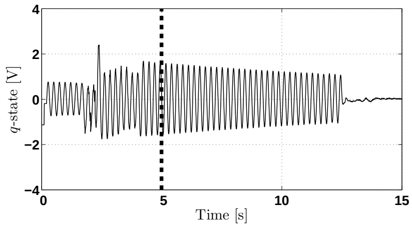
(e) $2\zeta\omega\dot{q} + \omega^2 q$

(f) $U_0\,\mathrm{sat}(s, \Delta_s)$

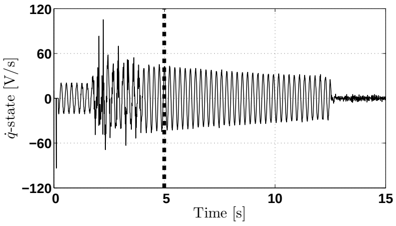
(g) *Unsaturated control action*

(h) *Control action*

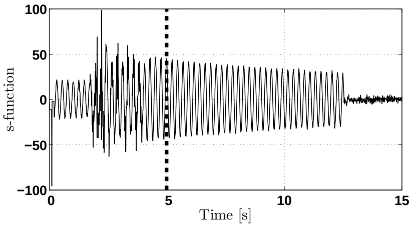
Figure D.10: *Case 4 measurements and controller functions (gain = 2 and $K = 92{,}869$)*



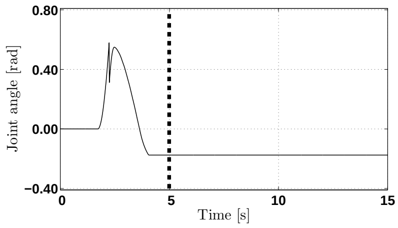
(a) *q-state*



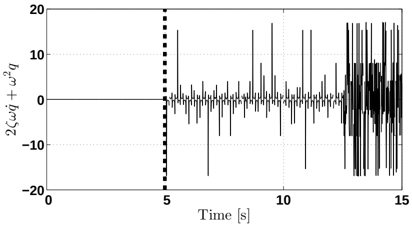
(b) *$\dot{q}$-state*



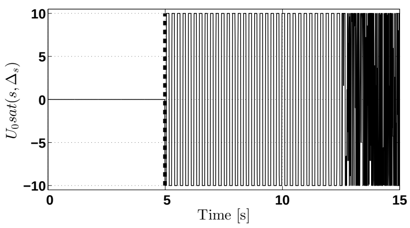
(c) *s-function*



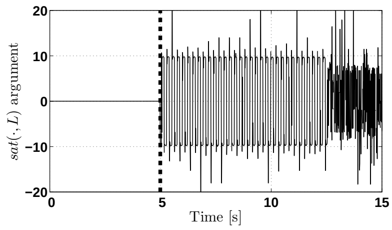
(d) *Joint angle measurements*



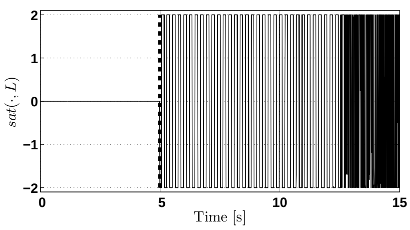
(e) $2\zeta\omega\dot{q}+\omega^2 q$



(f) $U_0\,\mathrm{sat}(s,\Delta_s)$

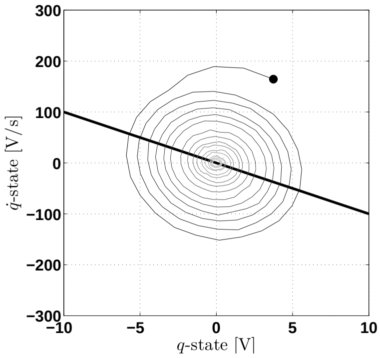


(g) *Unsaturated control action*

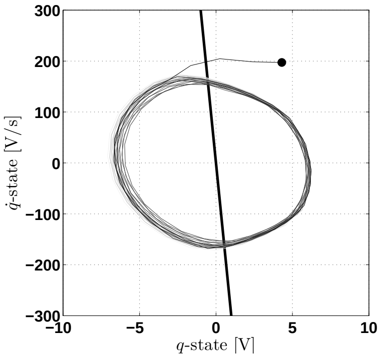


(h) *Control action*

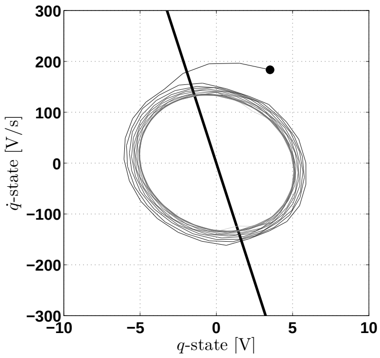
Figure D.11: *Case 5 measurements and controller functions (gain = 2 and $K = 300$)*



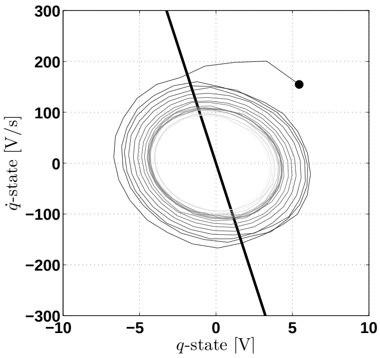
(a) *Case 1*



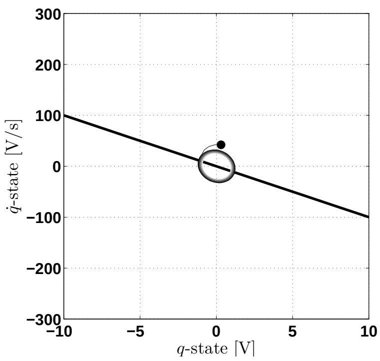
(b) *Case 2*



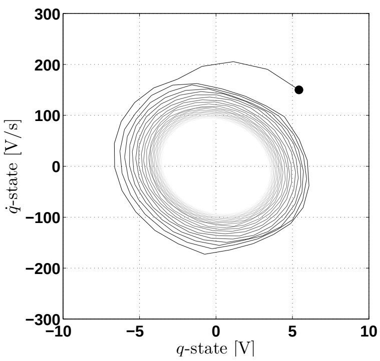
(c) *Case 3*



(d) *Case 4*

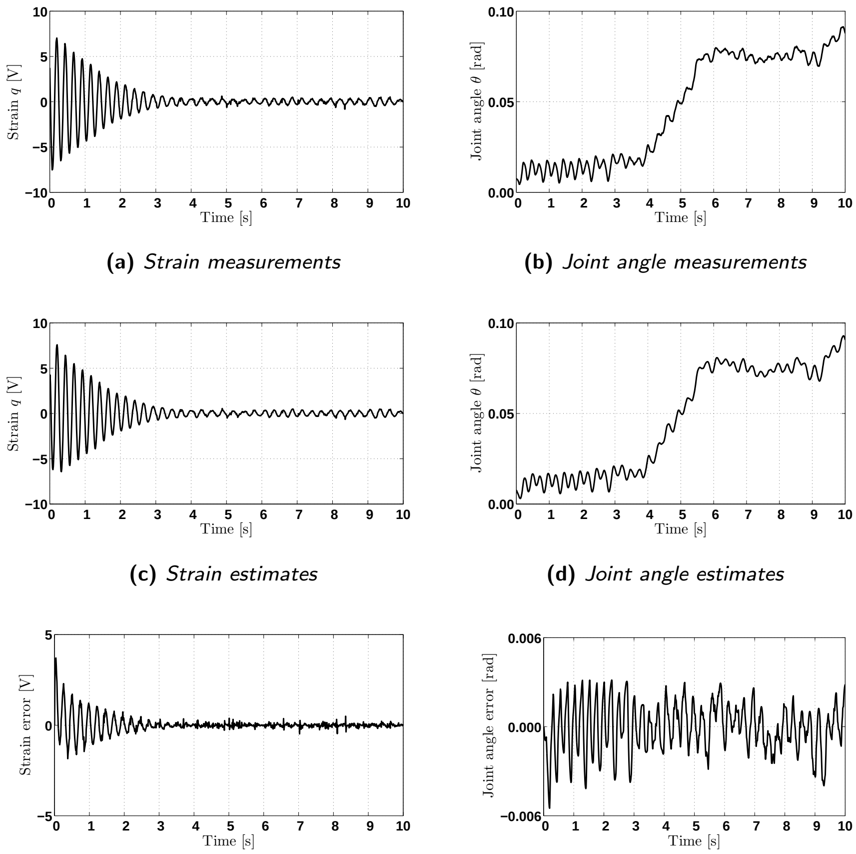


(e) *Case 5*



(f) *Case 6*

Figure D.12: *Phase portraits using 5 seconds of strain response*



(a) *Strain measurements* **(b)** *Joint angle measurements*

(c) *Strain estimates* **(d)** *Joint angle estimates*

(e) *Strain difference* **(f)** *Joint angle difference*

Figure D.13: *Measurements vs Kalman estimates*

Appendix E

Journal 5: Manipulator bandwidth

The dynamics of the manipulator must be investigated before applying control. This will show if the manipulator is capable of excite the tool with the necessary frequencies. An experiment must be designed to measure the magnitude response of the third axis of the REIS RV15 manipulator for a certain range of frequencies. The experimental procedure will be described in the sequel, and the results will follow immediately after.

E.1 Experimental procedure

When performing a frequency analysis of a system a sinusoidal input signal must be supplied. The output is then sampled, and the amplitude is measured. A magnitude plot can then be constructed as the ratio between input amplitude and output amplitude for a certain range of frequencies. For this particular case a frequency range of 0,2 Hz to 10,0 Hz has been selected with an increment of 0,2 Hz. This covers also the 4,20 Hz resonance of the flexible tool. Three different signals will be applied.

- Sinusoidal torque control signal of 1 V amplitude
- Sinusoidal velocity control signal of 1 V amplitude
- Sinusoidal velocity control signal of 2 V amplitude

The torque control applies a torque corresponding to the amplitude of the input signal amplitude, whereas the velocity control results in a joint velocity corresponding to the input signal amplitude. The results are shown in the sequel.

E.2 Results

Three different joint angle responses have been sampled during the experiments and the signals are depicted in figure E.1. Due to drifting of the joint (explained in details in appendix 10) the torque control experiment begins at a frequency of 2,6 Hz. For the velocity control experiments the drifting is almost linear, and has been removed using a moving-average filter of order 2000. The data can then be used by an FFT algorithm. However, the manipulator was not able to accept frequencies above 6,0 Hz (gain = 1) and 4,2 Hz (gain = 2), and the servo drivers automatically turned off.

The magnitude response estimates are shown in figure E.2. Each of the trends have approximately a slope of -20 dB/decade. Because the frequency is very low at 0,2 Hz the system is likely to behave as an integrator in stead of a low-pass filter. This can be caused by the high gear ratio of 1:100. The servo drivers may also provide dynamics.

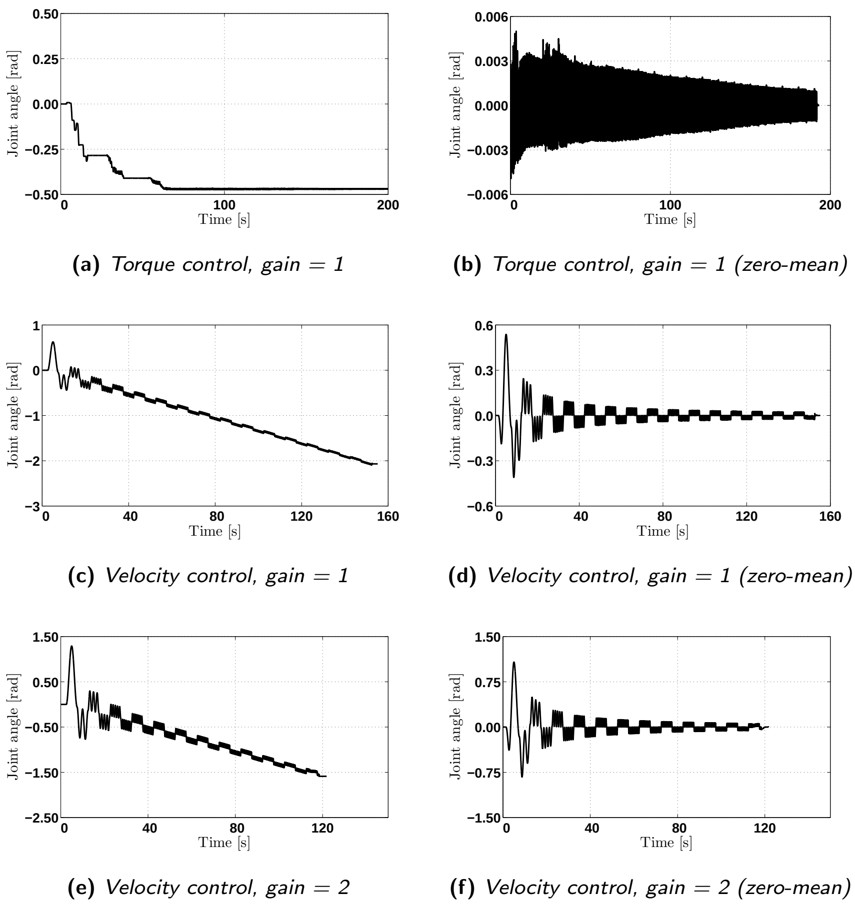


Figure E.1: *Time responses for magnitude response estimations*

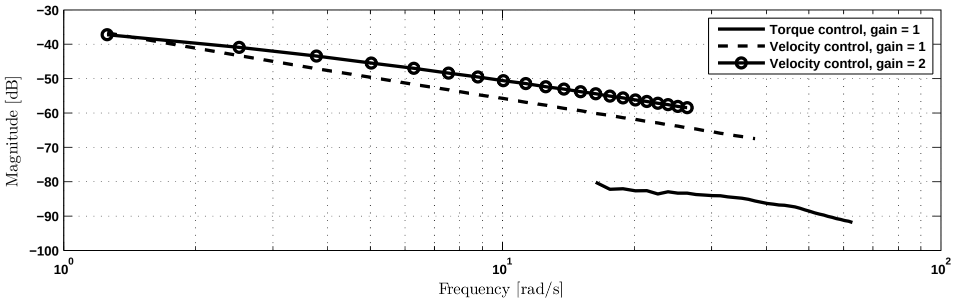


Figure E.2: *Magnitude response estimates from measurements in figure E.1*

E.3 Conclusion

The results are not uniquely stating that the system is behaving as an integrator system. Due to the drifting of the manipulator joint, the mean has been be subtracted in order to apply FFT, which may be a source of error. The entire frequency range was neither analyzed because either the servo drivers of the manipulator dynamics did not accept the control signals. However, the experiments showed, that it is possible to make excitations up to 10 Hz using torque control (which will be applied for control) even at unity gain. Amplification of the control signal may be needed for control purposes.

Appendix F

Journal 6: Strain/deflection relationship

A relation between beam deflection and measured strain [V] has been derived in subsection G.2.4. For a given time instance the output from the strain gauge amplifier v_{o2}^{str} is a linear function depending on an unknown scale factor. This will have to be estimated in order to relate strain measurements to actual strain/deflection. The linearity of the relation must also be verified. An experiment has been designed, which is described in the sequel.

F.1 Experimental procedure

In order to determine the ratio between measured strain and beam deflection, the beam is affected by a force at the tip. For each mm of deflection the strain is logged. This procedure is performed in both directions within the range ±20 mm. The range has been practically determined as the maximum allowable strain. To avoid any saturation of the measurements, a resistor $R_g = 200\,\Omega$ has been selected for the gauge amplifier circuit. This corresponds to a gain of ~ 248. The experimental setup is shown in figure F.1.

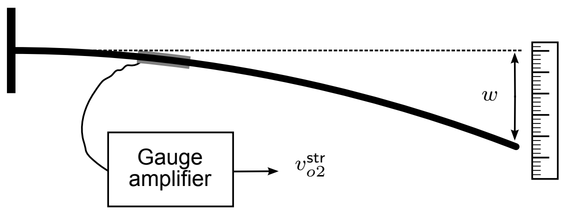


Figure F.1: *Experimental setup in journal 6*

F.2 Results

A total of 41 samples have been stored for the experiment, and table F.1 lists them all. Figure F.2 plots the relation between deflection [mm] and strain measurement

[V]. A linear regression has been used to establish a relation between the variables, and the estimate is given as

$$v_{\mathrm{o2}}^{\mathsf{str}}(t) = 0{,}1211 w(\ell, t) - 2{,}403 \quad [\mathrm{V}] \tag{F.1}$$

with the offset being a consequence of the initial condition of the beam and the initial gauge configuration. This relation will change if the beam is being deformed.

w [mm]	v_{o2}^{str} [V]	w [mm]	v_{o2}^{str} [V]	w [mm]	v_{o2}^{str} [V]	w [mm]	v_{o2}^{str} [V]
-20	-4,844	-10	-3,605	0	-2,441	10	-1,193
-19	-4,681	-9	-3,494	1	-2,298	11	-1,075
-18	-4,495	-8	-3,365	2	-2,186	12	-0,945
-17	-4,472	-7	-3,237	3	-2,061	13	-0,828
-16	-4,347	-6	-3,130	4	-1,918	14	-0,696
-15	-4,213	-5	-3,019	5	-1,807	15	-0,596
-14	-4,081	-4	-2,925	6	-1,668	16	-0,458
-13	-3,957	-3	-2,805	7	-1,561	17	-0,328
-12	-3,858	-2	-2,665	8	-1,429	18	-0,203
-11	-3,744	-1	-2,547	9	-1,293	19	-0,063
						20	0,019

Table F.1: *Strain samples for various deflections*

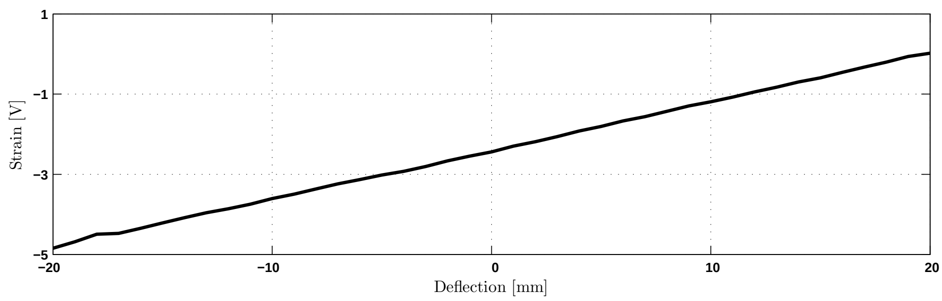


Figure F.2: *Strain as a function of tool tip deflection*

F.3 Conclusion

The relation between measured strain and beam deflection at the tip has been determined. A linear regression yields $R^2 = 0{,}9998$, which is enough to verify linearity within the tested range of deflections.

Appendix G

Hardware

Both the mechanical and the electrical systems will be described within this appendix in order to provide an overview of the technical equipment used for testing in this project. The mechanical system involves a 6-DOF industrial manipulator REIS RV15 [80] with revolute joints. It features a hydraulic damper on the first link, which is able to counteract the force of a fully extended manipulator in the horizontal plane. This eases the joint actuator by supporting some of the torque necessary to hold the manipulator in this position. In that way the link can be kept in an extended position for a longer period of time without unnecessarily stressing the actuator. All joint actuators are electric permanent magnet direct current motors, which are all driven by servo amplifiers [49]. The amplifiers are able to output a given torque proportional to a voltage input signal. Figure G.1 below shows the hardware accessible for this project, which will be described in details in the following.

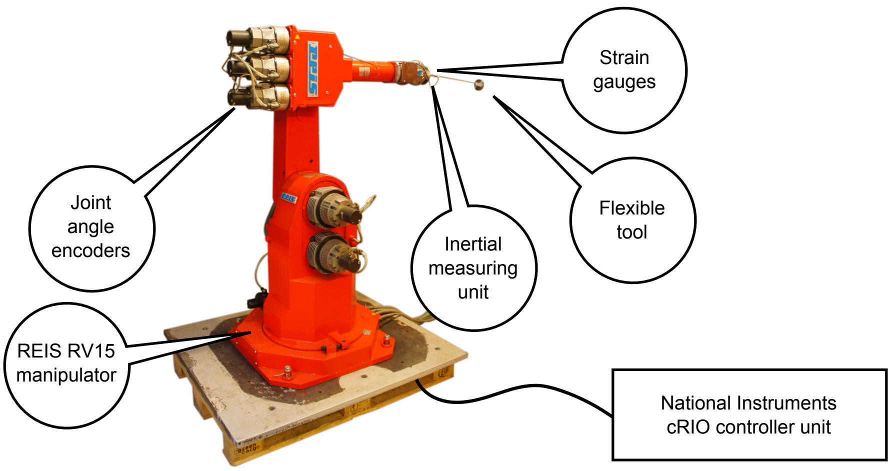


Figure G.1: *Overall hardware configuration*

The manipulator is controlled by a National cRIO-9074 controller system [68], which enables the current drivers to be controlled through a computer with LabVIEW as well as reading outputs from sensors mounted on the robot. Three NI 9411 digital-to-analog data acquisition cards [70] sample the voltage signals from the rotary encoders. Two NI 9263 analog output cards [67] provide signals to the current drivers by a given voltage signal. An NI 9201 data acquisition card [69] is used to sample signals from external sensors, which will be mounted on the manipulator for additional measurements. No separate section will be given for the National Instrument components, since the datasheets provide the information necessary to use them. All external sensors will be described later in this appendix. The mechanical and electronic system structure is shown in figure G.2.

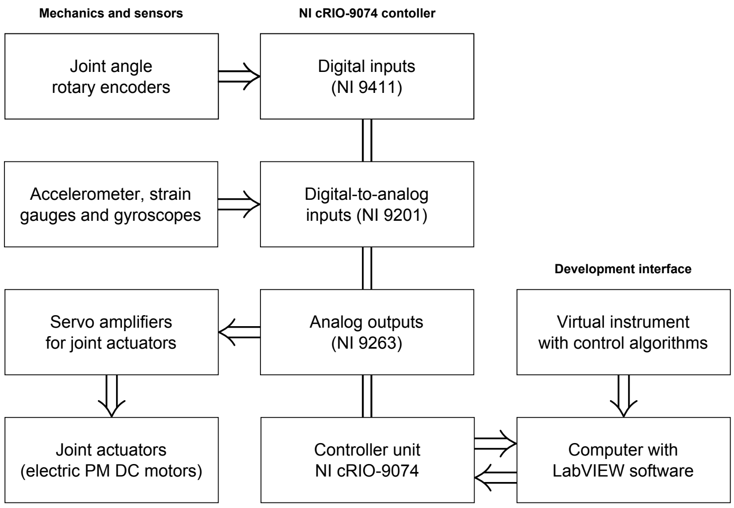


Figure G.2: *Block overview of hardware configuration*

From the REIS RV15 technical manuals [79, 80] a number of parameters are given on both geometry and dynamic limitations. The dynamic parameters are though not considered reliable, since the robot is not new (exposed to wear and smaller damages throughout the years). As a result, the parameters will only be used as initial conditions within the parameter estimation algorithms and otherwise used as guidelines for the actual values.

This appendix will first describe the manipulator hardware in details including the static and dynamic limitations including length of links, location of center of gravity and maximum allowable velocity of each joint actuator. All the sensors on the system will afterwards be described, including the original sensors (joint encoders) as well as the externally mounted sensors (accelerometers, gyroscopes and strain gauges). An equation is derived for each sensor, expressing the output as a stochastic process when including sensor noise. The distributions of the outputs will be used later when designing a Kalman filter to improve the dynamic model estimates using measurements.

The appendix is completed with a description of the flexible tool analog, which will be used to emulate a complex manipulator tool with flexible behavior. Before constructing a model of the flexible behavior, it is important to know relevant specifications of the tool including material, flexibility and dimensions.

G.1 Industrial robotic manipulator

The manipulator used for testing is a serial robot of the type REIS RV15 with 6 interconnected links (6 degrees of freedom) each with a joint being actuated by an electric motor. A general description of this type of robot is a *6DOF-manipulator*. This type of manipulator is capable of positioning objects in Euclidean space with an arbitrary orientation (within the limitations) and is therefore also capable of counteracting oscillations of a tool in all directions. The manipulator structure itself is assumed rigid, which is supported by the large ratio between self-weight of 310 kg and maximum payload of 15 kg [80]. This is often an indication that the manipulator is constructed in such a way, that it can be assumed rigid. Figure G.3 shows the manipulator.

Figure G.3: *REIS RV15 industrial manipulator*

Table G.1 lists the important data of the manipulator, which will be used when deriving a model and designing a controller.

	Axis 1	**Axis 2**	**Axis 3**	**Axis 4**	**Axis 5**	**Axis 6**
$\theta_{i,\max}$	330 °	150 °	270 °	360 °	240 °	360 °
$\dot{\theta}_{i,\max}$	180 °/s	180 °/s	180 °/s	215 °/s	215 °/s	215 °/s
$\theta_{i,0}$	-180 °	-90 °	90 °	-90 °	0 °	0 °
m_i	-	30 kg	35,5 kg	3 kg	3 kg	3 kg
$\theta_{i,\mathsf{SWmax},+}$	$+147$ °	-16 °	$+136$ °	$+356$ °	$+114$ °	$+356$ °
$\theta_{i,\mathsf{SWmax},\div}$	-165 °	-137 °	-181 °	-358 °	-115 °	-358 °
ℓ_i	-	600 mm	684 mm	-	100* mm	-
ℓ_i^{cog}	-	343 mm	37 mm	-	50* mm	-

Table G.1: *REIS RV15 data [79, 80] (* measured)*

The different variables are $\theta_{i,\max}$ (swivel range), $\dot{\theta}_{i,\max}$ (maximum joint velocities), $\theta_{i,0}$ (zero reference), m_i (link mass), $\theta_{i,\mathsf{SWmax},+}$ (software swivel upper limit), $\theta_{i,\mathsf{SWmax},\div}$ (software swivel lower limit), ℓ (length of link) and ℓ^{cog} (length from joint to center of gravity). The software limits were implemented by the original controller for the manipulator [79] and will be used as boundaries on the control signals to prevent hardware damage.

Two types of motors are driving the manipulator links with accessible data given in table G.2. No other data is available for the actuators, which will have to be estimated using system identification techniques. This issue will be treated in chapter 8.

	Axis 1-3	**Axis 4-6**
Actuator type	Mavilor MO800	Mavilor MO200
Velocity	3000 rpm	3000 rpm
Nominal current	9,2 A	6,5 A

Table G.2: *REIS RV15 actuator data [79]*

When controlling the TCP path it is important to know the admissible set of joint angles in joint space. Certain joint vectors (joint angles listed in θ vector) are not possible to reach, because the limits of the joint angles are depending on the current configuration of the manipulator. The *configuration space* is defined as the set of all admissible combinations of joint space entries [24]. In other words, it is the power set of the joint space, which includes all manipulator postures measured at the end-effector. The *obstacle space* is on the other hand a set of manipulator configurations that are to be avoided. This space can be static or dynamics depending on the surrounding environment.

By combining both spaces, the *workspace* (or *free space*) remains in which the manipulator is free to operate [24, 86]. Figure G.4 shows the REIS RV15 workspace, which has been generated on the basis of the software boundary limits from the *zero reference* joint angles. The zero reference is also referred to as the initial manipulator configuration, which is also shown in figure G.4. No tool is shown in the figure.

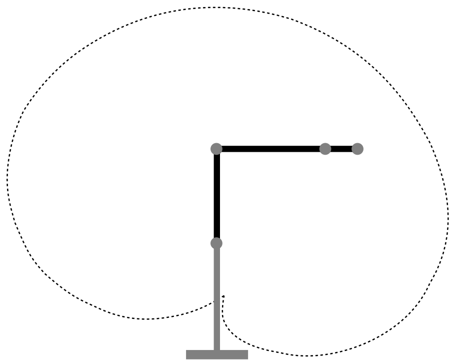


Figure G.4: *REIS RV15 simulated workspace*

The workspace can be converted into TCP-workspace in order to ensure that the scheduled TCP trajectory is within the admissible set of manipulator configurations. Because the scope of this project is concerned about damping oscillations in a local neighborhood around the destination of the tool, the workspace is not needed directly. No cartesian coordinates are given as references to reach with the TCP and all configurations are therefore reachable.

The REIS RV15 manipulator is included in the class of *holonomic* systems [26]. Specific for holonomic systems is the model constraints depending on the generalized coordinates alone. The number of actuators is also equal to the number of joint angles because the joints are free to operate independently. Given the joint angles of the manipulator, the position and orientation of the last link in the chain can be determined. *Nonholonomic* systems are on the other hand modeled using constraints, which are depending not only on the generalized coordinate vector, but also on its derivatives. This type of system will not be considered in this report.

Two related terms are further used to characterize the different classes of mechanical system constraints. Since the REIS RV15 manipulator itself is a constant rigid configuration, meaning that it does not change shape in time, the constraints are said to be *scleronomic* [15]. When considering the production cell as a system, the manipulator itself is still constrained by scleronomic constraints, but the manipulator base may be moving inside the cell, and therefore time dependent. These constraints are denoted as *rheonomic* constraints. The different system type notations will not be used explicitly throughout the report, but is only mentioned here to inform about the difference.

This ends the section about the manipulator hardware, and the data will be used in the sequel part. The following section describes the different sensor types mounted on the manipulator.

G.2 Sensor types

A number of sensors must be mounted on the manipulator to provide data for parameter estimation. Rotary encoders provide a pulse for every step of angular displacement of the axis through it. By counting the number of pulses, the absolute joint angles can be measured. Accelerometers and gyroscopes gathers data from specific locations on the manipulator, which must fit the dynamics model. Gyroscopes measure the change in angular orientation whereas accelerometers measure the changes in linear velocity. Both gyroscopes and accelerometers are collected on a single IMU board (inertial measuring unit) from SparkFun Electronics [88]. The IMU board is capable of measuring 6 different parameters at once by the 3-axis accelerometer and the two gyroscopes. Strain gauges will provide measurements of the tool strain caused by motion of the load and the end-effector. All sensors will be thoroughly described in the sequel subsections categorized according to the variable being measured. A picture of each individual sensor type, besides the integrated rotary encoders, is given at the end of the section in figure G.9. Current measurements are available from the servo drivers, but are not coupled to the controller, which is why they are not used.

G.2.1 Joint angles (rotary encoder)

Each joint on the manipulator is fitted with an encoder, which provides a read-out of the current joint angle. Due to the gearing between the encoder and the joint actuator the resolution of the measurement resolution is increased by a factor of 100 [25, 79]. The encoders are of the type LTN G70 from LTN Servotechnik GmbH [62, 46] with a resolution of 4000, but due to the manipulator gearing the effective count is 400000. By counting the number of pulses from the sensor and wrapping around for every 400000 counts, the angle can be determined as

$$\breve{\theta}_i = \frac{2\pi c_i}{4\text{E}5} + \theta_{i,\mathsf{off}}\ [\text{rad}], \quad c_i \pmod{4\text{E}5} \tag{G.1}$$

with c_i being the count from the i-th encoder and $\theta_{i,\mathsf{off}}$ an offset if the encoder is not reset at startup. According to the encoder datasheet[1] [62], the accuracy can be determined using a certain formula. Unfortunately, this formula is only applicable up and till 2500 counts per revolution. However, since there is no other data on the component, the formula will be used anyway, which is then given as

$$\varepsilon^{\mathsf{ja}} = \pm\frac{2\pi}{20 \cdot 4000} \approx \pm 7{,}85\text{E}{-5}, [\text{rad}] \tag{G.2}$$

and when assuming that $\varepsilon^{\mathsf{ja}} \sim \mathcal{N}(0, (7{,}85\text{E}{-5})^2)$, the joint angle is expressed as

$$\boxed{\breve{\theta}_i = \frac{2\pi c_i}{4\text{E}5} + \theta_{i,\mathsf{off}} + \varepsilon^{\mathsf{ja}} \quad [\text{rad}] \quad c_i \pmod{4\text{E}5}} \tag{G.3}$$

[1] The present datasheet is for the encoder of type G71, but according to the manufacturer, the output of the G70 is identical to the output of the G71 model

This sensor is the only one of the sensors used in this project that is not modeled as a voltage. A digital input card is used to count the number of pulses, which is then converted into an angle. All other sensors have analog outputs, and the voltage must be related to the sensor variable using a conversion equation. The next sensor is the gyroscope.

G.2.2 Change in orientation (gyroscope)

A 3-axis type gyroscope will be mounted close to the end-effector. This will provide the best chance of picking up the most information as possible (dynamics of links below will be included). The gyroscope is part of the IMU board [88] and consists of a 2-axis LPR530AL (pitch-roll) [90] and a 1-axis LY530ALH gyroscope (yaw) [89] both from STMicroelectronics. The gyroscopes LY530ALH and LPR530AL have the specifications given in table G.3 (the data are similar for both gyroscope types).

Supply voltage	2,7-3,6 V
Range (nominal)	±300 deg/s (1x) or ±1200 deg/s (4x)
Output sensitivity (nominal)	830 μVs/deg (1x) or 3300 μVs/deg (4x)
Zero-rate bias (nominal)	1230 mV
3dB bandwidth	140 Hz

Table G.3: *LY530ALH (1-axis) and LPR530AL (2-axis) gyroscope specifications [89, 90]*

The gyroscopes have two amplification setting, 1x and 4x, making it possible to measure more precisely but in 1/4 of the nominal range. No information about the distribution of the measurements is explicitly given from the datasheet, and a variable standard deviation σ^{gyro} [V] must be used for now in form of a random variable $\varepsilon^{\mathsf{gyro}} \sim \mathcal{N}(0, [\sigma^{\mathsf{gyro}}]^2)$. The actual distribution will be estimated from experiments (see appendix C) to be $2,7-3,1$ mV according to the axis of motion and the factor of amplification. A relation between angular rate and output voltage can be expressed with r_{gyro} [deg/s] describing the measured rate of change

$$\boxed{v_{\mathrm{o}}^{\mathsf{gyro}} = 830\text{E}{-6} r_{\mathsf{gyro}} + 1{,}23 + \varepsilon^{\mathsf{gyro}} \text{ [V]}} \tag{G.4}$$

for the 1x case. This equation will be applied in chapter 7 when improving the system states using a Kalman filter and practical measurements. One equation will be used for each of the three axes on the IMU. The reason for the unit of $\varepsilon^{\mathsf{gyro}}$ being [V] is because the standard deviation is measured directly from the sensor output in appendix C. A conversion must be performed when the deviation is used for Kalman filter design in chapter 7. From table C.3 in appendix C, the measurements from table G.4 are obtained.

IMU channel	Sample mean	Sample std.
XOUT-1X	1,2110 V	0,0029 V
YOUT-1X	1,2037 V	0,0031 V
ZOUT-1X	1,2052 V	0,0029 V
XOUT-4X	1,2529 V	0,0027 V
YOUT-4X	1,2252 V	0,0027 V
ZOUT-4X	1,2386 V	0,0029 V

Table G.4: *Results from IMU zero-rate measurements (gyroscopes only)*

The data shows that the following standard deviation estimates are used to express the noise distributions of uncertainty for all three axes

$$\sigma_x^{\mathsf{gyro}} = 0{,}0029, \quad \sigma_y^{\mathsf{gyro}} = 0{,}0031, \quad \sigma_z^{\mathsf{gyro}} = 0{,}0029 \quad [\mathrm{V}]$$

with all values provided in the unit [V]. In order to convert the standard deviations into the unit [rad/s], equation G.4 must be reformulated

$$r_{\mathsf{gyro}} = \frac{v_{\mathrm{o}}^{\mathsf{gyro}} - 1{,}23}{830\mathsf{E}{-}6} - \frac{\varepsilon^{\mathsf{gyro}}}{830\mathsf{E}{-}6}$$

meaning that the converted standard deviation σ_c^{gyro} of the gyroscope measurements are given from

$$\sigma_c^{\mathsf{gyro}} = \mathrm{std}\left(-\frac{\varepsilon^{\mathsf{gyro}}}{830\mathsf{E}{-}6}\right) = 1{,}2048\mathsf{E}3\sigma^{\mathsf{gyro}}$$

because of the new relationship

$$\varepsilon_c^{\mathsf{gyro}} \sim \mathcal{N}\left(0, \mathrm{std}^2\left(-\frac{\varepsilon^{\mathsf{gyro}}}{830\mathsf{E}{-}6}\right)\right)$$

This provides the following standard deviations in the unit [rad/s]

$$\sigma_{c,x}^{\mathsf{gyro}} = 0{,}0610, \quad \sigma_{c,y}^{\mathsf{gyro}} = 0{,}0652, \quad \sigma_{c,z}^{\mathsf{gyro}} = 0{,}0610 \quad [\mathrm{rad/s}] \tag{G.5}$$

For the 4x case, the standard deviations are

$$\sigma_{4c,x}^{\mathsf{gyro}} = 0{,}0143, \quad \sigma_{c,y}^{\mathsf{gyro}} = 0{,}0143, \quad \sigma_{c,z}^{\mathsf{gyro}} = 0{,}0153 \quad [\mathrm{rad/s}] \tag{G.6}$$

using the same procedure. The deviations can now be used for Kalman filter design in chapter 7. Next sensor to be described is the accelerometer used to measure linear acceleration.

G.2.3 Acceleration (accelerometer)

The accelerometer is a 3-axis type ADXL335 from Analog Devices, Inc. [4] and is also part of the IMU board. It has the specifications given in table G.7

Supply voltage	1,8-3,6 V
Range (nominal)	±3,6 g
Output sensitivity (nominal)	300 mV/g (@3 V)
Zero-g bias (nominal)	1500 mV
3dB bandwidth	50 Hz (0,1 μF caps.)

Table G.5: *ADXL335 3-axis accelerometer specifications [4]*

Similar to the gyroscope a linear relation between the rate of velocity r_{acc} [g] and the output voltage can be expressed

$$\boxed{v_{\mathrm{o}}^{\mathsf{acc}} = 0{,}3 r_{\mathsf{acc}} + 1{,}5 + \varepsilon^{\mathsf{acc}} \text{ [V]}} \tag{G.7}$$

with $\varepsilon^{\mathsf{acc}}(t) \sim \mathcal{N}(0, [\sigma^{\mathsf{acc}}]^2)$. The standard deviation σ^{acc} is estimated based on measurements as described in appendix C to be $2,5-3,8$ mV according to the direction of motion. From table C.3 in appendix C the measurements from table G.6 are obtained for the accelerometer outputs.

IMU channel	Sample mean	Sample std.
AXOUT	1,4705 V	0,0038 V
AYOUT	1,4353 V	0,0027 V
AZOUT	1,7669 V	0,0025 V

Table G.6: *Results from IMU zero-rate measurements (acclerometer)*

The data show that the following standard deviation estimates are used to express the noise distributions of uncertainty for all three axes

$$\sigma_x^{\mathsf{acc}} = 0{,}0038, \quad \sigma_y^{\mathsf{acc}} = 0{,}0027, \quad \sigma_z^{\mathsf{acc}} = 0{,}0025 \quad \text{[V]}$$

with all values provided in the unit [V]. Similar with the gyroscope the deviations must be converted to be used for Kalman filter design. In order to convert them into the unit [g] equation G.7 must be reformulated

$$r_{\mathsf{acc}} = \frac{v_{\mathrm{o}}^{\mathsf{acc}} - 1{,}5}{0{,}3} - \frac{\varepsilon^{\mathsf{acc}}}{0{,}3}$$

meaning that the converted standard deviation σ_c^{acc} of the accelerometer measurements are given from

$$\sigma_c^{\mathsf{acc}} = \mathrm{std}\left(-\frac{\varepsilon^{\mathsf{acc}}}{0{,}3}\right) = 3{,}3333\sigma^{\mathsf{acc}}$$

because of the new relationship

$$\varepsilon_c^{\mathsf{acc}} \sim \mathcal{N}\left(0, \mathrm{std}^2\left(-\frac{\varepsilon^{\mathsf{acc}}}{0{,}3}\right)\right)$$

This provides the following standard deviations in the unit [g]

$$\sigma_{c,x}^{\mathsf{acc}} = 0{,}0127, \quad \sigma_{c,y}^{\mathsf{acc}} = 0{,}0090, \quad \sigma_{c,z}^{\mathsf{acc}} = 0{,}0083 \quad [\mathrm{g}] \tag{G.8}$$

Similar with the converted standard deviations from the gyroscope subsection the deviations can now be used for Kalman filter design in chapter 7. The last sensor to be described is the strain gauge used to measure strain of the flexible tool.

G.2.4 Tool deflection (strain gauge)

The last sensor applied in the control system is the strain gauge of type N11-MA-5-120-11 from RS Components [27] with specifications given in table G.7. Since the strain gauge is a resistor element and not an integrated circuit with direct signal output, it must be used together with an electronic circuit. In this case a Wheatstone bridge is applied. Using four identical gauges the sensitivity can be determined from [31]

$$S = 0,25 n_s \kappa \epsilon$$

Gauge length	5 mm
Resistance (nominal)	120 Ω
Range	$\pm$(2 - 4 %)
	$\Delta\ell:\ \pm$(0,1 - 0,2 mm)
	$\Delta R:\ \pm$(25,2 - 50,4 Ω)
Gauge factor	2,1

Table G.7: *N11-MA-5-120-11 strain gauge specifications [27]*

where κ is the gauge factor, ϵ the strain of the sensor and n_s the *bridge factor* (number of strain gauges in the bridge, maximum 4). Each strain gauge will experience a change in resistance of $\Delta R = \kappa\epsilon R_0$, with R_0 denoting the nominal resistance of 120 Ω [6]. The bridge voltage output is thus given as

$$v_{\mathrm{o}}^{\mathsf{str}} = S v_s^{\mathsf{str}} + \varepsilon^{\mathsf{str}} = 0,25 n_s \kappa \epsilon v_s^{\mathsf{str}} + \varepsilon^{\mathsf{str}} \ [\mathrm{V}] \tag{G.9}$$

with v_s^{str} [V] denoting the supply voltage of 1,5 V. The stochastic behavior of the measurement is denoted by $\varepsilon^{\mathsf{str}}$. The strain gauge datasheet provides no information about tolerances, thus the only explicit source of error in the circuit is the 5 % gain resistor R_g, which affects the gain of the amplifier part. Equation (G.9) is therefore substituted with the amplifier gain [5]

$$v_{\mathrm{o2}}^{\mathsf{str}} = G_a S v_s^{\mathsf{str}} = \left(\frac{49400}{R_g + \varepsilon^{\mathsf{str}} - 1}\right) 0{,}25 n_s \kappa \epsilon v_s^{\mathsf{str}} \ [\mathrm{V}] \tag{G.10}$$

with G_a denoting the gain of the instrumental amplifier when taking stochastic behavior into consideration. No other sources of error are expressed, when assuming the gain resistor to make the largest contribution to uncertainty. This equation holds whenever $Z_o \sim \infty\,\Omega$ when the amplifier is not loading the bridge circuitry; calling for incorrect measurements. Depending on the size of R_g, the distribution of $\varepsilon^{\mathsf{str}}$ can be determined as

$$\varepsilon^{\mathsf{str}} \sim \mathcal{N}(0, q_c^2 R_g^2) \tag{G.11}$$

by assuming a component tolerance of $q_c = 5$ %. The sensor configuration and amplifier circuit is given in figure G.5.

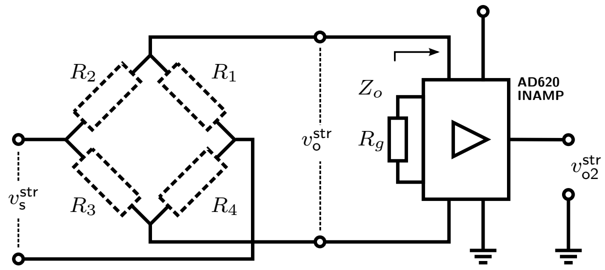


Figure G.5: *Strain gauge configuration circuit using a Wheatstone bridge*

Resistors R_i are representing the strain gauges 1 through 4. By using four identical strain gauges, the circuit is more tolerable to temperature variances and four times as sensitive to strain when compared to a single strain gauge. The AD620 is an instrumental amplifier (INAMP) from Analog Devices [5]. It is configured as a differential amplifier with dual supply, since the positive and negative strain is causing the output to change sign as well. An additional 33 nF has been used on the output terminal to decouple noise. The strain of a single gauge is determined as [48, 13, 31, 103]

$$\epsilon(x) \triangleq \frac{\Delta\ell}{\ell} = -\frac{h\kappa_w(x)}{2}\,\mathrm{sgn}\,\tfrac{\mathrm{d}w}{\mathrm{d}x} = -\frac{h}{2}\left|\tfrac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right|\mathrm{sgn}\,\tfrac{\mathrm{d}w}{\mathrm{d}x} \tag{G.12}$$

where κ_w is the curvature of the beam and w the deflection. The notations are introduced in chapter 4. This formulation also shows that if $h = 0$ the strain of the beam is zero. $h/2$ is the distance from the center line (*neutral axis*) of the beam to the top of the beam. See figure G.6 for the configuration of a Wheatstone bridge for measuring bending in one direction.

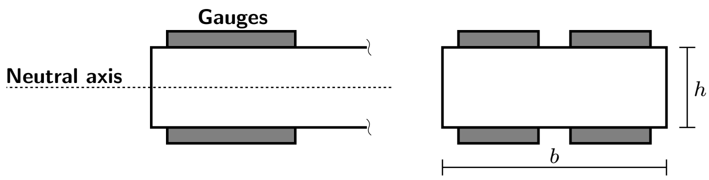


Figure G.6: *Strain configuration on beam*

The strain is therefore determined from a local curvature of the beam by approximating an osculating circle with radius $r_w = 1/\kappa_w$ for that specific point x, where [36]

$$\kappa_w(x) = \left|\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right| \left(1 + \left(\frac{\mathrm{d}w}{\mathrm{d}x}\right)^2\right)^{-3/2} \approx \left|\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right| \tag{G.13}$$

with the approximation based on the assumption $1 \gg \left(\frac{\mathrm{d}w}{\mathrm{d}x}\right)^2$. Only the notation $w(x)$ is used, but the actual arguments are $w(x,t)$. Knowing this relation, (G.10) can be expressed as [6]

$$\boxed{\begin{aligned} v_{o2}^{\mathsf{str}}(t) &= -\left(\frac{49400}{R_g + \varepsilon^{\mathsf{str}} - 1}\right) 0,25 n_s \kappa \frac{h}{2} \left|\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}\right|_{x=\ell^{\mathsf{str}}} \operatorname{sgn} \frac{\mathrm{d}w}{\mathrm{d}x}_{x=\ell^{\mathsf{str}}} v_s^{\mathsf{str}} \\ &= a_c \left| \sum_{i=1}^{M} q_i(t) \frac{\mathrm{d}^2 \phi_i}{\mathrm{d}x^2}\bigg|_{x=\ell^{\mathsf{str}}} \right| \operatorname{sgn}\left(\sum_{i=1}^{M} q_i(t) \frac{\mathrm{d}\phi_i}{\mathrm{d}x}\bigg|_{x=\ell^{\mathsf{str}}} \right) \quad [\mathrm{V}] \end{aligned}} \tag{G.14}$$

given the constant

$$a_c = -0{,}125 \left(\frac{49400}{R_g + \varepsilon^{\mathsf{str}} - 1}\right) n_s \kappa h v_s^{\mathsf{str}}$$

and knowing that ℓ^{str} is the distance between tool frame and strain gauges. The origin of the different parts of the equation involving the flexible tool dynamics will be treated thoroughly in chapter 4. However, the expressions are needed now to model the output of the strain gauge circuitry. It is then possible to determine the sensor output at any time instance, when given the dynamics of the flexible tool. Even though the constant is given from model parameters alone, an empirical gain can be added to adjust the amplitude to a given setup. In order to determine the relation between strain and deflection for the practical configuration, an experiment has been conducted. The experiment is described in appendix F and yields the relation

$$v_{o2}^{\mathsf{str}}(t) = 0{,}1211 w(\ell, t) - 2{,}403 \quad [\mathrm{V}] \tag{G.15}$$

which has been determined using $R_g = 200\ \Omega$. The offset is a consequence of the initial condition of the beam. This relation will be used in chapter 9 when tuning the controller. All the sensors introduced above, besides the joint internal encoders, must be installed at specific locations on the manipulator to achieve satisfactory measurements. This is the topic of the sequel section.

G.3 Sensor configurations

In order to achieve satisfactory results of the measurements, the sensors must be configured correctly. Especially the strain gauges will yield useless results if not located in the right distance from the tool mount. This is due to the modes of the tool structure, which causes certain points to be stationary. Based on theory derived in chapter 4 the dead spots can be determined. A dead spot is a spatial location on the beam, where no strain can be measured - even if the rest of the beam is bending. Because a point has no defined length, the finite length strain gauge will span a collection of points. A single point of zero strain is therefore not a problem. However, since the slope is changing sign around the point, the strain gauge will be both stretched and compressed, which is resulting in a zero strain read-out in the symmetrical case.

The gauges must therefore be located away from dead spots to avoid any misreading. Figure G.7 shows a plot of $\frac{\mathrm{d}^2 w}{\mathrm{d}x^2}$ which is used to calculate the strain. The best possible location of the gauges is therefore at $x = 0$, which will provide the most sensitive sensor configuration. From a practical view point, the gauges have been mounted at 7,56 % up the beam ($\ell^{\mathsf{str}} = 30$ mm of 397 mm tool without tip mass), which is considered far enough from the nearest dead spot. The figure is simulated using (4.4) without time dynamics.

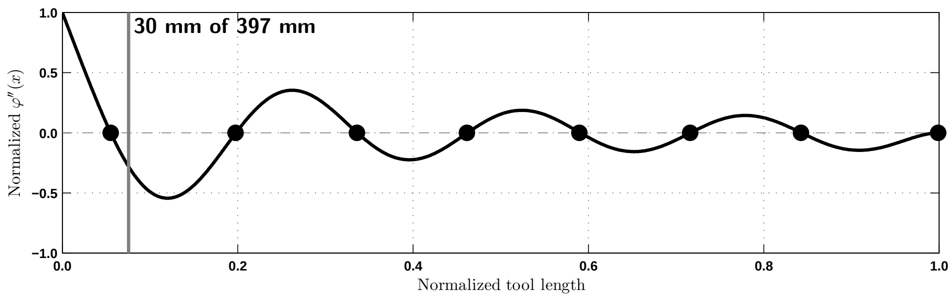


Figure G.7: *Dead spots on beam depending on the location (inspired by fig. 3.5 in [6])*

This method is, however, not providing a unique solution. All mode shape functions in the simulation are normalized at $x = \ell$, because the exact scaling is not known. A different scaling will provide a different result. Furthermore, because the resonance frequencies are unlikely to be multiples of each other, the dead spots will depend on time. Since the point of zero dynamics of two eigenmodes is not coinciding, there will not be a single point in which to place the gauges. However, it is assumed that the variation is not critical since the gauges have a length of 5 mm. No more theory will be given about this topic. Empirical observations have shown, that the issue is of no important regarding this project anyway. A principle sketch of the accelerometer and gyroscope locations on the IMU-board is given in figure G.8 along with locations of strain gauges to measure deflection in two directions.

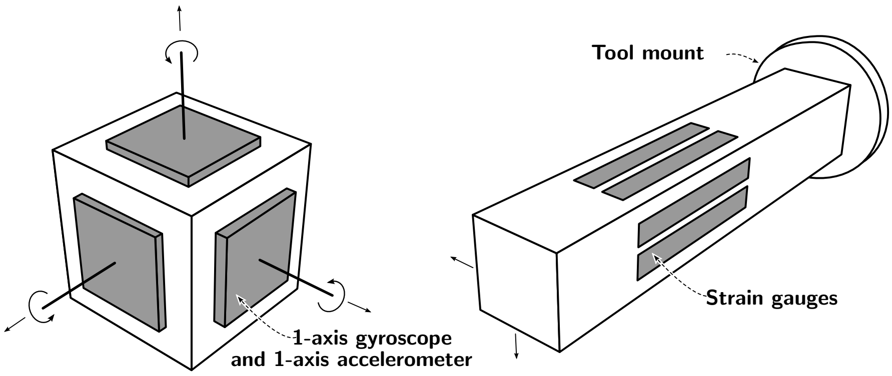
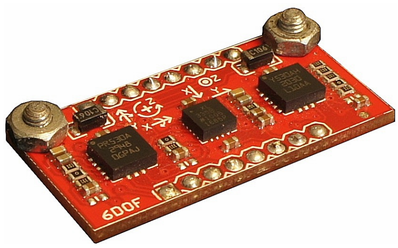
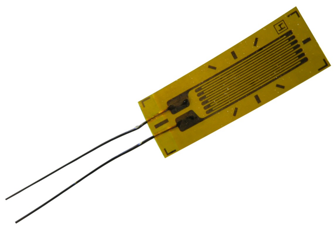


Figure G.8: *IMU construction (left) and configuration of strain gauges (right)*

The cube shape is an illustrative example of the orientation of the sensors on the IMU sensor board. As will be given in figure G.11 later on the strain gauges are located at ℓ^{str}, which is selected to avoid locations of zero-dynamics up to a certain eigenmode. The different mode shapes and points of zero dynamics are described in details in chapter 4. An image of the IMU-board is shown in figure G.9 together with a single strain gauge. Figure G.10 shows the IMU installed on the manipulator.



(a) *IMU board*



(b) *Strain gauge*

Figure G.9: *Images of sensors from figure G.8*

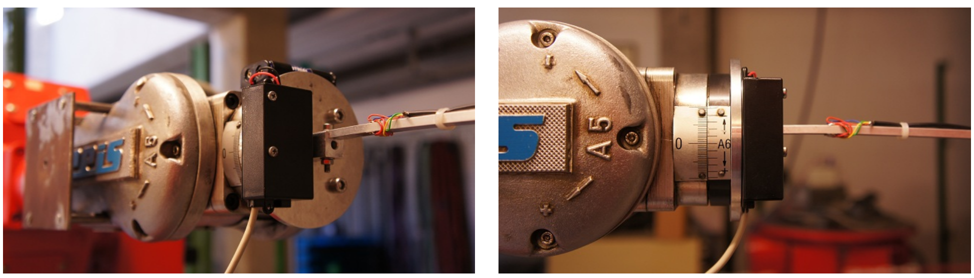


Figure G.10: *IMU installed on REIS RV15*

Setup of the IMU board is also crucial in order to achieve useful measurements. Due to the hardware configuration it is not possible to locate the sensors at the origo of the tool frame $\mathbb{F}_\tau$. Instead the board will have to be translated by the transformation ${}^{\tau}_{\mathsf{IMU}}\mathbf{T}$ (derivation and interpretation of kinematic transforms is given in chapter 6), which is given as

$$ {}^{\tau}_{\mathsf{IMU}}\mathbf{T} = \begin{bmatrix} 0 & 0 & 0 & x^{\mathsf{IMU}} \\ 0 & 0 & 0 & y^{\mathsf{IMU}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{G.16} $$

with the position $(x^{\mathsf{IMU}}, y^{\mathsf{IMU}})$ given as distances from the tool frame origo to the center of the IMU-board. The board is 35×18 mm and assumed of no size when stating the location of the individual sensors. All sensors have been introduced and modeled, and the remainder of this appendix is dedicated to the flexible tool analog. It describes the physical configuration and the material properties needed to estimate the beam dynamics in chapter 4.

G.4 Flexible tool specifications

A flexible tool has been constructed for testing on the manipulator. The specifications are denoted as the Ξ-system throughout the report and contains the following terms and values

Material density[2] $\rho = 2700$ kg/m^3 (aluminum) [105]
Dimensions[3] $w_b \times h_b \times \ell = 0{,}01 \times 0{,}0048 \times \{0{,}381; 0{,}397\}$ m
Cross sectional area $a_b = w_b \times h_b = 5\text{E}{-}5\ m^2$
Young's modulus[4]: $E = 73{,}1\text{E}9$ N/m^2 [105]
Second moment of inertia: $I = \frac{w_b h_b^2}{12} = 1{,}0417\text{E}{-}10$ m^4 [6]
Tip masses[5]: $m_{l1} = 0{,}351$ kg (ratio[6] 7,1) and $m_{l2} = 0{,}522$ kg (ratio 10,6)

with w_b, h_b and ℓ denoting the beam width, height and length, respectively. An illustration of the tool design is depicted in figure G.11. Strain gauges are mounted on the beam as shown in figure G.11 with $\ell^{\mathsf{str}} = 30$ mm. The distance is the location where the largest strain can be experienced from a practical viewpoint. Furthermore, the location is not a points of zero dynamics (eigenfunctions equal zero), which will otherwise cause the output from the gauges to be zero independent of the beam deflection. Figure G.12 shows the actual beam used for testing.

[2] The actual alloy of the flexible tool is unknown, and is therefore based on an arbitrary selected type - in this case aluminum 6061-T6 [105]
[3] Lengths are denoted for the case with and without tip mass, respectively
[4] The actual alloy of the flexible tool is unknown, and is therefore based on an arbitrary selected type - in this case aluminum 6061-T6 [105]
[5] Measured on AAU86759 Kern FCB 12K1, mass includes mounting screw
[6] The ratio is introduced in chapter 4 and is relating beam mass and tip mass

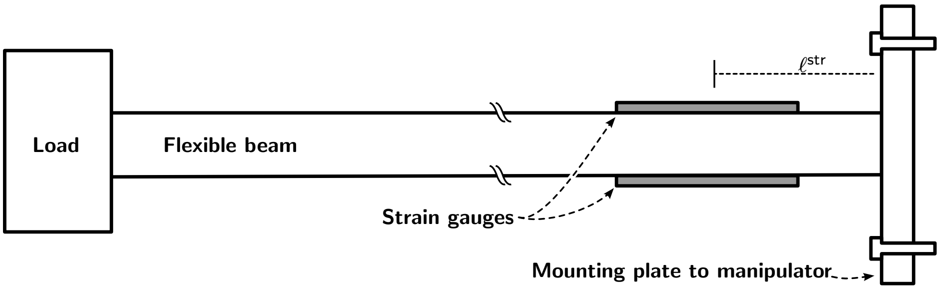


Figure G.11: *Flexible tool configuration (illustration)*

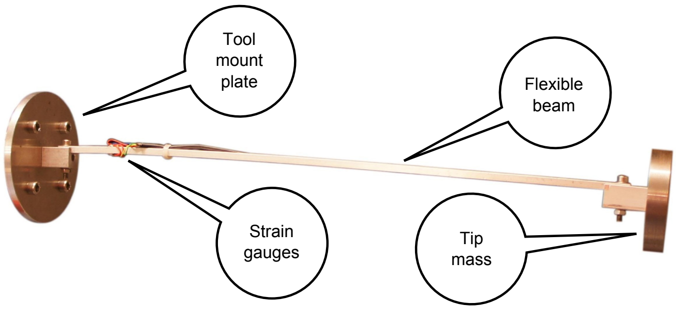


Figure G.12: *Flexible tool configuration (image from laboratory)*

Due to assumption 3 (introduced in chapter 4), strain gauges will only be mounted in one direction. Bending around one axis is therefore the only strain to be measured. This ends the hardware description, describing the manipulator configuration, sensors and the flexible tool.

Appendix H

Kinematic manipulator model

A *kinematic model* is used to describe the spatial position of any point on a manipulator as a function of the joint space. The so-called *DH-parameters* are used to specify the configuration of a particular manipulator type, and will be used to express a *generalized transformation matrix*. This matrix describes the position and orientation of one frame with respect to the previous one. Lastly, a complete transformation will be derived to describe the relation between the generalized coordinate vector θ and the tool center point within the production cell coordinate system. Figure 3.5 is repeated in figure H.1 to show the location of the different frames referred to in the sequel.

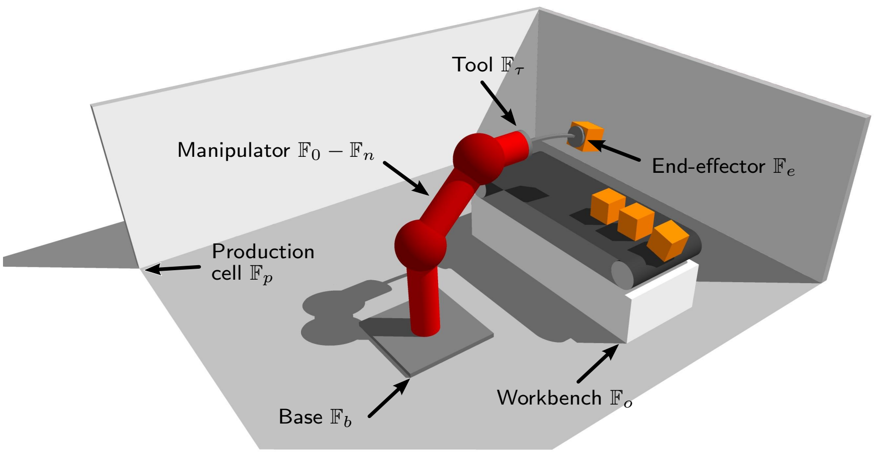


Figure H.1: *Sketch of frames in production cell*

H.1 Frame orientations

Depending on the order of rotations and translations from one link to the next, the transformation matrix can be expressed in different ways according to the applied convention. The general convention concerning the order of rotation and translation used throughout the report is denoted in convention 1. Furthermore, the terms translation and displacement will be used interchangeably throughout the report. If other conventions are applied, it will be explicitly announced.

A generalized transformation matrix can be derived from the convention by inserting the matrices corresponding to the rot(·) and dis(·) functions. The rotation matrices are each defined as [86]

$$\begin{aligned}
\operatorname{rot}(X_k, \theta_k)^* &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\,\theta_k & -\mathrm{s}\,\theta_k \\ 0 & \mathrm{s}\,\theta_k & \mathrm{c}\,\theta_k \end{bmatrix} \\
\operatorname{rot}(Y_k, \theta_k)^* &= \begin{bmatrix} \mathrm{c}\,\theta_k & 0 & \mathrm{s}\,\theta_k \\ 0 & 1 & 0 \\ -\mathrm{s}\,\theta_k & 0 & \mathrm{c}\,\theta_k \end{bmatrix} \\
\operatorname{rot}(Z_k, \theta_k)^* &= \begin{bmatrix} \mathrm{c}\,\theta_k & -\mathrm{s}\,\theta_k & 0 \\ \mathrm{s}\,\theta_k & \mathrm{c}\,\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{H.1}$$

The rotation matrices are elementary rotations, since the rotation is not performed around an arbitrary axis but instead around one of the three elementary axes. An asterisk * is used to indicate, that the rotations cannot be inserted into the definition from the convention directly because $\operatorname{rot}(X_k, \theta_k)^* \in \mathbb{R}^{3\times 3}$ while ${}^{i}_{j}\mathbf{T} \in \mathbb{R}^{4\times 4}$. The conversion is shown shortly. A translation is on the other hand a zero vector with one non-zero element in the displacement direction given as

$$\begin{aligned}
\operatorname{dis}(X_k, d_k)^* &= \begin{bmatrix} d_k & 0 & 0 \end{bmatrix}^\intercal \\
\operatorname{dis}(Y_k, d_k)^* &= \begin{bmatrix} 0 & d_k & 0 \end{bmatrix}^\intercal \\
\operatorname{dis}(Z_k, d_k)^* &= \begin{bmatrix} 0 & 0 & d_k \end{bmatrix}^\intercal
\end{aligned} \tag{H.2}$$

A general translation in an arbitrary direction can of course also be defined, but this is not used for the purpose of deriving a general transformation description using DH-parameters. Similar with the rotation, an asterisk * is added to the definitions. This is due to the fact, that the translation vectors given cannot be multiplied together with the rotational matrices to provide the desired result from convention 1, since they must be added in their current form. A vector ${}^{j}q$ described in frame j must be firstly rotated to obtain equal orientation as frame i, and then offset by the displacement between the two frame origo. The operation is given by ${}^{i}q = \operatorname{rot}(v_k, \theta_k)\,{}^{j}q + \operatorname{dis}({}^{i}\nu_j, ||{}^{i}\nu_j||_2)$ with $v_k \in \mathbb{R}^3$ denoting the axis of rotation to align the two frames through the rotation θ_k, and ${}^{i}\nu_j \in \mathbb{R}^3$ the direction of displacement by the rate $||{}^{i}\nu_j||_2$. This last measure is the distance between the frame origo of frame i and j.

Both the rotation matrix and the translation vector can be combined into what is known as a *homogeneous transform* [30] by padding them with ones and zeros in the following way

$$\operatorname{srw}(v_k, d_k, \theta_k) = \operatorname{dis}(v_k, d_k)\operatorname{rot}(v_k, \theta_k) = \left[\begin{array}{c|c} \operatorname{rot}(v_k, \theta_k)^* & \operatorname{dis}(v_k, d_k)^* \\ \hline 0 & 1 \end{array}\right] \tag{H.3}$$

with $\operatorname{srw}(v, d_k, \theta_k) \in \mathbb{R}^{4\times 4}$ defining the *screw* of a frame around an arbitrary axis $v_k \in \mathbb{R}^3$ by the amount $\theta \in [-\pi/2, \pi/2]$ along with a displacement $d_k \in \mathbb{R}$ in the direction of v_k. Both the rotation and the translation are then combined into a single matrix operation by extending the position vectors from $\mathbb{R}^3$ to $\mathbb{R}^4$, and now $\operatorname{rot}(\cdot), \operatorname{dis}(\cdot) \in \mathbb{R}^{4\times 4}$. The expanded operations will not be given explicitly, since the $\operatorname{srw}(\cdot)$ mapping is the only one needed in the sequel. A mapping $C : \mathbb{R}^4 \mapsto \mathbb{R}^4$ is thus constructed to map between two frames represented in 4 dimensions. Equation H.3 can be substituted into the equation from convention 1 to give

$$\begin{aligned} {}^{i}_{j}\mathbf{T} &= \operatorname{srw}(X_i, \ell^{\mathsf{DH}}_{Xi}, \varphi^{\mathsf{DH}}_{Xi}) \operatorname{srw}(Z_j, \ell^{\mathsf{DH}}_{Zj}, \varphi^{\mathsf{DH}}_{Zj}) \\ &= \begin{bmatrix} \cos\varphi^{\mathsf{DH}}_{Zj} & -\sin\varphi^{\mathsf{DH}}_{Zj} & 0 & \ell^{\mathsf{DH}}_{Xi} \\ \cos\varphi^{\mathsf{DH}}_{Xi}\sin\varphi^{\mathsf{DH}}_{Zj} & \cos\varphi^{\mathsf{DH}}_{Xi}\cos\varphi^{\mathsf{DH}}_{Zj} & -\sin\varphi^{\mathsf{DH}}_{Xi} & -\ell^{\mathsf{DH}}_{Zj}\sin\varphi^{\mathsf{DH}}_{Xi} \\ \sin\varphi^{\mathsf{DH}}_{Xi}\sin\varphi^{\mathsf{DH}}_{Zj} & \sin\varphi^{\mathsf{DH}}_{Xi}\cos\varphi^{\mathsf{DH}}_{Zj} & \cos\varphi^{\mathsf{DH}}_{Xi} & \ell^{\mathsf{DH}}_{Zj}\cos\varphi^{\mathsf{DH}}_{Xi} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{H.4}$$

which is coinciding with the definition of [30] because an equivalent order of rotation and translation was applied during the derivation. After expressing all transformations from the previous link to the next, a complete description from the end-effector frame $\mathbb{F}_e$ to the frame attached to the production cell $\mathbb{F}_p$ can be obtained. This complete transformation matrix depends on the reachability of the manipulator including revolution angles and/or prismatic strokes. Using the newly derived linear mapping $\{C : {}^{i}_{j}\mathbf{T}\}$ from one frame to a neighboring one, a mapping from the base to the n-th frame can be expressed for an n-DOF manipulator

$${}^{0}_{n}\mathbf{T}(\theta) = \prod_{\substack{j=1 \\ i=j-1}}^{n} {}^{i}_{j}\mathbf{T}(\theta_j) \tag{H.5}$$

which is depending on the generalized coordinate vector θ. Using this resulting transformation (or representation of orientation and position of the n-th frame), the end-effector frame $\mathbb{F}_e$ can be expressed relative to the production cell frame $\mathbb{F}_p$ [86]

$${}^{o}_{e}\mathbf{T}(t, \theta, \varpi) = {}^{p}_{o}\mathbf{T}^{-1}(t)\, {}^{p}_{b}\mathbf{T}(t)\, {}^{b}_{0}\mathbf{T} \left(\prod_{j=1}^{n} {}^{i}_{j}\mathbf{T}(\theta_j) \right) {}^{n}_{\tau}\mathbf{T}\, {}^{\tau}_{e}\mathbf{T}(\varpi, t) \tag{H.6}$$

o	Workbench frame $\mathbb{F}_o$	p	Production cell frame $\mathbb{F}_p$
b	Base frame $\mathbb{F}_b$	$[0, n]$	Manipulator frames $\mathbb{F}_i$
τ	Tool frame $\mathbb{F}_\tau$	e	End-effector frame $\mathbb{F}_e$

which is also given in (3.2). The time varying transformations ${}^{p}_{b}\mathbf{T}(t) \in \mathbb{R}^{4\times4}$, ${}^{b}_{0}\mathbf{T} \in \mathbb{R}^{4\times4}$ and ${}^{n}_{e}\mathbf{T}(t) \in \mathbb{R}^{4\times4}$ representing the manipulator base in reference with the production cell, the base frame in reference with the zero frame and the end-effector e in reference with the n-th frame, respectively. Time dependency of ${}^{p}_{0}\mathbf{T}(t)$ can be caused by a moving manipulator within the production cell, and time dependency of ${}^{n}_{e}\mathbf{T}(t)$ caused by the flexibility of the end-effector (represented by the $(n+1)$-th link). Other arguments can be added to yield desirable behavior. The inverse transform ${}^{p}_{o}\mathbf{T}(t)^{-1}$ has been applied to express a workbench o within the production cell. Additional frames can be added if needed. Next section describes how the DH-parameters are located on the manipulator.

H.2 Manipulator geometry

Based on the hardware configuration used for this project, the DH-parameters of the REIS RV15 manipulator are given in table 3.1. The general structure of the DH-parameters was defined based on pictures of the manipulator, and the parameters $\ell^{\mathsf{DH}}_{X_i}$ and $\ell^{\mathsf{DH}}_{Z_j}$ were given from physical measurements on the REIS RV15 as well as from a previous project using the same robot [25]. The reason for the non-unique approach to form a description of the manipulator configuration is caused by different uses of the final model. If the link exact link positions in between the base and the end-effector frame are not needed, several frames can be coinciding with each other to simplify the model. For the purpose of this project, a model is expressed by the DH-parameters from [25] with some sign changes.

Using the generalized transformation matrix from equation H.4, a complete transform from base to the 6-th link can be derived. Firstly, each of the separate transforms are explicitly expressed

$$
{}^{0}_{1}\mathbf{T}(\theta_1) = \begin{bmatrix} \mathrm{c}\,\theta_1 & -\mathrm{s}\,\theta_1 & 0 & 0 \\ \mathrm{s}\,\theta_1 & \mathrm{c}\,\theta_1 & 0 & 0 \\ 0 & 0 & 1 & \ell^{\mathsf{DH}}_{Z1} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad
{}^{1}_{2}\mathbf{T}(\theta_2) = \begin{bmatrix} \mathrm{c}\,\theta_2 & -\mathrm{s}\,\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\mathrm{s}\,\theta_2 & -\mathrm{c}\,\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}\mathbf{T}(\theta_3) = \begin{bmatrix} \mathrm{c}\,\theta_3 & -\mathrm{s}\,\theta_3 & 0 & \ell^{\mathsf{DH}}_{X2} \\ \mathrm{s}\,\theta_3 & \mathrm{c}\,\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad
{}^{3}_{4}\mathbf{T}(\theta_4) = \begin{bmatrix} \mathrm{c}\,\theta_4 & -\mathrm{s}\,\theta_4 & 0 & 0 \\ 0 & 0 & 1 & \ell^{\mathsf{DH}}_{Z4} \\ -\mathrm{s}\,\theta_4 & -\mathrm{c}\,\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}\mathbf{T}(\theta_5) = \begin{bmatrix} \mathrm{c}\,\theta_5 & -\mathrm{s}\,\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\mathrm{s}\,\theta_5 & -\mathrm{c}\,\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad
{}^{5}_{6}\mathbf{T}(\theta_6) = \begin{bmatrix} \mathrm{c}\,\theta_6 & -\mathrm{s}\,\theta_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \mathrm{s}\,\theta_6 & \mathrm{c}\,\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

A multiplication of the individual representations of orientation between consecutive links will provide the resulting orientation of the 6-th frame $\mathbb{F}_6$ in reference with the manipulator base frame $\mathbb{F}_b$

$$
{}^{0}_{6}\mathbf{T}(\theta) = {}^{0}_{1}\mathbf{T}(\theta_1){}^{1}_{2}\mathbf{T}(\theta_2){}^{2}_{3}\mathbf{T}(\theta_3){}^{3}_{4}\mathbf{T}(\theta_4){}^{4}_{5}\mathbf{T}(\theta_5){}^{5}_{6}\mathbf{T}(\theta_6) \tag{H.7}
$$

and with pre-multiplication of production cell dependent orientations and post-multiplication of tool related transforms form the final transform from (H.6). The complete transformation matrix will not be given explicitly, as it is rather complex. The mapping ${}^{0}_{6}\mathbf{T}$ is given explicitly in appendix N, and the zero configuration of the manipulator is given by $\theta = \mathbf{0}$. Other mappings are not used for the purpose of this project, as the manipulator is not moving, and no workbench is present in the laboratory. Figure H.2 shows the involved frames based on the DH-parameters.

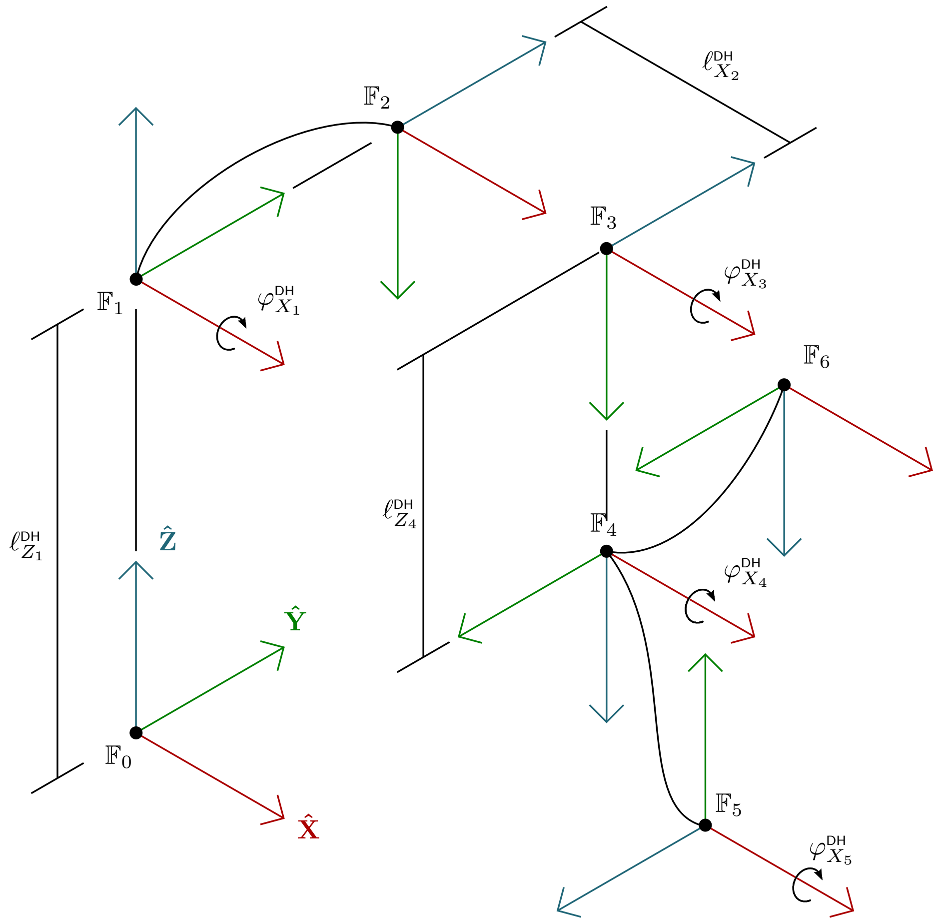


Figure H.2: *Graphical representation of frames describing the manipulator by DH-parameters*

The above theory has been derived for the entire 6-DOF manipulator, but as a consequence of remark 1 on page 18, there is no need to include the rotation axes 1, 4 and 6. Only the swivel axes 2, 3 and 5 are necessary [80]. Accordingly, the kinematics becomes simpler, and to further simplify things, the inactive axes will be removed from the link indices as well. Therefore, the links are now enumerated as follows.

θ_2 is renamed to θ_1
θ_3 is renamed to θ_2
θ_5 is renamed to θ_3
$\theta_1, \theta_4, \theta_6$ are ignored

A new set of DH-parameters can be derived, which is given in the main report in table 3.2 with corresponding transformation matrices expressed on page 22 from the base to the third frame. Before ending the chapter, a short example 1 is presented. It shows how to determine the tip position of the manipulator (origo of frame $\mathbb{F}_6$) as a function of θ. The example is using the non-reduced 6-DOF manipulator model.

Example 1: *The tip-position* ${}^0\nu_6(\theta)$*, defined as the origo of frame* $\mathbb{F}_6$ *in reference with the manipulator zero frame* $\mathbb{F}_0$*, is determined as a function of the generalized coordinate vector* θ *using the complete transformation matrix in* (H.7)*, yielding*

$$ {}^0\nu_6(\theta) = {}^0_6\mathbf{T}\,(\theta)\nu_6 = {}^0_6\mathbf{T}\,(\theta)\begin{bmatrix}\mathbf{0} & 1\end{bmatrix}^\mathsf{T} $$

with the zero-one vector being a consequence of the use of homogeneous transformations, which include a translation term. Using the DH-parameters from table 3.1, the origo of the 6-th link can be explicitly expressed as

$$ {}^0\nu_6(\theta) = \begin{bmatrix} \ell^{\mathsf{DH}}_{X_2}\,\mathrm{c}\,\theta_1\,\mathrm{c}\,\theta_2 - \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{c}\,\theta_1\,\mathrm{c}\,\theta_2\,\mathrm{s}\,\theta_3 + \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{c}\,\theta_1\,\mathrm{c}\,\theta_3\,\mathrm{s}\,\theta_2 \\ \ell^{\mathsf{DH}}_{X_2}\,\mathrm{c}\,\theta_2\,\mathrm{s}\,\theta_1 - \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{c}\,\theta_2\,\mathrm{s}\,\theta_1\,\mathrm{s}\,\theta_3 + \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{c}\,\theta_3\,\mathrm{s}\,\theta_1\,\mathrm{s}\,\theta_2 \\ \ell^{\mathsf{DH}}_{Z_1} - \ell^{\mathsf{DH}}_{X_2}\,\mathrm{s}\,\theta_2 - \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{c}\,\theta_2\,\mathrm{c}\,\theta_3 - \ell^{\mathsf{DH}}_{Z_4}\,\mathrm{s}\,\theta_2\,\mathrm{s}\,\theta_3 \end{bmatrix} $$

with the permuted 1 removed from the vector. Contributions from θ_4 *through* θ_6 *will take place only when a vector* $v \neq \bar{0}$ *is added to the last frame. This will be the case when adding a tool to the manipulator.* □

Next, the transformations will be used to express linear and angular velocities of specific points and bodies on the manipulator. This will come in handy when describing the dynamics involved in the next chapter.

H.3 Derivatives of position and joint angle

When modeling the dynamics of the manipulator, derivatives of positions and orientations are needed. Based on the definition given in chapter 3, a rotation is always performed around the third axis in a Cartesian frame description. The link rotation vector can therefore be described as [30]

$$ \bar{\omega}_k \equiv \dot{\theta}_k\mathbf{e}_3 = \begin{bmatrix}0 & 0 & \dot{\theta}_k\end{bmatrix}^\mathsf{T} \tag{H.8} $$

where $\bar{\omega}_k$ denotes the *angular velocity vector* of link k in reference with frame k itself. The process of determining the rotation of the following link (towards the end-effector, from base to n-th link) consists of adding additional rotational contributions to the previous link, described by

$$\bar{\omega}_j = {}^j\bar{\omega}_i + \dot{\theta}_j \mathbf{e}_3 = {}^j_i\mathbf{R}\,\bar{\omega}_i + \begin{bmatrix} 0 & 0 & \dot{\theta}_j \end{bmatrix}^{\mathsf{T}} \tag{H.9}$$

with ${}^j_i\mathbf{R} = \mathrm{rot}(v_k, \theta_k)^*$ expressing the pure rotation part of the homogeneous transformation ${}^j_i\mathbf{T}$ from equation H.3. Thus to find the resulting angular rotation of link j (as seen from frame j), add the rotation of link j to the angular rotation of link i in reference with frame j. Figure H.3 interprets equation (H.9) graphically.

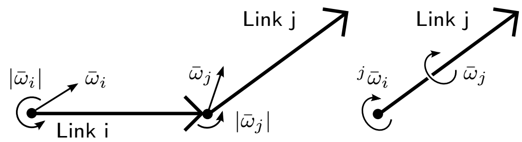


Figure H.3: *Graphical representation of* (H.9) *in chain and in free-body diagram*

Therefore, by changing the reference of the angular velocity $\bar{\omega}_i$ from frame i to j, the velocities can be added. The effect is verified, when selecting $\bar{\omega}_i = -\bar{\omega}_j$, making the resulting angular velocity equal to the zero vector, and thus no rotation is experienced from the j-th link point of view. The first subfigure shows both the axis of rotation $\bar{\omega}_k$ and the rate of rotation $|\bar{\omega}_k|$. Second subfigure, shows implicitly the angular rotation by the compact notation $\bar{\omega}_k$.

Because of the DH-notation with rotation of a link around the z-axis $\bar{\omega}_k = \dot{\theta}_k \mathbf{e}_3$, the linear velocity of the point of rotation of the j-th link can be obtained by

$$\bar{v}_j = {}^j_i\mathbf{R}\,{}^i\bar{v}_j = {}^j_i\mathbf{R}\left[\bar{v}_i + \bar{\omega}_i \times {}^i\nu_j\right] \tag{H.10}$$

Note again that the expression $\bar{v}_j$ is always implicitly given with respect to its own frame j. The velocity $\bar{v}_i$ is another expression of the first derivative of the corresponding position vector ${}^i\nu_j$ added a velocity contribution from the previous link. This is similar to adding initial conditions to a process. Similar with the angular velocity, equation (H.10) can be illustrated graphically, see figure H.4.

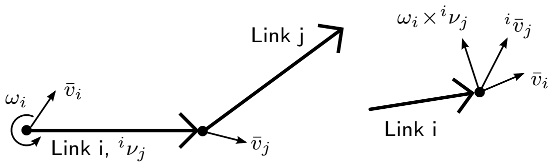


Figure H.4: *Graphical representation of* (H.10) *in chain and in free-body diagram*

Explicit expressions are given in appendix N. The position vectors ${}^i\nu_j$ are given for the original REIS RV15 manipulator configuration as (here without the padded one at the end)

$${}^0\nu_1 = \ell^{\mathsf{DH}}_{Z1}\mathbf{e}_3, \quad {}^1\nu_2 = \bar{0}, \quad {}^2\nu_3 = \ell^{\mathsf{DH}}_{X2}\mathbf{e}_1, \quad {}^3\nu_4 = \ell^{\mathsf{DH}}_{Z4}\mathbf{e}_2, \quad {}^4\nu_5 = \bar{0}, \quad {}^5\nu_6 = \bar{0}$$

When applying remark 1 from page 18, the position vectors ${}^i\nu_j$ can be written in a compact form

$$ {}^{i}\nu_j = \ell^{\mathsf{DH}}_{Xi}\mathbf{e}_1 \tag{H.11} $$

because the manipulator is operating in the plane (X_0, Y_0) only. Based on this information, the center of gravity vectors p_i, which will be used in the dynamics part, can be expressed as

$$ p_i = \tfrac{1}{2}{}^{i}\nu_j = \tfrac{1}{2}\ell^{\mathsf{DH}}_{Xi}\mathbf{e}_1 \tag{H.12} $$

when assuming a homogeneous mass distribution of each link. If the original manipulator configuration was maintained, the center of gravity vectors would have been more complicated because the frame origo are not coinciding with each joint.

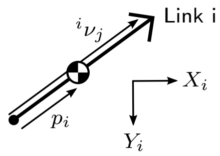


Figure H.5: *Center of gravity vector location on manipulator link*

The basic kinematics has been explained, and the remainder of this section will apply the theory in a number of examples, see example 2 and 3. Later, the singular points of the manipulator are analyzed on the basis of the derived kinematics.

> **Example 2:** *A location on the manipulator can be described according to any other point expressed in a second frame. A vector ${}^{b}q$ describes a vector q expressed in terms of the b-th frame. The same vector can be described in another frame by ${}^{a}q = {}^{a}_{b}\mathbf{T}\,{}^{b}q$. When expressed according to another point ${}^{a}v$, the full transformation is given by*
>
> $$ {}^{a}q = \left({}^{a}_{b}\mathbf{T} + \begin{bmatrix}\mathbf{0}^{[4\times3]} & {}^{a}v\end{bmatrix}\right){}^{b}q = {}^{a}_{b}\tilde{\mathbf{T}}{}^{b}q $$
>
> *which shows the benefits of using 4×4 transformation matrices, since all operations can be collected within a single matrix multiplication ${}^{a}_{b}\tilde{\mathbf{T}}$. This theory comes in handy when considering velocities for points that are not an origo location of a DH-frame.* □

Example 3: *The velocity of a point* ${}^{b}v$ *in reference with frame* b *is given by* ${}^{b}\dot{v} = \bar{v}_b + {}^{b}\bar{\omega}_p \times {}^{b}v$, *and for expressing this velocity according to frame* a *(given that* $a = b - 1$*) the definition from equation* (H.10) *yields*

$${}^{a}\dot{v} = \bar{v}_a + \bar{\omega}_a \times {}^{a}\nu_b + {}^{a}_{b}\mathbf{R}({}^{b}\bar{\omega}_p \times {}^{b}v)$$

with the middle term taking care of coupling the simultaneous translations and rotation. In the case when the constraint is altered to $a < b - 1$

$${}^{a}\dot{v} = \bar{v}_a + \sum_{k=a+1}^{b-1} {}^{a}_{k}\mathbf{R}\,(\bar{\omega}_k \times {}^{k}\nu_{k+1}) + {}^{a}_{b}\mathbf{R}({}^{b}\bar{\omega}_p \times {}^{b}v)$$

This expansion of the definition in (H.10) *becomes important for determining the velocity of the link center of gravity, which is related with the physical frames and not the DH-frames.* □

After deriving the basic kinematics of a manipulator, the singular points are to be identified from that description. These are necessary to know before applying a controller to the system, since they define extreme locations of the convex hull covering the configuration space.

H.4 Singularity analysis

By organizing $\bar{v}_j$ and $\bar{\omega}_j$ in a matrix structure, singularity issues can be revealed. The Jacobian structure is the target here [86]

$$\begin{bmatrix} \bar{v} & \bar{\omega} \end{bmatrix}^{\mathsf{T}} = \mathbf{J}(\theta)\dot{\theta} = \begin{bmatrix} \frac{\partial \bar{v}}{\partial \theta} & \frac{\partial \bar{\omega}}{\partial \theta} \end{bmatrix}^{\mathsf{T}} \dot{\theta} \tag{H.13}$$

with $\mathbf{J}$ denoting the Jacobian of linear and angular velocity expressions. The equation must be generalized to handle non-linearities, since the velocity equations (H.9) and (H.10) in general are nonlinear. It can therefore be evaluated around an operation point $\bar{\theta}$ similar to using a *multivariate Taylor approximation*, and therefore

$$\begin{bmatrix} \bar{v} & \bar{\omega} \end{bmatrix}^{\mathsf{T}} = \mathbf{J}(\theta)\big|_{\bar{\theta}}\dot{\theta} \tag{H.14}$$

For singularity analysis, it is important that $\mathbf{J}(\theta) \in \mathbb{R}^{n \times n}$ such that the determinant is valid. A singularity is defined as points in the system domain, where one or several of the states cannot be driven to the output by anything but infinite control signals. This is a consequence of taking the inverse of the Jacobian, and the case if a limitation in the mechanical reachability arises. Therefore, applying control close to singularities must be avoided, because one degree of freedom is removed. The specific points can be located from $\det \mathbf{J}(\theta) = 0$ [30], where $\operatorname{rank}(\mathbf{J}) < n$. In order to prevent confusions about what $\bar{v}$ and $\bar{\omega}$ to use, the expression in equation (H.13) is re-written to

$$\begin{bmatrix}\bar{v}_k & \bar{\omega}_k\end{bmatrix}^\mathsf{T} = {}^k\mathbf{J}(\theta)\dot{\theta} = \begin{bmatrix}\frac{\partial \bar{v}_k}{\partial \theta} & \frac{\partial \bar{\omega}_k}{\partial \theta}\end{bmatrix}^\mathsf{T}\dot{\theta} = \begin{bmatrix} \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_1} & \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_2} & \cdots & \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_1} & \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_2} & \cdots & \frac{\partial \bar{v}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_n} \\ \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_1} & \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_2} & \cdots & \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_1}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_1} & \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_2} & \cdots & \frac{\partial \bar{\omega}_k^\mathsf{T}\mathbf{e}_3}{\partial \theta_n} \end{bmatrix}\dot{\theta} \tag{H.15}$$

By substituting known expressions of linear and angular velocity into the Jacobian, the determinant can be expressed explicitly. Due to the complexity of the Jacobian, the determinant equation will not be solved analytically. Instead, the problem can be solved using knowledge of the manipulator structure and numerical methods. Also, the control is confined to a small region around the zero position. Next appendix will express the dynamics of the manipulator.

Appendix I

Dynamic manipulator model

A kinematic model is only useful to calculate steady-state positions of the manipulator. A dynamics model is on the other hand capable of expressing the manipulator dynamics in terms of differential equations. The general dynamics model is given from (3.5) and is repeated here [86]

$$\mathbf{M}(\theta, t)\ddot{\theta} = \tau(t) - \mathbf{C}(\theta, \dot{\theta}) - \mathbf{F}(\dot{\theta}) - \mathbf{G}(\theta) \tag{I.1}$$

Different terms makes up the manipulator dynamics model, which will each be described in the sequel sections. When constructing the model, the terms will not be derived individually and combined later on. In stead the entire model will be generated using an iterative procedure as described at the end of the chapter.

I.1 Manipulator link inertia

The first term in the dynamics model described by equation (I.1) is the mass matrix, which is dependent on the geometry and materials of the manipulator hardware. A general expression of the mass matrix cannot be derived, because it involves the specific homogeneous transformations describing the manipulator. Instead, the *inertia tensors* describing the inertia of each link can be given explicitly, based on the following assumptions

- Homogeneous mass distribution of the manipulator links
- Geometry of links is thin rod with mass m_i

Based on these assumptions, the inertia tensor of each of the links can be determined from the moment of inertia I_{cm} of a rod around the center of gravity cm as [12]

$$I_{\mathsf{cm},i,k} = \begin{cases} 1/12 m_i \ell_i^2 & k\text{-axis non-parallel to link} \\ 0 & k\text{-axis parallel to link} \end{cases} \tag{I.2}$$

and the inertia tensor can thus be described as

$$\mathcal{I}_i = \mathrm{diag}(I_{\mathsf{cm},1,i}, I_{\mathsf{cm},2,i}, I_{\mathsf{cm},3,i}) \tag{I.3}$$

with the index notation cm has been excluded as it is given by implicit. The inertia tensor is diagonal, since no cross inertias are involved when assuming simple rod geometry and not complex asymmetric link shapes. In this case, the tensor is only a structure to handle moment of inertias around three different axes of rotation. Figure I.1 shows graphically how the inertias are expressed.

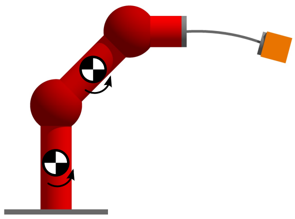


Figure I.1: *Definition of inertia for manipulator links*

An explicit description of a general mass matrix cannot be given, but whenever the inertia tensors have been derived for any link of the manipulator, and the joint space is expressed in terms of the generalized coordinate vector θ, the terms involving joint space accelerations $\ddot{\theta}$ can be collected within $\mathbf{M}(\theta)$. Next, the terms relating the motion between frames are considered.

I.2 Coriolis and centrifugal forces

Two effects caused by the motion of frames is the *Coriolis force* and the *centrifugal force*. The Coriolis force relates the motion of a point in one frame to another frame, and is generating a torque on the manipulator links if either of two neighboring links are moving [7]. All terms including a product between two joint velocities can thus be collected within matrix $\mathbf{C}(\theta, \dot{\theta})$. A general expression of the Coriolis force [14] transformed into a torque is given by

$$\tau_{\mathsf{cor},j} = -2m_j \, {}^{i}p_j \times (\bar{\omega}_i \times \bar{v}_{\mathsf{cm},j}) \tag{I.4}$$

with $v_{\mathsf{cm},j}$ = determinable using theory from chapter H. $\tau_{\mathsf{cor},j}$ is the torque affecting link j due to the Coriolis force. The second effect is the *centrifugal force*, which is not depending on interconnected links, but only on a single link. A general expression of the centrifugal force is given by [14]

$$\tau_{\mathsf{cen},j} = -m_j \, {}^{i}p_j \times \bar{\omega}_i \times (\bar{\omega}_i \times p_{\mathsf{cm},j}) \tag{I.5}$$

with $\tau_{\mathsf{cen},j}$ being the torque acting on the j-th link caused by the centrifugal effect on a moving object. Both there torques will be localizable within the system of dynamic equation of the manipulator in the term $\mathbf{C}(\theta, \dot{\theta})$ (see (3.5)), even though they are not used directly as shown here. A systematic approach is applied, which makes it possible to derive dynamic equations based on iterations from link to link. More on that in section I.6. Therefore, no further description of the forces will be addressed. Next, the gravitational pull on the links is described.

I.3 Manipulator gravity pull

Each link has a mass m_i, which will be pulled in negative z-direction of the inertial frame $\mathbb{F}_0$ from the center of gravity given by $\bar{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\intercal$, see figure I.2. The inertial frame is assumed by implicit, but must be expressed explicitly when applied. A term depending on the generalized coordinate vector θ must be included, which can be generalized in the following way to express the gravitational pull (as a torque to fit the form in (I.1)) of the i-th link center of gravity. No explicit term will be derived, because the model is automatically iterated.

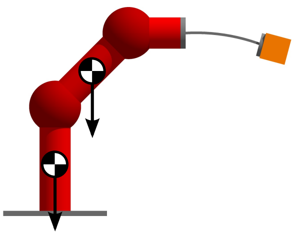


Figure I.2: *Gravity pull on manipulator links*

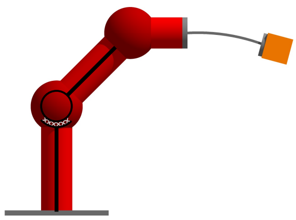


Figure I.3: *Manipulator link frictions*

Last part of the joint space model involves friction, which is emerging from the manipulator joints as depicted in figure I.3. Each manipulator joint will provide a torque contribution counteracting the positive motion of the joint. This is caused by friction in the joint bearings, which must be included by the model to provide approach real behavior from simulations. Since the bearings are rotating on the same axis as the joint actuators, the friction contribution from the bearings are included within the actuator model instead, which will be modeled in the sequel.

I.4 Actuator model

The actuator type installed on the manipulator is an electric DC-motor. Each joint is actuated by a torque generated by an actuator mounted directly on the axis of rotation. This implies the class of *direct drive manipulators*, which the REIS RV15 is part of [30]. Each drive is modeled independently and as a source of power to each individual link. The dynamics are then propagated through the links to the base. A model of the actuator dynamics is expressed in the following set of non-linear differential equations in the armature current $I_{a,i}$ and the joint angle $\theta_{a,i}$. The i-th actuator can be modeled as [41]

$$\dot{I}_{a,i} = -\alpha_i I_{a,i} - \beta_i \dot{\theta}_{a,i} + \kappa_i U_i \tag{I.6}$$

$$\ddot{\theta}_{a,i} = \gamma_i I_{a,i} - \delta_i \dot{\theta}_{a,i} + \tau_i(t, \theta_{a,i})/J_i \tag{I.7}$$

with U_i being the armature voltage [V] as well as

$$\alpha_i = R_i/L_i,\ \beta_i = K_{ei}/L_i,\ \gamma_i = K_{mi}/J_i,\ \delta_i = B_i/J_i,\ \kappa_i = 1/L_i \tag{I.8}$$

being motor specific constants, where R_i is the electrical armature resistance [Ω], L_i the armature inductance [H], J_i the armature inertia [kgm^2], B_i the friction [Nms] and $K_{ei} = K_{mi}$ the electrical and mechanical motor constant [Vs rad], respectively. No data is available on the actuator dynamics, and they will have to be estimated on the basis of sensor readouts. The nonlinear contribution is caused by the nonlinear dissipative function $\tau_i(t, \theta_{a,i})$. This is a collection of nonlinear friction terms and external load torques. Nonlinear friction forces are often described by discontinuous functions, but a continuously differentiable friction model has been suggested by [64] and is given as

$$\tau_i(t, \theta_i) = \tau_{L i+}(t, \theta_{i+}) - q_{1,i}[\tanh q_{2,i}\dot{\theta}_i - \tanh q_{3,i}\dot{\theta}_i] - q_{4,i} \tanh q_{5,i}\dot{\theta}_i \tag{I.9}$$

with the first term $\tau_{L i+}(t, \theta_{i+})$ added to include torque contributions from upper links and the payload. The index $i+$ indicates the set of indices $\{i+1, \ldots, n\}$, since the loading torque is a function of the upper links. The different parameters $q_{j,i}$ with $j = 1, \ldots, 5$ define the shape of the friction model. Three types of friction types are included in this model, whereas the viscous friction is linear in the angular velocity state $\dot{\theta}_i$ and therefore included as a direct term within (I.7). An example of the friction model shape is shown in figure I.4 with and without the viscous friction term $q_{6,i}\dot{\theta}_i$ added (in this case $B_i = q_{6,i}$), which provides a contribution proportional with the angular velocity. In order to apply the nonlinear friction model, the model must be fitted using system identification, because no data is available of the friction parameters for this system.

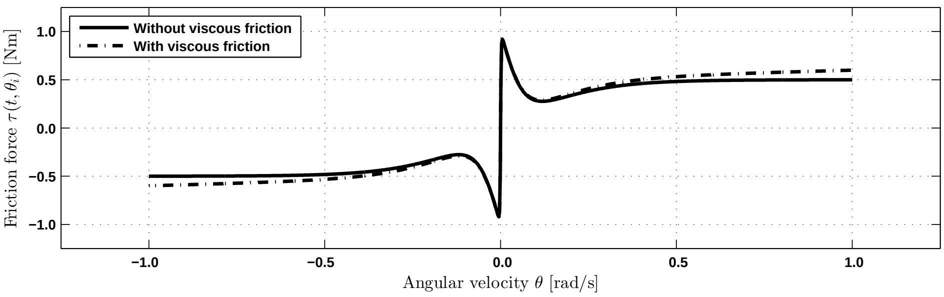


Figure I.4: *Friction with arbitrary parameters with and without viscous friction term*

I.5 Derivation of dynamic equations

In order to collect the terms of the dynamic equations into the matrices given in the previous sections, the equations must firstly be derived. The *Lagrangian* approach has been chosen here, which models the energy changes in the manipulator rather than the forces directly. A general formulation of the dynamics is given by the *Euler-Lagrange equation* [86]

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \mathcal{L}}{\partial \theta} = \xi \tag{I.10}$$

with $\mathcal{L} = \mathcal{T} - \mathcal{U}$ denoting the *Lagrangian*, which is the difference between the kinetic $\mathcal{T}$ and the potential energy $\mathcal{U}$. The vector $\xi = Q - \frac{\partial \mathcal{F}}{\partial \dot{\theta}}$ is expressing the generalized forces or external forces applied to the manipulator including actuator and friction forces. The friction forces are modeled using the Rayleigh dissipation function $\mathcal{F}$ [65]. A definition of the individual terms is explained in the sequel.

External forces Q applied from actuation are given directly from the model of the actuator. Equations (I.6) and (I.7) can thus be solved for the torque $Q_i = \alpha_i I_i$ of the i-th link. The energies $\mathcal{T}_m$ and $\mathcal{U}_m$, considering the manipulator without tool, are given by [30]

$$\mathcal{T}_m = \tfrac{1}{2}\sum_{i=1}^{n} m_i v_{\mathsf{cm},i}^{\mathsf{T}}\, v_{\mathsf{cm},i} + \omega_i^{\mathsf{T}}\, \mathcal{I}_i\, \omega_i \tag{I.11}$$

$$\mathcal{U}_m = -\tfrac{1}{2}\sum_{i=1}^{n} m_i\, \bar{g}_h^{\mathsf{T}}\, {}^0_i\mathbf{T}\, p_i + c \tag{I.12}$$

with $\bar{g}_h = -g\mathbf{e}_3$, $\bar{g}_h \in \mathbb{R}^4$ being the zero-padded gravity vector adapted to fit the homogeneous transformations. The reason why the fourth entry of $\bar{g}$ is not a one is because the gravity vector is oriented in the same direction independent of of the transformation applied to it. p_i expresses the center of gravity in reference with the i-th frame, and they are given, as introduced in chapter H, as

$$p_1 = -\ell_1\mathbf{e}_3, \quad p_2 = \ell_2\mathbf{e}_1 + \ell_3\mathbf{e}_3, \quad p_3 = \ell_4\mathbf{e}_2, \quad p_4 = \bar{0}, \quad p_5 = \bar{0}, \quad p_6 = \bar{0}$$

with ℓ_1 through ℓ_4 shown in figure I.5 (the lengths does not resemble the lengths described by the DH-parameters). The illustration is based on a pictorial interpretation of the manipulator structure, and not based on the actual structure.

The kinetic energy is based on the scalar case $E_{\mathsf{kin}} = \frac{1}{2}mv^2 + \frac{1}{2}J\omega^2$, but has been expanded to the matrix/vector case. Similar, for the scalar case of the potential energy $E_{\mathsf{pot}} = mgh + c$, with c being a constant offset used for special cases. The height is instead given by the position vector ${}^0_j\mathbf{T}\, p_j$ from base to the center of mass of the j-th link. A negative sign has been given the summation, since the potential energy must be positive for this configuration (zero-level or c assumed at the base).

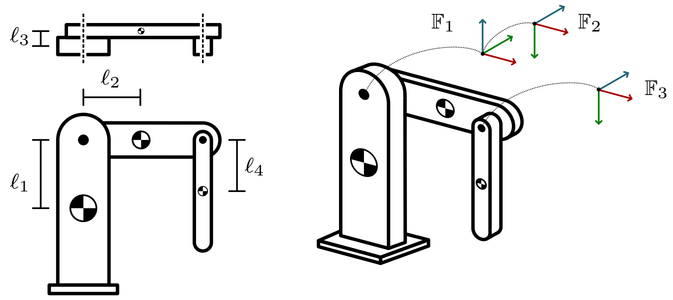


Figure I.5: *Illustration of parameters ℓ_1 through ℓ_4 used to define center of gravity vectors p_k*

A simplification has been made to further limit the complexity of the model, which is described in assumption 11.

> **Assumption 11:** *When calculating the kinetic energy contribution of the tool, the inertia is assumed located at the point of the end-effector, since the configuration used for testing features a tip mass much heavier than the beam mass, $m_l \gg m_b$. Thus, the inertia tensor is given as in* (I.15) *with a lumped mass. For more complex tools, $\mathcal{I}_t$ will include cross elements.* □

A definition of tool mass must be used in the sequel and thus $m_t \triangleq m_l + m_b$ is a combination of the load and the beam mass. Figure I.6 shows a graphical representation of the expressions under the summation sign in (I.11) and (I.12).

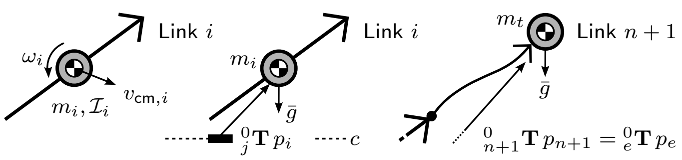


Figure I.6: *Expressions for kinetic energy, potential energy (link $[1, n]$) and potential energy of tool*

The position vector p_{n+1} describing the center of gravity of the tool (beam + load) is given from calculations on beam deflections in section 5.1. Because of assumption 11, the tool center of gravity is also the TCP given as the origo of the end-effector frame. To include the tool energy in the Lagrangian, two additional energy equations must be expressed, and due to the assumption of a lumped tool mass, the potential energy of the tool is determined from a point mass m_t

$$\mathcal{T}_t = \tfrac{1}{2} m_t v_{\mathsf{cm},t}^{\mathsf{T}}\, v_{\mathsf{cm},t} + \tfrac{1}{2}\int_0^{\ell} m(x) \left(\frac{\partial w(x,t)}{\partial t}\right)^2 \mathrm{d}x + \tfrac{1}{2}\omega_t^{\mathsf{T}}\, \mathcal{I}_t\, \omega_t \tag{I.13}$$

$$\mathcal{U}_t = -\tfrac{1}{2} m_t\, \bar{g}_h^{\mathsf{T}}\, {}^0_e\mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}} \tag{I.14}$$

where the last integral term is given from [94] describing the kinetic energy contribution from an oscillating beam given a mass distribution $m(x)$ and a deflection function $w(x)$. If the tool oscillation is small, the kinetic energy may be omitted. The inertia tensor of the tool $\mathcal{I}_t$ is given as

$$\mathcal{I}_t = \begin{bmatrix} m_t\ell & 0 & 0 \\ 0 & m_t\ell & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{I.15}$$

This will be used in a short-hand notation of the kinetic energy adopted from [72], which is able to express the combined energy equations given as

$$\mathcal{T} = \mathcal{T}_m + \mathcal{T}_{n+1} = \tfrac{1}{2}\sum_{i=1}^{n+1} \Omega_i^{\mathsf{T}} \mathcal{M}_i \Omega_i + \mathcal{T}_{n+1}(x) \tag{I.16}$$

$$\mathcal{U} = \mathcal{U}_m + \mathcal{U}_t = -\tfrac{1}{2}\sum_{i=1}^{n} m_i\, \bar{g}_h^{\mathsf{T}}\, {}^0_j\mathbf{T}\, p_i - \tfrac{1}{2} m_t\, {}^0\bar{g}_h^{\mathsf{T}}\, {}^0_e\mathbf{T} \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}} \tag{I.17}$$

with $\mathcal{M}_i$ representing the *generalized inertia matrix* and Ω_i a *generalized velocity vector*, both given as

$$\mathcal{M}_i = \begin{bmatrix} m_i I_3 & 0 \\ 0 & \mathcal{I}_i \end{bmatrix}, \quad \Omega_i = \begin{bmatrix} v_{\mathsf{cm},i} \\ \omega_i \end{bmatrix} \tag{I.18}$$

where the constant c from equation I.12 has been omitted. Position vectors are used to determine the linear velocity vectors $v_{\mathsf{cm},i}$, but the form of each individual vector will have to be derived on the basis of the DH-frames from appendix H. This provides the following vectors

$$\begin{aligned}
&v_{\mathsf{cm},1} = \bar{0}, \quad v_{\mathsf{cm},2} = \tfrac{1}{2}(\omega_2 \times {}^2\nu_3) = \tfrac{1}{2}(\dot{\theta}_1 + \dot{\theta}_2)\ell^{\mathsf{DH}}_{X2}\mathbf{e}_2 \\
&v_{\mathsf{cm},3} = v_3 + \tfrac{1}{2}(\omega_3 \times {}^3\nu_4) = v_3 - \tfrac{1}{2}(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\ell^{\mathsf{DH}}_{Z4}\mathbf{e}_1 \\
&v_{\mathsf{cm},4} \sim v_4, \quad v_{\mathsf{cm},5} \sim v_5, \quad v_{\mathsf{cm},6} \sim v_6
\end{aligned}$$

The last three vectors are expressed in terms of frame velocities, because the centers of gravity are assumed to be coinciding with the frame origo (formulated in assumption 12).

Assumption 12: *Velocity of the center of gravity is assumed similar to the linear velocity of the corresponding DH-frames for links 4,5 and 6 by $v_{\mathsf{cm},i} \sim v_i$ due to the assumed small moment of inertia and the coinciding between the frames. This also yields $\mathcal{I}_i = \mathbf{0}$ for the last three axes.* □

In order to express the complete set of dynamic equations, the initial conditions are stated as $v_0 = \bar{0}$, $\omega_0 = \bar{0}$. They are introduced in order to initiate the iterative process of finding the linear and angular velocity vectors using equations (H.10) and (H.9), respectively. Whenever the velocities have been found, the energies can be expressed using the previously derived equations (I.11) and (I.12).

Remark 7 (model derivation): *Because the dynamic manipulator model is too complex for performing the model derivation by hand, a symbolic tool must be applied to do the calculations. The MATLAB Symbolic Math Toolbox is available as the computational tool, but unable to differentiate time dependent functions. This calls for an iterative procedure for determining the dynamic equations instead. A method, which is not based on energy, must be applied, and will be described in the sequel section.* □

As described in remark 7, the energy based approach is not being applied to model the system. A regular Newtonian approach must be applied instead to derive the dynamic equations. This method is described in the sequel section.

I.6 Manipulator model by Newtonian approach

Two iterative equations have already been introduced in (H.9) and (H.10) to determine angular and linear velocity, respectively. Several other iterations will be needed to compute the dynamics of the manipulator. The recursive equations will not be derived but are given directly from equation 6.45 to equation 6.53 in [30], and they are given in the following way (accommodated for the notation within this report)

$$
\begin{aligned}
&1)\ \omega_j = {}^j_i\mathbf{R}\,\omega_i + \dot{\theta}_j \mathbf{e}_3 \\
&2)\ \alpha_j = {}^j_i\mathbf{R}\,\alpha_i + {}^j_i\mathbf{R}\,\omega_i \times \dot{\theta}_j \mathbf{e}_3 + \ddot{\theta}_j \mathbf{e}_3 \\
&3)\ a_j = {}^j_i\mathbf{R}\,\alpha_i \times {}^i\nu_j + {}^j_i\mathbf{R}\,\omega_i \times \omega_i \times {}^i\nu_j + {}^j_i\mathbf{R}\,a_i \\
&4)\ a_j^{\mathsf{cm}} = \alpha_j \times {}^i p_j + \omega_j \times \omega_j \times {}^i p_j + a_j \\
&5)\ F_j = m_j a_j^{\mathsf{cm}} \\
&6)\ N_j = \mathcal{I}_j \alpha_j + \omega_j \times \mathcal{I}_j \omega_j \\
&7)\ f_i = {}^i_j\mathbf{R}\,f_j + F_i \\
&8)\ n_i = N_i + {}^i_j\mathbf{R}\,n_j + p_i \times F_i + {}^i p_j \times {}^i_j\mathbf{R}\,f_j \\
&9)\ \tau_{\mathcal{N},i} = n_i^{\mathsf{T}} \mathbf{e}_3
\end{aligned}
$$

with the different expressions explained in the following list

1) *Angular velocity of consecutive links provided rotation around only* $\mathbf{e}_3$ *axes*
2) *Angular acceleration of link* j
3) *Linear acceleration of link* j
4) *Linear acceleration of center of gravity of link* j
5) *Force on link* j *center of gravity*
6) *Torque on link* j *center of gravity*
7) *Force propagation through manipulator*
8) *Torque propagation through manipulator*
9) *Resulting torque on joint* i *using the Newtonian approach*

Some of the variables are clashing with variables in the main report, but they will only be used here. Notice from the recursive equations, that the linear velocity of (H.10) is not applied directly, but used implicitly in the expression of the linear acceleration.

Convention 5: *Two mathematical conventions must be introduced to prevent misinterpretation of the math involved in this section. The expression* $\omega_i \times \omega_i \times p_j$ *involves two cross products, and they are evaluated from right to left, which can be explicitly denoted by* $\omega_i \times (\omega_i \times p_j)$*. Also, the expression* ${}^j_i\mathbf{R}\,\omega_i \times {}^i\nu_j$ *involves both a cross product and a change of basis by* ${}^j_i\mathbf{R}$*. The cross product is evaluated before the transformation, which can be stated explicitly as* ${}^j_i\mathbf{R}\,(\omega_i \times {}^i\nu_j)$*. Both notations are introduced to simplify the expressions.*$\diamond$

A set of initial conditions must be expressed in order to run the iterative process, and they are given as

$$\omega_0 = \bar{0} \Rightarrow \alpha_0 = \bar{0} \quad \text{and} \quad a_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^\intercal$$

The last condition will introduce gravity to the model [30], which is otherwise not given by any of the iterative equations. However, using the Lagrangian approach automatically includes the gravity term in the model, because the potential energy is explicitly expressed. Origo vectors ν_i, center of gravity vectors p_i, transformations $\mathbf{R}$, inertia tensors $\mathcal{I}_i$ and masses m_i are given in the previous chapters and sections. The resulting model is given in appendix N for the 3-DOF manipulator configuration, and the model is used in chapter 3.

Appendix J

Solutions to eigenvalue problem

The deflection $w(x,t)$ is solved in chapter 4, but in order to determine the mode shape functions $\phi_i(x)$, the first eigenvalue problem from (4.18) must be solved with the problem expressed as

$$\frac{\partial^4 \phi_i(x)}{\partial x^4} = \lambda_i^4 \phi_i \tag{J.1}$$

This problem must be solved for the eigenvalues λ_i. Applying a Laplace transform to the eigenvalue problem yields

$$\text{(J.1)} \overset{\mathcal{L}}{\rightarrow} s^4\phi_i(s) - s^3\phi_i(0) - s^2\phi_i^{(1)}(0) - s\phi_i^{(2)}(0) - \phi_i^{(3)}(0) = \lambda_i^4\phi_i(s) \tag{J.2}$$

Solving for the eigenfunction yields

$$\phi_i(s) = \frac{s^3\phi_i(0) + s^2\phi_i^{(1)}(0) + s\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{s^4 - \lambda_i^4} \tag{J.3}$$

The eigenvalue $\lambda_i \in \mathbb{R}$ is assumed real, since it relates to the eigenfrequency $\omega_i \in \mathbb{R}$ through the relation in (4.19). Since the solution to the problem has a periodic solution, the eigenvalues $\lambda_{i,j} \in \mathbb{C}$ where $\lambda_{i,1} = \lambda_i$ can be expressed together with the Laplace operator as

$$\lambda_{i,j} = |\lambda_{i,j}|\mathrm{e}^{\mathbf{j}\xi_{i,j}} \qquad s = |s|\mathrm{e}^{\mathbf{j}\xi_s} \tag{J.4}$$

making the poles of the fraction expressed through the relation $s^4 - \lambda_i = 0$

$$\mathrm{e}^{\mathbf{j}4\xi_s} = \mathrm{e}^{\mathbf{j}\xi_{i,j}} \Rightarrow \xi_s = \tfrac{1}{4}\xi_{i,j} + \tfrac{p\pi}{2},\ p \in \mathbb{Z} \tag{J.5}$$

Since the eigenvalues $\lambda_i \in \mathbb{R}$ the angle $\xi_{i,1} = \xi_i = 0$, the four eigenvalues can be located on the unit circle (assuming for now $|\lambda| = 1$) as shown in figure J.1. The plane on the left shows the case when $\lambda_i \in \mathbb{C}$, which makes it possible for $\xi_i \neq 0$, while the plane on the right assumes a real λ_i yielding $\xi_i \equiv 0$.

This indicates, that the eigenvalues (when assuming unit length) are laying on the perimeter of the unit circle separated by $\pi/2$ rad. The eigenvalues in question, when

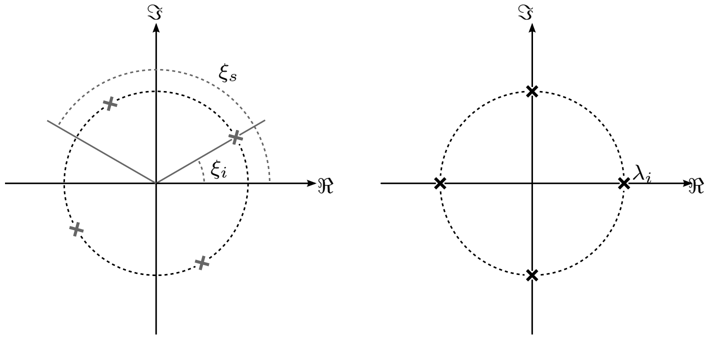


Figure J.1: *Location of eigenvalues in eigenvalue problem*

considering only a single i, are caused by periodicity of the solution to the eigenvalue problem. Due to the assumption of real eigenvalues λ_i, all four solutions can be expressed as

$$\lambda_{i,1} = \lambda_i \quad \lambda_{i,2} = \mathbf{j}\lambda_i \quad \lambda_{i,3} = -\lambda_i \quad \lambda_{i,4} = -\mathbf{j}\lambda_i \tag{J.6}$$

In order to express the eigenfunction in time domain, the fraction (J.3) must be expanded using the theory of *partial fraction expansion*, given as

$$\phi_i(s) = \sum_j \frac{P_j(s)}{s - \lambda_{i,j}}, \quad P_j(s) = \left.\frac{\phi_i(s)(s^4 - \lambda_i)}{\prod_{k \neq j}(s - \lambda_{i,k})}\right|_{s=\lambda_{i,j}} \tag{J.7}$$

with $j, k \in [1, 4]$ and $\phi_i(s)(s^4 - \lambda_i)$ expressing only the numerator polynomial of $\phi_i(s)$. Using this approach on equation (J.3), the eigenfunction becomes

$$\phi_i(s) = \frac{P_1(s)}{s - \lambda_{i,1}} + \frac{P_2(s)}{s - \lambda_{i,2}} + \frac{P_3(s)}{s - \lambda_{i,3}} + \frac{P_4(s)}{s - \lambda_{i,4}} \tag{J.8}$$

By substituting the solutions from (J.6), (J.8) yields

$$\phi_i(s) = \frac{P_1(s)}{s - \lambda_i} + \frac{P_2(s)}{s - \mathbf{j}\lambda_i} + \frac{P_3(s)}{s + \lambda_i} + \frac{P_4(s)}{s + \mathbf{j}\lambda_i} \tag{J.9}$$

where the rational fractions $P_j(s)$ becomes

$$P_1(s) = \left.\frac{s^2\phi_i^{(1)}(0) + s\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(s - \mathbf{j}\lambda_i)(s + \lambda_i)(s + \mathbf{j}\lambda_i)}\right|_{s=\lambda_i} = \frac{\lambda_i^2\phi_i^{(1)}(0) + \lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(\lambda_i - \mathbf{j}\lambda_i)(\lambda_i + \lambda_i)(\lambda_i + \mathbf{j}\lambda_i)}$$

$$P_2(s) = \left.\frac{s^2\phi_i^{(1)}(0) + s\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(s - \lambda_i)(s + \lambda_i)(s + \mathbf{j}\lambda_i)}\right|_{s=\mathbf{j}\lambda_i} = \frac{-\lambda_i^2\phi_i^{(1)}(0) + \mathbf{j}\lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(\mathbf{j}\lambda_i - \lambda_i)(\mathbf{j}\lambda_i + \lambda_i)(\mathbf{j}\lambda_i + \mathbf{j}\lambda_i)}$$

$$P_3(s) = \left.\frac{s^2\phi_i^{(1)}(0) + s\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(s - \lambda_i)(s - \mathbf{j}\lambda_i)(s + \mathbf{j}\lambda_i)}\right|_{s=-\lambda_i} = \frac{\lambda_i^2\phi_i^{(1)}(0) - \lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(-\lambda_i - \lambda_i)(-\lambda_i - \mathbf{j}\lambda_i)(-\lambda_i + \mathbf{j}\lambda_i)}$$

$$P_4(s) = \left.\frac{s^2\phi_i^{(1)}(0) + s\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(s - \lambda_i)(s - \mathbf{j}\lambda_i)(s + \lambda_i)}\right|_{s=-\mathbf{j}\lambda_i} = \frac{-\lambda^2\phi_i^{(1)}(0) - \mathbf{j}\lambda\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{(-\mathbf{j}\lambda - \lambda_i)(-\mathbf{j}\lambda - \mathbf{j}\lambda_i)(-\mathbf{j}\lambda + \lambda_i)}$$

when $\phi_i(0) = 0$ has been substituted. The expressions can be further simplified by expansion of the denominators (the polynomials are no longer functions of s but λ_i)

$$
\begin{aligned}
P_1(\lambda_i) &= \frac{\lambda_i^2\phi_i^{(1)}(0) + \lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{4\lambda_i^3} &&= \frac{Q_1 + Q_2 + Q_3}{4\lambda_i^3} \\
P_2(\lambda_i) &= \frac{-\lambda_i^2\phi_i^{(1)}(0) + \mathbf{j}\lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{-4\mathbf{j}\lambda_i^3} &&= \frac{-\mathbf{j}Q_1 - Q_2 + \mathbf{j}Q_3}{4\lambda_i^3} \\
P_3(\lambda_i) &= \frac{\lambda_i^2\phi_i^{(1)}(0) - \lambda_i\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{-4\lambda_i^3} &&= \frac{-Q_1 + Q_2 - Q_3}{4\lambda_i^3} \\
P_4(\lambda_i) &= \frac{-\lambda^2\phi_i^{(1)}(0) - \mathbf{j}\lambda\phi_i^{(2)}(0) + \phi_i^{(3)}(0)}{4\mathbf{j}\lambda_i^3} &&= \frac{\mathbf{j}Q_1 - Q_2 - \mathbf{j}Q_3}{4\lambda_i^3}
\end{aligned}
$$

with the definitions

$$
Q_1 \triangleq \lambda_i^2\phi_i^{(1)}(0), \quad Q_2 \triangleq \lambda_i\phi_i^{(2)}(0), \quad Q_3 \triangleq \phi_i^{(3)}(0)
$$

An inverse Laplace transformation brings the eigenfunctions $\phi_i(s)$ (from equation (J.8)) back into the x-domain

$$
\phi_i(x) = P_1(\lambda_i)\mathrm{e}^{\lambda_i x} + P_2(\lambda_i)\mathrm{e}^{\mathbf{j}\lambda_i x} + P_3(\lambda_i)\mathrm{e}^{-\lambda_i x} + P_4(\lambda_i)\mathrm{e}^{-\mathbf{j}\lambda_i x} \tag{J.10}
$$

which becomes

$$
\begin{aligned}
\phi_i(x) = {} & \frac{Q_1 + Q_2 + Q_3}{4\lambda_i^3}\mathrm{e}^{\lambda_i x} + \frac{-\mathbf{j}Q_1 - Q_2 + \mathbf{j}Q_3}{4\lambda_i^3}\mathrm{e}^{\mathbf{j}\lambda_i x} + \cdots \\
& \cdots + \frac{-Q_1 + Q_2 - Q_3}{4\lambda_i^3}\mathrm{e}^{-\lambda_i x} + \frac{\mathbf{j}Q_1 - Q_2 - \mathbf{j}Q_3}{4\lambda_i^3}\mathrm{e}^{-\mathbf{j}\lambda_i x} \qquad \text{(J.11)} \\
= {} & \frac{Q_1 + Q_3}{4\lambda_i^3}\left(\mathrm{e}^{\lambda_i x} - \mathrm{e}^{-\lambda_i x}\right) + \frac{Q_2}{4\lambda_i^3}\left(\mathrm{e}^{\lambda_i x} + \mathrm{e}^{-\lambda_i x}\right) + \cdots \\
& \cdots + \mathbf{j}\frac{-Q_1 + Q_3}{4\lambda_i^3}\left(\mathrm{e}^{\mathbf{j}\lambda_i x} - \mathrm{e}^{-\mathbf{j}\lambda_i x}\right) - \frac{Q_2}{4\lambda_i^3}\left(\mathrm{e}^{\mathbf{j}\lambda_i x} + \mathrm{e}^{-\mathbf{j}\lambda_i x}\right) \qquad \text{(J.12)}
\end{aligned}
$$

after substitution of the rational fractions from above. Using the complex trigonometric relations

$$
\begin{aligned}
\mathrm{e}^{\mathbf{j}\lambda_i x} &= \cos\lambda_i x + \mathbf{j}\sin\lambda_i x & \mathrm{e}^{-\mathbf{j}\lambda_i x} &= \cos\lambda_i x - \mathbf{j}\sin\lambda_i x \\
2\cosh\lambda_i x &= \mathrm{e}^{\lambda_i x} + \mathrm{e}^{-\lambda_i x} & 2\sinh\lambda_i x &= \mathrm{e}^{\lambda_i x} - \mathrm{e}^{-\lambda_i x}
\end{aligned}
$$

with the last two expressing hyperbolic cosine and hyperbolic sine, respectively, the eigenfunctions can be expressed in the following way

$$
\begin{aligned}
\phi_i(x) &= \frac{Q_1 + Q_3}{2\lambda_i^3}\sinh\lambda_i x + \frac{Q_2}{2\lambda_i^3}\cosh\lambda_i x - \frac{-Q_1 + Q_3}{2\lambda_i^3}\sin\lambda_i x - \frac{Q_2}{2\lambda_i^3}\cos\lambda_i x \\
&= -\frac{Q_2}{2\lambda_i^3}\cos\lambda_i x + \frac{Q_2}{2\lambda_i^3}\cosh\lambda_i x - \frac{-Q_1 + Q_3}{2\lambda_i^3}\sin\lambda_i x + \frac{Q_1 + Q_3}{2\lambda_i^3}\sinh\lambda_i x
\end{aligned}
$$

Before deriving the remaining constants, the mode is written as

$$\phi_i(x) = W_1 \cos\lambda_i x + W_2 \cosh\lambda_i x + W_3 \sin\lambda_i x + W_4 \sinh\lambda_i x \tag{J.13}$$

In order to satisfy the boundary conditions $\phi_i(0) = 0$ and $\phi_i^{(1)}(0) = 0$ then

$$\begin{aligned}
-\frac{Q_2}{2\lambda_i^3}\cos\lambda_i x + \frac{Q_2}{2\lambda_i^3}\cosh\lambda_i x - \frac{-Q_1+Q_3}{2\lambda_i^3}\sin\lambda_i x + \frac{Q_1+Q_3}{2\lambda_i^3}\sinh\lambda_i x = 0\\
-\frac{-Q_1+Q_3}{2\lambda_i^3}\cos\lambda_i x + \frac{Q_1+Q_3}{2\lambda_i^3}\cosh\lambda_i x + \frac{Q_2}{2\lambda_i^3}\sin\lambda_i x + \frac{Q_2}{2\lambda_i^3}\sinh\lambda_i x = 0
\end{aligned}$$

based on

$$\begin{aligned}
\phi_i^{(1)}(x) = \frac{Q_2}{2\lambda_i^3}\sin\lambda_i x + \frac{Q_2}{2\lambda_i^3}\sinh\lambda_i x - \cdots\\
\cdots - \frac{-Q_1+Q_3}{2\lambda_i^3}\cos\lambda_i x + \frac{Q_1+Q_3}{2\lambda_i^3}\cosh\lambda_i x
\end{aligned}$$

When evaluated in zero

$$-\frac{Q_2}{2\lambda_i^3} + \frac{Q_2}{2\lambda_i^3} = 0 \quad \text{and} \quad -\frac{-Q_1+Q_3}{2\lambda_i^3} + \frac{Q_1+Q_3}{2\lambda_i^3} = 0$$

which is only satisfied when $Q_1 = 0$. The constants W_1 through W_4 from the general definition in (J.13) are assigned the following relations

$$W_1 + W_2 = 0 \Rightarrow W_2 \triangleq \frac{Q_2}{2\lambda_i^3} \quad \text{and} \quad W_3 + W_4 = 0 \Rightarrow W_4 \triangleq \frac{Q_3}{2\lambda_i^3} \tag{J.14}$$

The remaining terms are therefore on the form

$$\phi_i(x) = W_2(\cosh\lambda_i x - \cos\lambda_i x) + W_4(\sinh\lambda_i x - \sin\lambda_i x) \tag{J.15}$$

In order to satisfy the remaining boundary conditions $\phi_i^{(2)}(\ell) = 0$ and $\phi_i^{(3)}(\ell) = 0$ then the second and third space derivatives must be used

$$\begin{aligned}
\phi_i^{(2)}(x) &= \lambda_i^2 W_2(\cosh\lambda_i x + \cos\lambda_i x) + \lambda_i^2 W_4(\sinh\lambda_i x + \sin\lambda_i x)\\
\phi_i^{(3)}(x) &= \lambda_i^3 W_4(\cosh\lambda_i x + \cos\lambda_i x) + \lambda_i^3 W_2(\sinh\lambda_i x - \sin\lambda_i x)
\end{aligned}$$

When evaluated in $x = \ell$ and equated zero

$$W_2(\cosh\lambda_i\ell + \cos\lambda_i\ell) + W_4(\sinh\lambda_i\ell + \sin\lambda_i\ell) = 0 \tag{J.16}$$

$$W_4(\cosh\lambda_i\ell + \cos\lambda_i\ell) + W_2(\sinh\lambda_i\ell - \sin\lambda_i\ell) = 0 \tag{J.17}$$

Two definitions are needed for the rest of the derivation

$$W_+ = W_2 + W_4 \quad \text{and} \quad W_\div = W_2 - W_4 \tag{J.18}$$

The sum and difference of the equations (J.16) and (J.17) must still equal zero, and thus

$$W_{+}\cos\lambda_i\ell + W_{+}\cosh\lambda_i\ell - W_{\div}\sin\lambda_i\ell + W_{+}\sinh\lambda_i\ell = 0 \tag{J.19}$$

$$W_{\div}\cos\lambda_i\ell + W_{\div}\cosh\lambda_i\ell + W_{+}\sin\lambda_i\ell - W_{\div}\sinh\lambda_i\ell = 0 \tag{J.20}$$

and they can be written as

$$\cos\lambda_i\ell + \mathrm{e}^{\lambda_i\ell} = \tfrac{W_{\div}}{W_{+}}\sin\lambda_i\ell \tag{J.21}$$

$$\cos\lambda_i\ell + \mathrm{e}^{-\lambda_i\ell} = -\tfrac{W_{+}}{W_{\div}}\sin\lambda_i\ell \tag{J.22}$$

Since both sides of the equations are equal individually, then the product between the two equations must also hold, thus

$$\begin{gathered}\cos^2\lambda_i\ell + 1 + \cos\lambda_i\ell\mathrm{e}^{\lambda_i\ell} + \cos\lambda_i\ell\mathrm{e}^{-\lambda_i\ell} = -\sin^2\lambda_i\ell \;\Updownarrow\\ 2 + 2\cos\lambda_i\ell\cosh\lambda_i\ell = 0 \;\Updownarrow\\ \boxed{\cos\lambda_i\ell\cosh\lambda_i\ell = -1}\end{gathered} \tag{J.23}$$

This is the frequency equation of a cantilever beam without tip mass, which is independent of the boundary conditions. The equation will be used in subsection 4.2.1.

Appendix K

Solving flexible tool dynamics using FEM

When the flexible tool was modeled in chapter 4, the mass was initially included within the partial differential equation itself. However, it was moved from the equation to a boundary condition, because it appeared as a Dirac function. This chapter will show the basic tools for numerically solving the model PDE describing the flexible tool behavior. The model could have been solved in this way while still keeping the mass distribution and the original boundary conditions.

Solving the problem analytically, however, requires splitting the partial differential equation into two, which must be solved simultaneously. In order to show, that the technique of moving the lumped mass approximation to the boundary conditions, the original problem can be solved numerically, and the two solutions must then coincide. Another reason for introducing a numerical method alongside the analytical derivation is to show, that in cases with more complicated flexible dynamics, the problem can only be solved numerically.

The applied method for solving the problem is the *finite element method* (FEM), which divides the tool configuration into a number of spaces with well-defined boundaries [52]. Basically, the method applies the technique of *modal decomposition* (introduced in chapter 4), which separates the dynamics into time and space [102]. Also, the differential equations describing the flexible tool dynamics are transformed into easily solvable linear systems. Only the basic theory will be explained to show the idea. Similar to the approach of solving the PDE in chapter 4, where the solution was assumed on the form given by (4.4), the FEM approach assumes a solution on the form [52]

$$w(x,t) = \sum_{i=1}^{M} \phi_i'(x)q_i'(t) + \phi_i''(x)q_i''(t) \tag{K.1}$$

with $\phi_i'(t)$, $\phi_i''(t)$ denoting time dynamics and $q_i'(t)$, $q_i''(t)$ denoting *interpolation function*. They approximate the mode shape functions that describe the shape of the flexible tool dynamics. Marks has been added as to avoid confusion with the original $\phi_i(x)$ and $q_i(t)$. Each element representing a part of the tool must be described by a total of four functions, but when multiple elements are combined, identical functions can be used to represent the behavior at the point of connection. The interpolation functions can be selected arbitrary, but according to [52], the function

$$\phi_i'(x) = \sum_{k=0}^{3} \alpha_k' x^k \quad \text{and} \quad \phi_i''(x) = \sum_{k=0}^{3} \alpha_k'' x^k \tag{K.2}$$

is most commonly used. Based on the interpolation function and the time dynamics, a model can be constructed for the k-th element on similar form as for the analytical case, namely equation 4.25 in [52]

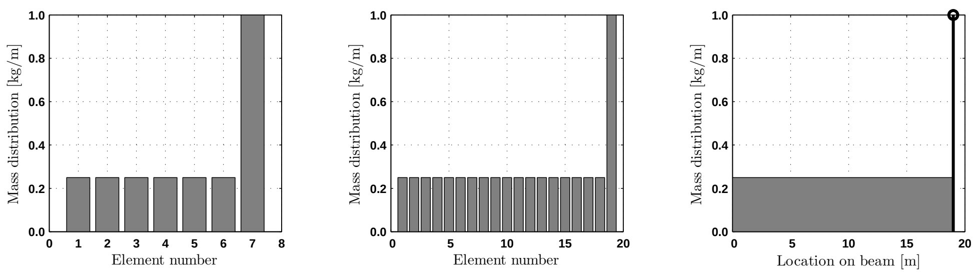
$$\frac{\mathrm{d}^2 q}{\mathrm{d}t^2} \int_0^{\ell_k} \rho_k \varphi \varphi^\intercal \,\mathrm{d}x + q \int_0^{\ell_k} E_k I_k \frac{\mathrm{d}^2 \varphi}{\mathrm{d}x^2} \frac{\mathrm{d}^2 \varphi^\intercal}{\mathrm{d}x^2} \,\mathrm{d}x = \int_0^{\ell_k} f(x,t) \varphi \,\mathrm{d}x \tag{K.3}$$

where φ is a vector containing all spatial functions $\phi_i'(x)$ and $\phi_i''(x)$. $f(x,t)$ is the distributive force, which in this case will be a force at $x = 0$ only. It can therefore be written as $f(t)$. However, it cannot be interchanged in (K.3), because it only affects the first element $k = 1$. In order to model the Dirac delta function term from the model in (4.3), a large number of elements must be selected. By selecting p elements and assigning them the following properties

$$\begin{aligned} E_k I_k &= EI \,\forall k < p \qquad && \text{and} \qquad && E_p I_p \to \infty \\ \ell_k &= \ell \,\forall k < p && \text{and} && \ell_p \to 0 \end{aligned}$$

the system can be solved close to the analytical version. The first $p - 1$ elements represent the flexible beam, whereas the last element given by $E_p I_p$ and ℓ_p represents the lumped tip mass. By making the flexural rigidity large to prevent any dynamics of the tip mass itself as well as decreasing the length of it to represent a lumped point mass, the model becomes closer to the analytical one. Similarly, the finite element model must be solved under identical boundary conditions as the analytical model.

The original boundary conditions can then be applied to the problem, since the tip mass has been included in the expression itself. It will therefore seem as if the boundary conditions of the last element includes the mass. Figure K.1 shows how the selection of more elements will improve the representation of mass along the beam compared with the continuous model. The mass distribution is normalized to one at the end of the beam.



(a) *FEM using 7 elements* **(b)** *FEM using 19 elements* **(c)** *Continuous model*

Figure K.1: *Mass distribution using 7 and 19 elements compared with continuous distribution*

This numerical method will not be analyzed in details, but for complex systems, the methods can be applied to avoid the time consuming math involved in solving partial differential equations with several conditions. Only the original boundary conditions of a cantilever beam are used, and the different forces can be added to the respective finite elements. If the flexible beam itself is not limited in motion by obstacles, the shape of the beam is not relevant as long as the end-effector position is known relative to the manipulator tool frame. This simplifies the equations as well as the complexity of the control problem in comparison with the analytical approach for solving the equations.

Appendix L

Generation of test signals

Proper test signals must be applied for system identification properties. This ensures the best possible responses from the hardware, and thus also the best parameter estimations close to the correct ones. The composition of useful test signals is considered in [107, 104], stating that the signals must include a slow and a random component. A slow component ensures actual motion of the manipulator, whereas the random signal includes the entire frequency range of the system. A so-called PRBS (*pseudo random binary sequence*) is used as the random signal.

Additionally, an offset can be applied in case of a relatively large Coulomb friction force generated by the actuators. The model to test is on the form (from (3.10)) $U_0 u \triangleq \dot{\boldsymbol{\tau}} = \mathbf{NF}\ddot{\theta}$. U_0 is a scale factor from the input voltage to the servo controllers to the applied torque derivative, which is also unknown. Using the expression $\dot{\boldsymbol{\tau}}$ instead of $\boldsymbol{\tau}$ is because a state space representation is needed, see section 3.4. The control signal u is being composed by a *multisine signal* with random phases [104] and a PRBS with sufficient amplitude [107]

$$u_i = \frac{\mathrm{d}}{\mathrm{d}t}\left\{\sum_{k=1}^{n_k} \alpha'_{i,k}\sin(\omega'_{i,k}t + \phi'_{i,k}) + \beta'_i\xi_{\mathrm{PRBS}}\right\} \tag{L.1}$$

where α', ω'_k, ϕ'_k, $\beta' \in \mathbb{R}$ denotes sine amplitude, frequency, phase and random signal amplitude, respectively. The actual sizes of the factors must be empirically determined, as different test signals also results in different models. ξ_{PRBS} is the PRBS signal of unit amplitude. The PRBS is beneficial over the use of a white noise process, because the normed amplitude is constant. It is therefore possible to excite nonlinear effects in a specific frequency range with a controlled amplitude. This avoids unwanted peaks in the test signals.

To include only manipulator dynamics, the identification process is carried out in open-loop, which excludes the controller from the parameter estimation. This is possible, because the system is naturally stable, and thus do not require any stabilizing controller in the loop to yield useful measurement data.

The shape of the test signals have been established, and the actual function cannot be given without conducting a number of experiments on the hardware. Different sets of parameters are used to estimate the parameters as described in appendix B. Figure L.1 shows an example of a test signal with arbitrary parameters and figure L.2 shows the integrated version. The different frequency components are clearly visible, and the effect of the added PRBS produces an oscillating overlay.

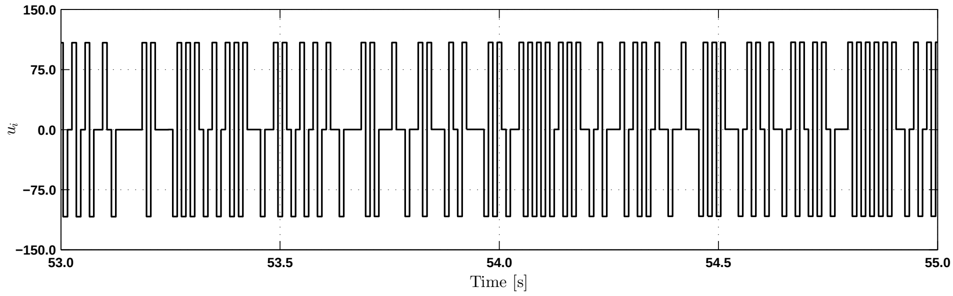


Figure L.1: *Arbitrary test signal based on* (L.1)

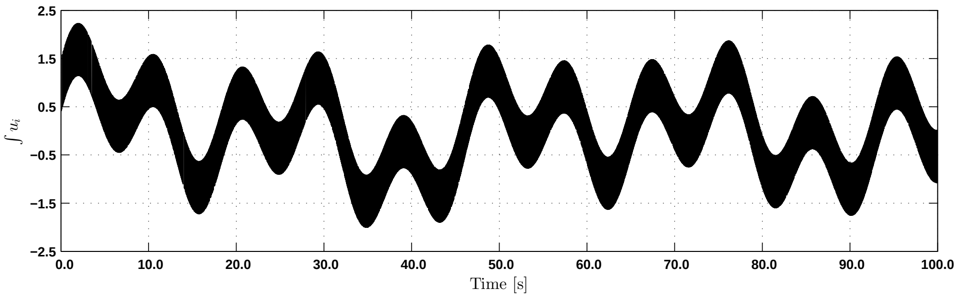
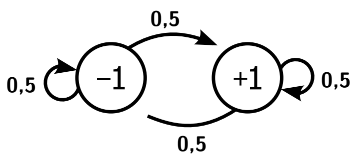


Figure L.2: *Arbitrary test signal based on integrated* (L.1)

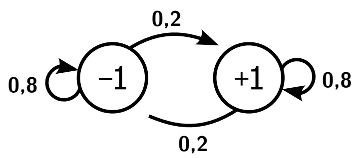
Due to the relation $u \triangleq \dot{\tau}$, the signal must be differentiated before applied to the servo amplifiers. Two different test signals will be constructed: one to estimate the manipulator parameters and one to estimate the flexible tool parameters. Table L.1 describes how the different signals are designed for each purpose.
The signals contain a number of primary frequencies in the low-frequency range and random components in the entire frequency span. By increasing the magnitude β_i' of the PRBS, the nonlinear effects of the unknown system can be excited more. Using several sets of test signals provides the best possible parameter estimation result when all signals are combined. Different approaches can be used to generate the PRBS, which is shown in figure L.3 as three Markov chains with different characteristics.

Test signals using PRBS	
Manipulator parameters	A signal containing 5 primary frequencies superimposed with a PRBS without band limitations will be used to estimate the manipulator parameters. The primary frequencies will be selected by random, and will be in the range of $0{,}1 \leq f \leq 1{,}0$ Hz to move the links at a slow speed in order to estimate the friction terms.
Flexible tool parameters	Unlike the test signal for estimation of the manipulator parameters, the flexible tool parameters can be determined using only the signal u_3 (controlling only link 3). A single primary frequency in the range of $0{,}1 \leq f \leq 1{,}0$ Hz will be applied to ensure some motion of the third link. This counteracts stiction effects. The PRBS must be band limited around the first resonance (calculated version) in a band defined by $0{,}9\omega_1 \leq \omega \leq 1{,}1\omega_1$.

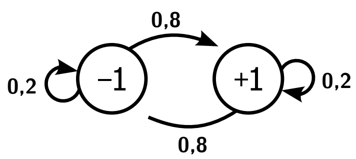
Table L.1: *Two different types of test signals for parameter estimation (using PRBS)*



(a) *Uncorrelated samples*



(b) *Lowpass*



(c) *Highpass*

Figure L.3: *Markov chains used for PRBS generation*

Even though the Markov sources are capable of generating random sequences with either low or high frequency content, the control of the individual frequencies can be difficult. For the purpose of estimating parameters for the manipulator itself, the Markov chain from figure L.3a is used. All nonlinear effects must be investigated, and thus the test signal must contain all frequencies (limited by sampling frequency). However, when investigating the dynamics of the flexible tool, only the primary mode function parameters must be estimated. Since calculations have predicted the location of the eigenfrequency as well as experiments have estimated the modal damping, a bandpass filtered test signal can be used instead. This concentrates the estimation effort within assumed parameter bounds.

The method for constructing the bandlimited signal begings with the generation of a random white noise sequence $y_{\mathcal{N}}$. Afterwards, a bandpass filter with a passband in the range $0,9\omega_1 \leq \omega \leq 1,1\omega_1$ is applied to produce $y'_{\mathcal{N}}$, and the resulting PRBS is given as $\xi_{\text{PRBS}} = \operatorname{sgn} y'_{\mathcal{N}}$ [61]. The test signal from L.1 will be applied to the system as illustrated in figure L.4.

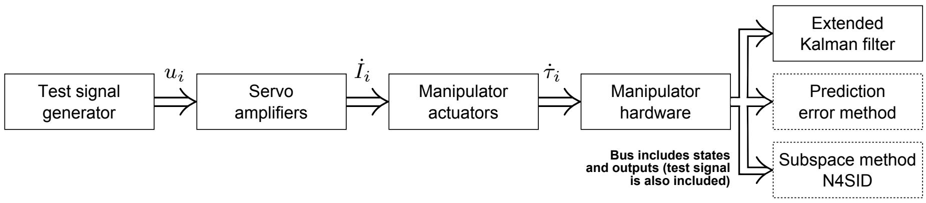


Figure L.4: *Application of test signals for parameter estimation and model verification*

A detailed look inside the test signal generator block is shown in figure L.5.

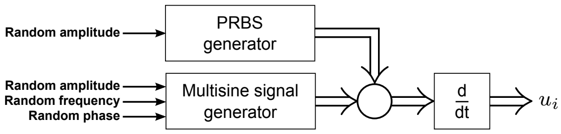


Figure L.5: *Test signal generator block diagram*

The differentiation is carried out to generate the torque derivative, which is the model input. A relation between the different parameters is given as $\tau_i \propto I_i$. Similarly, the relation can be differentiated, yielding $\dot{\tau}_i \propto \dot{I}_i \propto \triangleq u_i$ with u_i being the applied voltage signal to the servo amplifiers in this case.

> **Remark 8** (PRBS derivative)**:** *From the above derivation of test signals it is given by implicit, that it is possible to differentiate a pseudo-random binary sequence, which it is not. Only due to the discrete implementation of the sequence, the derivative can be approximated. However, a continuously differentiable analog must considered in order to avoid this approximation. As described in [61, 104], the PRBS method is frequently used for identifying manipulator systems.* □

Because of remark 8, table L.1 can be changed to table L.2

Description (using continuous signals)	
Manipulator parameters Flexible tool parameters	A signal constructed from 2 low-frequency harmonics and 4 harmonics with frequencies close to the expected resonance frequency of the flexible tool. In this case then $n_k = 10$. The first 2 frequencies are selected by random from within the range of $0{,}1 \leq f \leq 1{,}0$ Hz to move the links at a slow speed in order to estimate the friction terms. The remaining frequencies are gathered from the range $0{,}8f \leq f \leq 1{,}2f$ Hz to excite the tool dynamics for the first eigenfrequency f.

Table L.2: *Two different types of test signals for parameter estimation (continuous signals)*

Similar to the method using PRBS, an explicit expression of the signal can be given, which is on the form

$$u_i = \frac{\mathrm{d}}{\mathrm{d}t}\left\{\sum_{k=1}^{n_k} \alpha'_{i,k}\sin(\omega'_{i,k}t)\right\} = \sum_{k=1}^{n_k}\alpha'_{i,k}\omega'_{i,k}\cos(\omega'_{i,k}t) \tag{L.2}$$

Unlike (L.1), there is no phase involved here, because the differentiation removes the phase component of the harmonic function. Furthermore, the signal has been constructed from a finite number of harmonics, which have been selected on the basis of the model and knowledge about the system. From (L.2) a signal (one for each of the three axes) can be generated as shown in figure L.6 with a total of 6 harmonics.

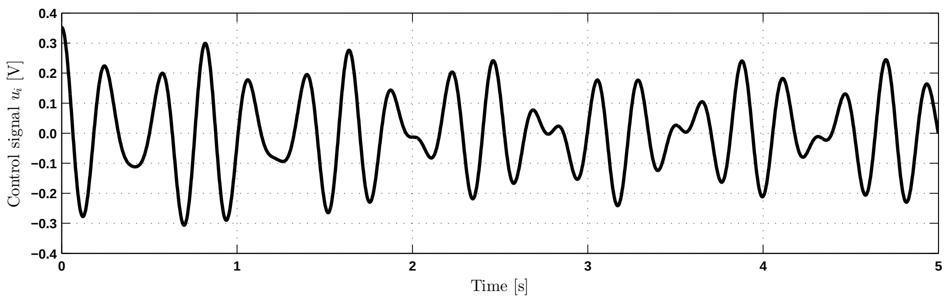


Figure L.6: *Arbitrary test signal based on* (L.2)

The arbitrary constants $\alpha'_{i,k}$ have been selected to follow the guidelines in table L.2 as well as generating resulting amplitudes no higher than 1 V. This is done to prevent huge excitation signals around the resonance, which can cause damage to the tool. The ranges of the signals are given as

$$\frac{1}{5n_k\omega_{j+}} \leq \alpha'_{i,k} \leq \frac{1}{n_k\omega_{j+}} \quad \text{and} \quad \omega_{j-} \leq \omega'_{i,k} \leq \omega_{j+} \tag{L.3}$$

in order to limit both frequency range and amplitude. These ranges also respects the guidelines from table L.2. The index j is used to divide the two parts of the signal - one part ($j = 1$) for low-frequency excitation and another part ($j = 2$) for near-resonance excitation. ω_- and ω_+ denotes minimum and maximum harmonic frequency, respectively. The 1 V limit can be seen by the relation

$$\left|\sum_{i=1}^{n_k}\alpha'_{i,k}\omega'_{i,k}\cos(\omega'_{i,k}t)\right| \leq \left|\sum_{i=1}^{n_k}\frac{\omega'_{i,k}}{n_k\omega_{j+}}\right| = \sum_{i=1}^{n_k}\frac{1}{n_k} = 1 \tag{L.4}$$

Because the flexible tool dynamics follows a second order system by assumption, only two points are needed to determine the shape of the frequency response. This will in turn provide both the damping ratio and the resonance frequency. Figure L.7 shows how four points on either the real bode plot or the approximated bode plot can uniquely determine the dynamics. Four points are used instead of two to secure the correct response is estimated [91].

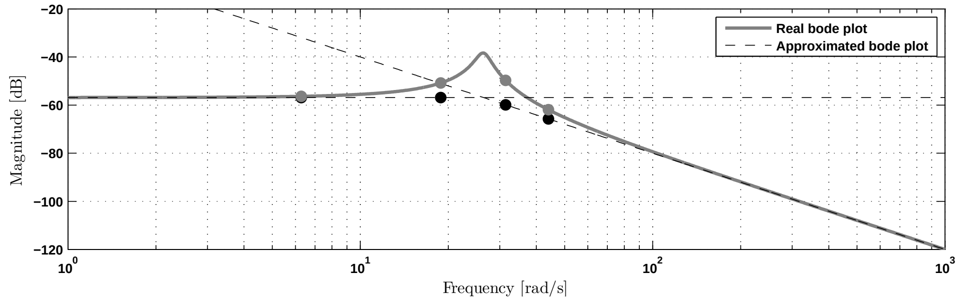


Figure L.7: *Bode plot of flexible tool dynamics and corresponding approximation*

States, system outputs and input signals are stored during the experiments and used in the different blocks in figure L.4. The first block is the extended Kalman filter, which must be verified to perform as good as the second block. Third block is used to verify the model structure. Only the EKF is an actual part of the system, whereas the two remaining blocks are used only in the design phase.

In chapter 10 the parameter estimation results will be presented with a comparison between the online EKF and the offline PEM. Data sample sets have been derived using the method described in this section, and more information regarding this is given in appendix B. Only with agreeing results, the EKF method can be applied for the online parameter estimation process on the controller hardware.

Appendix M

Static beam models

The necessary theory for modeling time dependent flexible behavior of the tool is based on beam bending theory for simple beams. A simple beam is characterized by the fact, that it is not composed from several parts. It is only a single beam made from a homogeneous material. The simple beam model can afterwards be affected by a generalized force, which will make it bend in different ways. Steady-state solutions are presented in this chapter for three different cases of beam configurations. Time dynamics will be included in chapter 4. The cases are listed below

- Beam bending under self-weight
- Beam bending under tip load
- Beam bending under generalized force

The first case involves a beam configuration that is only affected by gravity, whereas the second configuration adds the mass of a payload on the free end of the beam. Lastly, a generalized force is applied to the entire length of the beam. Figure M.1 shows graphically the three different configurations. All configurations are based on a cantilever beam setup with a fixed and a free end.

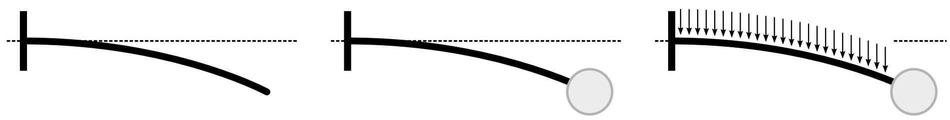


Figure M.1: *Flexible beam configurations (self-weight, tip load and generalized force)*

A generalized force can consist of several individual masses affecting the beam or a homogeneous distribution e.g. if the beam is affected by wind (this would be the case if aerodynamics are included in the model). The first case to consider is the bending of a beam under self-weight.

M.1 Beam bending under self-weight

The general Timoshenko beam equation from equation (4.1) is reduced to equation (4.2), which is repeated here [18, 74]

$$EI\frac{\partial^4 w(x,t)}{\partial x^4} + a\rho\frac{\partial^2 w(x,t)}{\partial t^2} = f(x)$$

and because a steady state solution is wanted, the acceleration term $\partial^2 w(x,t)/\partial t^2$ (time dependent) can be neglected. This yields the equation

$$EI\frac{\mathrm{d}^4 w(x)}{\mathrm{d}x^4} = f(x) \tag{M.1}$$

where time has been removed for this simple case. A set of initial boundary conditions must be included to define the specific cantilever setup, thus [108]

$$\begin{aligned} w(0) = \frac{\mathrm{d}w(0)}{\mathrm{d}x} = 0 \qquad & \textit{No deflection or non-tangential derivative at the point of beam attachment } x = 0 \\ \frac{\mathrm{d}^2 w(\ell)}{\mathrm{d}x^2} = \frac{\mathrm{d}^3 w(\ell)}{\mathrm{d}x^3} = 0 \qquad & \textit{No bending moment or shear force at the end-point of the beam } x = \ell \end{aligned} \tag{M.2}$$

A generalized candidate solution to the homogeneous version of (M.1) must be expressed in order to solve it. Also, it must be in terms of x and one order lower than the ODE [108]. This causes the fourth derivative to be zero, and the term f/EI can be added to the solution. A candidate can be a third order polynomial on the general form $w_h(x) = \mathcal{P}_3$ with coefficients $\psi_i \in \mathbb{R}$. A particular solution $w_p(x)$ is derived from integrating the applied force function $f(x)$ (the non-homogeneous part of (M.1)) four times, and the complete solution can thus be expressed as

$$\begin{aligned} w(x) &= w_h(x) + w_p(x) = \mathcal{P}_3 + \frac{f}{EI}\int \cdots \int_4 \mathrm{d}x \\ &= \psi_4 x^4 + \psi_3 x^3 + \psi_2 x^2 + \psi_1 x + \psi_0 \end{aligned} \tag{M.3}$$

given $\psi_4 = f_b/4!EI$ if the beam is affected by gravity pull alone expressed by the generalized force f_b. This explicit solution is only valid if f_b is not a function of x, which is the case when defining gravity as a location independent force $f_b = -m_b g/\ell$ with m_b [kg] denoting the mass of the beam and g [m/s^2] the gravitational constant. Using the initial conditions, the first two ψ_i coefficients can be found to be $\psi_0 = \psi_1 = 0$, and the others can be determined from

$$\begin{aligned} &\frac{\mathrm{d}^2 w(\ell)}{\mathrm{d}x^2} = 12\psi_4 x^2 + 6\psi_3 x + 2\psi_2 = 0 \\ &\frac{\mathrm{d}^3 w(\ell)}{\mathrm{d}x^3} = 24\psi_4 x + 6\psi_3 = 0 \\ &\Rightarrow\ \psi_2 = \tfrac{f_b \ell^2}{4EI},\ \psi_3 = -\tfrac{f_b \ell}{6EI} \end{aligned}$$

which is characteristic for a cantilever beam. Selecting the generalized force $f(x) = f_b(x) = m_b g/\ell$ allows for determination of tip deflection $w_b(\ell)$ under beam self-weight. The force term $f_b(x)$ has the unit [N/m] and describes the force per unit length of beam. Forces expressed within this chapter are respecting definition 1 regarding direction of forces. Using the solution from (M.3), it can be expressed as

$$w_b(\ell) = \frac{f_b}{24EI}x^4 - \frac{f_b\ell}{6EI}x^3 + \frac{f_b\ell^2}{4EI}x^2\Big|_{x=\ell} = -\frac{m_b g\ell^3}{8EI} \quad [\mathrm{m}] \tag{M.4}$$

This equation can only be used to determine the deflection of the beam under self-weight. The sequel section determines the beam deflection under tip load alone.

M.2 Beam bending under tip load

By subjecting a cantilever beam to an tip load, the equation to solve is [20]

$$\frac{\mathrm{d}^4 w(x)}{\mathrm{d}x^4} = \frac{f_l}{EI}\delta(x-\ell) \tag{M.5}$$

with $f_l = -m_l g$ being the static scalar force applied only to the tip of the beam and δ the *Dirac delta* function (continuous impulse) [1/m]. Solving this kind of problem introduces a Laplace transformation of the equation, that allows the impulse function to be treated as a continuous function. Four Laplace transform pairs are needed to convert between time and Laplace domain now and in the following model derivations [47, 51, 35] (shown here in the time variable t)

$$\begin{aligned}
&\eta H(t-T) \overset{\mathcal{L}}{\to} \frac{\eta \mathrm{e}^{-sT}}{s}, && \eta\delta(t-T) \overset{\mathcal{L}}{\to} \eta\mathrm{e}^{-sT} \\
&\frac{1}{s^k} \overset{\mathcal{L}^{-1}}{\to} \frac{t^{k-1}}{(k-1)!}, && \frac{1}{s^k}\mathrm{e}^{-sT} \overset{\mathcal{L}^{-1}}{\to} \frac{(t-T)^{k-1}}{(k-1)!}H(t-T) \\
&\frac{\mathrm{d}^k w}{\mathrm{d}t^k} \overset{\mathcal{L}}{\to} s^k W(s) - s^{k-1}w(0) - \cdots - sw^{(k-2)}(0) - w^{(k-1)}(0)
\end{aligned}$$

with $H(t)$ denoting the *Heaviside function.* The Laplace transformation of (M.5) yields the following expression in the Laplace variable s

$$W(s) = \frac{f_l}{EIs^4}\mathrm{e}^{-s\ell} + \sum_{k=0}^{3}\frac{w^{(k)}(0)}{s^{k+1}}$$

with the summation representing the Laplace transformation of a homogeneous version of (M.1) and the last term representing $\mathcal{L}\{\delta(x-\ell)\}$. A short notation $w^{(k)}(0)$ is applied to express the boundary condition of the k-th derivative. Transforming back into time-domain yields (after inserting the known boundary conditions $w(0) = 0$ and $w^{(1)}(0)$)

$$w_l(x) = \frac{w^{(2)}(0)}{2}x^2 + \frac{w^{(3)}(0)}{6}x^3 + \frac{f_l}{6EI}(x-\ell)^3 H(x-\ell) \tag{M.6}$$

The unknown boundary conditions $w^{(2)}(0)$ and $w^{(3)}(0)$ can be found from the known boundary conditions $w^{(2)}(\ell) = 0$ and $w^{(3)}(\ell) = 0$ by differentiating (M.6) two and three times followed by setting them equal to zero (given by initial boundaries) [20]. This yields the following tip deflection for a tip mass m_l

$$w_l(\ell) = \frac{f_l \ell}{2EI}x^2 - \frac{f_l}{6EI}x^3 + \frac{f_l}{6EI}(x-\ell)^3 H(x-\ell)\Big|_{x=\ell} = -\frac{m_l g \ell^3}{3EI} \quad [\mathrm{m}] \tag{M.7}$$

The last thing to do is to combine the distributed force and the point force. This enables calculation of the deflection of any point of the beam, which is covered in the sequel section.

M.3 Beam deflection under generalized force

Multiple forces can affect the deflection of the beam in different ways. To combine both tool mass and gravity contributions for this specific configuration, $f(x)$ must be concatenated in the following way

$$f(x) = f_c(x) = \tfrac{m_b g}{\ell} + m_t g \delta(x-\ell) \tag{M.8}$$

and the differential equation from (M.1) can once again be applied. Using similar algebra as shown above, the deflection at the tip can be expressed

$$\begin{aligned} W(s) &= \frac{m_l g}{EIs^4}\mathrm{e}^{-s\ell} + \frac{m_b g}{EI\ell s^5} + \sum_{k=0}^{3} \frac{w^{(k)}(0)}{s^{k+1}} \quad \downarrow \mathcal{L}^{-1} \\ w_c(x) &= \frac{w^{(2)}(0)}{2}x^2 + \frac{w^{(3)}(0)}{6}x^3 + \frac{m_l g}{6EI}(x-\ell)^3 H(x-\ell) + \frac{m_b g}{24EI\ell}x^4 \end{aligned}$$

with the new boundary conditions given as

$$w^{(2)}(0) = \frac{m_b g \ell}{2EI} + \frac{m_l g \ell}{EI}, \quad w^{(3)}(0) = -\frac{g}{EI}(m_l + m_b)$$

The deflection at the tip can thus be expressed as

$$\begin{aligned} w_c(x) &= \left(\frac{m_b g \ell}{4EI} + \frac{m_l g \ell}{2EI}\right)x^2 - \left(\frac{g}{6EI}(m_l+m_b)\right)x^3 + \cdots \\ \cdots &+ \frac{m_l g}{6EI}(x-\ell)^3 H(x-\ell) + \frac{m_b g}{24EI\ell}x^4\Big|_{x=\ell} = \frac{g\ell^3(8m_l + 3m_b)}{24EI} \end{aligned} \quad [\mathrm{m}] \tag{M.9}$$

where the unit step function $H(x-\ell) = 0, \forall x \in [0, \ell[$. The final equation (M.9) shows to be a summation of the two previous models from (M.3) and (M.6). However, this is not a general statement when using multiple point forces. Different situations will require the force distribution to be given explicitly and (M.1) must be solved for each case.

Three different static models have been expressed in the above. Basic methods for solving partial differential equations describing the bending of beams have been introduced, which may be used to solve more complex problems in chapter 4. However, the third order polynomial $\mathcal{P}_3$ is an approximation of the deflection along the beam length. Different shapes can be applied, which will still provide identical solutions at the boundaries but differ elsewhere along the beam length. The purpose of the model determines what approximation to use. Table M.1 summarizes the results from this chapter.

Beam subjected to self-weight
$w_b(x) = \frac{m_b g}{24EI\ell}x^4 - \frac{m_b g}{6EI}x^3 + \frac{m_b g\ell}{4EI}x^2$
Beam subjected to tip load
$w_l(x) = -\frac{m_l g}{6EI}x^3 + \frac{m_l g\ell}{2EI}x^2$
Beam subjected to both load types
$w_c(x) = \frac{m_b g}{24EI\ell}x^4 - \left(\frac{g}{6EI}(m_l + m_b)\right)x^3 + \left(\frac{m_b g\ell}{4EI} + \frac{m_l g\ell}{2EI}\right)x^2$

Table M.1: *Beam deflection when subjected to different load types*

Appendix N

Explicit model expressions

This appendix includes the explicit expressions of the following items

- Complete transformation ${}^0_6\mathbf{T}$ (full model)
- Complete transformation ${}^b_\tau\mathbf{T}$ (reduced model)
- Manipulator dynamics model (reduced model)

N.1 Complete transformation after simplification

Due to remark 1 on page 18, the swivel axes are made inactive, and the resulting transformation from base to tool frame is therefore on the form

$$ {}^b_\tau\mathbf{T} = \begin{bmatrix} t_{(1,1)} & t_{(1,2)} & 0 & t_{(1,4)} \\ 0 & 0 & 0 & 0 \\ t_{(3,1)} & t_{(3,2)} & 0 & t_{(3,4)} \\ 0 & 0 & 0 & 0 \end{bmatrix} $$

with the individual elements given as

$$ \begin{aligned} t_{(1,1)} &= +\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 - \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2 \\ t_{(1,2)} &= +\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 \\ t_{(1,4)} &= +\,\ell_1\,\mathrm{c}_1 + \ell_2\,\mathrm{c}_1\,\mathrm{c}_2 - \ell_2\,\mathrm{s}_1\,\mathrm{s}_2 + \ell_3\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 \\ &\quad - \ell_3\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \ell_3\,\mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 - \ell_3\,\mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2 \end{aligned} $$

$$ \begin{aligned} t_{(3,1)} &= +\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 \\ t_{(3,2)} &= +\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 + \mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2 \\ t_{(3,4)} &= +\,\ell_0 - \ell_1\,\mathrm{s}_1 - \ell_2\,\mathrm{c}_1\,\mathrm{s}_2 - \ell_2\,\mathrm{c}_2\,\mathrm{s}_1 \\ &\quad + \ell_3\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \ell_3\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 - \ell_3\,\mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2 - \ell_3\,\mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 \end{aligned} $$

N.2 Complete kinematic transformation

Defining the structure of the complete transformation matrix ${}^0_6\mathbf{T}$ as

$$
{}^0_6\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{N.1}
$$

the elements are given by the following expressions

$$
\begin{aligned}
t_{11} &= -\,\mathrm{s}_6\,\mathrm{c}_4\,\mathrm{s}_1 - \mathrm{s}_4\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{c}_6\,\mathrm{c}_5\,\mathrm{s}_1\,\mathrm{s}_4 + \mathrm{c}_4\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_5\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2\\
t_{12} &= \mathrm{s}_6\,\mathrm{c}_5\,\mathrm{s}_1\,\mathrm{s}_4 + \mathrm{c}_4\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_5\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2 - \mathrm{c}_6\,\mathrm{c}_4\,\mathrm{s}_1 - \mathrm{s}_4\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3\\
t_{13} &= -\,\mathrm{s}_5\,\mathrm{s}_1\,\mathrm{s}_4 + \mathrm{c}_4\,\mathrm{c}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{c}_5\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2\\
t_{14} &= \ell^{\mathsf{DH}}_{Z3}\,\mathrm{s}_1 - \ell^{\mathsf{DH}}_{Z4}\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_1\,\mathrm{c}_3\,\mathrm{s}_2 + \ell^{\mathsf{DH}}_{X2}\,\mathrm{c}_1\,\mathrm{c}_2
\end{aligned}
$$

$$
\begin{aligned}
t_{21} &= \mathrm{s}_6\,\mathrm{c}_1\,\mathrm{c}_4 + \mathrm{s}_4\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 + \mathrm{c}_6\,\mathrm{c}_5\,\mathrm{c}_1\,\mathrm{s}_4 - \mathrm{c}_4\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 + \mathrm{s}_5\,\mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2\\
t_{22} &= \mathrm{c}_6\,\mathrm{c}_1\,\mathrm{c}_4 + \mathrm{s}_4\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 - \mathrm{s}_6\,\mathrm{c}_5\,\mathrm{c}_1\,\mathrm{s}_4 - \mathrm{c}_4\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 + \mathrm{s}_5\,\mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2\\
t_{23} &= \mathrm{s}_5\,\mathrm{c}_1\,\mathrm{s}_4 - \mathrm{c}_4\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_1 - \mathrm{c}_5\,\mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2\\
t_{24} &= \ell^{\mathsf{DH}}_{X2}\,\mathrm{c}_2\,\mathrm{s}_1 - \ell^{\mathsf{DH}}_{Z3}\,\mathrm{c}_1 - \ell^{\mathsf{DH}}_{Z4}\,\mathrm{c}_2\,\mathrm{s}_1\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_1\,\mathrm{s}_2
\end{aligned}
$$

$$
\begin{aligned}
t_{31} &= -\,\mathrm{c}_6\,\mathrm{s}_5\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_4\,\mathrm{c}_5\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_2 - \mathrm{s}_4\,\mathrm{s}_6\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_2\\
t_{32} &= \mathrm{s}_6\,\mathrm{s}_5\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_2\,\mathrm{s}_3 - \mathrm{c}_4\,\mathrm{c}_5\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_2 - \mathrm{c}_6\,\mathrm{s}_4\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_2\\
t_{33} &= \mathrm{c}_5\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_2\,\mathrm{s}_3 + \mathrm{c}_4\,\mathrm{s}_5\,\mathrm{c}_2\,\mathrm{s}_3 + \mathrm{c}_3\,\mathrm{s}_2\\
t_{34} &= \ell^{\mathsf{DH}}_{Z1} + \ell^{\mathsf{DH}}_{Z4}\,\mathrm{c}_2\,\mathrm{c}_3 - \mathrm{s}_2\,\mathrm{s}_3 + \ell^{\mathsf{DH}}_{X2}\,\mathrm{s}_2
\end{aligned}
$$

N.3 Manipulator dynamics model

The dynamics model is given on the general form

$$
\mathbf{M}(\theta)\ddot{\theta} = \tau - \mathbf{C}'(\theta)\dot{\theta}_{\text{quad}} - \mathbf{C}''(\theta)\dot{\theta}_{\text{prod}} - F(\dot{\theta}) - G(\theta) \ \Updownarrow
$$

$$
\begin{aligned}
\begin{bmatrix} m_{(1,1)} & m_{(1,2)} & m_{(1,3)} \\ m_{(2,1)} & m_{(2,2)} & m_{(2,3)} \\ m_{(3,1)} & m_{(3,2)} & m_{(3,3)} \end{bmatrix} \ddot{\theta} = \tau - \begin{bmatrix} c'_{(1,1)} & c'_{(1,2)} & c'_{(1,3)} \\ c'_{(2,1)} & c'_{(2,2)} & c'_{(2,3)} \\ c'_{(3,1)} & c'_{(3,2)} & c'_{(3,3)} \end{bmatrix} \dot{\theta}_{\text{quad}} - \cdots \\
\cdots - \begin{bmatrix} c''_{(1,1)} & c''_{(1,2)} & c''_{(1,3)} \\ c''_{(2,1)} & c''_{(2,2)} & c''_{(2,3)} \\ c''_{(3,1)} & c''_{(3,2)} & c''_{(3,3)} \end{bmatrix} \dot{\theta}_{\text{prod}} - \begin{bmatrix} f_{(1)} \\ f_{(2)} \\ f_{(3)} \end{bmatrix} - \begin{bmatrix} g_{(1)} \\ g_{(2)} \\ g_{(3)} \end{bmatrix}
\end{aligned} \tag{N.2}
$$

with the vectors $\dot{\theta}_{\text{quad}}$ and $\dot{\theta}_{\text{prod}}$ being defined as (structure given from [30])

$$\theta_{\text{quad}} \triangleq \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \\ \dot{\theta}_3^2 \end{bmatrix} \quad \text{and} \quad \theta_{\text{prod}} \triangleq \begin{bmatrix} \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_3 \\ \dot{\theta}_2\dot{\theta}_3 \end{bmatrix} \tag{N.3}$$

The vector $\dot{\theta}_{\text{quad}}$ and $\dot{\theta}_{\text{prod}}$, corresponding to quadratic velocities and product between velocities [86] together with the corresponding matrices $\mathbf{C}'$ and $\mathbf{C}''$, respectively, are not part of a linearization but used to show how the different vectors/matrices are composed. The vectors and matrices are given explicitly in the sequel subsections. The torque vector τ will be given explicitly, but rather as an input within the explicit description of the entire model on the form $\dot{x} = \mathbf{A}_l x + \mathbf{B}_l u + \mathbf{A}_n$, which contains both linear and nonlinear components. The state vector is defined as

$$x = \begin{bmatrix} I_1 & I_2 & I_3 & \theta_1 & \theta_2 & \theta_3 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 \end{bmatrix}^\intercal$$
$$\dot{x} = \begin{bmatrix} \dot{I}_1 & \dot{I}_2 & \dot{I}_3 & \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_3 & \ddot{\theta}_1 & \ddot{\theta}_2 & \ddot{\theta}_3 \end{bmatrix}^\intercal$$

with the linear system matrix given as

$$\mathbf{A}_l = \begin{bmatrix} -\alpha_1 & 0 & 0 & 0 & 0 & 0 & -\beta_1 & 0 & 0 \\ 0 & -\alpha_2 & 0 & 0 & 0 & 0 & 0 & -\beta_2 & 0 \\ 0 & 0 & -\alpha_3 & 0 & 0 & 0 & 0 & 0 & -\beta_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ m'_{(1,1)}N_1\gamma_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m'_{(2,2)}N_2\gamma_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m'_{(3,3)}N_3\gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where the $m'_{(i,i)}$-terms are derived from

$$\mathbf{M}^{-1} = \begin{bmatrix} m'_{(1,1)} & m'_{(1,2)} & m'_{(1,3)} \\ m'_{(2,1)} & m'_{(2,2)} & m'_{(2,3)} \\ m'_{(3,1)} & m'_{(3,2)} & m'_{(3,3)} \end{bmatrix}$$

The linear input matrix is given as

$$\mathbf{B}_l = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & \kappa_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The nonlinear part is not written explicitly, but given in compact form as

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -\mathbf{M}^{-1}\left(\mathbf{C}'(\theta)\dot{\theta}_{\text{quad}} + \mathbf{C}''(\theta)\dot{\theta}_{\text{prod}} + \mathbf{N}F(\dot{\theta}) + G(\theta)\right) \end{bmatrix}$$

Lastly, all vectors and matrices of the general dynamics model are given explicitly.

Quadratic velocities

$$\begin{aligned} c'_{(1,1)} = &- m_3\ell_1^2 \,\mathrm{c}_2^3\,\mathrm{s}_3 + m_3\ell_1^2\,\mathrm{c}_2^2\,\mathrm{c}_3\,\mathrm{s}_2 - m_3\ell_1^2\,\mathrm{c}_2\,\mathrm{s}_2^2\,\mathrm{s}_3 + m_3\ell_1^2\,\mathrm{c}_3\,\mathrm{s}_2^3 \\ &- \ell_2 m_3\ell_1\,\mathrm{c}_2^2\,\mathrm{s}_3 + 2\ell_2 m_3\ell_1\,\mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_2 + \ell_3 m_3\ell_1\,\mathrm{c}_2\,\mathrm{s}_2 - \ell_3 m_3\ell_1\,\mathrm{c}_2\,\mathrm{s}_3\,/2 \\ &- \ell_2 m_3\ell_1\,\mathrm{c}_3^2\,\mathrm{s}_2 - \ell_3 m_3\ell_1\,\mathrm{c}_3\,\mathrm{s}_2\,/2 + \ell_2 m_3\ell_1\,\mathrm{s}_2^2\,\mathrm{s}_3 - \ell_2 m_3\ell_1\,\mathrm{s}_2\,\mathrm{s}_3^2 \end{aligned}$$

$$c'_{(1,2)} = + \ell_1\ell_2 m_2\,\mathrm{s}_2\,/2 + \ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_2 - \ell_1\ell_2\,\mathrm{c}_2^2\,m_3\,\mathrm{s}_3 + \ell_1\ell_2 m_3\,\mathrm{s}_2^2\,\mathrm{s}_3 + 2\ell_1\ell_2\,\mathrm{c}_2\,\mathrm{c}_3\,m_3\,\mathrm{s}_2$$

$$c'_{(1,3)} = + \ell_2\ell_3 m_3\,\mathrm{s}_3\,/2 + \ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_2$$

$$c'_{(2,1)} = - \ell_1\ell_2 m_3\,\mathrm{s}_2\,\mathrm{c}_3^2 - \ell_1\ell_3 m_3\,\mathrm{s}_2\,\mathrm{c}_3\,/2 - \ell_1\ell_2 m_3\,\mathrm{s}_2\,\mathrm{s}_3^2 - \ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_3\,/2 - \ell_1\ell_2 m_2\,\mathrm{s}_2\,/2$$

$$c'_{(2,3)} = + \ell_2\ell_3 m_3\,\mathrm{s}_3\,/2$$

$$c'_{(3,1)} = - \ell_2\ell_3 m_3\,\mathrm{s}_3\,/2 - \ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_3\,/2 - \ell_1\ell_3\,\mathrm{c}_3\,m_3\,\mathrm{s}_2\,/2$$

$$c'_{(3,2)} = - \ell_2\ell_3 m_3\,\mathrm{s}_3\,/2$$

Friction forces

$$\begin{aligned} f_{(1)} &= -F_1\dot{\theta}_1 \\ f_{(2)} &= -F_2\dot{\theta}_2 \\ f_{(3)} &= -F_3\dot{\theta}_3 \end{aligned}$$

Mass matrix

$$
\begin{aligned}
m_{(1,1)} = &+ m_3\ell_1^2 \,\mathrm{c}_2^3\,\mathrm{c}_3 + m_3\ell_1^2\,\mathrm{c}_2^2\,\mathrm{s}_2\,\mathrm{s}_3 + m_2\ell_1^2\,\mathrm{c}_2^2 + m_3\ell_1^2\,\mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_2^2\\
&+ m_3\ell_1^2\,\mathrm{s}_2^3\,\mathrm{s}_3 + m_2\ell_1^2\,\mathrm{s}_2^2 + m_1\ell_1^2/4 + m_3\ell_1\ell_2\,\mathrm{c}_2^2\,\mathrm{c}_3\\
&+ m_3\ell_1\ell_2\,\mathrm{c}_2\,\mathrm{c}_3^2 + 2m_3\ell_1\ell_2\,\mathrm{c}_2\,\mathrm{s}_2\,\mathrm{s}_3 + m_3\ell_1\ell_2\,\mathrm{c}_2\,\mathrm{s}_3^2 + m_2\ell_1\ell_2\,\mathrm{c}_2\\
&- m_3\ell_1\ell_2\,\mathrm{c}_3\,\mathrm{s}_2^2 + m_3\ell_1\ell_3\,\mathrm{c}_2^2/2 + m_3\ell_1\ell_3\,\mathrm{c}_2\,\mathrm{c}_3/2 - m_3\ell_1\ell_3\,\mathrm{s}_2^2/2\\
&- m_3\ell_1\ell_3\,\mathrm{s}_2\,\mathrm{s}_3/2 + m_3\ell_2^2\,\mathrm{c}_3^2 + m_3\ell_2^2\,\mathrm{s}_3^2 + m_2\ell_2^2/4\\
&+ m_3\ell_2\ell_3\,\mathrm{c}_3 + m_3\ell_3^2/4
\end{aligned}
$$

$$
\begin{aligned}
m_{(1,2)} = &+ m_3\ell_2^2\,\mathrm{c}_3^2 + m_3\ell_2^2\,\mathrm{s}_3^2 + m_2\ell_2^2/4 + m_3\ell_2\ell_3\,\mathrm{c}_3\\
&+ \ell_1 m_3\ell_2\,\mathrm{c}_2^2\,\mathrm{c}_3 + 2\ell_1 m_3\ell_2\,\mathrm{c}_2\,\mathrm{s}_2\,\mathrm{s}_3 + \ell_1 m_2\ell_2\,\mathrm{c}_2/2 - \ell_1 m_3\ell_2\,\mathrm{c}_3\,\mathrm{s}_2^2\\
&+ m_3\ell_3^2/4 + \ell_1 m_3\ell_3\,\mathrm{c}_2^2/2 - \ell_1 m_3\ell_3\,\mathrm{s}_2^2/2
\end{aligned}
$$

$$
m_{(1,3)} = + \ell_3^2 m_3/4 + \ell_1\ell_3\,\mathrm{c}_2^2\,m_3/2 - \ell_1\ell_3 m_3\,\mathrm{s}_2^2/2 + \ell_2\ell_3\,\mathrm{c}_3\,m_3/2
$$

$$
\begin{aligned}
m_{(2,1)} = &+ m_3\ell_2^2\,\mathrm{c}_3^2 + m_3\ell_2^2\,\mathrm{s}_3^2 + m_2\ell_2^2/4 + m_3\ell_2\ell_3\,\mathrm{c}_3\\
&+ \ell_1\,\mathrm{c}_2\,m_3\ell_2\,\mathrm{c}_3^2 + \ell_1\,\mathrm{c}_2\,m_3\ell_2\,\mathrm{s}_3^2 + \ell_1\,\mathrm{c}_2\,m_2\ell_2/2 + m_3\ell_3^2/4\\
&+ \ell_1\,\mathrm{c}_2\,m_3\ell_3\,\mathrm{c}_3/2 - \ell_1 m_3\,\mathrm{s}_2\,\ell_3\,\mathrm{s}_3/2
\end{aligned}
$$

$$
m_{(2,2)} = + m_3\ell_2^2\,\mathrm{c}_3^2 + m_3\ell_2^2\,\mathrm{s}_3^2 + m_2\ell_2^2/4 + m_3\ell_2\ell_3\,\mathrm{c}_3 + m_3\ell_3^2/4
$$

$$
m_{(2,3)} = + m_3\ell_3^2/4 + \ell_2\,\mathrm{c}_3\,m_3\ell_3/2
$$

$$
m_{(3,1)} = + \ell_3^2 m_3/4 + \ell_2\ell_3\,\mathrm{c}_3\,m_3/2 + \ell_1\ell_3\,\mathrm{c}_2\,\mathrm{c}_3\,m_3/2 - \ell_1\ell_3 m_3\,\mathrm{s}_2\,\mathrm{s}_3/2
$$

$$
m_{(3,2)} = + m_3\ell_3^2/4 + \ell_2\,\mathrm{c}_3\,m_3\ell_3/2
$$

$$
m_{(3,3)} = + \ell_3^2 m_3/4
$$

Product of different velocities

$$c''_{(1,1)} = +\ell_1\ell_2 m_2\,\mathrm{s}_2 + 2\ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_2 - 2\ell_1\ell_2\,\mathrm{c}_2^2\,m_3\,\mathrm{s}_3 + 2\ell_1\ell_2 m_3\,\mathrm{s}_2^2\,\mathrm{s}_3 + 4\ell_1\ell_2\,\mathrm{c}_2\,\mathrm{c}_3\,m_3\,\mathrm{s}_2$$

$$c''_{(1,2)} = +\ell_2\ell_3 m_3\,\mathrm{s}_3 + 2\ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_2$$

$$c''_{(1,3)} = +\ell_2\ell_3 m_3\,\mathrm{s}_3 + 2\ell_1\ell_3\,\mathrm{c}_2\,m_3\,\mathrm{s}_2$$

$$c''_{(2,2)} = +\ell_2\ell_3 m_3\,\mathrm{s}_3$$

$$c''_{(2,3)} = +\ell_2\ell_3 m_3\,\mathrm{s}_3$$

$$c''_{(3,1)} = -\ell_2\ell_3 m_3\,\mathrm{s}_3$$

Terms with gravity

$$\begin{aligned} g_{(1)} = &- \ell_1\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2^3\,\mathrm{c}_3 + \ell_1 gm_3\,\mathrm{s}_1\,\mathrm{c}_2^3\,\mathrm{s}_3 - \ell_1 gm_3\,\mathrm{s}_1\,\mathrm{c}_2^2\,\mathrm{c}_3\,\mathrm{s}_2 - \ell_1\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2^2\,\mathrm{s}_2\,\mathrm{s}_3 \\ &- \ell_1\,\mathrm{c}_1\,gm_2\,\mathrm{c}_2^2 - \ell_2\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2\,\mathrm{c}_3^2 - \ell_1\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2\,\mathrm{c}_3\,\mathrm{s}_2^2 - \ell_3\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2\,\mathrm{c}_3\,/2 \\ &+ \ell_1 gm_3\,\mathrm{s}_1\,\mathrm{c}_2\,\mathrm{s}_2^2\,\mathrm{s}_3 - \ell_2\,\mathrm{c}_1\,gm_3\,\mathrm{c}_2\,\mathrm{s}_3^2 + \ell_3 gm_3\,\mathrm{s}_1\,\mathrm{c}_2\,\mathrm{s}_3\,/2 - \ell_2\,\mathrm{c}_1\,gm_2\,\mathrm{c}_2\,/2 \\ &+ \ell_2 gm_3\,\mathrm{s}_1\,\mathrm{c}_3^2\,\mathrm{s}_2 - \ell_1 gm_3\,\mathrm{s}_1\,\mathrm{c}_3\,\mathrm{s}_2^3 + \ell_3 gm_3\,\mathrm{s}_1\,\mathrm{c}_3\,\mathrm{s}_2\,/2 - \ell_1\,\mathrm{c}_1\,gm_3\,\mathrm{s}_2^3\,\mathrm{s}_3 \\ &- \ell_1\,\mathrm{c}_1\,gm_2\,\mathrm{s}_2^2 + \ell_2 gm_3\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3^2 + \ell_3\,\mathrm{c}_1\,gm_3\,\mathrm{s}_2\,\mathrm{s}_3\,/2 + \ell_2 gm_2\,\mathrm{s}_1\,\mathrm{s}_2\,/2 - \ell_1\,\mathrm{c}_1\,gm_1/2 \end{aligned}$$

$$\begin{aligned} g_{(2)} = &+ \ell_2 gm_2\,\mathrm{s}_1\,\mathrm{s}_2\,/2 - \ell_2\,\mathrm{c}_1\,\mathrm{c}_2\,gm_2/2 - \ell_2\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3^2\,gm_3 - \ell_2\,\mathrm{c}_1\,\mathrm{c}_2\,gm_3\,\mathrm{s}_3^2 \\ &+ \ell_2\,\mathrm{c}_3^2\,gm_3\,\mathrm{s}_1\,\mathrm{s}_2 + \ell_2 gm_3\,\mathrm{s}_1\,\mathrm{s}_2\,\mathrm{s}_3^2 - \ell_3\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3\,gm_3/2 + \ell_3\,\mathrm{c}_1\,gm_3\,\mathrm{s}_2\,\mathrm{s}_3\,/2 \\ &+ \ell_3\,\mathrm{c}_2\,gm_3\,\mathrm{s}_1\,\mathrm{s}_3\,/2 + \ell_3\,\mathrm{c}_3\,gm_3\,\mathrm{s}_1\,\mathrm{s}_2\,/2 \end{aligned}$$

$$g_{(3)} = + \ell_3\,\mathrm{c}_1\,gm_3\,\mathrm{s}_2\,\mathrm{s}_3\,/2 - \ell_3\,\mathrm{c}_1\,\mathrm{c}_2\,\mathrm{c}_3\,gm_3/2 + \ell_3\,\mathrm{c}_2\,gm_3\,\mathrm{s}_1\,\mathrm{s}_3\,/2 + \ell_3\,\mathrm{c}_3\,gm_3\,\mathrm{s}_1\,\mathrm{s}_2\,/2$$

Appendix O

Hardware gallery

Figure O.1: *Images of hardware configuration in laboratory*

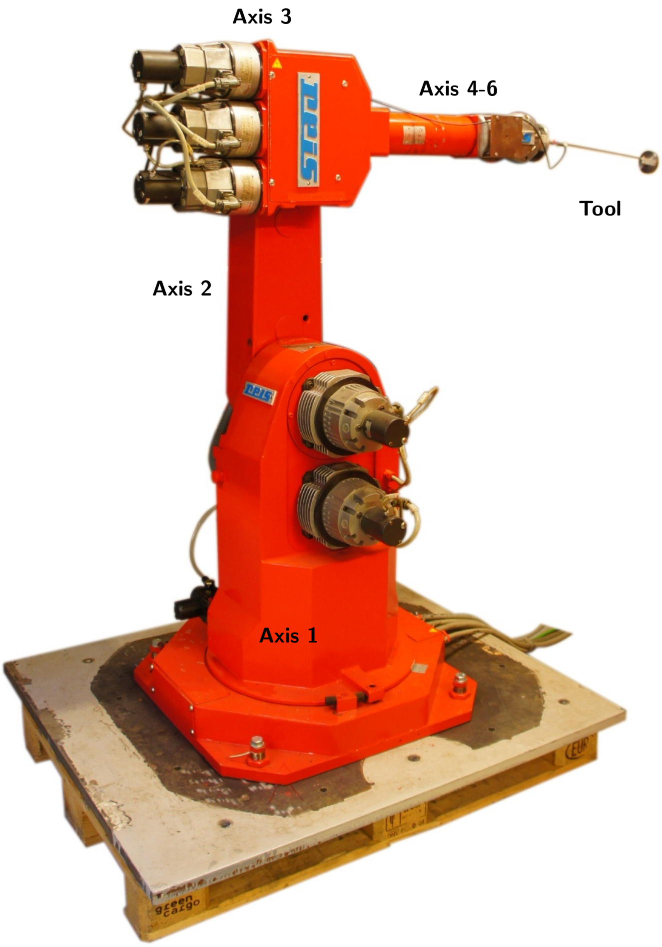


Figure O.2: *Image of REIS RV15 manipulator*

List of Figures

List of Tables

Bibliography

[1] Aalborg Universitet. AAU logos. http://www.design.aau.dk/logo%26segl/logo/, 2012.

[2] Jürgen Ackermann. *Robust Control: The Parameter Space Approach*, pages 110–111 (part of chapter 5). Springer-Verlag London Limited, 2 edition, 2002.

[3] M. Ameri and K. S. P. Kumar. Pattern matching algorithm for identification of load mass of an elastic robot arm. In *American Control Conference, 1988*, pages 963–964, june 1988.

[4] Analog Devices, Inc. Small, Low Power, 3-Axis ±3 g Accelerometer - ADXL335. http://www.analog.com/static/imported-files/data_sheets/adxl335.pdf, Rev. B, 2010.

[5] Analog Devices, Inc. Low Cost Low Power Instrumentation Amplifier - AD620. http://www.analog.com/static/imported-files/data_sheets/ad620.pdf, Rev. H, 2011.

[6] Lars K. Andersen, Jan S. Jensen, Allan M. Madsen, and Flemming B. Nielsen. Modellering og regulering af fleksibel robotarm. Aalborg University student report, 1995.

[7] Haruhiko Asada and Jean-Jacques E. Slotine. *Robot analysis and control*, pages 100–103 (subsection 5.1.3). John Wiley & Sons, Inc., 1 edition, 1986.

[8] K.J. Astrom, T.H. Lee, K.K. Tan, and K.H. Johansson. Recent Advances in Relay Feedback Methods - A Survey. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 3, pages 2616–2621, oct 1995.

[9] A.K.M. Azad and M.O. Tokhi. Modelling of a single-link flexible manipulator system: Theoretical and practical investigations. In M.O. Tokhi and A.K.M. Azad, editors, *Flexible Robot Manipulators: Modelling, simulation and control*, pages 23–46. The Institution of Engineering and Technology, 2008.

[10] W.P. Bailón, E.B. Cardiel, I.J. Campos, and A.R. Paz. Mechanical Energy Optimization in Trajectory Planning for Six DOF Robot Manipulators Based on Eighth-Degree Polynomial Functions and a Genetic Algorithm. In *Electrical Engineering Computing Science and Automatic Control (CCE), 2010 7th International Conference on*, pages 446–451, sept. 2010.

[11] Rüdiger G. Ballas. *Piezoelectric Multilayer Beam Bending Actuators - Static and Dynamic Behavior and Aspects of Sensor Integration*, pages 103–122 (chapter 6). Microtechnology and MEMS. Springer-Verlag Berlin Heidelberg, 2007.

[12] Millard F. Beatty. *Principles of Engineering Mechanics: Volume 2 Dynamics - The Analysis of Motion*, pages 363–367 (section 9.5). Springer Science+Business Media, Inc., 2006.

[13] Ferdinand P. Beer and E. Russell Johnston, Jr. *Mechanics of Materials*, page 269 (part of chapter 4 review). McGraw-Hill Inc., 2 edition, 1992.

[14] Walter Benenson, John W. Harris, Horst Stocker, and Holger Lutz. *Handbook of Physics*, pages 59–63 (section 2.3). Springer-Verlag New York, Inc., 2002.

[15] A. M. Bloch. *Nonholonomic Mechanics and Control*, volume 24 of *Interdisciplinary Applied Mathematics*, pages 1–12 (part of chapter 1). Springer New York, 2003.

[16] I. Boiko, L. Fridman, and R. Iriarte. Analysis of Chattering in Continuous Sliding Mode Control. In *American Control Conference, 2005. Proceedings of the 2005*, volume 4, pages 2439–2444, june 2005.

[17] Igor Boiko, Leonid Fridman, Alessandro Pisano, and Elio Usai. A Comprehensive Analysis of Chattering in Second Order Sliding Mode Control Systems. In Giorgio Bartolini, Leonid Fridman, Alessandro Pisano, and Elio Usai, editors, *Modern Sliding Mode Control Theory*, volume 375 of *Lecture Notes in Control and Information Sciences*, pages 23–49. Springer Berlin / Heidelberg, 2008.

[18] William J. Bottega. *Engineering Vibrations*, pages 552 (part of subsection 9.6.5), 603–610 (example 10.7–10.8). Taylor & Francis Group, LLC, 2006.

[19] J. A. Burton and A. S. I. Zinober. Continuous approximation of variable structure control. *International journal of Systems Science*, 17(6):875–885, may 1986.

[20] Jim Caldwell and Douglas Ng. *Mathematical Modelling - Case Studies and Projects*, volume 28 of *Texts in the Mathematical Sciences*, pages 67–84 (project B3). Springer Netherlands, 2004.

[21] Claudio Canuto and Anita Tabacco. *Mathematical Analysis I*, pages 404–410 (section 11.4). Universitext. Springer Milan, 2008.

[22] F.S. Cheng. The Method of Recovering Robot TCP Positions in Industrial Robot Application Programs. In *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pages 805–810, aug. 2007.

[23] Ji-wung Choi, R. Curry, and G. Elkaim. Path Planning Based on Bézier Curve for Autonomous Ground Vehicles. In *World Congress on Engineering and Computer Science 2008, WCECS '08. Advances in Electrical and Electronics Engineering - IAENG Special Edition of the*, pages 158–166, oct. 2008.

[24] Howie M. Choset. *Principles of Robot Motion: theory, algorithms, and implementations*, pages 40–47 (section 3.1–3.2). Massachusetts Institute of Technology, 2005.

[25] Esben Toke Christensen, Jonas Heidemann Sjølund, Jeppe Bjørn Jørgensen, Jens Ammitzbøll Glud, Henrik Okkels Birk, and Emil Krabbe Nielsen (group Fib.16-2.109). Design af robot-controller: Systemdesign. Aalborg University student report, 2011.

[26] Woojin Chung. *Nonholonomic Manipulators*, volume 13 of *Springer Tracts in Advanced Robotics*, pages 1–15 (chapter 1). Springer Berlin / Heidelberg, 2004.

[27] RS Components. Foil strain gauges and accessories (terminals/cement). http://docs-europe.electrocomponents.com/webdocs/0077/0900766b80077de6.pdf, May 1981.

[28] J. Connors and G. Elkaim. Analysis of a Spline Based, Obstacle Avoiding Path Planning Algorithm. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 2565–2569, april 2007.

[29] D. Constantinescu and E. A. Croft. Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators along Specified Paths. *Journal of Robotic Systems*, 17(5):233–249, 2000.

[30] John J. Craig. *Introduction to Robotics - Mechanics and Control*. Pearson Education, Inc., 3 edition, 2005.

[31] Dan Mihai Ştefănescu. *Handbook of Force Transducers - Principles and Components*, pages 347–360 (chapter 18). Springer Berlin Heidelberg, 2011.

[32] Alessandro De Luca, Valerio Caiano, and Dionisio Del Vescovo. Experiments on Rest-to-rest Motion of a Flexible Arm. In Bruno Siciliano and Paolo Dario, editors, *Experimental Robotics VIII*, volume 5 of *Springer Tracts in Advanced Robotics*, pages 338–349. Springer-Verlag Berlin Heidelberg, 2003.

[33] R.A. DeCarlo, S.H. Zak, and G.P. Matthews. Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial. *Proceedings of the IEEE*, 76(3):212–232, mar 1988.

[34] P. Dutkiewicz, K.R. Kozlowski, and W.S. Wróblewski. Experimental Identification of Robot and Load Dynamic Parameters. In *Control Applications, 1993., Second IEEE Conference on*, volume 2, pages 767–776, sept. 1993.

[35] C. Henry Edwards and David E. Penney. *Differential Equations & Linear Algebra*, pages 584–585 (part of section 10.2). Prentice-Hall, Inc., 2001.

[36] C. Henry Edwards and David E. Penney. *Calculus - Early Transcendentals*, pages 865–879 (section 11.6). Pearson Education, Inc., 7 edition, 2008.

[37] A. Ekrekli and D.J. Brookfield. Parameter estimation for a non-direct drive robot arm. In *Software Engineering for Real Time Systems, 1991., Third International Conference on*, pages 98–103, sep 1991.

[38] Farbod Fahimi. *Autonomous Robots*, pages 131–162 (chapter 5), 325–330 (appendix B). Springer US, 2009.

[39] J.J. Feliu, V. Feliu, and C. Cerrada. Load Adaptive Control of Single-Link Flexible Arms Based on a New Modeling Technique. *Robotics and Automation, IEEE Transactions on*, 15(5):793–804, oct 1999.

[40] Adrian Filipescu, Viorel Minzu, Adriana Filipescu, and Eugenia Minca. Discrete-Time Sliding-Mode Control of a Mobile Platform with Four Driving/Steering Wheels. In Gary Lee, editor, *Advances in Automation and Robotics, Vol.1*, volume 122 of *Lecture Notes in Electrical Engineering*, pages 401–409. Springer-Verlag Berlin Heidelberg, 2012.

[41] Gene F. Franklin, David J. Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems: International Version*, pages 42–43 (part of section 2.3), 107–115 (section 3.3). Pearson Education, 2008.

[42] Weibing Gao, Yufu Wang, and A. Homaifa. Discrete-Time Variable Structure Control Systems. *Industrial Electronics, IEEE Transactions on*, 42(2):117 –122, apr 1995.

[43] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering - Theory and Practice Using MATLAB*, pages 131–140 (part of chapter 4). John Wiley & Sons, Inc., 2007.

[44] P. Hamon, M. Gautier, and P. Garrec. Dynamic Identification of Robots with a Dry Friction Model Depending on Load and Velocity. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 6187–6193, oct. 2010.

[45] P. Hamon, M. Gautier, and P. Garrec. New Dry Friction Model with Load- and Velocity-Dependence and Dynamic Identification of Multi-DOF Robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1077–1084, may 2011.

[46] Steffen Hansen, Jens Grandjean Jørgensen, Jonas Mathias Laustsen, Kasper Mygind Madsen, and Casper Abildgaard Pedersen (group MP 3-026). Surgical robot with force feedback. Aalborg University student report, 2011.

[47] John W. Harris and Horst Stocker. *Handbook of Mathematics and Computational Science*, pages 756–759 (part of subsection 20.5.4). Springer-Verlag New York, Inc., 1998.

[48] R. C. Hibbeler. *Mechanics of Materials*, pages 299–302 (section 6.2). Pearson Prentice Hall, 7 edition, 2008.

[49] IRT SA. *Instruction manual - Series 600R servo-amplifiers braking modules racks*, 2009 (digital version, original version 1992).

[50] R. Isermann and M. Münchhof. *Identification of Dynamic Systems*, pages 548–549 (section 21.5). Springer-Verlag Berlin Heidelberg, 2011.

[51] Alan Jeffrey. *Handbook of mathematical formulas and integrals*, pages 293–299 (chapter 19). Academic Press, 1995.

[52] Jer-Nan Juang and Minh Q. Phan. *Identification and Control of Mechanical Systems*, pages 81–100 (section 4.2–4.3). Cambridge University Press, 2001.

[53] Tohru Katayama. *Subspace Methods for System Identification*, pages 1–14 (chapter 1). Communications and Control Engineering. Springer Berlin Heidelberg, 2005.

[54] K. J. Keesman. *System Identification - An Introduction*, pages 1–13 (chapter 1). Springer-Verlag London Limited, 2011.

[55] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*, pages 291–311 (chapter 12). Butterworth-Heinemann, 2004.

[56] A. Kiruluta, M. Eizenman, and S. Pasupathy. Predictive Head Movement Tracking Using a Kalman Filter. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(2):326 –331, apr 1997.

[57] Cristian Kunusch, Paul Puleston, and Miguel Mayosky. *Sliding-Mode Control of PEM Fuel Cells*, pages 35–71 (chapter 3). Springer London, 2012.

[58] P. A. A. Laura, J. L. Pombo, and E.A. Susemihl. A note on the vibrations of a clamped-free beam with a mass at the free end. *Journal of Sound and Vibration*, 37(2):161–168, 1974.

[59] T.H. Lee, Q.G. Wang, and K.K. Tan. A Modified Relay-Based Technique for Improved Critical Point Estimation in Process Control. *Control Systems Technology, IEEE Transactions on*, 3(3):330–337, sep 1995.

[60] Daniel Liberzon. *Switching in Systems and Control*, pages 3–15 (chapter 1). Systems & Control: Foundations & Applications. Birkhäuser Boston, 2003.

[61] Lennart Ljung. *System Identification: Theory for the User*, pages 418–419 (part of chapter 13). Prentice Hall PTR, 2 edition, 1999.

[62] LTN Servotechnik GmbH. Incremental Rotary Encoder G71. http://www.ltn.de/global/pdf/G71_gb_03_02.pdf, 17-04-2002.

[63] Henrik Madsen. *Time Series Analysis*, pages 163–166 (subsection 6.4.3). Chapman & Hall/CRC, 2008.

[64] C. Makkar, W.E. Dixon, W.G. Sawyer, and G. Hu. A New Continuously Differentiable Friction Model for Control Systems Design. In *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on*, pages 600–605, july 2005.

[65] Leonard Meirovitch. *Methods of Analytical Dynamics*, pages 88–91 (section 2.12). McGraw-Hill, Inc., 1970.

[66] Concepción A. Monje, YangQuan Chen, Blas M. Vinagre, Dingyü Xue, and Vicente Feliu. *Fractional-order Systems and Controls*, pages 59–62 (section 3.5). Advances in Industrial Control. Springer London, 2010.

[67] National Instruments Corporation. Operating instructions and specifications, *NI-9263*. http://www.ni.com/pdf/manuals/373781e.pdf, August 2009.

[68] National Instruments Corporation. Operating instructions and specifications, *CompactRIO™ cRIO*-9072/3/4. http://www.ni.com/pdf/manuals/374639e.pdf, June 2010.

[69] National Instruments Corporation. Operating instructions and specifications, *NI-9201/9221*. http://www.ni.com/pdf/manuals/373783g.pdf, May 2009.

[70] National Instruments Corporation. Operating instructions and specifications, *NI-9411*. http://www.ni.com/pdf/manuals/373506e.pdf, November 2009.

[71] Michael D O'Toole, Kaddour Bouazza-Marouf, and David Kerr. Chatter Suppression in Sliding Mode Control: Strategies and Tuning Methods. In Vincenzo Parenti-Castelli and Werner Schiehlen, editors, *ROMANSY 18 Robot Design, Dynamics and Control*, volume 524 of *CISM Courses and Lectures*, pages 109–116. Springer Vienna, 2010.

[72] Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, volume 49 of *Springer Tracts in Advanced Robotics*, pages 13–27 (chapter 2). Springer-Verlag Berlin / Heidelberg, 2008.

[73] No-Cheol Park, Hyun-Seok Yang, and Young-Pil Park. Position/vibration control of manipulators having flexible link with artificial pneumatic muscle actuators. In John X. Liu, editor, *Robots Manipulators: New Research*, pages 91–131. Nova Science Publishers, Inc., 2005.

[74] S. Park, W.K. Chung, Y. Youm, and J.W. Lee. Natural frequencies and open-loop responses of an elastic beam fixed on a moving cart and carrying an intermediate lumped mass. *Journal of Sound and Vibration*, 230(3):591–615, 2000.

[75] F. Pfeiffer and R. Johanni. A Concept for Manipulator Trajectory Planning. *Robotics and Automation, IEEE Journal of*, 3(2):115 –123, april 1987.

[76] Tim Piessens and Michiel Steyaert. *Design and Analysis of High Efficiency Line Drivers for xDSL*, volume 759 of *The Kluwer International Series in Engineering and Computer Science*, pages 57–73 (chapter 3). Springer US, 2004.

[77] K. Radkhah, D. Kulic, and E. Croft. Dynamic Parameter Identification for the CRS A460 Robot. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3842–3847, 29 2007-nov. 2 2007.

[78] Singiresu S. Rao. *Vibration of Continuous Systems*, pages 33–42 (section 2.1), 317–392 (chapter 11). John Wiley & Sons, Inc., 2007.

[79] Reis GmbH & Co Maschinenfabrik. *REIS manual with source code and specifications*, 1993.

[80] Reis GmbH & Co Maschinenfabrik. *Technical manual: Reis vertical articulated arm robots - RV12, RV12L, RV15*, 1993 (not indicated in manual but robot is from 1993).

[81] Hanz Richter. *Advanced Control of Turbofan Engines*, pages 112–120 (section 6.1). Springer Science+Business Media, LLC, 2012.

[82] Rush D. Robinett III, Clark Dohrmann, G. Richard Eisler, John Feddema, Gordon G. Parker, David G. Wilson, and Dennis Stokes. *Flexible Robot Dynamics and Control*, pages 288–302 (section 7.4). Kluwer Academic/Plenum Publishers, New York, 2002.

[83] Steven Roman. *Advanced Linear Algebra*, volume 135 of *Graduate Texts in Mathematics*, pages 205–226 (chapter 9). Springer New York, 3 edition, 2008.

[84] M. Shoham. Robotic surgery: Enabling technology? In Jacob Rosen, Blake Hannaford, and Richard M. Satava, editors, *Surgical Robotics - Systems Applications and Visions*, pages 247–255. Springer Science+Business LLC, 2011.

[85] Y.B. Shtessel and Young-Ju Lee. New Approach to Chattering Analysis in Systems with Sliding Modes . In *Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 4, pages 4014–4019, dec 1996.

[86] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics - Modelling, Planning and Control*. Springer London, 2009.

[87] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*, pages 179–187 (section 5.4), 276–309 (chapter 7). Prentice Hall, Inc., 1991.

[88] SparkFun Electronics. IMU Analog Combo Board Razor - 6DOF Ultra-Thin IMU. http://www.sparkfun.com/datasheets/sensors/imu/6dof-razor-v12.pdf, 2010 (version from 24/01).

[89] STMicroelectronics. LY530ALH - MEMS motion sensor: high performance ±300 °/s analog yaw-rate gyroscope. http://www.st.com/internet/com/technical_resources/technical_literature/datasheet/cd00237186.pdf, Doc ID 15801 Rev 2, July 2009.

[90] STMicroelectronics. LPR530AL - MEMS motion sensor: dual axis pitch and roll ±300 °/s analog gyroscope. http://www.st.com/internet/com/technical_resources/technical_literature/datasheet/cd00237209.pdf, Doc ID 15812 Rev 2, July 2009.

[91] Karl Johan Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*, pages 229–266 (chapter 8), 267–292 (chapter 9). Princeton University Press, 2008.

[92] H. Temeltas and G.M. Asher. Real time identification of robot dynamic parameters using parallel processing - Part 1: Theory. *Control Theory and Applications, IEE Proceedings -*, 142(4):359–368, july 1995.

[93] The MathWorks, Inc. System Identification Toolbox™- User's Guide, R2011b. http://www.mathworks.se/help/pdf_doc/ident/ident.pdf, 2011.

[94] Douglas Thorby. *Structural Dynamics and Vibration in Practice: An Engineering Handbook*, pages 1–28 (chapter 1), 181–235 (chapter 8). Elsevier Ltd., 2008.

[95] M. O. Tokhi, Wen-Jun Cao, Jian-Xin Xu, and H. Poerwanto. Vibration control of flexible manipulators. In Osman Tokhi and Sandor Veres, editors, *Active Sound and Vibration Control - theory and applications*, volume 62 of *IEE control engineering series*, pages 275–318. The Institution of Electrical Engineers, 2002.

[96] M.O. Tokhi, A.K.M. Azad, H.R. Pota, and K. Senda. Flexible manipulators - an overview. In M.O. Tokhi and A.K.M. Azad, editors, *Flexible Robot Manipulators: Modelling, simulation and control*, pages 1–21. The Institution of Engineering and Technology, 2008.

[97] Peter R. Turner. *Guide to Scientific Computing*, pages 171–178 (section 6.1). Macmillian Press Ltd., 2000.

[98] V. I. Utkin. Sliding Mode Control: Mathematical Tools, Design and Applications. In Andrei A. Agrachev, A. Stephen Morse, Eduardo D. Sontag, Héctor J. Sussmann, and Vadim I. Utkin, editors, *Nonlinear and Optimal Control Theory*, volume 1932 of *Lecture Notes in Mathematics*, pages 289–347. Springer Berlin / Heidelberg, 2008.

[99] Vadim Utkin. First Stage of VSS: People and Events. In Xinghuo Yu and Jian-Xin Xu, editors, *Variable Structure Systems: Towards the 21st Century*, volume 274 of *Lecture Notes in Control and Information Sciences*, pages 1–32. Springer-Verlag Berlin Heidelberg, 2002.

[100] Vadim Utkin, Jürgen Guldner, and Jingxin Shi. *Sliding Mode Control in Electro-Mechanical Systems*, pages 1–15 (chapter 1), 41–62 (chapter 3), 93–121 (chapter 5). CRC Press, 2 edition, 2009.

[101] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl. Time-Energy Optimal Path Tracking for Robots: a Numerically Efficient Optimization Approach. In *Advanced Motion Control, 2008. AMC '08. 10th IEEE International Workshop on*, pages 727–732, march 2008.

[102] David Wagg and Simon Neild. *Nonlinear Vibration with Control*, volume 170 of *Solid Mechanics and Its Applications*, pages 173–174 (section 5.1), 215–229 (section 6.1). Springer Netherlands, 2010.

[103] Robert B. Watson. Bonded Electrical Resistance Strain Gages. In William N. Sharpe, editor, *Springer Handbook of Experimental Solid Mechanics*, pages 283–334 (chapter 12). Springer US, 2008.

[104] E. Wernholt and S. Gunnarsson. Estimation of Nonlinear Effects in Frequency Domain Identification of Industrial Robots. *Instrumentation and Measurement, IEEE Transactions on*, 57(4):856 –863, april 2008.

[105] Jon S. Wilson. *Sensor Technology Handbook*, pages 619–623 (appendix G). Newnes, 2005.

[106] J. Xia and Q.J. Ge. An exact representation of effective cutting shapes of 5-axis CNC machining using rational Bézier and B-spline tool motions. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 342–347, 2001.

[107] Jian-jun Zhou and Finn Conrad. Identification and evaluation of hydraulic actuator models for a two-link robot manipulator. In Spyros G. Tzafestas, editor, *Robotic Systems: Advanced Techniques and Applications*, pages 577–584. Kluwer Academic Publishers, 1992.

[108] Dennis G. Zill and Michael R. Cullen. *Differential Equations with Boundary-Value Problems*, pages 212–220 (section 5.2). Thomson Brooks/Cole, 6 edition, 2005.

Index