Extending the Gillijns–De Moor Filter for Distributed Load Estimation on Wind-Turbine Blades

Master Thesis in Electronic Systems

Guillermo Valentín Gutiérrez Bea

Aalborg University
Electronic Systems

AALBORG UNIVERSITET
STUDENTERRAPPORT

Electronic Systems
Aalborg University
http://www.aau.dk

Title:
Extending the Gillijns–De Moor Filter for Distributed Load Estimation on Wind-Turbine Blades

Theme:
Master Thesis

Project Period:
Spring Semester 2025

Project Group:
Group 1026

Participant(s):
Guillermo Valentín Gutiérrez Bea

Supervisor(s):
Torben Knudsen
Szymon Gres
Mohamad Al Ahdab

Copies: 0

Page Numbers: 56

Date of Completion:
June 4, 2025

Abstract:

The technical problem that is tackled in this project is the estimation of distributed inputs from only acceleration data. In this pursuit, four setups are proposed that utilize the Gillijns–De Moor Filter (GDF), aiming to overcome the limitations of the filter in this scenario and reduce the number of sensors needed. Numerical simulations are provided, although only with a beam model due to time constraints. The methods are subjected to scenarios with loads of different dynamical properties and measurement noise levels, and their performance is compared. Further testing must be performed using a wind turbine blade model to validate its applicability to the target scenario.

Preface

This document constitutes the Master's thesis submitted in the tenth semester of the Electronic Systems programme, prepared under the invaluable guidance of my supervisors, whose expertise and support throughout the process are gratefully acknowledged.

Aalborg University, June 4, 2025

Guillermo Valentín Gutiérrez Bea

Contents

Introduction 1

Structural-dynamics studies are particularly active in the wind-energy field. Researchers must analyse highly stochastic aerodynamic and hydrodynamic loads, and many critical components become difficult to inspect or repair after turbine commissioning. For this reason, virtual numerical replicas [1] (termed Digital Twins) are developed to evaluate component performance. When predictive algorithms are incorporated into these twins, they provide probabilistic failure forecasts and enable maintenance planning that shortens repair periods, decreases unplanned interventions, and improves spare-parts logistics, thereby lowering costs. In such predictive-maintenance schemes, accurate estimation of fatigue life is essential, especially for offshore turbines whose parts experience continuous wind- and wave-induced cyclic loading. Detailed time-series records of material stresses and other fatigue-relevant indicators supply the data required to determine the remaining useful life of each structural element[2].

To make these precise models of wind turbine components, their precise dynamic response must be captured. Using a few sensors at sparse locations, virtual sensing (VS) can be used to estimate this response even where no sensor is placed [3]. That is usually achieved by combining a Finite Element (FE) model of the component with the sensor data. It is preferable to use accelerometer data only due to its convenience in structural dynamics applications [4]. One common strategy for performing VS uses Modal Expansion for this purpose, it has been successfully applied even with accelerometer measurement only [5].

Some widely used strategies are based on the Kalman filter, which is a recursive Bayesian estimator that blends a linear state-space model with Gaussian-noisy measurements to deliver minimum-mean-square-error state estimates in real-time. The challenge with this method is the fact that the forces acting on the system are not known and cannot be directly measured; these are unknown inputs to the Kalman filter. This problem was tackled

differently by the research community. Many use augmented state techniques to estimate the unknown loads. In [6], such a filter was presented while the input force was modelled as a random walk, and they identified a limitation of the method in estimating the slow-changing parts of the load when using acceleration measurements only. This issue was analysed by [4], where the stability of the filter was theoretically proved to be unstable in that scenario. As a solution, the introduction of dummy position measurements to render stable estimates is proposed. Another attempt on improving the stability of the filter in input force and state identification when using accelerometer data only was proposed in [7] where, instead of using a random walk model for the input forces, a stable dynamic model was used to recover the detectability of the system and hence obtain a stable filter. In particular, they obtain a state space representation of a process given a desired stationary covariance matrix, hence a non-parametric Gaussian process model. To improve the estimation of the AKF, [2] proposed a method that consists of adaptive tuning of the process noise covariance matrix.

In a different research direction, to avoid the difficulties of finding an accurate model of the inputs in practice [8] introduces a method that does not make any assumptions about the load's dynamics or statistical properties and offers an unbiased minimum variance estimator of the input for every time step of the Kalman filter; the so called Gillijns De Moore Filter filter (GDF). Thereafter, another approach was developed in [9] that uses a Dual Kalman filter formulation, which avoids un-observability and rank deficiency issues of the AKF and shows that by an adequate choice of noise covariance matrix of the input is able to avoid drift which is reported by some authors when using the GDF filter.

Since wind turbines are subjected to distributed loads instead of point loads, the specific scenario of estimating the distributed load is explored by trying to apply the GDF filter for that purpose. But as an early development of the methods, they will be applied and tested on a simple Finite element (FE) model of a beam. Due to time constraints, the method was not applied to a more realistic model of the wind turbine blade or with a more realistic wind load.

Problem Statement 2

The GDF filter is designed as an unbiased minimum variance estimator, this theoretical property makes it attractive for input-state estimation. Moreover, the fact that no assumption is made on the dynamics or statistical properties of the input makes it flexible and practical to adapt to any scenario, especially if the nature of the load is not fully known. Despite this, some challenges might arise when estimating dynamic distributed loads.

The main assumptions the filter makes are:

- **Plant and signal model**
  - Linear, time-varying, discrete-time:

$$x_{k+1} = A_k x_k + G_k d_k + w_k$$
$$y_k = C_k x_k + H_k d_k + v_k$$

  - System matrices A_k, G_k, C_k, H_k known.
  - Unknown input d_k arbitrary (no prior model).
- **Noise and initial state**
  - w_k, v_k: zero-mean, white, uncorrelated; covariances $Q_k \succeq 0$, $R_k \succ 0$.
  - Initial estimate $\hat{x}_0$ unbiased with known P_0^x; x_0 independent of $\{w_\ell, v_\ell\}$.
- **Structural / identifiability**
  - Full-column rank: $\text{rank}(H_k) = m$, where m is the number of measurements. Implying that the number of sensors must be lower than the number of inputs estimated.
  - Observable (or detectable) pair (A_k, C_k).

  Since a Finite Element (FE) model of the physical system will be used, the inputs to be estimated will be equal to the number of free nodes. That means that if the filter tries to estimate directly, for example, 14 nodal forces, it needs at least 14 sensors providing independent information about the inputs. This constraint on the

minimum number of sensors needed makes the filter impractical for estimating a distributed load a priori. Moreover, as modal reduced models are normally used in structural dynamics applications, some extra numerical challenges will arise, and in that regard, the work of [10] is used.

This leaves the problem statement as: *Developing a method for estimating an arbitrary dynamic distributed load with the Gillijns De Moore Filter while avoiding an excessive amount of sensors and using acceleration measurements only*

Methods 3

The most essential theory used in this project is introduced in the first part of the section as well as references to the appendix for more information. In the last part of the section, the limitations of the GDF are discussed, which feature a discussion on the limitations that appear when working with reduced-order models and the ones that appear when estimating a distributed input. Finally, some proposed methods using the GDF are proposed and described.

3.1 Elastic Beam Modelling

Mechanical and structural systems can be modeled by discretizing the system with finite element (FE) and enforcing the following governing equation:

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} = \{f(t)\} \tag{3.1}$$

where $\{u(t)\}$ contains the displacement of each element. The matrices correspond to: the global mass $[M]$, the global damping $[C]$, and the global stiffness $[K]$.

However, using this representation can pose some computational and conditioning problems since it requires dealing with a large state space of as many degrees of freedom as free nodes. To simulate large systems modal analysis can be used to find a more powerful representation that can be used to effectively reduce the dimensions of the state space by choosing the most relevant modes.

3.1.1 Modal Analysis

Following the modal expansion theorem, the vector $\{u(t)\}$ can be represented as a superposition of mode shapes $\{\phi_r\}$. Each mode will contribute to the response with various degrees of intensity over time which is determined by its modal coordinate $q_r(t)$. These form a new basis for $\{u(t)\}$ which is orthonormal.

$$\{u(t)\} = \{\phi_1\}q_1(t) + \{\phi_2\}q_2(t) + \cdots + \{\phi_n\}q_n(t) = [\Phi]\{q(t)\} \tag{3.2}$$

By substituting this expression and pre-multiplying by $\{\phi_r\}^T$ to achieve a similarity transform, projecting the equation into the new basis, the system is represented into an interesting structure. Due to the orthogonality of the new basis with respect to $[M]$ and $[D]$, the new matrices $m_r = \{\phi_r\}^T[M]\{\phi_r\}$, $k_r = \{\phi_r\}^T[K]\{\phi_r\}$ and in some cases $c_r = \{\phi_r\}^T[C]\{\phi_r\}$ become diagonal. This means that the new representation presents decoupled states independent from each other. Each mode is now a 1 DOF spring-mass system. In this work, this representation will be used for simplicity. More information in how to obtain this representation is found in the appendix A.

Introducing Damping

Depending on the type of damping used, the MDOF systems can be categorized as [11]:

- Undamped system, when $[C] = 0$
- Proportionally damped systems, with any of the following forms:
  * Rayleigh damping, $[C] = \beta[K] + \gamma[M]$
  * Modal damping
  * General proportional damping
- Non proportionally damped systems

For simplicity, the damped system is described as a proportionally damped system. This is done to obtain uncoupled and damped modes of vibration. Specifically, the method of obtaining the proportional damping via modal damping is chosen.

Modal damping is found based on the assumption

$$c_r = \{\phi_r\}^T[C]\{\phi_r\} = diag\{2\zeta_k\omega_k\} \tag{3.3}$$

Under this assumption only the eigenvalues change and not the eigenvectors [11]. According to the formula, the damping ratio ζ_k is applied to the kth mode of vibration (eigenvector). This leads to a characteristic polynomial per mode of the form

$$\lambda^2 + 2\zeta_k\omega_k\lambda + \omega_k^2 = 0 \tag{3.4}$$

By finding their roots one can find the eigenvalues associated with each eigenvector

$$\lambda_{\pm k} = -\zeta_k \omega_k \pm i\sqrt{1-\zeta_k^2}\omega_k \tag{3.5}$$

3.1.2 Modal Decomposition of State-Space Models

In the previous part of this section, the modal analysis was introduced and the power of describing the system in modal coordinates was also presented. Due to the convenience of representing the system in a state space model for state estimation, the state space model itself it is transformed into modal coordinates with a similarity transform.

First, equation 3.1 is posed in state-space form by defining the state vector $\{x(t)\} = [\{u(t)\}^T, \{\dot{u}(t)\}^T]^T$ with displacement and velocity of the elements. And the input vector of forces $\{f(t)\}$.

By isolating $\ddot{u}$ in the equation 3.1 one obtains the derivative of the states of velocity which is the lower row of equation 3.6.

$$\begin{bmatrix} \dot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \mathbf{f}(t) \tag{3.6}$$

To transform it to a modal basis a similarity transform is needed. This transformation allows for changing the basis of the state space representation of the system.

$$\mathbf{z}(t) = v^{-1}\mathbf{x}(t) \tag{3.7}$$

Where v is the transformation matrix. Specifically, if the aim is to represent the state space in a modal basis, the columns of v should be chosen as the eigenvectors of the state matrix A (where $\mathbf{x}(t) = \begin{bmatrix} \mathbf{u}(t) \\ \dot{\mathbf{u}}(t) \end{bmatrix}$ and

$$A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}).$$

That is, let v satisfy

$$Av = \Lambda v, \tag{3.8}$$

where Λ is a diagonal matrix containing the eigenvalues of A.

Substituting $\mathbf{x}(t) = v\mathbf{z}(t)$ into equation (3.6) gives

$$v\dot{\mathbf{z}}(t) = A(v\mathbf{z}(t)) + B\mathbf{f}(t), \tag{3.9}$$

with

$$B = \begin{bmatrix} 0 \\ \\ M^{-1} \end{bmatrix}.$$

Multiplying both sides by v^{-1} leads to

$$\dot{\mathbf{z}}(t) = v^{-1}Av\,\mathbf{z}(t) + v^{-1}B\mathbf{f}(t) = \Lambda\mathbf{z}(t) + \tilde{B}\mathbf{f}(t), \tag{3.10}$$

where $\tilde{B} = v^{-1}B$.

Furthermore, the input matrix can be made in terms of the eigenvalues and eigenvectors in the following way [12]

$$\tilde{B} = \begin{bmatrix} I \\ \\ -I \end{bmatrix} (\bar{\Lambda} - \bar{\Lambda}^*)^{-1}\bar{v}^T F \tag{3.11}$$

Summarizing, this similarity transformation converts the original state-space representation into a modal form, in which the dynamics are decoupled. Each modal coordinate in $\mathbf{z}(t)$ evolves independently according to its corresponding eigenvalue (contained in Λ). This decoupled (modal) form greatly simplifies the analysis and control design of the system. These modal coordinates are related to the modal coordinates mentioned in the modal expansion theorem 3.2 by the following expression $\mathbf{z}(t) = [\mathbf{q}(t)\ \dot{\mathbf{q}}(t)]^T$.

However, not all modes need to be included in the final model. The importance of each mode in the response depends on the nature of the excitation and more specifically its spectral content. That means that modes appearing in frequencies where the input has no or very little energy are not excited and can be disregarded. By removing some of the least relevant modes modal reduction is performed; which allows for a more efficient representation of the system.

Measurement model

The measurements consist of acceleration measurements. The model itself can be used to calculate the acceleration and using a matrix C_a can be used to select the accelerations of the nodes of interest. As shown in equation 3.12, it leads to an output/measurement model with a direct feedthrough matrix.

$$\mathbf{y}(t) = C_a\ddot{\mathbf{u}}(t) = C_a\begin{bmatrix} -M^{-1}K & -M^{-1}C \end{bmatrix}\mathbf{x}(t) + C_aM^{-1}\mathbf{f}(t) = C\mathbf{x}(t) + D\mathbf{f}(x) \tag{3.12}$$

Performing the modal decomposition as in the dynamic model of the system, the similarity transform is applied to go from the coordinates $\mathbf{x}(t)$ to the coordinates $\mathbf{z}(t)$

$$y(t) = Cv\mathbf{z}(t) + Df(t) \simeq \bar{C}\mathbf{z}(t) + \bar{D}\mathbf{f}(t) \tag{3.13}$$

where $\bar{C}$ and $\bar{D}$ are modally reduced matrices. This is their definition in terms of the included eigenvalues and eigenvectors.

$$\bar{C} = C_a\begin{bmatrix} \bar{v}\bar{\Lambda}^2 & \bar{v}(\bar{\Lambda}^*)^2 \end{bmatrix} \tag{3.14}$$

$$\bar{D} = C_a\bar{v}\bar{v}^T \tag{3.15}$$

3.1.3 Discretization of a State Space System

Finding the equivalent discrete state space formulation of a continuous state space comes down to finding the matrices A_d and B_d from equation 3.16.

$$\mathbf{x}(n+1) = A_d\,\mathbf{x}(n) + B_d\,\mathbf{u}(n) \tag{3.16}$$

Depending on the assumptions made, different techniques can be employed to find these matrices. Here, the exact discretization method is employed, and the ZOH assumption is made, which means that between samples the inputs are assumed to be held constant. To construct the matrices, the solution of the continuous state space in 3.17 is used. Essentially, the solution is computed by taking the previous state as the initial condition and assuming a constant input during the sample interval for every sample interval. That way, the dynamics are still continuous, and the only inaccuracy comes from the fact that the input may not be constant during the sample interval.

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-\tau)} B\mathbf{u}(\tau)\, d\tau \tag{3.17}$$

Given the ZOH assumption, since the input is constant, it can be taken out from the integration and the following A_d and B_d matrices can be obtained:

$$\mathbf{x}((n+1)\,T_s) = \underbrace{e^{AT_s}}_{A_d}\mathbf{x}(nT_s) + \underbrace{\left(\int_0^{T_s} e^{A\tau}\, d\tau\right) B}_{B_d}\, \mathbf{u}(nT_s) \tag{3.18}$$

and after performing the integration:

$$B_d = (e^{AT_s} - I)A^{-1}B \tag{3.19}$$

Discretization of the Modal Decomposed State-Space

In the previous section, the state-space formulation of the system was transformed into modal form. Fortunately, after the transformation, the dynamics are decoupled, making A a diagonal matrix. This is advantageous as the matrix exponential of a diagonal matrix simplifies to a diagonal matrix whose entries are the scalar exponentials of the original diagonal elements. Therefore, A_d is simply:

$$e^{\Lambda T_s} = \begin{bmatrix} e^{\lambda_1 T_s} & & & & & \\ & \ddots & & & & \\ & & e^{\lambda_k T_s} & & & \\ & & & e^{\lambda_1^* T_s} & & \\ & & & & \ddots & \\ & & & & & e^{\lambda_k^* T_s} \end{bmatrix} \tag{3.20}$$

To obtain the B_d matrix, equation 3.19 is used where the continuos A matrix is Λ and the continuous B matrix is given in equation 3.11.

However, in this basis, the modal coordinates are complex-valued. To obtain real-valued states as $\mathbf{z}_{\text{real}} = [\text{Re}\{\mathbf{z}\}, \text{Im}\{\mathbf{z}\}]^T$, a similarity transform $\mathbf{z}_{\text{real}} = \tilde{T}\mathbf{z}$ can be performed. Such a transformation matrix is given in equation 3.21.

$$\tilde{T} = \frac{1}{2}\begin{bmatrix} I & -iI \\ I & iI \end{bmatrix} \tag{3.21}$$

After applying the transformation, the state matrix becomes $\tilde{T}^{-1}A_d\tilde{T}$, the input matrix becomes $\tilde{T}^{-1}B_d$ and the C matrix becomes $C\tilde{T}$.

3.1.4 Discrete Model of a Distributed Input

Due to using a finite-element model, the inputs can only be applied to the nodes and not within the elements. This requires the discretization of the distributed load. Ideally, the equivalent discretized load should produce the same external work as the distributed load for each possible displacement function that the shape functions allow. Fortunately, such discretization is possible, and it is said to be energy-consistent[13].

For an arbitrary virtual displacement $\delta v(x) = [N]\,\delta\{d\}$ the external virtual work due to the distributed traction $q(x)$ is

$$\delta W_{\text{ext}} = \int_\Gamma q(x)\,\delta v(x)\,d\Gamma = \delta\{d\}^{\mathsf{T}}\left(\int_\Gamma [N]^{\mathsf{T}} q(x)\,d\Gamma\right), \tag{3.22}$$

where Γ is the portion of the spatial domain where the distributed load is applied.

However, one must remember that the finite-element interpolation approximates the displacement field along the loaded edge (or line element)

$$v(x) = [N]\,\{d\}, \tag{3.23}$$

where $[N]$ is the *row* matrix of shape functions $N_i(x)$ and $\{d\}$ is the *column* vector of nodal degrees of freedom (DOF).

This leads to the following expression of the external virtual work from an unknown nodal load vector $\{f\}$

$$\delta W_{\text{nodal}} = \delta\{d\}^{\mathsf{T}}\,\{f\}. \tag{3.24}$$

Requiring $\delta W_{\text{ext}} = \delta W_{\text{nodal}}$ for every admissible $\delta\{d\}$ leads directly to the consistent (energy-conjugate) nodal load vector

$$\boxed{\{f\} = \int_\Gamma [N]^{\mathsf{T}} q(x)\,d\Gamma} \tag{3.25}$$

Component-wise,

$$f_i = \int_\Gamma N_i(x)\, q(x)\, d\Gamma. \tag{3.26}$$

In practice, the integral for each f_i is evaluated on every element patch where $N_i \neq 0$ and the results are summed during assembly. After that, all contributions from each element are superimposed at the shared nodes. I.e., if two adjacent elements share a node, their contributions accumulate, so the final global vector $\{F\}$ contains the total nodal forces that are energetically equivalent to the original distributed load; given the assumed interpolation given by the shape functions.

3.1.5 Selecting the Number of Elements

The objective of the analysis is to find the minimal number of elements to divide the beam to get a sufficiently accurate dynamical response for a particular load. To do so, the acceleration measurement was computed for several seconds on the free end of the beam, since all modes have significant participation in the motion of that particular node. Note that only five modes of the beam are included due to the nature of the excitation that the beam is subjected to, in this case, wind. The response is measured from models with different numbers of elements. That way, the relative root mean squared error (RRMSE) between the response of models with a subsequent number of elements can be computed until the RRMSE falls below the convergence criterion, signifying that increasing the number of elements would not yield a significantly different response.

An example of a constantly distributed load in space but with white noise dynamics is imposed on the beam model consisting of 4 modes. The test shows that with 8 elements, the RRMSE falls below the convergence criterion of 1% as figure 3.1 shows. The RRMSE is 0.835%.

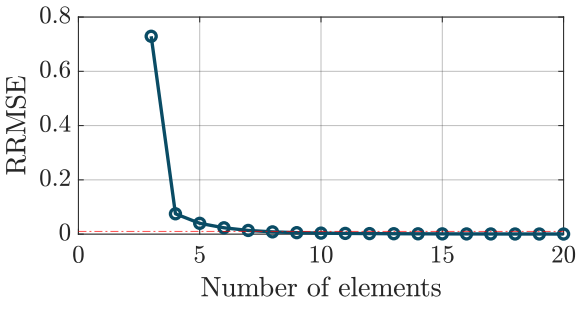


Figure 3.1: RRMSE between the acceleration responses of the end node of models with a subsequent number of elements. The threshold is 1% and is crossed at 8 elements.

3.2 Input-state estimation for linear discrete-time systems with direct feedthrough using the GDF

This section explains the estimator proposed by [8] together with a discussion about its limitations when dealing with reduced-order models.

Having a system formulated in state-space form, where the system model in equation 3.27 and the measurement model in equation 3.28 are shown.

$$x_{k+1} = A_k x_k + G_k d_k + w_k \tag{3.27}$$

$$y_k = C_k x_k + H_k d_k + v_k \tag{3.28}$$

Where the matrices A_k, G_k, C_k, H_k are known, the noise terms w_k and v_k are uncorrelated with mean zero white noise terms (the case with correlated measurement noise is discussed in A.1.1). The state is represented as x_k, and the unknown disturbance input as d_k. The disturbance input can also be viewed as a systematic error or bias term in the models, which directly affects both the system model and the measurement model.

The objective presented in the paper [8] is to find an optimal recursive filter that estimates both the system state x_k and the input based on the initial estimate $\hat{x}_0$ and the sequence of measurements from y_0 to y_k, without any prior knowledge of the input or its structure.

It is worth noting that due to the direct feedthrough, the estimation of d_k can be done without a one-step delay since the measurement will contain the current value of the unknown input.

To achieve this goal a certain recursive filter structure is proposed that consists in three steps:

1. Time Update:
$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + G_{k-1}\hat{d}_{k-1} \tag{3.29}$$
2. Estimation of the unknown inputs:
$$\hat{d}_k = M_k(y_k - C_k\hat{x}_{k|k-1}) \tag{3.30}$$

3. Measurement Update:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k(y_k - C_k\hat{x}_{k|k-1}) \tag{3.31}$$

This narrows down the problem to finding the unknown matrices M_k and L_k such that a minimum variance unbiased estimator is found for both the states and the unknown inputs.

3.2.1 Input Estimation

Focusing on the input estimation of equation 3.30. The innovation $\tilde{y}_k$ difference between the measurement and the predicted measurement is used for the following property:

$$\tilde{y}_k := y_k - C_k\hat{x}_{k|k-1} = C_k\tilde{x}_{k|k-1} + H_k d_k + v_k = H_k d_k + e_k \tag{3.32}$$

Where an error term $e_k := C_k\tilde{x}_{k|k-1} + v_k$ can be defined with the contribution of the error from the state estimation and from the random noise v_k. With this the expectation of the innovation can be calculated and, as depicted in equation 3.33, it is proportional to the expected value of d_k. This proves that the innovation can be used to obtain an unbiased input estimation.

$$\mathbb{E}[\tilde{y}_k] = \mathbb{E}[H_k d_k] + \mathbb{E}[e_k] = H_k\mathbb{E}[d_k] \tag{3.33}$$

Note that $\mathbb{E}[e_k] = 0$ since the $\hat{x}_{k|k-1}$ is unbiased and the random noise v has mean zero.

Using Least Squares (LS) one can find the ordinary least squares estimator $\hat{d}_k$ for the linear regression problem stated at 3.32 where the objective is to find the d_k that would minimize the cost function in equation 3.34.

$$\underset{d_k}{\arg\min} \, \|\tilde{y}_k - H_k d_k\|^2 \tag{3.34}$$

The estimator that satisfies this is $\hat{d}_k = (H_k H_k^T)^{-1} H_k^T \tilde{y}_k$ which is the LS solution. This requires that $M_k = (H_k H_k^T)^{-1} H_k^T$ which is the pseudo-inverse of H_k.

This estimator would be unbiased and minimum variance if the Gauss-Markov assumption were met. However, the issue is that the error term e_k likely has $Var[e_k] \neq cI$, which violates one of the assumptions and makes the innovation correlated. In fact:

$$\tilde{R}_k := \mathbb{E}[e_k e_k^T] = C_k P_{k|k-1} C_k^T + R_k \neq cI. \tag{3.35}$$

This implies that the estimator, even though it is unbiased, it is not necessarily minimum variance. To overcome this issue, the problem can be framed as a Generalized Least Squares problem [14] which first transforms the model to a representation where the correlation in the measurements is eliminated and then the ordinary least squares problem can be applied to this transformed model.

Define the whitening matrix as

$$W_k = \tilde{R}_k^{-1/2}, \tag{3.36}$$

which satisfies

$$W_k \tilde{R}_k W_k^T = I.$$

and leads to the whitening transform

$$\bar{y}_k = \tilde{R}_k^{-1/2} \tilde{y}_k, \quad \bar{H}_k = \tilde{R}_k^{-1/2} H_k, \tag{3.37}$$

so that the model becomes

$$\bar{y}_k = \bar{H}_k d_k + \bar{e}_k, \tag{3.38}$$

with the transformed error term

$$\bar{e}_k = \tilde{R}_k^{-1/2} e_k,$$

which now satisfies

$$\mathbb{E}[\bar{e}_k \bar{e}_k^T] = cI.$$

In the whitened model (3.38), the ordinary least squares solution can be applied, leading to the GLS estimator:

$$\hat{d}_k = \left(H_k^T \tilde{R}_k^{-1} H_k\right)^{-1} H_k^T \tilde{R}_k^{-1} \tilde{y}_k. \tag{3.39}$$

And therefore $M^* = \left(H_k^T \tilde{R}_k^{-1} H_k\right)^{-1} H_k^T \tilde{R}_k^{-1}$ will make the estimator both minimum variance and unbiased.

In summary, the estimation of the unknown input d_k is performed using the innovation $\tilde{y}_k$ which, as shown in equation 3.32, is given by

$$\tilde{y}_k = H_k d_k + e_k.$$

Although a simple least squares solution

$$\hat{d}_k = (H_k H_k^T)^{-1} H_k^T \tilde{y}_k$$

provides an unbiased estimate, the presence of correlated noise (with covariance $\tilde{R}_k = C_k P_{k|k-1} C_k^T + R_k \neq cI$) means that this estimator does not have minimum variance. The GLS approach remedies this by transforming the model, effectively a change of basis via the whitening transformation in (3.37), so that the noise becomes white, and the minimum variance property is restored in the transformed space. The resulting GLS estimator in (3.39) is thus the minimum-variance unbiased estimator for d_k.

3.2.2 Measurement Update

Similarly, the objective in the measurement update step is to find a matrix L_k that produces an unbiased minimum variance estimate of the state. To ensure that the estimate is unbiased it must cancel the bias term d_k in

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k(y_k - C_k \hat{x}_{k|k-1}) = (I - L_k C_k)\tilde{x}_{k|k-1} - L_k H_k d_k - L_k v_k \tag{3.40}$$

which means that

$$L_k H_k = 0 \tag{3.41}$$

As discussed in [8], an MVU state estimator is obtained by determining the gain matrix L_k that minimizes the trace of the covariance matrix of the estimation while satisfying the unbiasedness constraint in 3.41. Where the covariance matrix is presented in 3.42. Note that the trace of the covariance matrix corresponds to the variance terms of the states' prediction.

$$P_{x_{k|k}} = (I - L_k C_k) P_{x_{k|k-1}} (I - L_k C_k)^T + L_k R_k L_k^T \tag{3.42}$$

The solution yields the following equation:

$$L_k = K_k (I - H_k M_k) \tag{3.43}$$

where

$$K_k = P_{x_{k|k-1}} C_k^T \tilde{R}_k^{-1} \tag{3.44}$$

3.2.3 Limitations of the Estimator

For the linear estimator to be unbiased, a necessary and sufficient condition is that the rank of H_k is equal to the number of inputs, as discussed in [8]. This implies two conditions:

- The number of sensors should be at least as great as the number of estimated inputs.
- The columns of the direct feedthrough matrix H_k must be linearly independent, which implies that the regressors are linearly independent [14].

A lower number of independent measurements than inputs to estimate makes an underdetermined system of equations. This means that there are infinite solutions. Not having independent columns means that not every input independently affects the measurement, and therefore, they cannot be distinguished easily. Also known as multicollinearity, it causes large variances and covariances on the least-squares estimators of the regression coefficients [14].

The former condition limits the applicability by limiting the amount of forces estimated to the number of sensors available. The latter directly limits the minimum number of modes that need to be included in the reduced order model of the system. The effect of reducing the modes on the rank reduction of H_k is manifested as a projection matrix to the subspace spanned by the modal coordinates in the definition of the reduced version of H_k in equation 3.15. Therefore the rank of the reduced H_k cannot be larger than the number of modes of the reduced order model. This projection occurs because the acceleration measurement is calculated using the reduced order model itself presented in equation 3.12.

Moreover, in [10] the numerical issues arising from applying this estimator to a reduced order model in the context of structural dynamics were discussed and tackled with some additions to the algorithm. The summary of the issues related to the reduced order model can be summarized as follows:

- The number of inputs should not be greater than the number of modes ($n_d \leq n_{modes}$). If the condition is broken the matrix $\left(H_k^T \tilde{R}_k^{-1} H_k\right)^{-1}$ in equation 3.39 becomes singular due to the rank deficiency of the matrix being inverted.
- The number of measurements should not be greater than the number of modes ($n_y \leq n_{modes}$). Otherwise, the matrix $\tilde{R}_k^{-1}$ becomes nearly singular.
- If $n_y \geq min(n_d, n_m)$ then $HP_{x_{k|k}}H^T$ becomes rank deficient.

Table 3.1: Conditions for numerical stability in the UMV estimator (adapted from [10])

Condition	Numerical Issue if Violated
$n_d \leq n_{\text{modes}}$	$\left(H_k^T \tilde{R}_k^{-1} H_k\right)^{-1}$ becomes singular
$n_y \leq n_{\text{modes}}$	$\tilde{R}_k^{-1}$ becomes ill-conditioned or singular
$n_y \leq \min(n_d, n_{\text{modes}})$	$HP_{x_{k\|k}}H^T$ becomes rank deficient

To avoid these, some additions to the original UMV estimator were proposed in the paper. It involves using spectral truncation techniques, where the problematic matrices are regularized by retaining only the dominant eigenmodes, and a pseudo-inverse is performed instead of an inverse. The result is that only meaningful information is passed from one time step of the filter to the next one [10] and the issues stated in table 3.1 are thus avoided. Nevertheless, violating these conditions does not come without a degradation in performance.

Methodological Limitations for Distributed Input Estimation

Even though some numerical issues can be mitigated as previously discussed, some of the limitations of the filter may present some challenges in the estimation of distributed loads.

Its main limitations identified for estimating a distributed load are:

- Number of estimated forces should be smaller than or equal to the number of measurements.
- Having underestimated nodal forces where the distributed load is being applied will be treated as process noise. Which poses two main problems:
  * The unestimated forces are not necessarily mean zero.
  * If the inputs are treated as process noise and using the formulation that uncorrelates the measurement and process noise A.1.1 the input matrix associated to the unestimated forces must be included in the definition of the covariance matrices where the process noise has an effect and leads to its inversion i.e., the inversion of a non-square matrix as can be seen in appendix C.2.

Consequently, an ideal method should allow the estimation of a reduced number of forces that is enough to reconstruct the distributed load; avoiding the need for

a large quantity of sensors and without leaving any unestimated nodal forces where the distributed load is applied.

To validate the hypothetical impracticability of leaving unestimated forces, a setup where some unestimated forces will also be constructed and subjected to the same tests as the other methods.

3.3 Proposed Methods for Estimating Distributed Loads with the GDF

Three different methods are proposed to deal with the aforementioned limitations. The common feature among them is the avoidance of unestimated nodal forces. Then, an extra method is proposed where some forces are left unestimated to compare with the other methods and validate if, in fact, the other methods are more reliable.

- Estimation of the modal forces instead. And then converting them back to nodal forces after the estimation is done.
- Making a coarser discretized FEM model and estimate all of the nodal forces where the load is acting.
- Parameterizing the distributed input. Basis functions can be used to describe the distributed input. The estimation then consists of estimating the amplitude coefficient of each basis function.

Only in the third method is the spatial interpolation embedded into the estimation. Therefore, offline spatial interpolation must be performed on the other two methods to obtain an estimate of the continuous load.

3.3.1 Method 1

In this case, the estimator tries to estimate the modal forces instead and uses them within the filter. That significantly reduces the number of forces needed to be estimated to the number of modes and could provide a more stable filter implementation. Moreover, it also changes the minimal required number of sensors to be equal to or larger than the number of modes included.

Then, some or all the other forces can be computed from the estimated modal forces.

3.3.2 Method 2

Here, all nodal forces are estimated, but to do so, the number of elements of the model has to be greatly reduced. And the sensors in the courser discretized model should be placed in the same physical coordinate as the ones in the finer discretized model that produced the response measured. The algorithm used to easily do so is the following:

- **Create the detailed beam model:**
  - Define the number of elements as n_E (e.g., 40 elements).
- **Create the coarser beam model for estimation:**
  - Use fewer elements than the detailed model.
  - The number of elements must divide n_E evenly (i.e., n_E/n, where n is an integer).
  - Example: If $n_E = 40$ and $n = 4$, the coarser model has $40/4 = 10$ elements.
- **Sensor placement:** Sensors must be placed at the same physical location in both the detailed and coarse models.
  - Since in the coarse model the sensors are placed in every node of the model (except at the support) to have as many sensors as forces to estimate, the problem becomes finding the equivalent location of the sensors in the finer discretized model. The following formula can be used to relate the nodes where the sensor is placed in the corase model to the nodes in the fine model:

$$n_f = (n_c - 1)n + 1 \tag{3.45}$$

    where n_f is the node where the sensor is placed on the fine model in the same physical location as the node n_c in the coarse model. Note that the following relationship relates the number of nodes and elements $nE = n_{nodes} - 1$
  - Example:
    - For $n_E = 40$ and $n = 4$, possible sensor locations in the fine model are at multiples of 4 (i.e., 4, 8, 12, ..., 40), avoiding node 1 if it is supported at that node.

3.3.3 Method 3

In this setup, the distributed load is approximated with basis functions.

$$q_k(x) \approx \sum_{i=1}^{n} b_{i,k}\phi_i(x) \tag{3.46}$$

Then it is converted into equivalent discrete nodal forces using the energy equivalent approach in section 3.1.4. After obtaining the equivalent nodal force vector for the profile of each basis function, they can be conveniently stacked in a matrix and multiplied by the vector of coefficients to obtain the approximation of the input by linear combination.

$$\mathbf{d}_k \approx \begin{bmatrix} \mathbf{f}_{b1}^T & \mathbf{f}_{b2}^T & \cdots & \mathbf{f}_{bN}^T \end{bmatrix} \begin{bmatrix} b_{1,k} \\ b_{2,k} \\ \vdots \\ b_{N,k} \end{bmatrix} \tag{3.47}$$

The benefit of this method is that the number of coefficients needed to be estimated can be significantly reduced, at the cost of making assumptions about how the distributed load behaves spatially.

The basis functions can be obtained either experimentally from data by performing SVD on the spatio-temporal data and selecting the singular vectors associated with spatial evolution, or by using another generic type of basis function associated with a regression method, such as polynomial regression.

3.3.4 Method D

This is the only one of the proposed methods that leaves some forces unestimated. The model used by the method assumes that the distributed load is discretized in bins where the load is constant within each bin. The error is then the error of the approximation.

If the area of the distributed load is divided into rectangular bins with equal width, the location of the equivalent load acts in the middle of the rectangle. The height of the rectangle is the magnitude of the function at the middle point of the rectangle. The smallest area segment that can be done is, in consequence, the area enclosed by

3 nodes. Then a bigger segment will consist of 5 nodes, 7 nodes, 9 nodes, and so forth. More generally, all the segment areas possible consist of the number of nodes given by $N_{area} = 3 + 2\gamma$ where $\gamma \in \mathbb{Z}^+$.

Knowing the size of the area segments the following empirical relationship can also be obtained to check the possible number of subdivisions that can be made given the number of nodes by using one of the subdivisions.

$$N_{subdivisions} = \frac{N_{nodes} - 1}{N_{area} - 1} \tag{3.48}$$

$$N_{subdivisions} = \frac{N_{nodes} - 1}{2(\gamma + 1)} \tag{3.49}$$

where $\gamma \in \mathbb{Z}^+$ and $N_{subdivisions} \in \mathbb{Z}^+$. From this relationship, it is clear that the $N_{subdivisions}$ is restricted and not all combinations of N_{nodes} and γ are valid. In fact, since the number of elements $N_{nodes} - 1$ must be divisible by an even number, the N_{nodes} must be odd.

The method will supposedly increase in how accurately it discretizes the load as many equivalent forces are added. An example could be a beam with 50 elements, which has 51 nodes, and it can be divided into 2, 4, 5, 10, and 20 areas of the same width. A comparison of the response of the 50-element model with 12 modes is shown in Figure 3.2 with the two ways of discretizing the input. The approximation is almost exact at the low frequencies. However, this is only the case if the load is constant or linearly increasing within the bins. If the function changes greatly in a nonlinear fashion, the approximation of the magnitude coming from the difference of the area of the rectangle and the real function over the beam will differ, and the approximation will not be as precise.

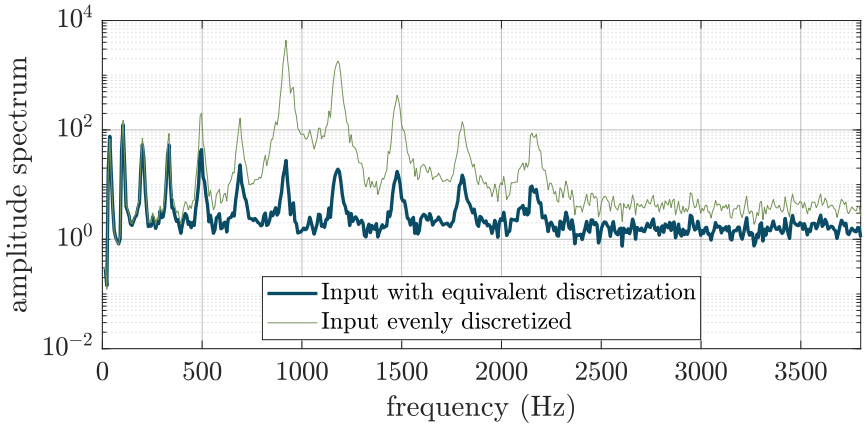


Figure 3.2: Comparison of the spectral content of the measured acceleration of a 50-element model subjected to the load discretized with the energy-consistent discretization against the response to the even discretized load (in 5 loads). Twelve Modes are included in the model.

Test Design 4

To test the suitability of the proposed methods in estimating the distributed load acting on the beam, different test setups must be constructed. The following points are common to all test setups:

- To generate the measurement data, the model with the number of elements found in section 3.1.5 is used to generate an accurate response when the first five modes are considered. It is then subjected to the discretized load using the energy-equivalent method described in section 3.1.4 as it is calculated to produce the same equivalent external work as the original distributed load.
- Different types of load are estimated by combining spatial properties with dynamic properties:
  * Spatial properties: exponentially increasing.
  * Dynamical properties: random smooth dynamics according to a GP (more information in the appendix A.2.2) and white noise dynamics. The smooth dynamics correspond to a more realistic scenario when considering a load like wind, and the white noise dynamics are employed to excite all the modes of the system and analyze its performance on this idealized case, where the chosen modes for the model are greatly excited.
- The system is subjected to different levels of measurement noise. And the short and long term stability is experimentally evaluated.
  * Measurement noise variance: additive noise of 0.1%, 1%, 5% and 10% of the measurements variance.
- Only acceleration measurements are allowed.

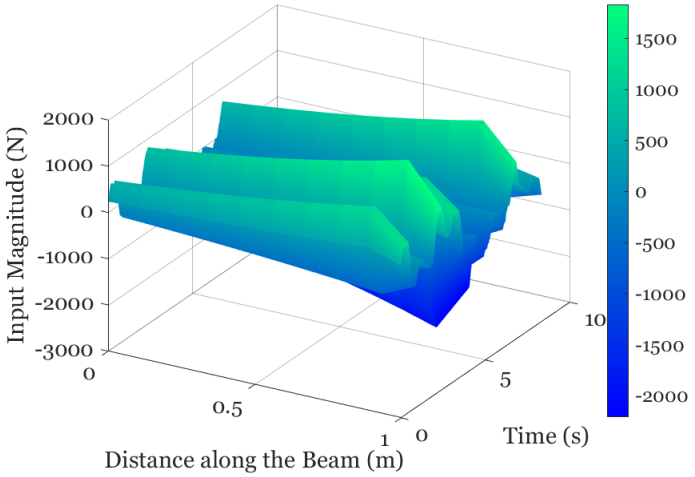


Figure 4.1: Distributed input that behaves spatially as an exponential function and dynamically as a Gaussian Process with Matré covariance with 0.5 length-scale and variance 10.

Four methods are tested, which consist of the three proposed methods, which avoid leaving unestimated forces, and the method where some forces are left unestimated. Then a proper comparison can be made between them to asses which one of the methods performs best in the estimation of dynamic distributed loads. Two acceptance tests are performed to asses if the methods are suitable for the estimation of the two kinds of distributed load discussed with the different noise levels.

- *Acceptance test 1: Can the proposed methods estimate a distributed load with only accelerometer measurements when all modes of the reduced order model are greatly excited? (white noise dynamics)*
  - Check if the acceptance test applies for the following noise conditions:
    (a) $\sigma_v = 0.1\%\sigma_Y$
    (b) $\sigma_v = 1\%\sigma_Y$
    (c) $\sigma_v = 10\%\sigma_Y$
- *Acceptance test 2: Can the proposed methods estimate a distributed load with only accelerometer measurements when all modes of the reduced order model are excited by a dynamically smooth function?*
  - Check if the acceptance test applies for the following noise conditions:
    (a) $\sigma_v = 0.1\%\sigma_Y$
    (b) $\sigma_v = 1\%\sigma_Y$
    (c) $\sigma_v = 10\%\sigma_Y$

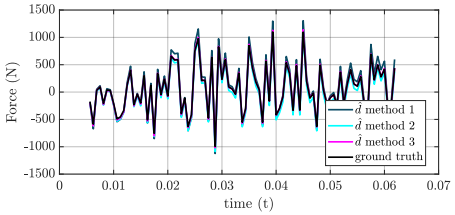
Results 5

A series of experiments is carried out further to explore the strengths and limitations of the proposed methods. Throughout the tests in this experiment, the model that produces the data consists of 4 modes and 12 elements. However, in method 2, the estimator has a model with 4 elements instead. Method 3 assumes that the load behaves spatially as a 3rd degree polynomial.

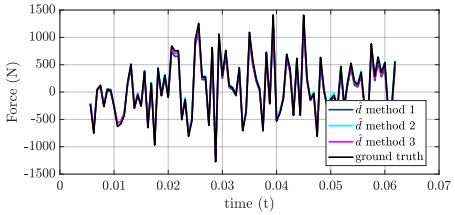
5.1 Acceptance test 1

Comparison of Methods 1, 2, 3

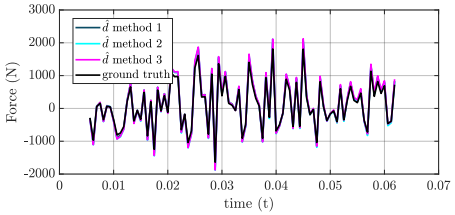
Figures 5.1 5.2 correspond to the cases where the load with white noise dynamics was estimated with noise with a 0.1% and 1% of variance of the measured signal. It is noticeable that estimating the load with white noise dynamics is significantly easier for the estimator, and the estimations are significantly more robust to sensor noise than the case of smooth dynamics.



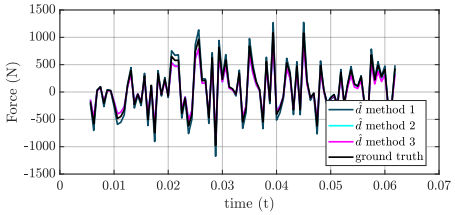
(a) Estimated force at node 1



(b) Estimated force at node 2

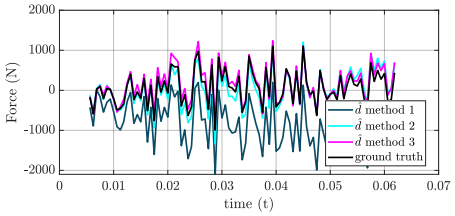


(c) Estimated force at node 3

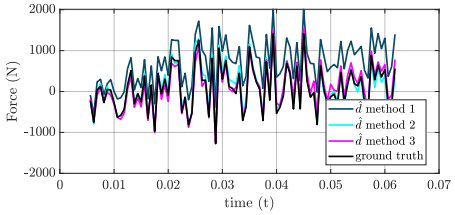


(d) Estimated force at node 4

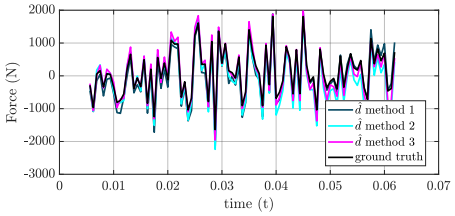
Figure 5.1: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 1\%\sigma_Y$ when subjected to a load with white noise dynamics.



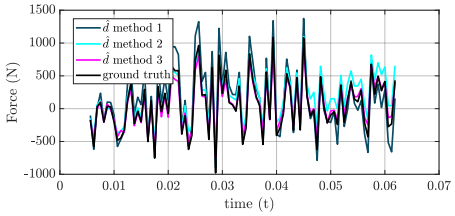
(a) Estimated force at node 1



(b) Estimated force at node 2



(c) Estimated force at node 3



(d) Estimated force at node 4

Figure 5.2: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 10\%\sigma_Y$ when subjected to a load with white noise dynamics.

Evaluation on Method D

This method uses a model with 12 elements and 4 modes, as the model that produced the data, but it only has 4 inputs. The appendix section D.1 contains the figures regarding this test. During the test, a scaling issue appeared in the estimate and after some inspection, it seemed that the resulting estimates have been attenuated by a

factor that corresponds to the number of elements that each bin has. This happens since the estimator estimates the forces of that bin as an aggregate force. Therefore, a factor corresponding to the number of elements of the bin was introduced.

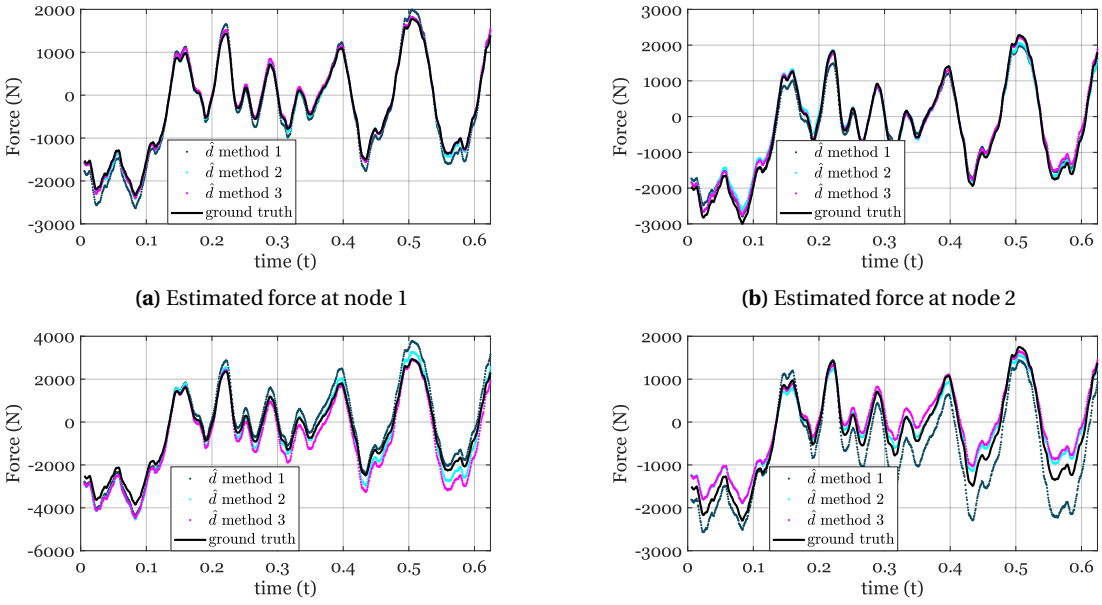
By visual inspection, it can be concluded that the method presents similar robustness to noise as the other methods since it cannot widstand 10% noise, but it can handle 1% and 0.1%.

5.2 Acceptance test 2

The estimators' performance in estimating a smooth load is evaluated with varying levels of noise.

Comparison of Methods 1, 2, 3

Figures 5.3, 5.4 correspond to the cases where the load with smooth dynamics was estimated with noise with a 0.1% and 1% of variance of the measured signal.



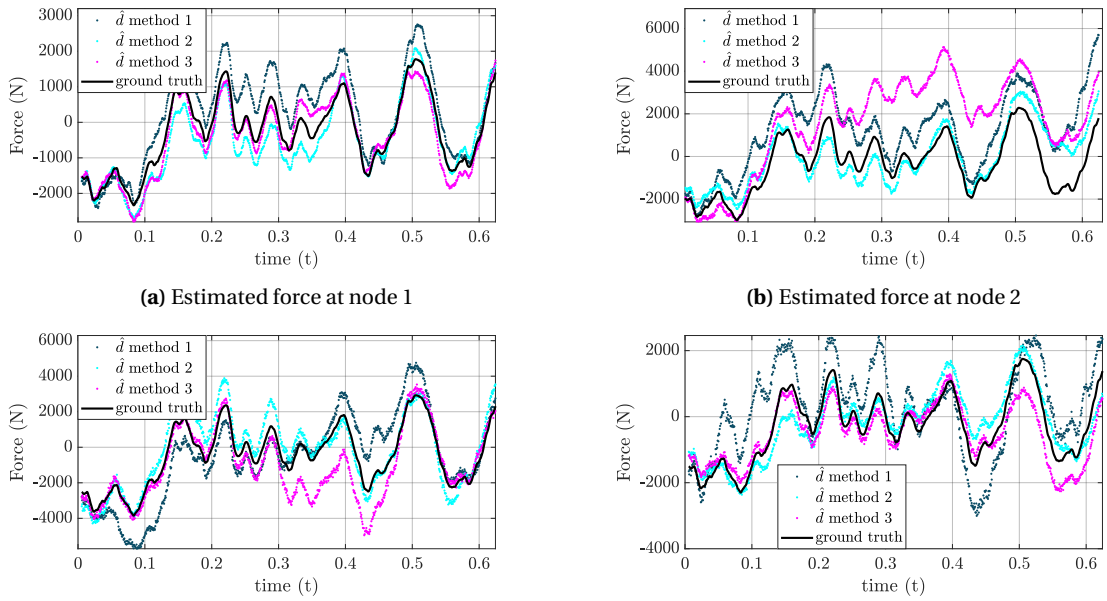
(a) Estimated force at node 1

(b) Estimated force at node 2

(c) Estimated force at node 3

(d) Estimated force at node 4

Figure 5.3: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 0.1\%\sigma_Y$ when subjected to a load with smooth dynamics.



(a) Estimated force at node 1

(b) Estimated force at node 2

(c) Estimated force at node 3

(d) Estimated force at node 4

Figure 5.4: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 1\%\sigma_Y$ when subjected to a load with smooth dynamics.

Out of visual inspection, methods 2 and 3 clearly outperform method 1 in all tests, whereas the most reliable method over the tested measurement noise levels is method 2. The tests also show that if enough measurement noise is present, the estimator can suffer from stability issues in the long term, as shown in figure 5.5.

Evaluation on Method D

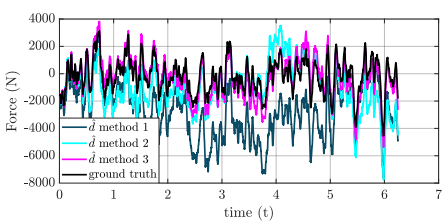
The appendix section D.1 contains the figures regarding this test. The scaling factor is also added here.

Similarly to the other estimators, acceptable estimates can be obtained with noise up to 1%. With mild low-frequency deviations in the long term.

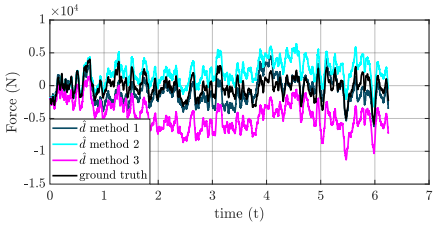
The result of the acceptance test is summarized in Table 5.1.

5.3 Other tests

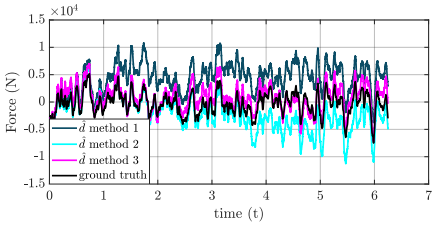
Due to time constraints, a comparison with other estimators, such as an augmented Kalman filter proposed in [7], was not possible. The main reason being the fact that the estimator presented some issues when using accelerometer data only, although



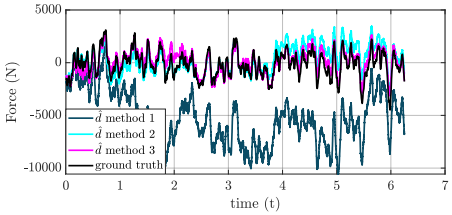
(a) Estimated force at node 1



(b) Estimated force at node 2



(c) Estimated force at node 3



(d) Estimated force at node 4

Figure 5.5: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 0.1\%\sigma_Y$ when subjected to a load with smooth dynamics in a longer period.

Acceptance-test item	**Method 1**	**Method 2**	**Method 3**	**Method D**
1.a $(\sigma_v = 0.1\%\sigma_Y)$	✓	✓	✓	✓
1.b $(\sigma_v = 1\%\sigma_Y)$	✓	✓	✓	✓
1.c $(\sigma_v = 10\%\sigma_Y)$	✗	✓	✓	✓
2.a $(\sigma_v = 0.1\%\sigma_Y)$	✓	✓	✓	✓
2.b $(\sigma_v = 1\%\sigma_Y)$	✗	✓	✗	✗
2.c $(\sigma_v = 10\%\sigma_Y)$	✗	✗	✗	✗

Table 5.1: Pass (✓) or fail (✗) for each acceptance-test item across the four load-estimation methods.

not enough troubleshooting was possible to determine wether it was an issue of the method or of the implementation. In the appendix A.2 the implementation of the augmented Kalman Filter with a latent force modelled as a Gaussian process can be found which includes a section where the stability of the filter when only using accelerometer data is discussed A.2.3. Using the Rauch-Tung-Striebel (RTS) Smoother discussed in appendix A.3, some improved results can be obtained as shown in figure 5.6; but only if position measurements are used, which is not the objective of this work.

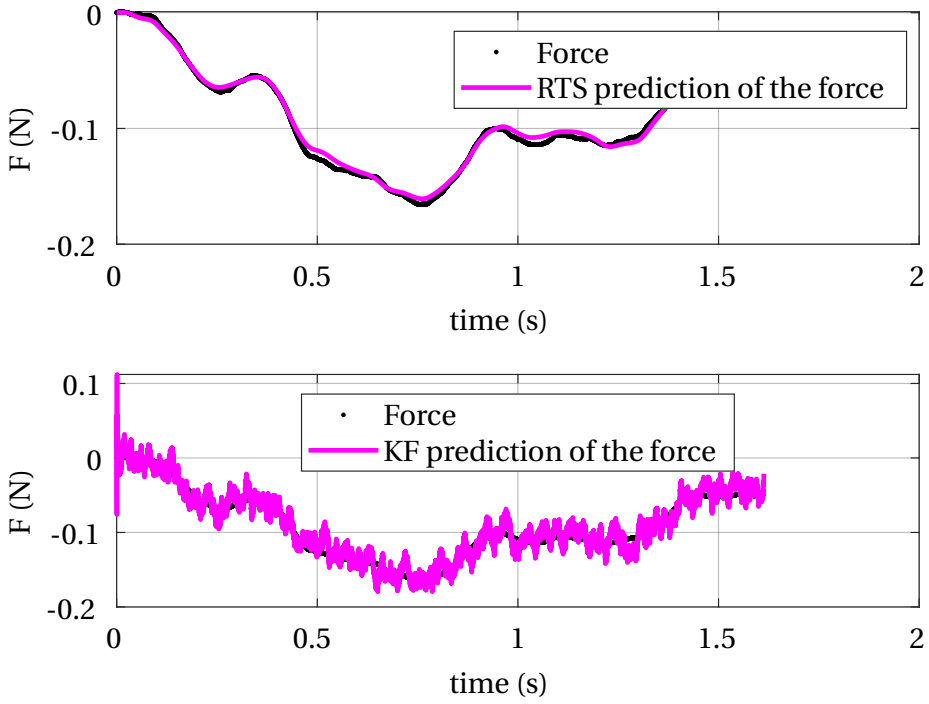


Figure 5.6: Estimation of the force acting on the last node with measurement $\sigma_v = 0.1\%\sigma_Y$. The estimate of the RTS smoother is compared with the performance of the regular augmented Kalman Filter proposed in [7].

Discussion & Conclusion 6

This work designed and implemented some methods, using the GDF, for the estimation of a dynamic distributed load acting on a beam. In total, four methods were proposed and tested with numerical simulations. Generally, the estimators are more robust to measurement noise when the modes used in the model are greatly excited. Out of all methods, Method 2 was the method that passed the most acceptance tests, and specifically, it appears to be more robust to measurement noise in the estimation of the smooth load. More target-specific testing is required with a wind turbine blade model under realistic wind conditions to prove the suitability of the methods for the application.

Future Work 7

There are numerous directions to improve, they can be summarized in the following points:

- Optimization of sensor placement if the method allows it. Specifically, methods 1, 3, and D give freedom to the user to place the sensors on the beam.
- Optimizing the choice of basis functions for method 3.
- Test with a realistic simulation of a wind turbine blade and simulated wind.
- Exploiting spatial correlation in the KF and trying other methods.
- Exploring how other input-state estimation methods handle distributed load estimation,
- Making a comparison of other methods and the GDF to asses which one is more successful.

Bibliography

[1] F Sayer et al. “ReliaBlade project: A material’s perspective towards the digitalization of wind turbine rotor blades”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 942. 1. IOP Publishing. 2020, p. 012006.

[2] Silvia Vettori et al. “An adaptive-noise Augmented Kalman Filter approach for input-state estimation in structural dynamics”. In: *Mechanical Systems and Signal Processing* 184 (2023), p. 109654.

[3] J Kullaa. “Bayesian virtual sensing in structural dynamics”. In: *Mechanical Systems and Signal Processing* 115 (2019), pp. 497–513.

[4] Frank Naets, Javier Cuadrado, and Wim Desmet. “Stable force identification in structural dynamics using Kalman filtering and dummy-measurements”. In: *Mechanical Systems and Signal Processing* 50 (2015), pp. 235–248.

[5] Alexandros Iliopoulos et al. “A modal decomposition and expansion approach for prediction of dynamic responses on a monopile offshore wind turbine using a limited number of vibration sensors”. In: *Mechanical Systems and Signal Processing* 68 (2016), pp. 84–104.

[6] E Lourens et al. “An augmented Kalman filter for force identification in structural dynamics”. In: *Mechanical systems and signal processing* 27 (2012), pp. 446–460.

[7] Rajdip Nayek, Souvik Chakraborty, and Sriram Narasimhan. “A Gaussian process latent force model for joint input-state estimation in linear structural systems”. In: *Mechanical Systems and Signal Processing* 128 (2019), pp. 497–530.

[8] Steven Gillijns and Bart De Moor. “Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough”. In: *Automatica* 43.5 (2007), pp. 934–937.

[9] Saeed Eftekhar Azam, Eleni Chatzi, and Costas Papadimitriou. “A dual Kalman filter approach for state estimation via output-only acceleration measurements”. In: *Mechanical systems and signal processing* 60 (2015), pp. 866–886.

[10] E Lourens et al. “Joint input-response estimation for structural systems based on reduced-order models and vibration data from a limited number of sensors”. In: *Mechanical Systems and Signal Processing* 29 (2012), pp. 310–327.

[11] Bingen Yang. “Stress, Strain, and Structural Dynamics”. In: *An Interactive Handbook of Formulas, Solutions, and MATLAB Toolboxes* (2005).

[12] Javier Cara. “Computing the modal mass from the state space model in combined experimental–operational modal analysis”. In: *Journal of Sound and Vibration* 370 (2016), pp. 94–110.

[13] Daryl L Logan. *A first course in the finite element method*. Vol. 4. Thomson, 2011.

[14] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2006.

[15] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[16] Jouni Hartikainen and Simo Särkkä. “Kalman filtering and smoothing solutions to temporal Gaussian process regression models”. In: *2010 IEEE international workshop on machine learning for signal processing*. IEEE. 2010, pp. 379–384.

[17] Carl Edward Rasmussen. “Gaussian processes in machine learning”. In: *Summer school on machine learning*. Springer, 2003, pp. 63–71.

Appendix A - Background Theory A

A.1 Modal Analysis - Derivations

First, the undamped and unforced response of the system is analyzed. This corresponds to setting the damping and external force to zero on the original governing equation 3.1 which results in equation A.1.

$$[M]\{\ddot{u}(t)\} + [K]\{u(t)\} = 0 \tag{A.1}$$

Finding a basis where each state is decoupled from each other, and therefore, orthogonal to the matrix [M] and [K] is achieved by solving the generalized eigenvalue problem.

$$[K]\phi = \lambda[M]\phi \tag{A.2}$$

where the eigenvectors ϕ (or mode shapes) form the desired modal basis. In fact, there is a physical interpretation of the obtained eigenvalues and eigenvectors.

Assume a harmonic solution to the undamped, unforced differential equation of the form

$$u(t) = \phi e^{j\omega t}, \tag{A.3}$$

where ϕ is a constant vector (the mode shape) and ω is the frequency of oscillation. For the undamped, unforced system, the governing equation is

$$[M]\,\ddot{u}(t) + [K]\,u(t) = 0. \tag{A.4}$$

Substitute $u(t) = \phi e^{j\omega t}$ into this equation. Noting that

$$\ddot{u}(t) = -\omega^2 \phi e^{j\omega t}, \tag{A.5}$$

we obtain:

$$[M](-\omega^2\phi e^{j\omega t}) + [K](\phi e^{j\omega t}) = 0. \tag{A.6}$$

Factoring out the nonzero term $e^{j\omega t}$ yields:

$$-\omega^2[M]\phi + [K]\phi = 0, \tag{A.7}$$

which can be rearranged as:

$$[K]\phi = \omega^2[M]\phi. \tag{A.8}$$

Comparing this result with the generalized eigenvalue in equation A.2, we see that

$$\lambda = \omega^2. \tag{A.9}$$

Thus, the eigenvalues λ are the squares of the natural frequencies ω of the system. This shows that each mode shape ϕ corresponds to a natural frequency, providing a direct physical interpretation: when the system is disturbed, it oscillates naturally at these frequencies, and each mode shape describes the corresponding pattern of vibration.

A.1.1 UMV with Correlated Process and Measurement Noise

The filter explained in this section is optimal under the assumption that the process noise and the measurement noise are uncorrelated with zero mean among other assumptions. However, in some systems, these two noise sources are correlated.

Starting from the UMV kalman filter. A de-correlating framework can be used to find a representation where the process noise is uncorrelated from the measurement noise. Such an approach is explained in [15]. This model can be used in the time update step, that way, with the process noise successfully uncorrelated the remaining steps of the KF remain the same.

The dynamic model is modified by introducing a term; that does not violate the equality since $y_k - C_k x_k - D_k d_k - v_k = 0$.

$$x_{k+1} = A_k x_k + G_k d_k + w_k + T_k\left(y_k - C_k x_k - D_k d_k - v_k\right) \tag{A.10}$$

Furthermore, it can be modified so that a new definition of w_k is uncorrelated with the measurement noise v_k with a proper choice of T_k. This new definition of w_k is denoted as w_k^*. The representation of the pseudo-process is

$$x_{k+1} = A_k^* x_k + w_k^* + T_k y_k + G_k^* d_k \tag{A.11}$$

where

$$\begin{aligned} A_k^* &= A_k - T_k C_k, \\ w_k^* &= w_k - T_k v_k, \\ G_k^* &= G_k - T_k D_k \end{aligned} \tag{A.12}$$

A T_k that eliminates the correlation $\mathbb{E}[w_k^* v_k^T] = 0$. Given

$$\mathbb{E}[w_k^* v_k^T] = \mathbb{E}[(w_k - T_k v_k) v_k^T] = S - T_k R_k, \tag{A.13}$$

must be

$$T_k = S_k R_k^{-1} \tag{A.14}$$

That completes the definition of the uncorrelated process noise w_k^* with covariance matrix

$$\mathbb{E}[w_k^* w_k^{*T}] = \mathbb{E}[(w_k - T_k v_k)(w_k - T_k v_k)^T] = Q - T_k R_k T_k^T = Q - S_k R_k^{-1} S_k^T \tag{A.15}$$

The derivation of the covariance matrices in equation A.21 and A.23 can be found in the appendix in equations C.1 and C.2. A defined T_k also completes the definition of the pseudo-process where the process noise is uncorrelated from the measurement noise.

Finally, the modified time step of the UMV Kalman filter can be presented

$$\hat{x}_{k|k-1} = A_{k-1}^* \hat{x}_{k-1|k-1} + G_{k-1}^* \hat{d}_{k-1} + T_{k-1} y_{k-1} \tag{A.16}$$

with a modified error covariance matrix that includes the uncorrelated error w_k^* instead

$$\begin{aligned} P_{k|k-1}^x &= \mathbb{E}[\tilde{x}_k \tilde{x}_k^T] \\ &= \begin{bmatrix} A_{k-1}^* & G_{k-1}^* \end{bmatrix} \mathbb{E}[\tilde{x}_{k-1} \tilde{x}_{k-1}^T] \begin{bmatrix} A_{k-1}^{*T} \\ G_{k-1}^{*T} \end{bmatrix} + \mathbb{E}[w_k^* w_k^{*T}] \\ &= \begin{bmatrix} A_{k-1}^* & G_{k-1}^* \end{bmatrix} \begin{bmatrix} P_{k|k-1}^x & P_k^{xd} \\ P_k^{dx} & P_k^d \end{bmatrix} \begin{bmatrix} A_{k-1}^{*T} \\ G_{k-1}^{*T} \end{bmatrix} + Q - S_k R_k^{-1} S_k^T. \end{aligned} \tag{A.17}$$

The full derivation can be found in appendix equation C.3.

A.1.2 GDF with Correlated Process and Measurement Noise - With unkown forces as process noise

The filter explained in this section is optimal under the assumption that the process noise and the measurement noise are uncorrelated with zero mean among other assumptions. However, in some systems, these two noise sources are correlated.

Starting from the UMV kalman filter. A de-correlating framework can be used to find a representation where the process noise is uncorrelated from the measurement noise. Such an approach is explained in [15]. This model can be used in the time update step, that way, with the process noise successfully uncorrelated the remaining steps of the KF remain the same.

The dynamic model is modified by introducing a term; that does not violate the equality since $y_k - C_k x_k - D_k d_k - v_k = 0$.

$$x_{k+1} = A_k x_k + G_k d_k + \bar{G}_k w_k + T_k \left(y_k - C_k x_k - D_k d_k - v_k\right) \tag{A.18}$$

Furthermore, it can be modified so that a new definition of w_k is uncorrelated with the measurement noise v_k with a proper choice of T_k. This new definition of w_k is denoted as w_k^*. The representation of the pseudo-process is

$$x_{k+1} = A_k^* x_k + w_k^* + T_k y_k + G_k^* d_k \tag{A.19}$$

where

$$\begin{aligned} A_k^* &= A_k - T_k C_k, \\ w_k^* &= \bar{G} w_k - T_k v_k, \\ G_k^* &= G_k - T_k D_k \end{aligned} \tag{A.20}$$

A T_k that eliminates the correlation $\mathbb{E}[w_k^* v_k^T] = 0$. Given

$$\mathbb{E}[w_k^* v_k^T] = \mathbb{E}[(\bar{G}_k w_k - T_k v_k) v_k^T] = \bar{G}_k S_k - T_k R_k, \tag{A.21}$$

must be

$$T_k = \bar{G}_k S_k R_k^{-1} \tag{A.22}$$

That completes the definition of the uncorrelated process noise w_k^* with covariance matrix

$$\mathbb{E}[w_k^* w_k^{*T}] = \mathbb{E}[(\bar{G} w_k - T_k v_k)(\bar{G} w_k - T_k v_k)^T] = \ldots \tag{A.23}$$

The derivation of the covariance matrices in equation A.21 and A.23 can be found in the appendix in equations C.1 and C.2. A defined T_k also completes the definition of

the pseudo-process where the process noise is uncorrelated from the measurement noise.

Finally, the modified time step of the UMV Kalman filter can be presented

$$\hat{x}_{k|k-1} = A^*_{k-1}\hat{x}_{k-1|k-1} + G^*_{k-1}\hat{d}_{k-1} + T_{k-1}y_{k-1} \tag{A.24}$$

with a modified error covariance matrix that includes the uncorrelated error w^*_k instead

$$\begin{aligned} P^x_{k|k-1} &= \mathbb{E}[\tilde{x}_k\,\tilde{x}_k^T] \\ &= \begin{bmatrix} A^*_{k-1} & G^*_{k-1} \end{bmatrix} \mathbb{E}[\tilde{x}_{k-1}\,\tilde{x}_{k-1}^T] \begin{bmatrix} A^{*T}_{k-1} \\ G^{*T}_{k-1} \end{bmatrix} + \mathbb{E}[w^*_k\,w^{*T}_k] \\ &= \begin{bmatrix} A^*_{k-1} & G^*_{k-1} \end{bmatrix} \begin{bmatrix} P^x_{k|k-1} & P^{xd}_k \\ P^{dx}_k & P^d_k \end{bmatrix} \begin{bmatrix} A^{*T}_{k-1} \\ G^{*T}_{k-1} \end{bmatrix} + Q - S_k R_k^{-1} S_k^T. \end{aligned} \tag{A.25}$$

The full derivation can be found in the appendix equation C.3. Moreover, in the appendix C, the equations for the case where the process noise consists of the unestimated are found and derived.

A.2 Augmented State Framework for Input Estimation

Another method that could be used to estimate the input forces is to include them as unknown states instead. This method also features the possibility of embedding assumptions about the dynamics of the input if they are known.

A.2.1 Force without Known Dynamics

When the dynamics are unknown, the random walk assumption is usually imposed on them. That means that a random process drives the load dynamics. For simplicity, a zeroth-order random walk is used where each force has the following dynamics $\dot{f} = w_f$ where w_f is assumed to be a Gaussian random process with zero mean and variance σ_f^2.

A.2.2 Force with Dynamic Model

In [16], a method was presented to get a state space representation of a process with a particular stationary covariance matrix. Assuming that each force follows dynamics such that its stationary covariance matrix is equal to the Matérn family of covariance functions of specified smoothness. The state-space equations for the Matern process with $p = 1$ are:

$$\frac{d\mathbf{x}(t)}{dt} = A\mathbf{x}(t) + Bw(t)$$

$$A = \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix}, \quad L = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

where:

- $\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ is the state vector
- $w(t)$ is white noise with spectral density q given by [7]:

$$q = \frac{12\sqrt{3}\sigma^2}{l^3}$$

- $\lambda = \sqrt{3}/\ell$ is the characteristic frequency (inverse length-scale)
- σ^2 is the variance parameter
- ℓ is the length-scale parameter

Each force is modeled using this state space formulation, which means that the state vector is composed of the force and its rate of change.

An augmented state space can be obtained by combining the system's states and the force-related states into a combined state vector.

$$\mathbf{x} = \begin{bmatrix} \mathbf{z} \\ f_1 \\ \dot{f}_1 \\ \vdots \\ f_n \\ \dot{f}_n \end{bmatrix}$$

The augmented system given by [7] with the aforementioned dynamic model for the forces. The augmented state matrix A_{a} is organized in the following way:

$$A_{\text{a}} = \left(\begin{array}{c|c} \text{Modal state matrix } \tilde{T}^{-1}A\tilde{T} & \text{Force coupling} \\ z \in \mathbb{R}^{2n} & \tilde{T}^{-1}BS_f \in \mathbb{R}^{2n\times u} \\ \hline \text{Zero block} & \text{Force dynamics} \\ \mathbf{0} \in \mathbb{R}^{2u\times 2n} & A_F \in \mathbb{R}^{2u\times 2u} \end{array}\right)$$

Where:

- $z \in \mathbb{R}^{2n}$ represents all modal states (both real and imaginary parts)
- $\tilde{T}^{-1}A\tilde{T} \in \mathbb{R}^{2n\times 2n}$ is the block-diagonal modal state transition matrix in continuous time, but after the transformation that turns it into real states.
- $\tilde{T}^{-1}BS_f \in \mathbb{R}^{2n\times d}$ is the real part of the input-to-modal coupling
- $S_f \in \mathbb{R}^{d\times 2d}$ selects only the force components (not their derivatives)
- $A_F \in \mathbb{R}^{2d\times 2d}$ is the block-diagonal force dynamics matrix (Matern p=1)
- n = number of modes, d = number of inputs

After its construction, it is discretized using the ZOH assumption.

The C_{aug} matrix must also be expanded to handle the new states by concatenating the previously defined $\bar{C}$ and $\bar{D}$ and excluding the derivative of the forces with S_f. Then the D_{aug} matrix is set to zero. This essentially constitutes a reformulation of the sensor model used in the UMV estimator, and therefore it is equivalent mathematically.

$$C_{\text{a}} = \begin{bmatrix} \bar{C} & \bar{D}S_f \end{bmatrix} \tag{A.26}$$

It is important to note that the spatial correlation between the forces is not exploited in this case.

Discretization of the Augmented Kalman filter

Applying exact discretization with the ZOH assumption, the following discrete state space representation is obtained:

$$\begin{aligned} x_k^a &= F_{ad}x_{k-1}^a + w_{k-1}^a \\ y_k &= H_{ad}x_k^a + v_k \end{aligned} \tag{A.27}$$

where:

- $x_k^a \in \mathbb{R}^{2n+2d}$ is the discrete augmented state vector at time k
- $F_{ad} = e^{A_a \Delta t} \in \mathbb{R}^{(2n+2d)\times(2n+2d)}$ is the discrete state matrix
- $H_{ad} = C_a \in \mathbb{R}^{y\times(2n+2d)}$ is the output matrix
- $w_{k-1}^a = \begin{bmatrix} \tilde{w}_{k-1} & w_{k-1} \end{bmatrix}^T$ where the first term is the process noise from the states and the second therm is a discrete zero-mean Gaussian white noise vector for the force dynamics and has a covaraince matrix

$$Q_d = \int_0^{\Delta t} \Psi(\Delta t - \tau) L q L^T \Psi(\Delta t - \tau)^T d\tau \tag{A.28}$$

where $\Psi(\tau) = e^{A_a \tau}$. Therefore, the process noise of the augmented state is given by:

$$Q^a = Q_d + \begin{bmatrix} Q^x & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{A.29}$$

A.2.3 Stability properties of the Augmented Kalman filter

In [7], the stability properties of both the random walk and the Gaussian prior latent force function were discussed. Using the Popov-Belevitch-Hautus (PBH) criterion which states that system is detectable if and only if the PBH matrix

$$PBH = \begin{bmatrix} sI - A \\ C \end{bmatrix} \tag{A.30}$$

is a full column-rank for all $s \in \mathbb{C}$ or the undetectable modes are stable [**<empty citation>**]. Where A and C are the continuous state and output matrices, respectively. This test was performed on both augmented state systems, and the conclusion was that, for only acceleration measurements, the one that uses a Gaussian prior latent force is the only one that is detectable due to the unobservable modes being stable, contrary to the one using the random walk model. Detectability is sufficient for the continuous algebraic Ricatti equation A.31 to have at least one

positive semidefinite solution P, resulting in a steady-state stable Kalman filter [15]. Since the Gaussian prior latent force is proven detectable, a stable Kalman filter is granted by using this method.

$$-PC^T R^{-1} CP + AP + PA^T + Q = 0 \tag{A.31}$$

A.3 Fixed-Interval Smoothing

The standard Kalman filter estimates the state at a time step using measurements up to the current time step. However, better estimation results can be obtained if future measurements are included. This is the idea behind optimal smoothing. Depending on the type of problem, a different algorithm is required; they can be categorized into [15]:

- Fixed-point smoother: Where the state at a fixed time step is constantly refined with incoming measurements.
- Fixed-lag smoother: The system's state is estimated with a fixed time lag in a sliding window fashion. In other words, in every time step, the state is estimated including N future samples.
- Fixed-interval smoother: Measurements are taken during a time interval, and every state is estimated using all the available measurements.

In this work, a fixed-interval smoother called the Rauch, Tung, and Striebel (RTS) smoother is performed offline after sufficient data is collected. Conceptually, the fixed-interval smoother combines estimates of two independent estimators, one performing a forward pass and another performing a backward pass. The forward pass is performed by a standard Kalman filter, while the backward pass is handled by a separate Kalman filter operating in reverse time. That's exactly the idea behind the forward-backward smoothing, however, the RTS smoother performs an equivalent process while avoiding the explicit computations of the backwards estimates and covariances, which results in improved computational efficiency.

It is worth noting that the estimation's improvement due to smoothing is greater the smaller the measurement noise [15].

A.4 Gaussian Process Regression

Finding or approximating the underlying function that produced the observed data is typically done with regression. In many cases, this means assuming a certain parametric function form and finding the parameters that best fit the model. However, parametric methods do not encode the uncertainty of the predictions made by the model and its form is limited to the chosen model. Gaussian process regression is a method that not only has more flexibility in the regression model due to being non-parametric, but also gives an estimate of the uncertainty of its predictions.

Since the underlying model that produced the data is unknown, when assuming a certain parametric functional form, the uncertainty in the model is transferred to the parameters themselves. One can try to maximize the likelihood which is the probability of the data given the weights $p(X, w)$, or the a posterior probability which is the probability distribution over the weights given the data $p(w, x)$. However, there is a way to not focus on the probability distribution over the parameters but to focus on the probability distribution over the functions themselves. The dependency of the regression problem on the parameters is eliminated in the Bayesian approach by marginalizing w out when finding directly the posterior probability $P(y|x_{\text{test}}, X)$ of the data at the test points given the observed data. Equation A.32 shows that the posterior is, in fact, Gaussian if the prior is Gaussian. The GPR places a prior over functions, assuming that function values follow a multivariate Gaussian distribution. Instead of finding a single best function, we compute the posterior distribution over all possible functions given the data. Then the mean function with confidence intervals can be calculated when making predictions.

$$P(y|x_{\text{test}}, X) = \int_{\mathbf{w}} P(y|x_{\text{test}}, \mathbf{w}) P(\mathbf{w}|X) d\mathbf{w} = N(\mu, \Sigma) \tag{A.32}$$

This means that each sample from the multivariate Gaussian distribution corresponds to an entire function, where the number of evaluation points matches the dimensions of the Gaussian distribution. Although, in theory, the Gaussian process is defined over an infinite-dimensional space, since a continuous function has infinitely many points, we only evaluate it at a finite set of selected points. In essence, we are sampling an entire function from the underlying stochastic process.

Deciding on the mean function and covariance structure of the prior influences the characteristics of the functions sampled from the Gaussian process. In particular, the covariance matrix, which defines how function values correlate across input points, is determined by the choice of kernel. For instance, using a Radial Basis Function (RBF) kernel results in smooth functions, as it encodes the assumption that function values at nearby points are highly correlated.

Finally, the posterior can be updated when new evidence or data points are found.

A.4.1 Function-space Interpretation

First, the priory distribution over functions must be defined. Usually for notational simplicity a zero mean function assumption over the distribution is taken; which does not mean that the functions of the distribution have zero mean, it means that if all the values of those functions are averaged out at a particular point the result would be zero [17]. Then the covariance matrix is specified using a kernel $k(x, x')$, which will encode the properties of the functions that will be sampled.

$$f \sim GP(0, k(x, x')) \tag{A.33}$$

An example of such a process could be a simple Bayesian Linear Regression model $f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w}$ where the transformation $\phi(\mathbf{x})$ can project the input into a higher dimensional space so that more interesting functions than a liner function can be obtained. And with prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$.Note that it is still linear with respect to the parameters $\mathbf{w}$ [17]. It can also be viewed as a linear combination of basis functions.

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^T\mathbb{E}[\mathbf{w}] = 0 \tag{A.34}$$

$$\mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] = \phi(\mathbf{x})^T\mathbb{E}[\mathbf{w}\mathbf{w}^T]\phi(\mathbf{x}') = \phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}') \tag{A.35}$$

Since working with infinite variables is impractical, a sample vector from the distribution of functions can be defined and the covariance matrix can be obtained element-wise from the kernel.

$$\mathbf{f}_{test} \sim \mathcal{N}((\mathbf{0}, K(X_{test}, X_{test}))) \tag{A.36}$$

If some noisy data is obtained and modeled as $y = f(\mathbf{x}) + \varepsilon$, where ε is independent identically distributed Gaussian noise with variance σ_n^2. A joint distribution between the known data and the unknown test values whose value we want to find can be stated

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_{test} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 \mathbf{I} & K(X, X_{test}) \\ K(X_{test}, X) & K(X_{test}, X_{test}) \end{bmatrix}) \tag{A.37}$$

To update the prior distribution of functions with the known noisy data observations and obtain a posterior distribution of functions, the joint Gaussian prior distribution must be conditioned on the observations. Due to the joint distribution being Gaussian the conditional probability distribution is also Gaussian and has the following form

$$\mathbf{f}_{test} | X_{test}, X, \mathbf{y} \sim \mathcal{N}(K(X_{test}, X)[K(X,X)+\sigma_n^2\mathbf{I}]^{-1}\mathbf{y}, K(X_{test}, X_{test}) - K(X_{test}, X)[K(X,X)+\sigma_n^2\mathbf{I}]^{-1}K(X, X_{test})) \tag{A.38}$$

Using the formula, the value of the mean function over the posterior distribution can be found for the test points x_{test}.

The algorithm in algorithm 1 from [17] computes the predictive mean and covariance at the test inputs, and it also evaluates the log marginal likelihood, which is essential for hyperparameter optimization. A Cholesky factorization is employed to improve numerical stability during the inversion of the covariance matrix and to efficiently compute the log determinant required for the log marginal likelihood.

A.4.2 Hyperparameter Optimization

The marginal likelihood is obtained by marginalizing over (or integrating out) the latent function (noise-free) $\mathbf{f}$:

$$p(\mathbf{y} \mid X, \theta) = \int p(\mathbf{y} \mid \mathbf{f}, X, \theta)\, p(\mathbf{f} \mid X, \theta)\, d\mathbf{f}, \tag{A.39}$$

where θ represents the set of hyperparameters (e.g., the kernel parameters and the noise variance σ_n^2).

We assume that the observed data $\mathbf{y}$ is the sum of the latent function and Gaussian noise, so that

$$\mathbf{y} \mid \mathbf{f}, X, \theta \sim N(\mathbf{f}, \sigma_n^2 I).$$

Algorithm 1 Predictions and log marginal likelihood for Gaussian process regression

Input: Training inputs X, targets y, covariance function $k(\cdot,\cdot)$, noise variance σ^2, test inputs X_*
Output: Predictive mean f_*, predictive variance $\mathrm{Var}(f_*)$, log marginal likelihood
$K_y \leftarrow k(X,X) + \sigma^2 I$ ▷ Compute the training covariance matrix
$L \leftarrow \mathrm{chol}(K_y)$ ▷ Cholesky factorization: $K_y = LL^\top$
$\alpha \leftarrow L^\top \backslash (L \backslash y)$ ▷ Solve $K_y\alpha = y$ via forward/backward substitution
$f_* \leftarrow k(X_*, X)\,\alpha$ ▷ Predictive mean
$V \leftarrow L \backslash k(X, X_*)$ ▷ Intermediate for predictive variance
$\mathrm{Var}(f_*) \leftarrow k(X_*, X_*) - V^\top V$ ▷ Predictive variance
$\log p(y \mid X) \leftarrow -\frac{1}{2}\, y^\top \alpha - \sum_i \log(L_{ii}) - \frac{n}{2}\log(2\pi)$ ▷ Log marginal likelihood

Similarly, the prior over the latent function is given by a Gaussian process with zero mean and covariance defined by the kernel:

$$\mathbf{f} \mid X, \theta \sim N(\mathbf{0}, K(X, X; \theta)).$$

Since both the likelihood and the prior are Gaussian, the integral in (1) can be performed analytically, yielding a closed-form expression for the marginal likelihood:

$$p(\mathbf{y} \mid X, \theta) = N\Big(\mathbf{y} \,\Big|\, \mathbf{0},\, K(X, X; \theta) + \sigma_n^2 I\Big).$$

Taking the logarithm, we obtain the log marginal likelihood:

$$\log p(\mathbf{y} \mid X, \theta) = -\frac{1}{2}\mathbf{y}^\top \left(K(X, X; \theta) + \sigma_n^2 I\right)^{-1} \mathbf{y} - \frac{1}{2}\log\Big|K(X, X; \theta) + \sigma_n^2 I\Big| - \frac{n}{2}\log(2\pi).$$

This log marginal likelihood is then maximized with respect to the hyperparameters θ (e.g., by gradient-based optimization methods) to find the values that best explain the observed data.

Appendix B - Derivations B

Derivation of the covariance matrix between the modified process and measurement noise

$$\mathbb{E}\big[w_k^* v_k^T\big] = \mathbb{E}\big[(w_k - T_k v_k) v_k\big] = \mathbb{E}\big[w_k v_k^T - T_k v_k v_k^T\big] = \mathbb{E}\big[w_k v_k^T\big] - T_k \mathbb{E}\big[v_k v_k^T\big] = S - T_k R_k \tag{B.1}$$

Derivation of the covariance matrix of the modified process noise

$$\begin{aligned}
\mathbb{E}\big[w_k^* w_k^{*T}\big] &= \mathbb{E}\big[(w_k - T_k v_k)(w_k - T_k v_k)^T\big] \\
&= \mathbb{E}\big[w_k w_k^T\big] - \mathbb{E}\big[w_k (T_k v_k)^T\big] - \mathbb{E}\big[T_k v_k w_k^T\big] + \mathbb{E}\big[T_k v_k (T_k v_k)^T\big] \\
&= \mathbb{E}\big[w_k w_k^T\big] - \mathbb{E}\big[w_k v_k^T\big] T_k^T - T_k \mathbb{E}\big[v_k w_k^T\big] + T_k \mathbb{E}\big[v_k v_k^T\big] T_k^T \\
&= Q - S_k T_k^T - T_k S_k^T + T_k R_k T_k^T \\
&= Q - T_k R_k T_k^T - T_k (T_k R_k)^T + T_k R_k T_k^T \\
&= Q - T_k R_k T_k^T \\
&= Q - S_k R_k^{-1} S_k^T
\end{aligned} \tag{B.2}$$

Derivation of the error covariance matrix of the time step prediction

$$\begin{aligned}
P_{k|k-1}^x &= \mathbb{E}[\tilde{x}_k \tilde{x}_k^T] \\
&= \mathbb{E}[(A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-1})(A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-1})^T] \\
&= \begin{bmatrix} A_{k-1}^* & G_{k-1}^* \end{bmatrix} \begin{bmatrix} \mathbb{E}[\tilde{x}_{k-1} \tilde{x}_{k-1}^T] & \mathbb{E}[\tilde{x}_{k-1} \tilde{d}_{k-1}^T] \\ \mathbb{E}[\tilde{d}_{k-1} \tilde{x}_{k-1}^T] & \mathbb{E}[\tilde{d}_{k-1} \tilde{d}_{k-1}^T] \end{bmatrix} \begin{bmatrix} A_{k-1}^{*T} \\ G_{k-1}^{*T} \end{bmatrix} + \mathbb{E}[w_k^* w_k^{*T}] \\
&= \begin{bmatrix} A_{k-1}^* & G_{k-1}^* \end{bmatrix} \begin{bmatrix} P_{k|k-1}^x & P_k^{xd} \\ P_k^{dx} & P_k^d \end{bmatrix} \begin{bmatrix} A_{k-1}^{*T} \\ G_{k-1}^{*T} \end{bmatrix} + Q - S_k R_k^{-1} S_k^T.
\end{aligned} \tag{B.3}$$

where

$$\tilde{x}_k = x_k - \hat{x}_k = A_k^* x_{k-1} + G_{k-1}^* d_{k-1} + T_{k-1} y_{k-1} + w_{k-1}^* - (A_k^* x_{k-1} + G_{k-1}^* d_{k-1} + T_{k-1} y_{k-1}) = A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-}^*$$

Appendix C - Derivations

C

Derivation of the covariance matrix between the modified process and measurement noise

$$\mathbb{E}\left[w_k^* v_k^T\right] = \mathbb{E}\left[(\bar{G}_k w_k - T_k v_k) v_k^T\right] = \mathbb{E}\left[\bar{G}_k w_k v_k^T - T_k v_k v_k^T\right] = \bar{G}_k \mathbb{E}\left[w_k v_k^T\right] - T_k \mathbb{E}\left[v_k v_k^T\right] = \bar{G}_k S - T_k R_k \tag{C.1}$$

Derivation of the covariance matrix of the modified process noise

$$\begin{aligned}
\mathbb{E}\left[w_k^* w_k^{*T}\right] &= \mathbb{E}\left[(\bar{G}_k w_k - T_k v_k)(\bar{G}_k w_k - T_k v_k)^T\right] \\
&= \bar{G}_k \mathbb{E}\left[w_k w_k^T\right] \bar{G}_k^T - \bar{G}_k \mathbb{E}\left[w_k (T_k v_k)^T\right] - \mathbb{E}\left[T_k v_k w_k^T\right] \bar{G}_k^T + \mathbb{E}\left[T_k v_k (T_k v_k)^T\right] \\
&= \bar{G}_k \mathbb{E}\left[w_k w_k^T\right] \bar{G}_k^T - \bar{G}_k \mathbb{E}\left[w_k v_k^T\right] T_k^T - T_k \mathbb{E}\left[v_k w_k^T\right] \bar{G}_k^T + T_k \mathbb{E}\left[v_k v_k^T\right] T_k^T \\
&= \bar{G}_k Q \bar{G}_k^T - \bar{G}_k S_k T_k^T - T_k S_k^T \bar{G}_k^T + T_k R_k T_k^T \\
&= \bar{G}_k Q \bar{G}_k^T - \bar{G}_k \bar{G}_k^{-1} T_k R_k T_k^T - T_k (\bar{G}_k^{-1} T_k R_k)^T \bar{G}_k^T + T_k R_k T_k^T \\
&= \bar{G}_k Q \bar{G}_k^T - \bar{G}_k \bar{G}_k^{-1} T_k R_k T_k^T - T_k R_k^T T_k \bar{G}_k^{-1\,T} \bar{G}_k^T + T_k R_k T_k^T \\
&= \bar{G}_k Q \bar{G}_k^T - T_k R_k T_k^T - T_k R_k^T T_k + T_k R_k T_k^T \\
&= \bar{G}_k Q \bar{G}_k^T - T_k R_k^T T_k \\
&= \bar{G}_k Q \bar{G}_k^T - \bar{G}_k S_k R_k^{-1} S_k \bar{G}_k^T
\end{aligned} \tag{C.2}$$

Note that the inverse cannot be calculated if the matrix $\bar{G}_k$ is not square.

Derivation of the error covariance matrix of the time step prediction

$$\begin{aligned}
P_{k|k-1}^x &= \mathbb{E}[\tilde{x}_k \tilde{x}_k^T] \\
&= \mathbb{E}[(A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-1}^*)(A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-1}^*)^T] \\
&= \begin{bmatrix} A_{k-1}^* & G_{k-1}^* \end{bmatrix} \begin{bmatrix} \mathbb{E}[\tilde{x}_{k-1} \tilde{x}_{k-1}^T] & \mathbb{E}[\tilde{x}_{k-1} \tilde{d}_{k-1}^T] \\ \mathbb{E}[\tilde{d}_{k-1} \tilde{x}_{k-1}^T] & \mathbb{E}[\tilde{d}_{k-1} \tilde{d}_{k-1}^T] \end{bmatrix} \begin{bmatrix} A_{k-1}^{*T} \\ G_{k-1}^{*T} \end{bmatrix} + \mathbb{E}[w_k^* w_k^{*T}]
\end{aligned} \tag{C.3}$$

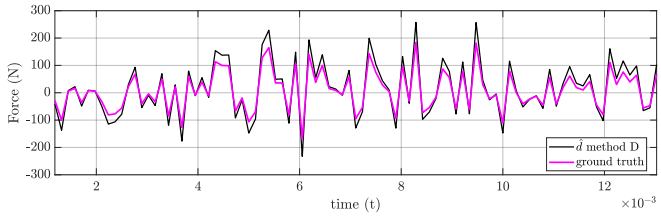
where

$$\tilde{x}_k = x_k - \hat{x}_k = A_k^* x_{k-1} + G_{k-1}^* d_{k-1} + T_{k-1} y_{k-1} + w_{k-1}^* - (A_k^* x_{k-1} + G_{k-1}^* d_{k-1} + T_{k-1} y_{k-1}) = A_k^* \tilde{x}_{k-1} + G_{k-1}^* \tilde{d}_{k-1} + w_{k-}^*$$

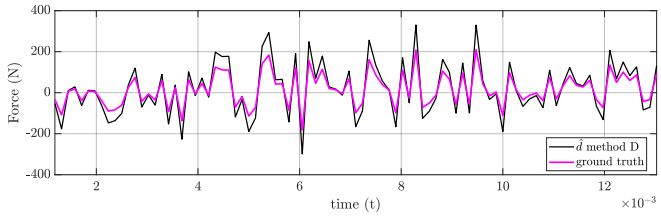
Appendix D - Extra Figures

D

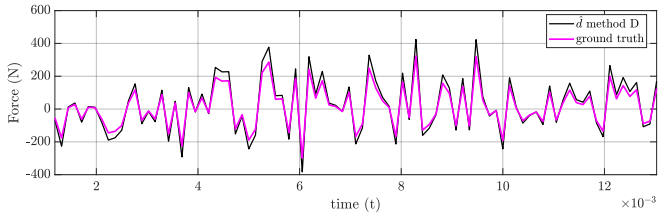
D.1 Estimation of the force with white noise dynamics



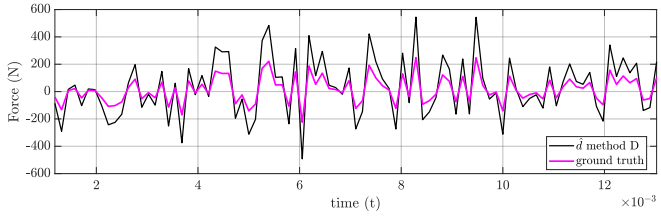
(a) Estimated force at node 1



(b) Estimated force at node 2

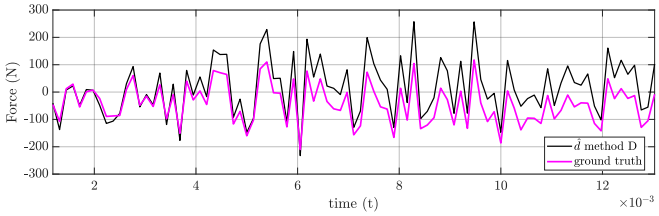


(c) Estimated force at node 3

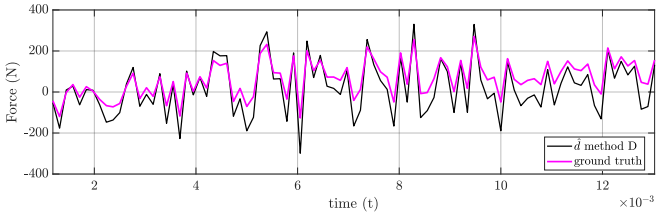


(d) Estimated force at node 4

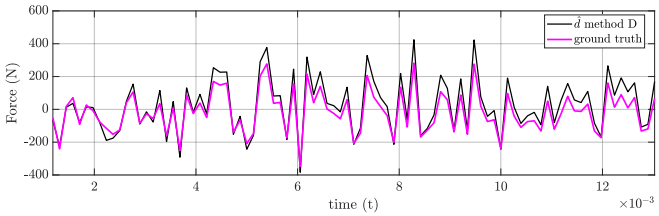
Figure D.1: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 1\%\sigma_Y$



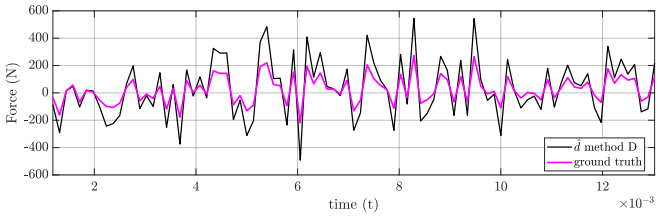
(a) Estimated force at node 1



(b) Estimated force at node 2



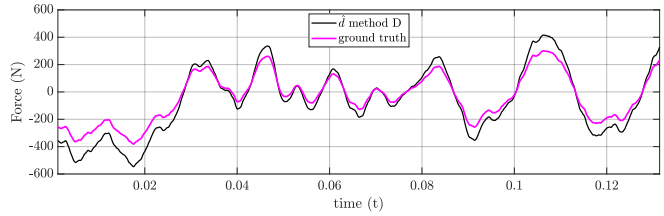
(c) Estimated force at node 3



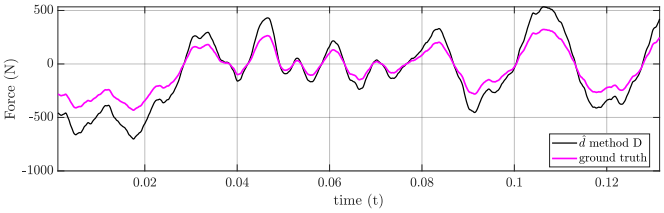
(d) Estimated force at node 4

Figure D.2: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 10\%\sigma_Y$

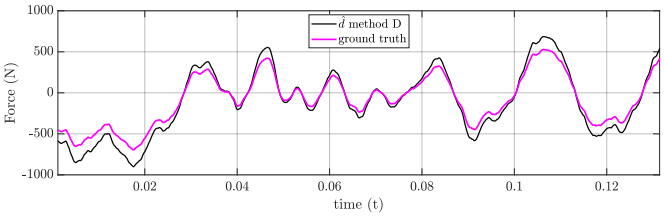
D.2 Estimation of the force with smooth dynamics



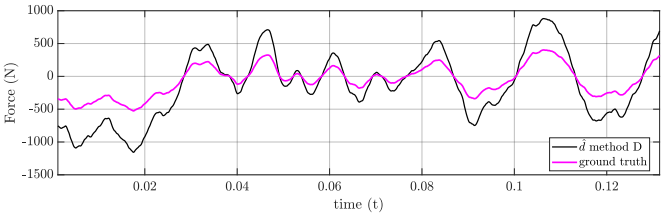
(a) Estimated force at node 1



(b) Estimated force at node 2

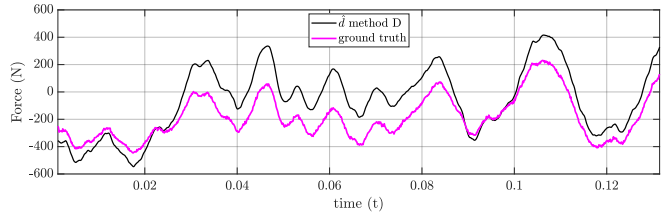


(c) Estimated force at node 3

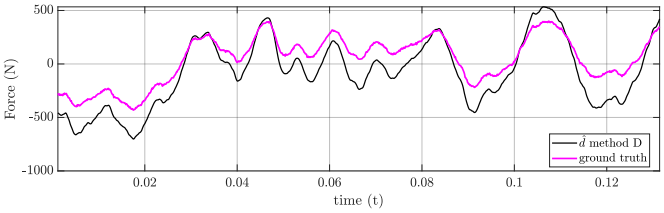


(d) Estimated force at node 4

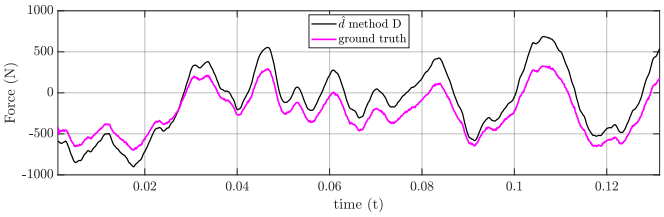
Figure D.3: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 0.1\%\sigma_Y$



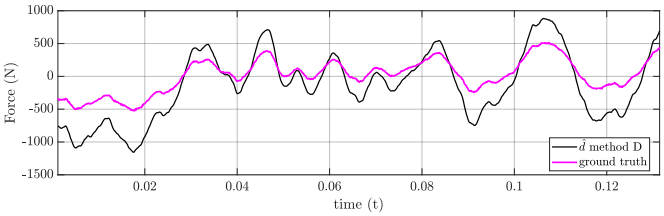
(a) Estimated force at node 1



(b) Estimated force at node 2



(c) Estimated force at node 3



(d) Estimated force at node 4

Figure D.4: Estimation of all free nodal forces of a 4-element beam model with measurement $\sigma_v = 1\%\sigma_Y$