Current Control with Self-Learning Ability for PMSM Drives

Project Report

PED4-1045

Aalborg University
Department of Energy Technology

AALBORG UNIVERSITY
STUDENT REPORT

Department of Energy Technology
Aalborg University
https://www.et.aau.dk/

Title:

Current Control with Self-Learning Ability for PMSM Drives

10th Semester Project:

Master Thesis

Project period:

October 2021 - January 2022

Project group:

PED4 - 1045

Participant:

Rui Cai

Rui Cai

Supervisor:
Kaiyuan Lu

Abstract:

Since firstly proposed in early 1970s, field-oriented-control (FOC) with a typical topology of an inner current control loop and an outer speed control loop control has gained much popularity. The current control is essential in FOC since it directly determines the machine output torque. For the time being, FOC current control is often realized with PI controllers, which have limits pursuing faster current response. To improve this, many research has been made. Among them, predictive current control that utilizes the machine model to calculate in advance the required voltage command has been a promising solution. However, the performance of predictive control is not robust with respect to machine parameter variation. In this paper, an adaptive PID controller with self-learning ability is analyzed and designed for FOC based PMSM drive system. Compared with FOC current control with traditional PI controller, the designed FOC current control with adaptive PID controller not only gives a faster current response, but is also robust against machine parameter variation. The effectiveness of the proposed method is validated through experimental results.

Page count: 24
Appendix count: 0
Completed: 01/2/2022

By signing this document, each member of the group confirms participation on equal terms in the process of writing the project. Thus, each member of the group is responsible for the all contents in the project.

Nomenclature

$\omega_{1\bar{,}2,3}$	Normalized APID weight coefficient 1, 2, 3	$[-]$
χ_1	APID internal variable 1	$[-]$
χ_2	APID internal variable 2	$[-]$
χ_3	APID internal variable 3	$[-]$
Δe	Change of APID input current error during one sampling period	$[A]$
η_D	Step length for differential component	$[-]$
η_I	Step length for integral component	$[-]$
η_P	Step length for proportional component	$[-]$
λ_{mpm}	Peak value of permanent magnet flux linkage	$[Wb]$
ω_1	APID weight coefficient 1	$[-]$
ω_2	APID weight coefficient 2	$[-]$
ω_3	APID weight coefficient 3	$[-]$
ω_r	Rotor angular electrical speed	$[rad/sec]$
τ_d	System delay	$[sec]$
e	APID input current error	$[A]$
f_{sw}	Switching frequency	$[Hz]$
i_{ds}	d-axis current	$[A]$
i_{ds}^*	d-axis current command	$[A]$
i_{qs}	q-axis current	$[A]$
i_{qs}^*	q-axis current command	$[A]$

J	Moment of Inertia	$[mKgm^2]$
K	APID gain	$[-]$
Ki	Classic PI controller integral gain	$[-]$
Kp	Classic PI controller proportional gain	$[-]$
L_d	d-axis Inductance	$[H]$
L_q	q-axis Inductance	$[H]$
n	Noise signal	$[-]$
N_{pp}	Number of pole pairs	$[-]$
p	Differential operator	$[-]$
R_s	Stator resistance for each phase	$[\Omega]$
T_s	Sampling period	$[sec]$
u_{ds}	d-axis stator voltage	$[V]$
u_{qs}	q-axis stator voltage	$[V]$

Acronyms

ADC Analog to Digital Conversion

APID Adaptive Proportional-Integral–Derivative

DC Direct Current

DSP Digital Signal Processor

FOC Field Oriented Control

PI Propotional-Integral

PID Proportional-Integral–Derivative

PM Permanent Magnet

PMSM Permanent Magnet Synchronous Motor

VSI Voltage Source Inverter

Contents

Preface

This report is presented by *PED4 - 1045* at the Department of Energy Technology at Aalborg University. It is a 10^{th} semester project, Master Thesis. Throughout this project following software has been used:

- **Overleaf** - Report writing
- **MATLAB & Simulink** - Control system modelling
- **dSPACE Controldesk** - Laboratory setup controlling
- **Microsoft Visio & Draw i/o** - Figures drawing and editing

Reader's Guide: This project report consists of 6 chapters. Chapter 1 begins with a brief introduction followed by the main objectives and limitations of the project. Chapter 2 offers an analysis of the similar research conducted in the project related field.

The first part of Chapter 3 presents the design and analysis of FOC current loop with traditional PI controller. The rest analyzes the design of FOC current loop with APID controller.

Chapter 4 firstly gives a brief introduction of the test setup and PMSM parameters. Then it compares the current response of a traditional current loop to the designed one with APID controller. Later it validates the proposed current loop with APID controller of its good robustness against parameter change. At the end of this chapter, it concludes an experience-based tuning guidance.

Chapter 5 and Chapter 6 gives the conclusion based on the experimental results and future work respectively.

Introduction

1.1 Background

The present and future market for motors places high value on power density, operating efficiency, reliability, variable speed operation and low cost [1]. Permanent Magnet (PM) motors thus play a more important role in nowadays drive systems. As for control method, Field Oriented Control (FOC) proposed firstly in early 1970s has gained a wide popularity [2].

The typical control topology for FOC is composed with an inner current loop and an outer speed loop. In this manner, the current loop is essential since it regulates the machine output torque. The current loop is often realized with Propotional-Integral (PI) controller. This control topology has been well developed for the passing 30 years and shows satisfying performance in many applications.

However riding with the improvement in magnet technology, PMs are made able to withstand higher temperature without demagnetization [1]. This allows the drive system to run at higher speed, but also raise a claim for the current controller being able to react faster regardless of machine parameter variation due to high temperature.

To improve the current transient response, predictive control method that makes use of machine model to calculate in advance the required voltage has been a promising solution. However, this method is not robust if the machine parameter varies.

To take one step further, this project investigates into an adaptive PID controller with self-learning ability for FOC controlled PMSM drive system. With self-learning ability, this type of smart PID controller is able to modify its parameter online with respect to variation of machine parameters. In this way, this adaptive PID controller is hopefully to improve the current transient response with good robustness against machine parameter variation.

1.2 Problem statement

The goal of this project is the implementation of an adaptive PID controller with proper self-learning rule which gives better performance during transient under varying machine parameters. The main research question for the project is:

How can an adaptive PID controller with self-learning ability for field-oriented-controlled PMSM be designed and utilized so as to improve the transient current response against machine parameter change during operation?

1.2.1 Project Objectives

The main objectives required to achieve the goal of this project:

- To implement the classic Propotional-Integral (PI) controller for FOC current loop.
- To investigate and implement the adaptive PID with proper self-learning rule via simulation.
- To test the designed adaptive PID controller regarding reasonable variation in machine parameters.
- Compare the results for both methods and draw conclusion, make recommendations on general design guidelines.

1.2.2 Project Limitations

Following are the limitations that have been made in the project.

- Speed control loop for FOC is neglected in this project since current control performance is the focus.
- Effect of system delay is not taken into consideration.

State of the Art

In order to improve the transient of current loop response, much effort has been put into research. In [3], a comparative study has been made on three kinds of predictive control methods for current loop, together with theoretical analysis, simulation and experimental validations. However, two of the compared predictive control methods presented in [4] and [5] are based on inverter side, which is beyond topic of this project. In this regard, only predictive current control based on machine model will be introduced in this chapter.

2.1 Model Predictive Current Control

For PMSM, stator voltage equations could be described as follows[6]:

$$\begin{bmatrix} u_{ds} \\ u_{qs} \end{bmatrix} = \begin{bmatrix} R_s + pL_d \\ R_s + pL_q \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \end{bmatrix} + \begin{bmatrix} -\omega_r L_q i_{qs} \\ \omega_r L_d i_{ds} + \omega_r \lambda_{mpm} \end{bmatrix} \tag{2.1}$$

where u_{ds}, u_{qs}, i_{ds}, i_{qs} are the stator $d-axis$ and $q-axis$ voltages and currents respectively; p is differential operator; L_d and L_q are the $d-axis$ and $q-axis$ inductance; λ_{mpm} is the peak value of rotor permanent magnet flux linkage.

During each sampling period, the rotor speed ω_r and PM flux linkage λ_{mpm} can be regarded as constant value [7]. Thus at k-th sampling period, the the discrete-time stator voltage equation is,

$$\begin{bmatrix} u_{ds}(k) \\ u_{qs}(k) \end{bmatrix} = \begin{bmatrix} R_s i_{ds}(k) + \frac{L_d}{T_s}[i_{ds}(k+1) - i_{ds}(k)] - \omega_r L_q i_{qs} \\ R_s i_{qs}(k) + \frac{L_q}{T_s}[i_{qs}(k+1) - i_{qs}(k)] + \omega_r L_d i_{ds} + \omega_r \lambda_{mpm} \end{bmatrix} \tag{2.2}$$

where T_s is system sampling period.

Now considering stator current required at $(k+1)$ sampling period, the corresponding stator voltage command could be calculated using equation 2.2 ,

$$\begin{bmatrix} u_{ds}(k) \\ u_{qs}(k) \end{bmatrix} = \begin{bmatrix} R_s i_{ds}(k) + \frac{L_d}{T_s}[i^*_{ds}(k+1) - i_{ds}(k)] - \omega_r L_q i_{qs} \\ R_s i_{qs}(k) + \frac{L_q}{T_s}[i^*_{qs}(k+1) - i_{qs}(k)] + \omega_r L_d i_{ds} + \omega_r \lambda_{mpm} \end{bmatrix} \tag{2.3}$$

where $i^*_{ds}(k+1)$ and $i^*_{qs}(k+1)$ are d,q-axis current reference at $(k+1)$'s moment respectively. It can be seen that in order to make $(k+1)th$ d,q-axis current measurements follow the command signals $i^*_{ds}(k+1)$ and $i^*_{qs}(k+1)$, a stator voltage can be calculated using equation 2.3 and applied at k-th sampling period. Above illustrates the basic idea of model predictive current control. However, it should be noted that the stator voltage command calculations relies on accurate machine parameters. In conditions that machine parameter could change, this method might lose accuracy and the current response even has a steady state error.

Besides model predictive control, research is also made in area of self-adaptive PID controllers. Most of them are using least square method in [8], [9]. However, they are made only offline. That is, tests need to be carried out to perform least square method and then the PID controller can be improved with more proper gains. These methods give a inspiration for this project to find an online self-adaptive method, which will be further discussed in design section.

Traditional PI vs. APID Design

Firstly in this chapter, control topology of a traditional q-axis current loop for FOC is shown, followed with its basic tuning method (zero-pole cancellation based). In the next section 3.2, a detailed analysis and design procedure for APID controller is illustrated. It is worth mentioning here that the classical PI controller will later be used as a comparison for experimental result discussion in Chapter. 4.

3.1 Traditional Current Loop with PI

The classical current loop design for FOC makes use of two PI controllers in order to regulate d and q-axis currents. The control topology for both d and q-axis is same while only controller parameter may be different to meet different performance requirements. Thus in the later part a current loop design for q-axis is taken as an example.

However, it should be noted that the derivative part of a normal PID controller is neglected. This is because the derivative calculation will enlarge the noise or measurement error in the current signal, which may even make system unstable.

3.1.1 Control Topology

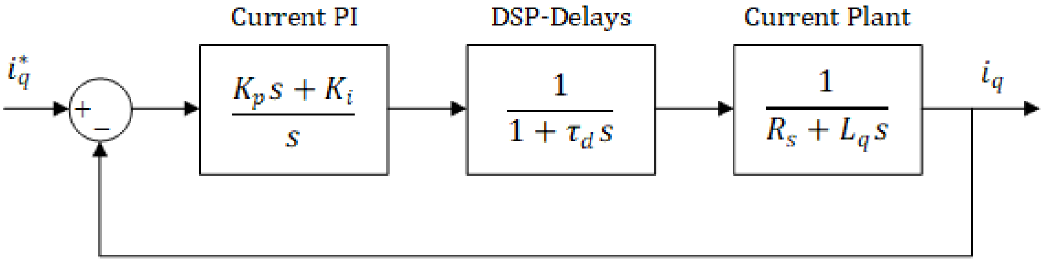

Fig. 3.1: q-axis current loop block diagram

Fig. 3.1 illustrates the design of a traditional current loop with PI for q-axis in order to realize FOC. K_p and K_i are the proportional and integral gain respectively, i_q is q-axis current while i_q^* is its reference, τ_d is time constant for the sum of system delay, R_s is stator resistance and L_q is q-axis inductance.

It can be observed that the current loop consists of a current PI, system delay and the current plant. The system delay is modelled as a first-order system with an equivalent time constant τ_d. The total system delay is mainly introduced by two components. One is the Analog to Digital Conversion (ADC) delay resulted from every time the control system trying to read a value, and the other one is calculation delay when DSP is trying to calculate the next switching commands for inverter. These two components take almost one sampling period respectively so the total system delay τ_d can be approximated as $\tau_d = 2T_s = 2/f_{sw}$ [10], where T_s is sampling period and f_{sw} is switching frequency.

Last but not least, the current plant is modelled with the machine transfer function from U_q to i_q.

3.1.2 PI Controller Tuning

The classical tuning method for current loop PI controller is zero-pole cancellation. This method makes the zero of current PI equal to the pole of the current plant so as to cancel them. Beside this, some other requirements like percent of overshoot, rise time and steady-state error can be specified to determine satisfying PI gains.

From Fig. 3.1, the open loop transfer function for q-axis current control loop can be written as,

$$G_{ol,i_q} = \frac{K_p s + K_i}{s} \frac{1}{\tau_d s + 1} \frac{1}{R_s + L_q s} \tag{3.1}$$

it can be observed from (3.1) that if controller zero $\frac{K_i}{K_p}$ equals to current plant pole $\frac{R_s}{L_q}$, cancellation can be performed. In this way the ratio between two PI gains can be determined. Hereby, the simplified closed current loop transfer function can be obtained,

$$G_{cl,i_q} = \frac{\frac{K_p/L_q}{s(\tau_d s+1)}}{\frac{K_p/L_q}{s(\tau_d s+1)} + 1} = \frac{K_p/L_q}{K_p/L_q + s(\tau_d s + 1)} = \frac{\frac{K_p/L_q}{\tau_d}}{s^2 + \frac{1}{\tau_d} s + \frac{K_p}{L_q \tau_d}} \tag{3.2}$$

where $K_p/L_q = \frac{K_p s + K_i}{R_s + L_q s}$ comes from the cancellation.

Compare this with typical second-order system transfer function,

$$G_s = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.3}$$

where ζ is damping ratio, ω_n is natural frequency and K is gain. Thus with (3.2), (3.1) and performance requirements selected, one can tune current PI controller (q-axis for example) using following results [11].

- System delay $\tau_d = 2T_s = 2/f_{sw}$
- Percent overshoot $= e^{-\zeta\pi/\sqrt{(}1-\zeta^2)}$
- Proportional gain $K_p = L_q\tau_d(\frac{1}{2\tau_d\zeta})^2$
- Integral gain $K_i = \frac{R_s}{L_q}K_p$

Beside the tuning procedure mentioned above, sisotool offered by MatLab is also a very useful and straightforward tuning helper. It can be used to improve the system response. However, it is shown in later Chapter 4 that there is a limitation for conventional current loop with PI. It is not capable enough since it only contains limited sets of parameters, and tuning of parameters asks for extra labor, not to mention adding derivative part into controller.

3.2 APID Analysis and Design

Besides traditional PI controller, a APID controller is investigated and designed for current control in FOC in this project. APID controller is essentially still a PID controller, except for its self-learning ability. During each sampling period, APID controller is capable to update its control parameters according to the input error signal, following the self-learning rule.

3.2.1 APID Controller Analysis and Design

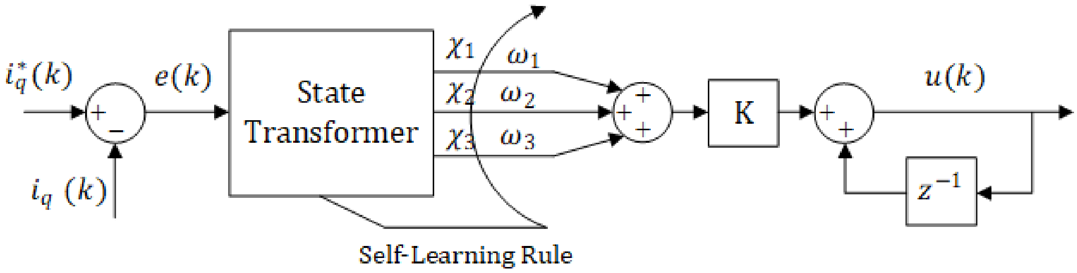

Fig. 3.2: APID controller schematic diagram

Unlike the PI controller modelled in section 3.1, APID controller is modelled in an incremental fashion to better illustrate its working principle. Fig. 3.2 shows schematic diagram of APID controller, where $i_q(k)$ and $i_q^*(k)$ are q-axis current and reference at k^{th} sampling period. The difference between them is denoted as $e(k)$, which is later processed by state transformer into three internal variables χ_1, χ_2 and χ_3. Self-learning rule here is set to specify how should the three weight coefficients ω_1, ω_2 and ω_3 be updated online through each sampling period. K is gain and $u(k)$ is the APID controller output at k^{th} sampling period.

Like shown in Fig. 3.2, taking system noise into consider, the current error at k^{th} sampling period can be expressed as

$$e(k) = i_q^*(k) - i_q(k) - n(k) \tag{3.4}$$

where $n(k)$ is system noise at k^{th} sampling period.

State transformer here transforms input error signal $e(k)$ at k^{th} sampling period into 3 internal variables as following.

$$\begin{cases} \chi_1(k) = e(k) - e(k-1) \\ \chi_2(k) = e(k) \\ \chi_3(k) = [e(k) - e(k-1)] - [e(k-1) - e(k-2)] \end{cases} \tag{3.5}$$

$\chi_1(k)$ here can be regarded as $\Delta e(k)$ while $\chi_3(k)$ can be regarded as $\Delta(\Delta e(k))$ at k^{th} sampling period. It is obvious that $\chi_1(k)$, $\chi_2(k)$ and $\chi_3(k)$ represent the proportional, integral and derivative component in traditional PID controller, at k^{th} sampling period, respectively.

After this, the output of APID controller can be written.

$$u(k) = u(k-1) + K\sum_{i=1}^{3}\omega_i(k)\chi_i(k) \tag{3.6}$$

By far, weight coefficients and self-learning rule for APID controller are left to be specified. In order to pursue faster current response, decreasing current error input can be set as an optimization goal. However, direct using of current error $e(k)$ will lead to the sign problem. Thus, a quadratic function of $e(k)$ is utilized to avoid this.

$$E(k) = \frac{1}{2}[e(k)]^2 = \frac{1}{2}[i_q^*(k) - i_q(k) - n(k)]^2 \tag{3.7}$$

where $E(k)$ is quadratic function of q-axis current error $e(k)$ and $n(k)$ is the system background noise at k^{th} sampling period.

Next, steepest descent algorithm is applied in order to minimize quadratic error function $E(k)$. The gradients of q-axis current error function $E(k)$ are found through partial differentiation,

$$\begin{cases} \frac{\partial E(k)}{\partial \omega_1(k)} = -e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_1(k)} \\ \frac{\partial E(k)}{\partial \omega_2(k)} = -e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_2(k)} \\ \frac{\partial E(k)}{\partial \omega_3(k)} = -e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_3(k)} \end{cases} \tag{3.8}$$

where $U_q^*(k)$ is the output of APID controller, q-axis command voltage.

With (3.8), during each sampling period the changes in weight coefficients should follow the opposite direction of gradients to reach steepest descent. Thus for each weight coefficient, the balancing of weight coefficients can be determined as follows:

$$\begin{cases} \Delta\omega_1(k) = -\eta_P\frac{\partial E(k)}{\partial \omega_1(k)} = \eta_P e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_1(k)} \\ \Delta\omega_2(k) = -\eta_I\frac{\partial E(k)}{\partial \omega_2(k)} = \eta_I e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_2(k)} \\ \Delta\omega_3(k) = -\eta_D\frac{\partial E(k)}{\partial \omega_3(k)} = \eta_D e(k)\frac{\partial i_q(k)}{\partial U_q^*(k)}\frac{\partial U_q^*(k)}{\partial \omega_3(k)} \end{cases} \tag{3.9}$$

where η_P, η_I and η_D are the proportional, integral and derivative learning step length for each sampling period.

Comparing items in controller output (3.6) and change of weight coefficients (3.9), following facts can be obtained.

$$\frac{\partial U_q^*(k)}{\partial \omega_i(k)} = K\chi_i(k),\, i = 1, 2, 3$$

Although the other item $\frac{\partial i_q(k)}{\partial U_q^*(k)}$ in (3.9) could be solved using voltage equation for PMSM, it will lead to voltage command calculation's inaccuracy when machine parameter changes. In order to avoid that, it is substituted with term $e(k)+\Delta e(k)$. This substitution is made according to engineering experience when tuning the controller parameter, term $e(k)$ and $\Delta e(k)$ should converge to zero fast, which hereby indicates the controller parameter tuning is associated with term $e(k)$ and $\Delta e(k)$ [12]. As for the exact value of term $\frac{\partial i_q(k)}{\partial U_q^*(k)}$, it is left for learning step length η_P, η_I and η_D to compensate. That is, step length η_P, η_I and η_D here have relationship with machine parameters.

Last but not least, weight coefficients $\omega_i(k)(i = 1, 2, 3)$ are normalized to make sure the convergence of expressions in (3.6) and (3.9), as shown below. Otherwise weight coefficients can go easily off boundary within certain number of sampling periods, so does APID controller output.

$$\begin{cases} \bar{\omega_1}(k) = \omega_1(k)/\sum_{i=1}^{3} |\omega_i(k)| \\ \bar{\omega_2}(k) = \omega_2(k)/\sum_{i=1}^{3} |\omega_i(k)| \quad i = 1, 2, 3 \\ \bar{\omega_3}(k) = \omega_3(k)/\sum_{i=1}^{3} |\omega_i(k)| \end{cases} \tag{3.10}$$

To sum up, self-learning rule specified for APID controller can be expressed as followed.

$$\begin{cases} \omega_1(k) = \omega_1(k-1) + \eta_P K\chi_1(k)e(k)[e(k) + \Delta e(k)] \\ \omega_2(k) = \omega_2(k-1) + \eta_I K\chi_2(k)e(k)[e(k) + \Delta e(k)] \\ \omega_3(k) = \omega_3(k-1) + \eta_D K\chi_3(k)e(k)[e(k) + \Delta e(k)] \end{cases} \tag{3.11}$$

And corresponding APID controller output can be rewritten as shown below.

$$u(k) = u(k-1) + K\sum_{i=1}^{3} \bar{\omega_i}(k)\chi_i(k) \tag{3.12}$$

With the specified self-learning rule above, APID controller is able to self-update its control parameter online according to the input current error signal during each sampling period until the error is eliminated, which benefits it to have better current transient response compared with a traditional PI controller. However, the complexity in algorithm brings more difficulty in tuning a APID controller, some experience based tuning guidelines will be mentioned at the end of Chapter 4.

Setup Description and Experimental Results

4.1 Setup Description and Machine Parameters

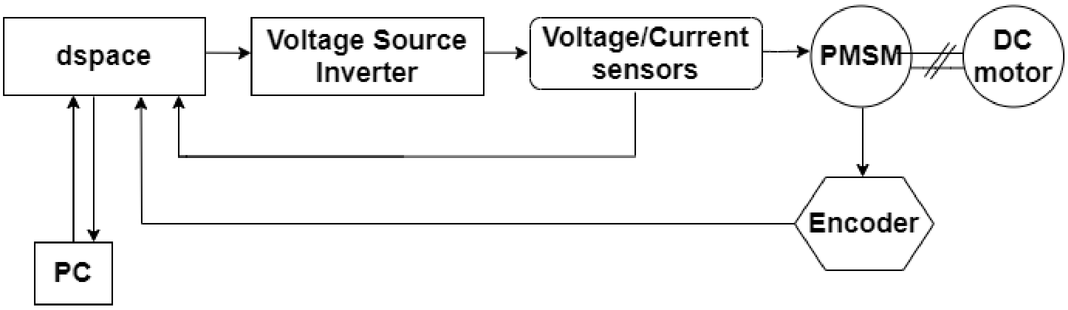

Fig. 4.1: Experimental setup

Fig. 4.1 above illustrates the block diagram of the experimental setup. It consists mainly of a dSPACE control box, Voltage Source Inverter (VSI), voltage and current sensors, encoder and PMSM coupled with a DC motor.

dSPACE control box communicates with a PC to generate proper command signals to control the whole setup. It receives measurements from voltage/current sensors and encoder, which are very important in order to full fill the control strategy. The encoder here can measure rotor position in real time.

Receiving gate signals from dSPACE, the VSI is then able to generate three phase voltage signals to actuate PMSM. Details for parameters of the used PMSM are listed below in table 4.1.

The DC motor is coupled with PMSM. It is used to produce load torque for PMSM in order to test the control performance.

Table 4.1: Technical parameters of Permanent Magnet Synchronous Motor

Parameter	Symbol	Value	Units
Power	P	400	W
Maximum phase Voltage	V	380	V
Rated current(RMS)	I_r	2.9	A
Rated speed	N_r	2850	rpm
Rated frequency	f_r	95	Hz
Stator resistance	R_s	2.3	Ω
d-axis inductance	L_d	6.9	mH
q-axis inductance	L_q	8.6	mH
Permanent magnet flux linkage	λ_m	0.12	Wb
Inertia of (PMSM)	$\mathcal{J}$	1	$mKgm^2$
No. of pole pairs	npp	2	-

4.2 Experimental Results

In this section, current response for conventional FOC current loop with PI is tested to show the limitation of response speed. Later on, FOC current loop with proposed adaptive PID controller is also tested on the same experiment setup and further evaluated. It should be noted that for both traditional and self-adaptive FOC current loop, experiments are carried out only for q-axis current as it is responsible for torque generation. Comparisons and discussions for the two methods are then made.

4.2.1 Traditional Current Loop with PI

In this subsection, traditional FOC current loop with PI is tested for current response, under the circumstances of:

- Different K_p, K_i gain sets, keeping the same ratio between them as discussed in subsection 3.1.2
- The gain sets ratio is calculated referring stator resistance R_s and q-axis inductance L_q in table 4.1 as $\frac{K_i}{K_p} = \frac{R_s}{L_q} = 267.4$
- Same q-axis current reference of a $1A$ step change
- Same rotor speed controlled by the coupled DC motor.

Kp = 5, Ki = 1337 q-Axis Current Step Response

Fig.4.2 illustrates q-axis current 1A step response at 370 rpm. The classic PI gains set is chosen to be $K_p = 5$ and $K_i = 1337$ (ratio 267.4). It can be observed from the figure that the rising time is about $0.0145s$, which is around 58 switching periods (4000 switching frequency). After $0.025s$ the q-axis current reaches steady state, but it takes some time for the system to completely remove steady state error.

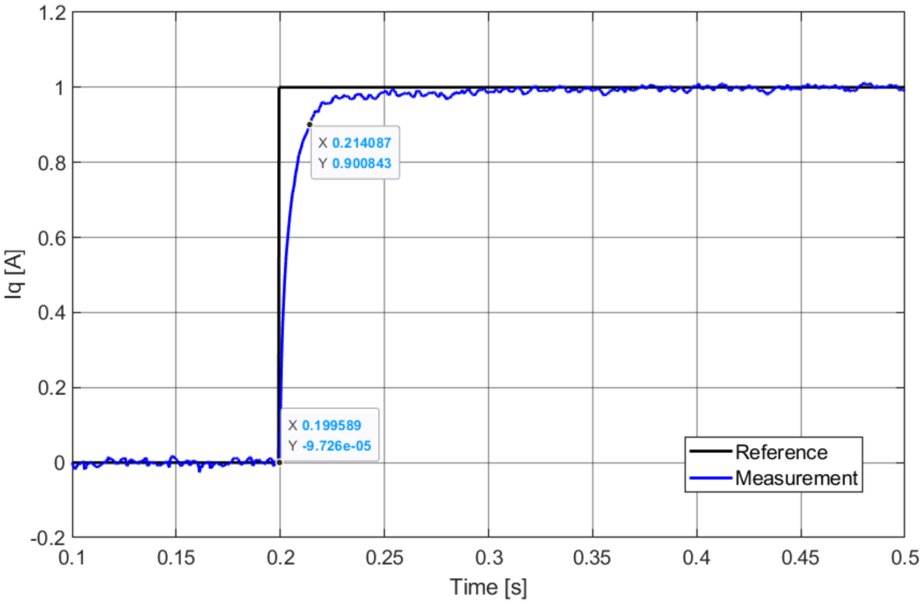

Fig. 4.2: Kp = 5, Ki = 1337, q-axis current loop step response of 1A, 370 rpm.

Kp = 10, Ki = 2674 q-Axis Current Step Response

Fig.4.3 below shows the q-axis current step response of 1A at 370 rpm. The classic PI gains set is chosen to be $K_p = 10$ and $K_i = 2674$ (ratio 267.4). By analysing Fig.4.3, the rising time is about $0.0073s$, which is around 29 switching periods (4000 switching frequency). After $0.018s$ the q-axis current reaches steady state and the I_q measurement converges faster to reference compared to Fig.4.2. This set of classic current controller's PI gains is also what has been calculated in section 3.1 using zero-pole cancellation method.

Kp = 15, Ki = 4011 q-Axis Current Step Response

Fig.4.4 below illustrates the q-axis current step response of 1A at 370 rpm. The classic PI gains set is chosen to be $K_p = 15$ and $K_i = 4011$ (ratio 267.4). The rising time is found out to be about $0.0038s$ and it accounts for 15 switching periods (4000 switching frequency). But there is a tiny current drop during rising (not obvious in figure), which might be not so good. After $0.014s$ the q-axis current reaches steady state and the I_q measurement converges even a bit faster to the reference value compared to Fig.4.3.

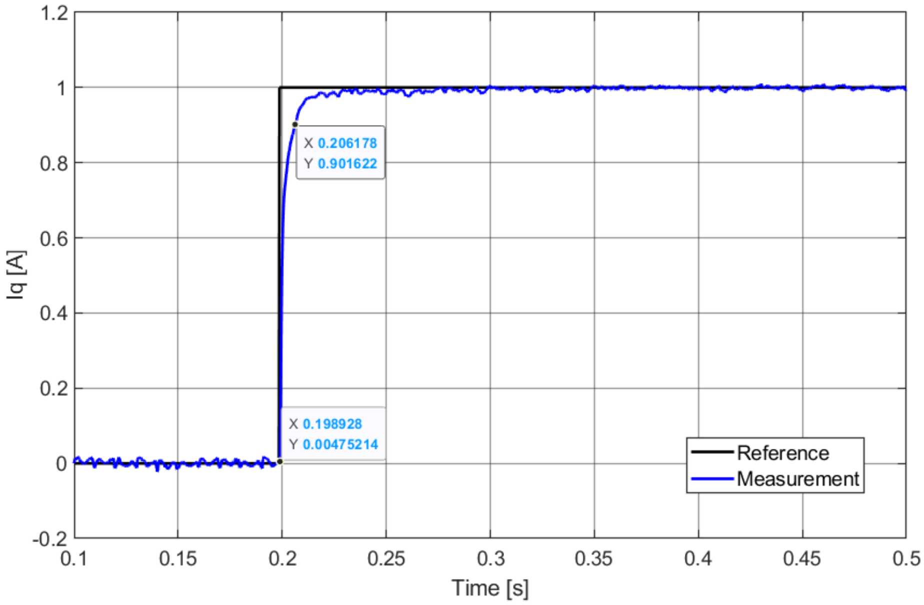

Fig. 4.3: Kp = 10, Ki = 2674, q-axis current loop step response of 1A, 370 rpm.

Kp = 20, Ki = 5348 q-Axis Current Step Response

The final PI gains are chosen to be $K_p = 20$ and $K_i = 5348$, the results are presented in Fig.4.5. It illustrates the q-axis current step response of 1A at 370 rpm. For this gain set, there are undesired oscillations and overshoot in I_q current measurement. Though it converges to reference value, the system takes more than $0.010s$ to reach steady state (more than 40 switching periods). This current response is worse compared with those in last two subsections.

Comparing all 4 current step response results in this section, it could be concluded that for FOC current loop with traditional PI controller, there is a performance limit that the fastest rising time occupies 15 switching periods with a tiny transient defect.

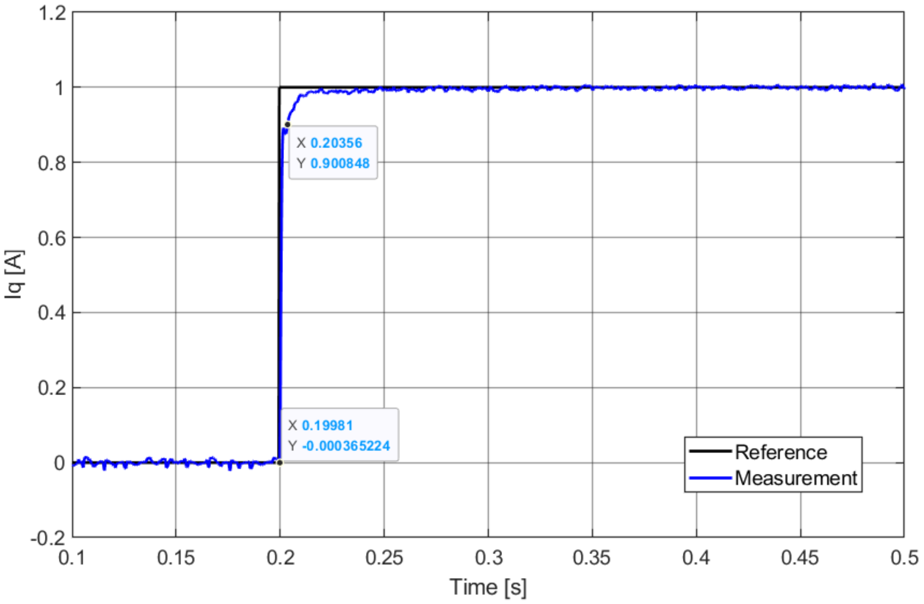

Fig. 4.4: Kp = 15, Ki = 4011, *q*-axis current loop step response of 1A, 370 rpm.

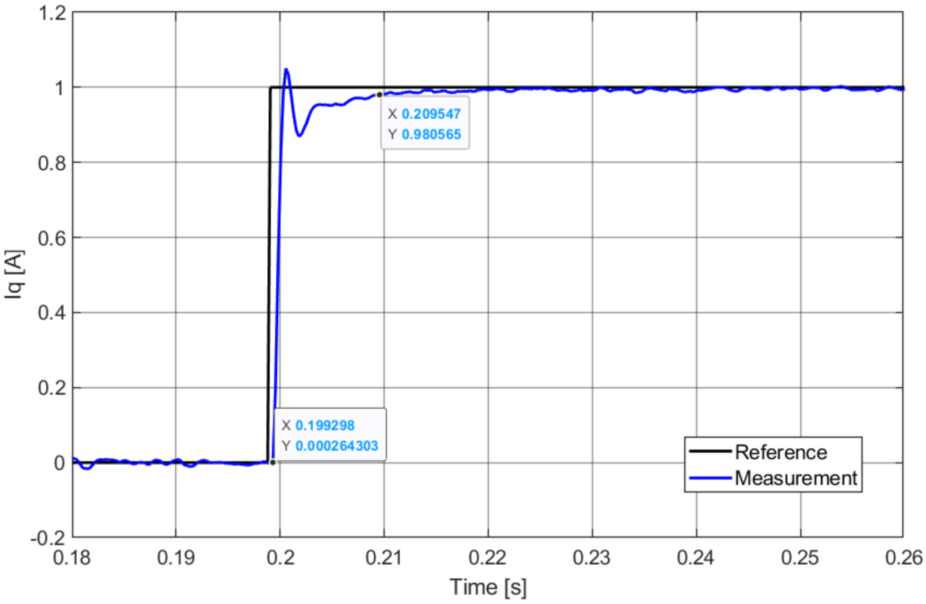

Fig. 4.5: Kp = 20, Ki = 5348, *q*-axis current loop step response of 1A, 370 rpm.

4.2.2 Current Loop with APID

In this subsection, FOC current loop with the proposed APID controller is firstly tested for current response, under the same circumstances mentioned above:

- Same test setup
- Same current reference of a $1A$ step change
- Same rotor speed controlled by the coupled DC motor.

Fig.4.6 below is q-axis current step response of FOC current loop with proposed APID controller. The system rising time could be found to be $0.0015s$, which is only 6 switching periods (4000 switching frequency). And there is no observable steady state error or overshoot, the oscillations in current measurement are also milder compared to those of results with traditional PI controller in last section.

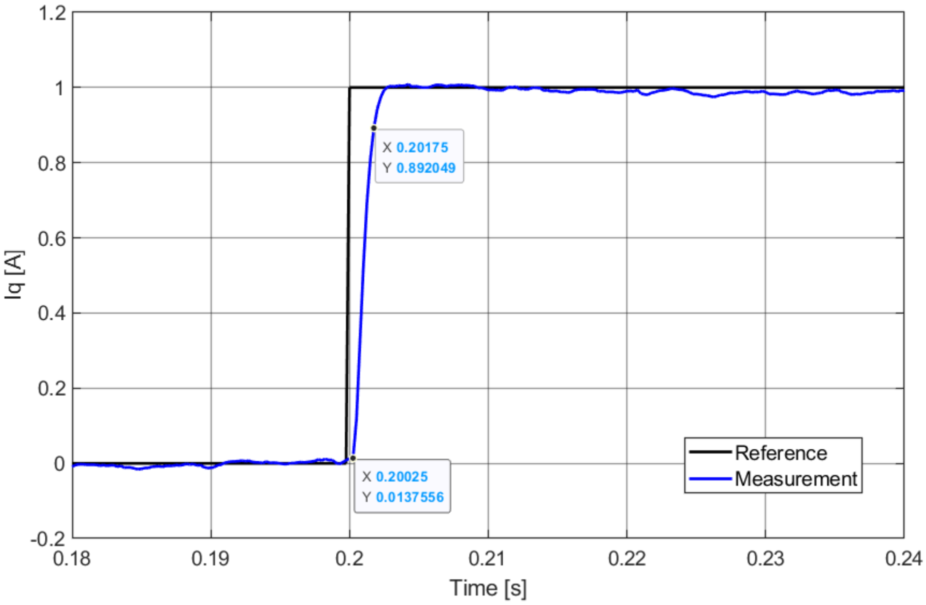

Fig. 4.6: q-axis current loop with APID, step response of 1A, 370 rpm.

Besides the result above, a further set of experiments are carried out to check the performance of FOC current loop with proposed APID controller against machine parameter changes due to different working condition e.g. temperature. In section 3.2, analysis reveal that step length η_P, η_I and η_D could have relationships with machine parameters. Thus, experiments below are designed as:

- A well tuned APID controller with $\eta_P = 500$, $\eta_I = 50$ and $\eta_D = 50$
- Different step length η_P, η_I and η_D (50, 100, 150 percent of well tuned value)
- Same q-axis current reference of a $1A$ step change
- Same rotor speed controlled by the coupled DC motor.

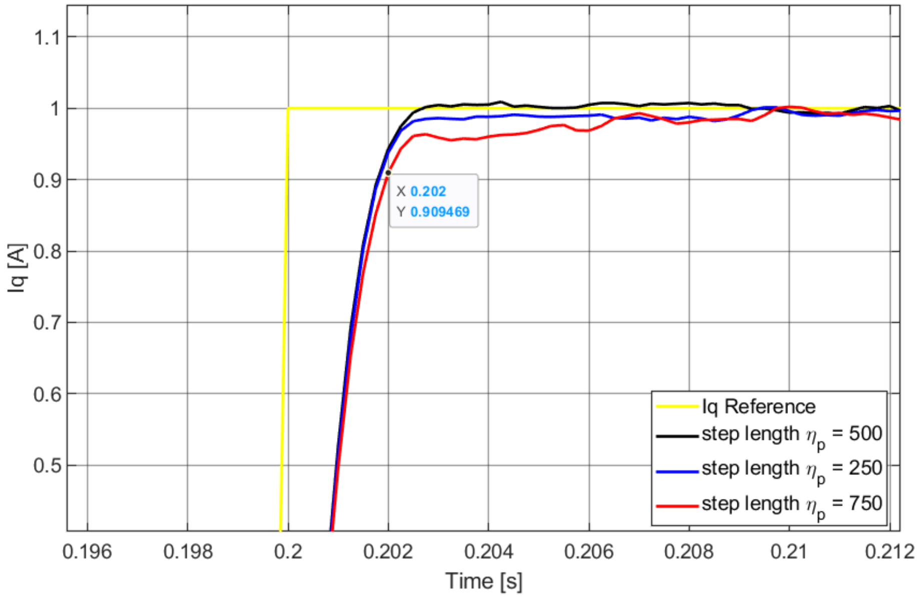

Fig. 4.7: q-axis current step response with APID, η_P= 250, 500, 750, at 130 rpm.

Fig.4.7 illustrates FOC q-axis current loop with APID, $1A$ step response for different step length η_P at 130 rpm. It could be found that for all three occasions, system rising times are almost the same (6 switching periods). While for step length $\eta_P = 750$, the system takes more time to reach steady state. And there is no significant steady state error for all three occasions.

The second experiment is carried out under different step length $\eta_I = 25, 50, 75$, showed in Fig.4.8. It could be seen obviously that for system rising time, all three occasions have very much similar results. However, both current responses with step length $\eta_I = 25$ and $\eta_I = 75$ have larger current oscillations in transient (about 25 switching periods). Besides, there are no steady state error for all three occasions.

The last set of experiments is carried out under different step length $\eta_D = 25, 50, 75$. The result is shown in Fig.4.9. It could be seen that only $\eta_D = 25$ lags a bit (around two switching periods) compared with the other two. And for $\eta_D = 25$, q-axis current

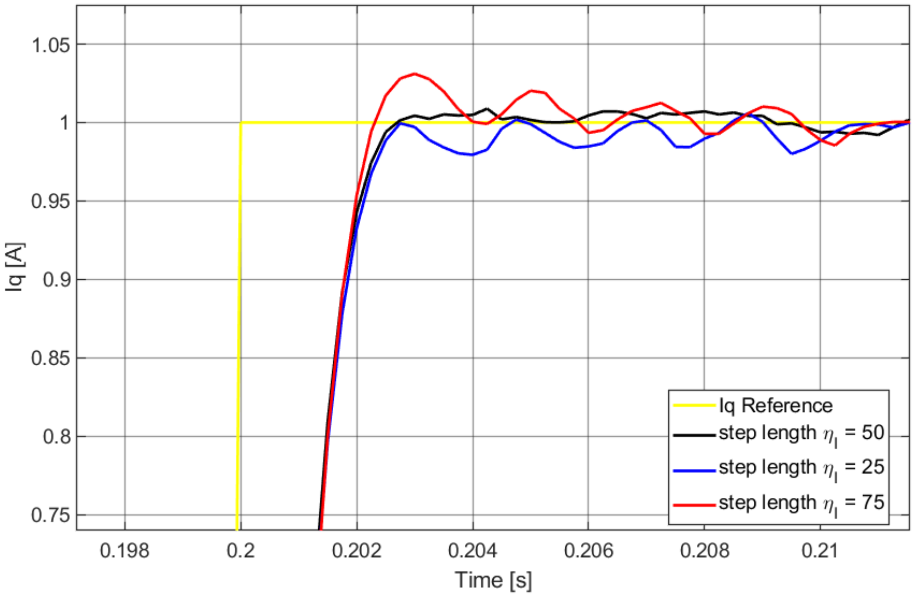

Fig. 4.8: q-axis current step response with APID, η_I= 25, 50, 75, at 130 rpm.

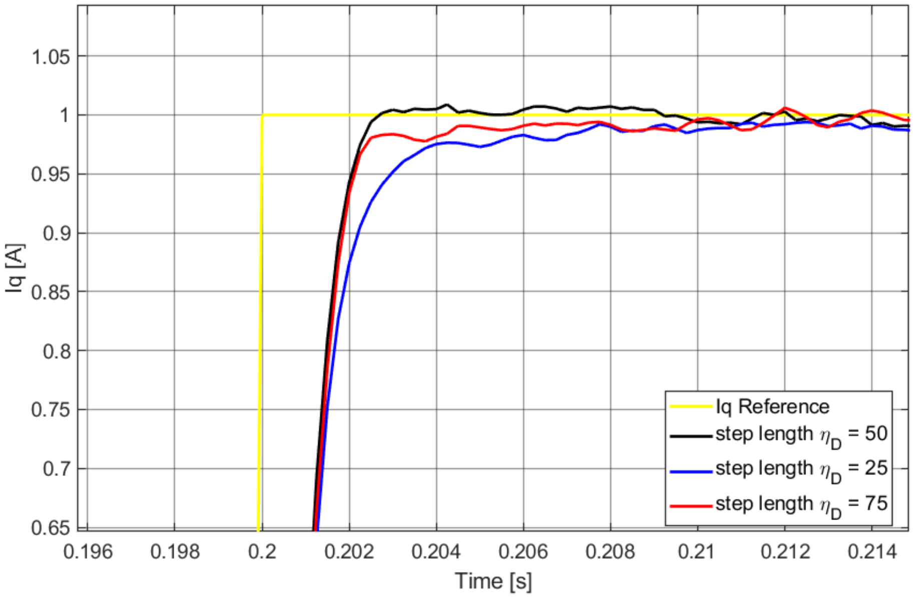

Fig. 4.9: q-axis current step response with APID, η_D= 25, 50, 75, at 130 rpm.

takes more time (about 14 switching periods) to reach steady state. In steady state, all of them shows no significant steady state error.

With the experiments mentioned above, FOC current loop with proposed APID controller shows good robustness against controller parameter change and faster current step response compared to traditional FOC current loop with PI controller. Besides, some experience-based tuning guidelines could also be given by analysing the results above.

4.2.3 APID Experience-Based Tuning Guidelines

The proposed APID controller basically has 7 parameters that need to be tuned: gain K, weight coefficients ω_1, ω_2 and ω_3, step length η_P, η_I and η_D. The tuning process could be complicated if trying blindly. In this section, tuning guidelines based on experience gained in this project will be concluded.

Before tuning APID, it should be noted that d-axis current loop's APID controller is better taken as a starting point. This is because for FOC based PMSM drive system, if the motor does no have saliency e.g. surface-mounted PMSM, $L_d = L_q$, d-axis current will not produce torque. In this way, disturbance of back-EMF can be avoided since the motor is not running. If the motor has saliency (e.g. interior PMSM), a classic q-axis current loop with PI controller could be adopted to control q-axis current to be 0 to achieve the same goal.

When tuning the proposed APID controller, firstly the gain K should be selected. Through observations during tuning and experiment process, the gain K's effect to the system response is quite similar to the damping ratio in a second-order system in classica control theory. A bigger gain K could lead to faster transient response but bigger oscillations at same time (even makes the system unstable), so it should be chosen properly according to experimental results.

After that, a set of initial weight coefficients ω_1, ω_2 and ω_3 need to be selected. Based on the results and tuning experience, they could be selected arbitrarily at first since their effect to system stability is not really huge. When a stable system response is obtained, they could be further tuned to make the system response faster.

Next, step length η_P, η_I and η_D need to be carefully selected based on the system response. In Fig.4.8, it could be found that different step length η_I may make the system reaching its steady state faster or slower. And it will also contributes to the oscillations in current response. This undesired effect could be compensated within limits by changing η_D.

Last but not least, the above steps should be repeatedly carried out to see if a more desirable controller performance could be achieved.

Conclusion

The major objectives of this project described in section 1.2.1 are achieved with the help of literature study and research analysis. A FOC current loop with proposed APID controller is implemented and validated through different experiments, and decent current tracking performance is evident from the experimental results presented in chapter 4.

In order to obtain a general understanding of the problem stated in section 1.2, a literature study is conducted to gain background knowledge of model predictive current control and existed self-updating PI controller in aspects of both advantages and shortcomings. Some knowledge from previous semesters is also utilized in this project, such as PMSM machine model, classic PI controller design for FOC based current loop etc.

Based on the knowledge from former semester project, a traditional FOC current loop with PI controller is analysed and designed. Besides that, a proper standard tuning method is described and utilized.

With the literature study made, a self-adaptive PID controller is then analyzed, designed and implemented. Following the project objectives, it is designed to be able to self-adapt online to perform a faster current response regardless machine parameter change.

Based on the efforts mentioned above, classic FOC current loop with traditional PI controller is tested with different K_p, K_i gains on the experimental setup, and limitations in current response are clear. The FOC current loop with implemented APID controller is also authentically validated with different controller parameters in order to simulate situation of machine parameter change.

The experiment results prove that the designed APID controller benefits FOC current loop with faster current response and good robustness against parameter variation. Furthermore, in order to make results of this project reproducible, a tuning guideline based on observations during both tuning and experiment process is elaborated.

Future Work

In this chapter, a variety of beneficial or interesting future work related to problem found during this project is described.

As formerly described in section 1.2.2, a speed loop could be designed with APID and then added to designed current loop to check the system performance whether similar improvements could be made. Besides, system delay could also be taken into consideration, since it is normally the defect when realizing model based predictive current control.

Next, APID controller analysis in section 3.2 uses $(e + \Delta e)$ to replace item $\frac{\partial i_q(k)}{\partial U_q^*(k)}$ which is based on engineering experience [12]. However, for PMSM drive system, it is also possible to directly calculate $\frac{\partial i_q(k)}{\partial U_q^*(k)}$ through stator voltage equation. It could be interesting to see the difference of these two methods and gain deeper understanding of APID controller.

Besides, in section 3.2, it is analyzed that three parameters (step length η_P, η_I and η_D) are in relation with machine parameters. It would be beneficial if a more exact relation can be defined (e.g. numerical calculations). In this way, when conducting experiments with machine parameter change, a more obvious condition can be made like how much the machine parameter (stator resistance R_s, dq-axis inductance L_d, L_q, etc) is varying other than condition made like 50, 100, 150 percent of well-tuned parameter in experimental results chapter.

Further, the effect of initial values for three weight coefficients could be investigated. There is possibly a method to find the recommended value, which can make the tuning guidelines mentioned in above chapter a more elaborated one.

Bibliography

[1] M. J. Melfi, S. Evon, and R. McElveen, "Permanent magnet motors for power density and energy savings in industrial applications," in *Conference Record of 2008 54th Annual Pulp and Paper Industry Technical Conference*, 2008, pp. 218–225. DOI: 10.1109/PAPCON.2008.4585822.

[2] T. Sawa and T. Kume, "Motor drive technology - history and visions for the future," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, vol. 1, 2004, 2–9 Vol.1. DOI: 10.1109/PESC.2004.1355703.

[3] F. Morel, X. Lin-Shi, J.-M. Retif, B. Allard, and C. Buttay, "A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2715–2728, 2009. DOI: 10.1109/TIE.2009.2018429.

[4] J. Rodriguez, J. Pontt, C. Silva, P. Cortes, U. Amman, and S. Rees, "Predictive current control of a voltage source inverter," in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, vol. 3, 2004, 2192–2196 Vol.3. DOI: 10.1109/PESC.2004.1355460.

[5] P. Cortes and J. Rodriguez, "Three-phase inverter with output lc filter using predictive control for ups applications," in *2007 European Conference on Power Electronics and Applications*, 2007, pp. 1–7. DOI: 10.1109/EPE.2007.4417385.

[6] P. Krause, O. Wasynczuk, S. D. Sudhoff, and S. Pekarek, *Analysis of Electric Machinery and Drive Systems*. IEEE Press, 2013.

[7] H.-T. Moon, H.-S. Kim, and M.-J. Youn, "A discrete-time predictive current control for pmsm," *IEEE Transactions on Power Electronics*, vol. 18, no. 1, pp. 464–472, 2003. DOI: 10.1109/TPEL.2002.807131.

[8] X. Liu, T. Huang, X. Tang, and H. Xin, "Design of self-adaptive pid controller based on least square method," in *2009 Third International Conference on Genetic and Evolutionary Computing*, 2009, pp. 527–529. DOI: 10.1109/WGEC.2009.65.

[9] S. D. Sahputro, F. Fadilah, N. A. Wicaksono, and F. Yusivar, “Design and implementation of adaptive pid controller for speed control of dc motor,” in *2017 15th International Conference on Quality in Research (QiR) : International Symposium on Electrical and Computer Engineering*, 2017, pp. 179–183. DOI: 10.1109/QIR.2017.8168478.

[10] AAU PED-842, “Auto-tuning of high performance vector controller for pmsm drive system,” *Aalborg University*, 2013.

[11] C. L. Phillips and R. D. Harbor, *Feedback Control Systems*.

[12] J. Liu, W. Wu, H. S.-H. Chung, and F. Blaabjerg, “Disturbance observer-based adaptive current control with self-learning ability to improve the grid-injected current for *lcl* -filtered grid-connected inverter,” *IEEE Access*, vol. 7, pp. 105 376–105 390, 2019. DOI: 10.1109/ACCESS.2019.2931734.