ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

RSL Robotic Systems Lab
Prof. Dr. Marco Hutter

Semester Thesis

Development of a Model-based Unified Torque Controller for PMSM-based Series Elastic Actuators

Autumn Term 2018

Supervised by:
Marco Hutter

Author:
Yifan Jiang

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Development of a Model-based Unified Torque Controller for PMSM-based Series Elastic Actuators

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Yifan	Jiang

Student supervisor(s)

Vassilios	Tsounis

Supervising lecturer

Marco	Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zurich, 07.01.2018
Place and date

Signature

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zurich, 07.01.2018	
Place and date	Signature

Contents

Acknowledgements

I would like to thank my supervisor Vassilios Tsounis for his patient guidance and his help with hardware setup, which inspired me to find the right direction in this project and enabled me to carry out my experiments smoothly. I am also grateful to Prof. Marco Hutter and other lab colleagues, who offered me inspiring suggestions and critics in this projects and during the presentations. This project is truly an invaluable learning experience for me.
Last but not least, I appreciate the mental support my girlfriend was giving me during my work, which helped me overcome many hard moments.

Abstract

In this thesis, a novel model-based torque controller for a Series Elastic Actuators (SEA) is developed and tested. The dynamics of the SEA is first modeled and implemented in Simulink to enable the verification of the controller in simulation. The model-based torque controller is then derived and verified in simulation. Experiments are then conducted on the ANYdrive to identify the parameters of the model and to examine the performance of the controller.

Symbols

Symbols

$\phi_m, \dot{\phi}_m, \ddot{\phi}_m$	position, velocity, and acceleration of the PMSM
$\phi_j, \dot{\phi}_j$	position and velocity of the joint
i_d, i_q	D- and Q-axis current of the PMSM
L_d, L_q	D- and Q-axis inductance of the PMSM
R_s	single-phase resistance of the PMSM
n_p	number of pole pairs of the PMSM
λ_m	rotor flux of the PMSM
K_τ	torque constant of the PMSM
τ_m	torque generated by the PMSM
θ_e	electrical angle of the PMSM
J_m	inertia of the rotor of the PMSM
J_L	load inertia on the joint
b_m	viscous damping of the PMSM
b_s	viscous damping of the spring
k_s	spring constant
n_g	reduction ration of the gearbox
η	efficiency of the gearbox
$\mathbf{x}$	state vector
$\mathbf{A_d}$	discrete time state transition matrix
$\mathbf{Q}$	stage cost matrix
$\mathbf{Q_T}$	terminal cost matrix
$\mathbf{R}$	control action cost matrix

Acronyms and Abbreviations

PMSM	Permanent Magnet Synchronous Motor
MPC	Model Predictive Controller
LQR	Linear Quadratic Regulator
RHC	Receding Horizon Controller
RTOS	Real Time Operating System
FOC	Field Oriented Controller
MBUTC	Model Based Unified Torque Controller

Chapter 1

Introduction

Series Elastic Actuators (SEAs) are a special class of actuators for robotic applications, where the output characteristics of a electrical motor is first modified by a transmission mechanism (e.g. gearbox), and next the load is coupled to the output of the transmission mechanism via a spring. First introduced in [1], this actuation concept gains popularity in the robotics community because of its ability to protect the relatively stiff gearbox from impacts and to enable easy interaction force/torque measurement. They are considered to be particularly useful in human-robot interactions, as the compliance introduced by the spring provides a buffer between the stiff transmission and human.

The conventional approach to control the output torque of the SEA is to employ a number of cascaded control loops, the first one of which is normally a current loop, which controls the electrical dynamics of the motor to generate desired motor current. Outer loops then control the output torque of the SEA, which is reflected as the deflection of the elastic element. Researchers have purposed various control techniques based on the cascaded loop architecture. For example, M.M. Williamson [1] purposed placing PID controller for output torque directly out of the current loop; while Vallery *et al.* [2] employs a velocity loop between torque loop and inner current loop; Sariyildiz *et al.* [3] and Paine *et al.* [4] employed controllers based on disturbance observers, which helps reject the impact of load disturbance to torque control.

The objective of this work is to develop and test a novel model-based unified output torque controller for the SEA, which removes the need of cascaded loops for torque control. Instead, both the electrical dynamics of the motor and the mechanical dynamics of the gearbox and the series elastic element are controlled using a single model-based controller. Output torque setpoints are provided to the controller, and based on the measured state of the SEA, the voltage to be applied to the motor is computed.

The work consists of the following stages. First, the ANYdrive, which is the SEA under consideration in this work, is modeled and analyzed. Second, a simplified nonlinear state-space model of the ANYdrive is derived, and the unified torque controller is formulated. Third, the performance of the controller is verified in simulation, using the model constructed earlier. Fourth, the parameters of the ANYdrive are identified in experiments, and the accuracy of the model is then verified against experimental results. Finally, unified torque controller is implemented in the firmware of the ANYdrive, and its performance is examined experimentally.

Chapter 2

Modeling of the ANYdrive

In this chapter, the modeling of the ANYdrive will be discussed. The model constructed in this chapter plays a number of important roles. It will be used for simulations, and as a virtual test bench to verify control algorithms. In addition, a simplified version of the model will be used as the basis of the derivation of the unified torque controller. The discussions in this chapter is based on the ANYdrive, but the same principles can be applied to a wider range of SEAs.
The full model of the SEA is provided in two versions. The first version is implemented in Simulink, and the second version is a C module, which can be used as a plugin in Gazebo, enabling the simulation of motor dynamics in whole-robot simulations. In addition, a simplified version of the full model is used for the derivation of the unified torque controller. In this chapter, the derivation of the full model and its implementation in Simulink and C will first be elaborated. Next, the derivation of the model for control purpose is discussed.

2.1 Basics of the ANYdrive

The ANYdrive is a compact series elastic actuator module developed by the Robotic Systems Lab at ETH Zurich [5]. It is a rotary actuator that is driven by a brushless permanent magnet synchronous motor (PMSM). The PMSM is coupled to a gearbox with a reduction ration of 50:1, which increases the torque and reduces and speed. A planar torsion spring then couples the output of the gearbox to the load.

2.2 Modeling for Simulation

2.2.1 Modeling of the PMSM

To achieve a more realistic simulation of the electrical dynamics of the PMSM, the electrical system is modeled directly in the 3-phase domain, contrary to more common formulations where 3-phase quantities (i.e. voltage and current) are transformed into the DQ domain to simplify analysis. This formulation provides the possibility of simulating the uneven spacing of phases and the difference in the impedances of phases, which are common imperfections of real-life PMSM. The formulation of the electrical system model is based on the work of David Ocen [6]. The differential equation of the electrical system is given by:

$$\frac{d}{dt}\boldsymbol{\lambda} + \mathbf{R_{abc}}\mathbf{i_{abc}} = \mathbf{v_{abc}} \tag{2.1}$$

where $\mathbf{i_{abc}}$ and $\mathbf{u_{abc}}$ are the 3-phase current and voltage values, $\mathbf{R_{abc}}$ is the single-phase resistance matrix, and $\boldsymbol{\lambda}$ is the flux link of the PMSM.

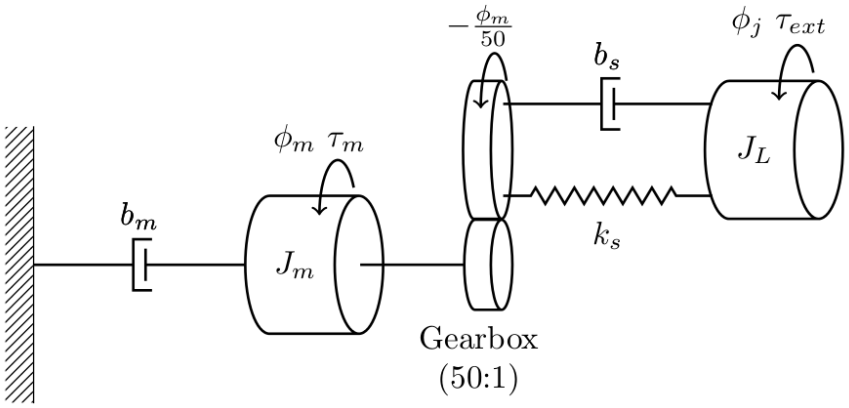

Figure 2.1: Free-body Diagram of the ANYdrive

$$\mathbf{i_{abc}} = \begin{bmatrix} i_a \\ i_c \\ i_c \end{bmatrix} \mathbf{v_{abc}} = \begin{bmatrix} v_a \\ v_c \\ v_c \end{bmatrix} \mathbf{R_{abc}} = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_b & 0 \\ 0 & 0 & R_c \end{bmatrix} \tag{2.2}$$

The total flux link of the PMSM is simply the summation of the stator and rotor fluxes. The stator (i.e. windings) flux is given by the product of the inductance matrix $\mathbf{L_{abc}}$ and the phase current vector $\mathbf{i_{abc}}$, and the rotor flux is simply the flux of the permanent magnets. The rotor flux vector, $\boldsymbol{\lambda}_m$, represents the components of the rotor flux in the directions of the phases. $\boldsymbol{\lambda}_m$ is a function of the electrical commutation angle of the motor, θ_e.

$$\boldsymbol{\lambda} = \mathbf{L_{abc}} \mathbf{i_{abc}} + \boldsymbol{\lambda}_m \tag{2.3}$$

2.2.2 Modeling of the Mechanical System

The mechanical aspect of the ANYdrive is modeled as a spring-mass-damper system with nonlinear friction. The free-body diagram of the ANYdrive is shown in Figure 2.1.

The mechanical system is modeled on the motor side. Because of the presence of the gearbox, the effects of the spring and the spring damping term on the motor are modified. Any torque that is applied to the joint side of the gearbox is also reflected on the motor side but modified by the gearbox. The effective values of the above-mentioned terms after the modification of the gearbox are summarized as follows:

$$\begin{aligned} k_{s,eff} &= \frac{k_s}{{n_g}^2 \eta} \\ b_{s,eff} &= \frac{b_s}{{n_g}^2 \eta} \\ \tau_{ext,eff} &= \frac{\tau_{ext}}{n_g \eta} \end{aligned} \tag{2.4}$$

The nonlinear friction is modeled as a continuous and differentiable function of velocities. Many different formulations are possible, as mentioned in [7]. For simplicity, the method proposed in [7] is modified as follows:

$$friction = b\dot{\phi} + F_s \tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}) \tag{2.5}$$

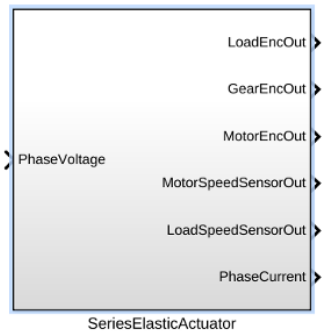

Figure 2.2: ANYdrive Simulink Model

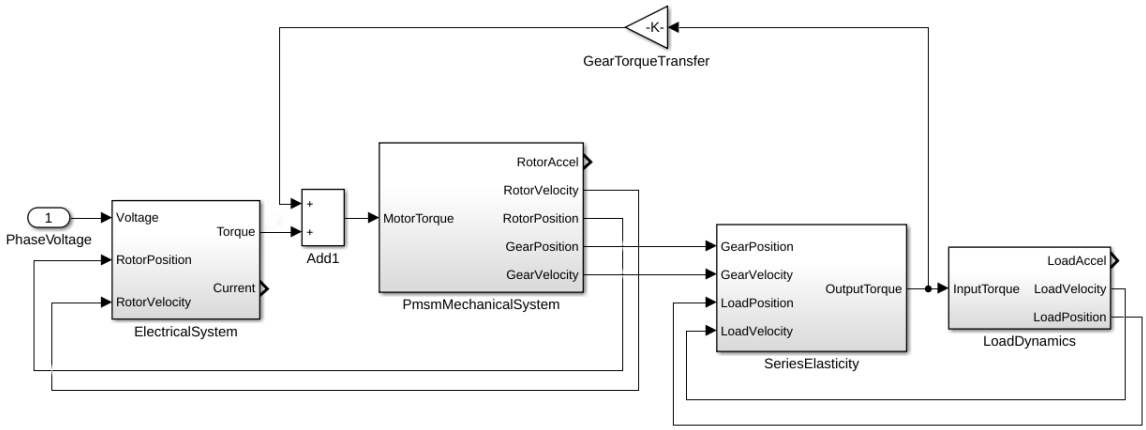

Figure 2.3: ANYdrive Simulink Model Details

It consists of a linear part, $b\dot{\phi}$, which represents viscous damping effect at high speed. In addition, the effect of static friction is modeled as a hyperbolic tangent function, which provide a smooth transition in friction torque when velocity changes direction. F_s is the magnitude of maximum static friction. The parameter ω_{bk} determines the velocity at which the magnitude of the nonlinear friction reaches 97% of the F_s. The choice of ω_{bk} will be discussed in later part of the chapter.
The equation of motion of the mechanical system can then be written down as follows:

$$J_m\ddot{\phi}_m+(b_m+\frac{b_s}{{n_g}^2\eta})\dot{\phi}_m+\frac{k_s}{{n_g}^2\eta}\phi_m+F_s\tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}_m)-\frac{k_s}{n_g\eta}\phi_j-\frac{b_s}{n_g\eta}\dot{\phi}_j=\tau_m \quad (2.6)$$

2.2.3 Implementation of the Model

The full model of the ANYdrive is first implemented as a Simulink model. The Simulink model of the ANYdrive is shown in Figures 2.2 and 2.3. The output of the Simulink model are processed with zero-order hold and quantization, which simulates the quantization effect of sensors in the real system. Figure 2.4 shows the implementation of the sensor model.
In order to integrate the electrical system of the PMSM into whole-body simulations for robots that use ANYdrive as joint actuator, the SEA full model is also implemented as a C module, which is then integrated into a Gazebo simulation.

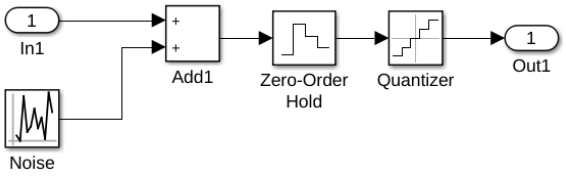

Figure 2.4: Model of Sensor

A number of differences exist between the C module and the Simulink model. First, only the dynamics of the electrical system of the PMSM is computed by the C module at this stage. Second, the C module contains its own numerical integration scheme, which allows the model to be integrated independent from Gazebo. A separate numerical integration scheme is necessary because the time constant of the electrical system is small compared to that of the mechanical system, which requires the use of smaller timestep. The timestep in the C module is fixed at 0.01ms, and each call to the following function advances the numerical integration by one timestep:

```
pmsm_advance(pmsm_input_data * contro_action)
```

When integrating the C module into simulations of mechanical systems, where the mechanical system is numerically integrated using a different timestep, the advance function should be called an appropriate number of times to ensure that both the mechanical and the electrical systems progress at the same rate.

2.3 Modeling for Control

The derivation of the unified torque controller calls for a model of the system dynamics that is as compact as possible for computational efficiency, yet captures the important behaviors of the system. It is also required that the model be continuous and differentiable. In order to obtain the model that meets the needs of controller design, additional assumptions are made and the full model is then simplified. The list below summarizes the key modifications:

- The electrical system is transformed to DQ-domain using Park transformation.
- Load dynamics is replaced with a first-order approximation based on sensor measurements.

The electrical system dynamics in DQ domain is given by:

$$\begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix} \begin{bmatrix} \dot{i_d} \\ \dot{i_q} \end{bmatrix} + \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} -i_q L_q \\ i_d L_d \end{bmatrix} \dot{\theta}_e = \begin{bmatrix} u_d \\ u_q \end{bmatrix} \tag{2.7}$$

Due to the fact that the PMSM used in the ANYdrive has 10 pole pairs, the relationship between the electrical velocity and the actual velocity of rotation of the PMSM is given by:

$$\dot{\theta}_e = 10\dot{\phi}_m \tag{2.8}$$

The torque generated by the PMSM is:

$$\tau_m = \frac{3}{2} n_p (\lambda_m i_q + (L_d - L_q) i_d i_q) \tag{2.9}$$

The mechanical system of the ANYdrive is formulated on the motor side of the gearbox, and is given by the following equation:

$$J_m\ddot{\phi}_m + (b_m + \frac{b_s}{{n_g}^2\eta})\dot{\phi}_m + \frac{k_s}{{n_g}^2\eta}\phi_m + f(\dot{\phi}_m) - \frac{k_s}{n_g\eta}\phi_j - \frac{b_s}{n_g\eta}\dot{\phi}_j = \tau_m \tag{2.10}$$

The nonlinear component of the friction of the PMSM is modeled by the $f(\dot{\phi}_m)$ term:

$$f(\dot{\phi}_m) = F_s \tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}_m) \tag{2.11}$$

In practive, the value of ω_{bk} should be at least 10 times larger than the resolution of velocity measurement, in order to avoid instabilities caused by the noise in measured velocity.
The vector of state variables is defined as:

$$\mathbf{x} = \begin{bmatrix} \phi_m & \dot{\phi}_m & i_d & i_q & \phi_j \end{bmatrix}^{\mathbf{T}} \tag{2.12}$$

The control action vector is defined as:

$$\mathbf{u} = \begin{bmatrix} u_d \\ u_q \end{bmatrix} \tag{2.13}$$

The linear state matrix is given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{k_s}{J_m {n_g}^2\eta} & -\frac{b_m}{J_m} & 0 & \frac{3n_p\lambda_m}{2J_m} & \frac{k_s}{J_m n_g\eta} \\ 0 & 0 & -\frac{R_s}{L_d} & 0 & 0 \\ 0 & -\frac{\lambda_m}{L_q}n_p & 0 & -\frac{R_s}{L_q} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.14}$$

The action matrix is given by:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \end{bmatrix} \tag{2.15}$$

The nonlinear terms of the state space system dynamics are collected in function g(q):

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 0 \\ \frac{3n_p}{2J_m}(L_d - L_q)i_d i_q - \frac{f(\dot{\phi}_m)}{J_m} \\ \frac{L_q}{L_d}\dot{\theta}_e i_q \\ -\frac{L_d}{L_q}\dot{\theta}_e i_d \\ \dot{\phi}_{j,mes} \end{bmatrix} \tag{2.16}$$

The simplified state space model of the SEA can then be written in a compact form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{g}(\mathbf{x}) \tag{2.17}$$

Chapter 3

The Model-based Unified Torque Controller

Based on the simplified model derived in the previous chapter, the model-based unified torque controller (MBUTC) is formulated. The MBUTC is analyzed in a Matlab simulation using the full model derived in the previous chapter. The MBUTC is implemented as part of the firmware of the ANYdrive, and is then optimized for computational efficiency in order to meet the real-time constraints of the firmware.

3.1 Formulation

The MBUTC is a model-predictive controller (MPC) for spring deflection, which is based on a local linearization of the nonlinear model. The output torque control problem is converted to a spring deflection control problem. Due to the well defined kinematic relationship of the gearbox, the spring deflection control problem can be considered as equivalent to a motor position control problem. As a result, the target joint torque setpoint is converted to a motor position setpoint using the following equation:

$$\phi_{m,des} = \frac{1}{n_g}(\frac{\tau_{des}}{k_s} + \phi_j) \tag{3.1}$$

At each timestep, the nonlinear model is linearized at the current state, which is obtained from sensor readings.The linearized model is then discretized and the controller outputs are computed by solving a finite horizon linear quadratic regulator (LQR) problem.
Recall that the state space model of the ANYdrive is given by:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{g}(\mathbf{x}) \tag{3.2}$$

Construct the current state, $\mathbf{x_0}$, using sensor measurements. The system dynamics can then be linearized about $\mathbf{x_0}$:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{g}(\mathbf{x_0}) + \mathbf{D_g}(\mathbf{x} - \mathbf{x_0}) \tag{3.3}$$

where

$$\mathbf{D_g} = \frac{d\mathbf{g}}{d\mathbf{x}}|_{\mathbf{x_0}} \tag{3.4}$$

Define reference state as $\mathbf{x_{des}} = [\phi_{m,des}, 0, 0, 0, 0]^T$, and then apply the following linear change of variable, which results the dynamics with respect to the difference between the current state and the reference state (the error dynamics):

$$\boldsymbol{\epsilon} = \mathbf{x} - \mathbf{x_{des}} \tag{3.5}$$

$$\begin{aligned}\dot{\boldsymbol{\epsilon}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}_{des} \\ &= \mathbf{Ax} + \mathbf{Bu} + \mathbf{g(x_0)} + \mathbf{D_g(x - x_0)} - \dot{\mathbf{x}}_{\mathbf{des}} \\ &= \mathbf{A}\boldsymbol{\epsilon} + \mathbf{Bu} + [\mathbf{D_g(x - x_0)} + \mathbf{Ax_{des}} - \dot{\mathbf{x}}_{\mathbf{des}}] \\ &= \mathbf{A}\boldsymbol{\epsilon} + \mathbf{Bu} + \mathbf{C}\end{aligned} \tag{3.6}$$

To obtain the time derivative of the reference state, $\dot{\mathbf{x}}_{des}$, only the joint velocity term is considered, as the time derivative of output torque setpoint is not always available.

$$\dot{\mathbf{x}}_{des} = [\frac{1}{n_g}\dot{\phi}_j, 0, 0, 0, 0]^T \tag{3.7}$$

The discretization of the error dynamics is defined as follows:

$$\boldsymbol{\epsilon}_{k+1} = \mathbf{A_d}\boldsymbol{\epsilon}_k + \mathbf{B_d}\mathbf{u_k} + \mathbf{C_d} \tag{3.8}$$

where

$$\begin{aligned}\mathbf{A_d} &= e^{\mathbf{A}dt} \simeq \mathbf{I} + \mathbf{A}dt + \frac{1}{2}\mathbf{A}dt^2 \\ \mathbf{B_d} &= \mathbf{B}dt \\ \mathbf{C_d} &= \mathbf{C}dt\end{aligned} \tag{3.9}$$

One additional step is needed to transform the discrete error dynamics into the standard form ($x_{k+1} = Ax_k + Bu_k$), which can be readily plugged in to the LQR formulation:

$$\begin{aligned}\bar{\boldsymbol{\epsilon}}_k &= \begin{bmatrix}\boldsymbol{\epsilon}_k \\ \mathbf{C_d}\end{bmatrix} \\ \bar{\mathbf{A}}_d &= \begin{bmatrix}\mathbf{A_d} & \mathbf{I_{5x5}} \\ \mathbf{0_{5x5}} & \mathbf{I_{5x5}}\end{bmatrix} \bar{\mathbf{B}}_d = \begin{bmatrix}\mathbf{B_d} \\ \mathbf{0_{5x2}}\end{bmatrix}\end{aligned} \tag{3.10}$$

$$\bar{\boldsymbol{\epsilon}}_{k+1} = \bar{\mathbf{A}}_d\bar{\boldsymbol{\epsilon}}_k + \bar{\mathbf{B}}_d u_k \tag{3.11}$$

At this moment, the finite horizon LQR problem can be formulated and solved, which will yield the control action to be applied to the PMSM. First, the finite horizon cost is define:

$$\mathbf{J} = \bar{\boldsymbol{\epsilon}}_N^T\mathbf{Q_T}\bar{\boldsymbol{\epsilon}}_N + \sum_{k=1}^{N-1}(\bar{\boldsymbol{\epsilon}}_k^T\mathbf{Q}\bar{\boldsymbol{\epsilon}}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k) \tag{3.12}$$

Then, the following finite horizon, discrete LQR problem can be solved to obtain a sequence of u_ks, which is the optimal action under this situation:

$$\begin{aligned}\bar{\boldsymbol{\epsilon}}_k^* \ , \ \mathbf{u}_k^* &= \arg\min_{\bar{\boldsymbol{\epsilon}}_k \ , \ \mathbf{u}_k}[\mathbf{J}] \\ s.t. \quad \bar{\boldsymbol{\epsilon}}_{k+1} &= \bar{\mathbf{A}}_d\bar{\boldsymbol{\epsilon}}_k + \bar{\mathbf{B}}_d u_k\end{aligned} \tag{3.13}$$

The cost matrices $\mathbf{Q_T}$ and $\mathbf{Q}$ must be positive semidefinite, while $\mathbf{R}$ must be positive definite to avoid infinite control actions. In the formulation of the MBUTC, the cost matrices are all diagonal matrices which take the following form:

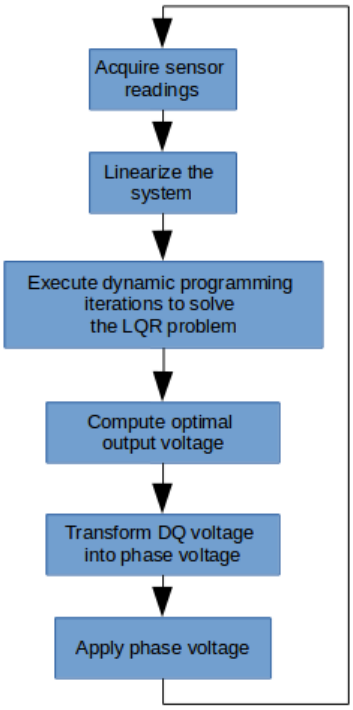

Figure 3.1: MBUTC Flow Chart

$$\mathbf{Q_T} = \begin{bmatrix} q_{T11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{T33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{R} = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix} \quad (3.14)$$

q_{T11} and q_{11} penalize a deviation from joint torque setpoint, q_{T33} and q_{33} penalize a non-zero D-current, and $\mathbf{R}$ penalizes the application of a too large control action. For state variables that are not related to torque tracking, the respective cost terms are set to zero. As Q-current should be generating the majority of motor torque, it is not penalized either.
The MBUTC algorithm is illustrated in the flow chart in Figure 3.1.
Each time the LQR problem is solved, it generates a sequence of optimal actions, $[u_k^*(1),\ ...,\ u_k^*(N)]$, for the duration of the horizon. Only the action computed for the first timestep, $u_k^*(1)$, is actually applied to the PMSM. The remaining actions are discarded. At the next timestep, the optimal actions are recomputed based on new sensor measurements. This scheme is referred to as a receding horizon control (RHC) scheme, and is illustrated in Figure 3.2 [8].
This formulation is selected due to a number of desirable properties. First, it requires a minimum number of state variables, which help conserve the precious computation power of the MCU in the ANYdrive. Second, this formulation has a closed form solution, which can be computed deterministically. The controller of ANYdrive has tight real-time constraints, which makes deterministic computation time a crucial property because it ensures the timing is not violated stochastically. However, attention should be paid to potential pitfalls. As the only constraint in the LQR problem is the system dynamics, the resulting actions might not always be

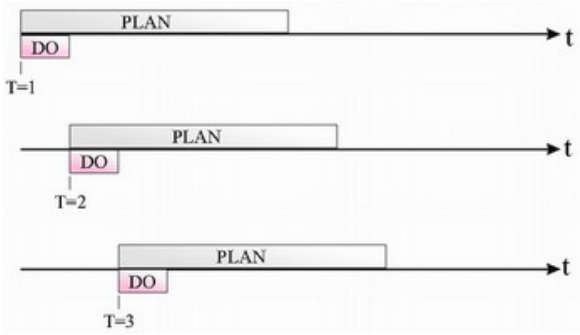

Figure 3.2: Receding Horizon Control

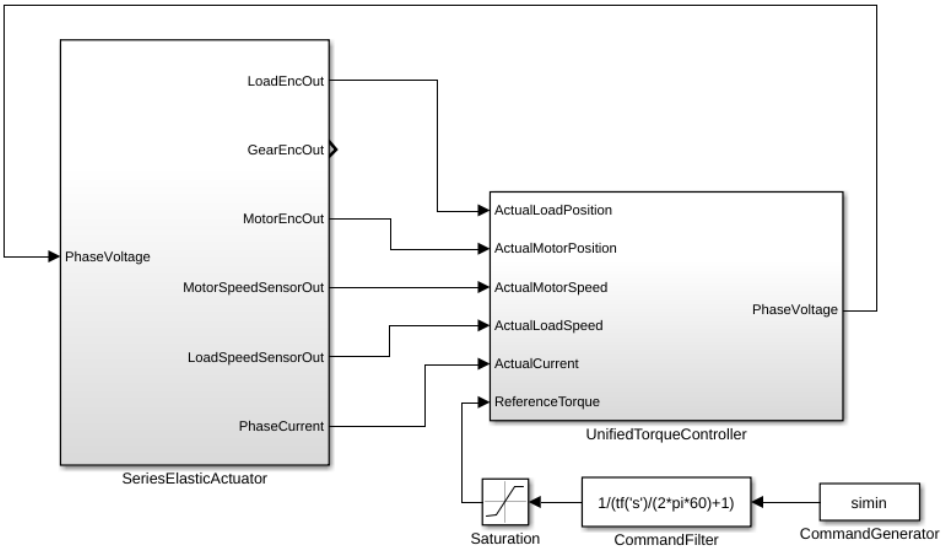

Figure 3.3: Setup of Simulation Verification

feasible, due to the constraint of the limited DC line voltage and the design of the power stage. A simple saturation function is currently used to limit the output to within the capacity of the power stage. However, when the saturation is triggered, the optimality of the solution will be lost, which could result in instability of the controller. This situation can in general be avoided by properly tuning the cost matrices.

3.2 Verification in Simulation

The MBUTC is examined in a Matlab simulation, to verify its functionality, stability, and performance. Figure 3.3 shows the block diagram of the simulation verification. The Simulink model of the ANYdrive constructed in the previous chapter is used.
The MBUTC is condifured as follows for this verification:

- Horizon: 14
- Discretization timestep: 0.1 ms
- Controller update rate: 2500 Hz

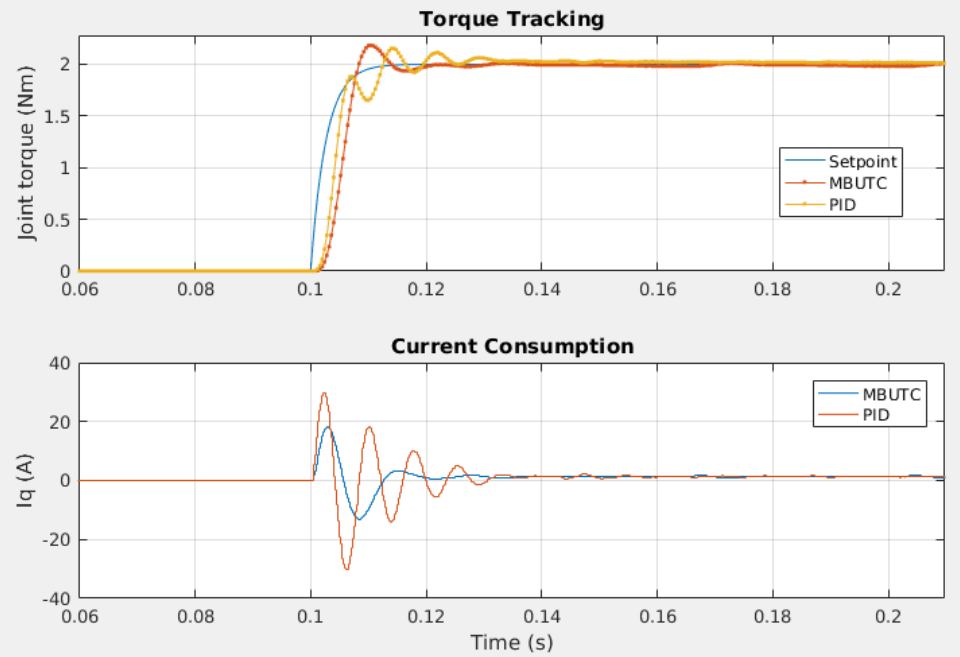

Figure 3.4: Step Response with Joint Blocked

In the first test case, the joint is assumed to be blocked, therefore no load dynamics enters the system. A step torque command is applied to the system, and the torque tracking performance is analyzed. The torque tracking performances of the MBUTC and a conventional PID controller with well tuned gains are compared in Figure 3.4. In comparison to the PIC controller, the MBUTC achieves a shorter settling time and less oscillations before settling. Meanwhile, the MBUTC consumes significantly less current than the PID controller.
In the second test case, the joint is still blocked, and a chirp torque command is applied. The bode plot of the torque controller is then obtained by taking the ratio between the discrete Fourier transformations of the actual torque and the torque command. The bode plots of the MBUTC and the PIC controller are compared in Figure 3.5. A obvious observation is that the response of the MBUTC has a higher gain in high frequency range. It corresponds to the slightly larger overshoot of this controller's step response.
In the third test case, a pendulum comprising a point mass of 15kg suspended on the tip of a rod of length 0.3m is attached to the joint to simulate a disturbance. The pendulum is initialized in the upright position, and a step torque command is applied. The pendulum will naturally be driven away from its unstable position and start to swing, and the torque tracking performance of the MBUTC under this disturbance is examined. The torque tracking performances of the MBUTC and the PID controller are compared in Figure 3.6. Under large joint movements, the MBUTC tracks torque with a slightly larger error than the PID controller, although the difference is not significant. Both controllers consume similar current during the process (Figure 3.7).

3.3 Implementation Issues in ANYdrive Firmware

The control law of MBUTC is rather computationally expensive, yet it has to be implemented as a add-on module in the firmware of the ANYdrive. The controller

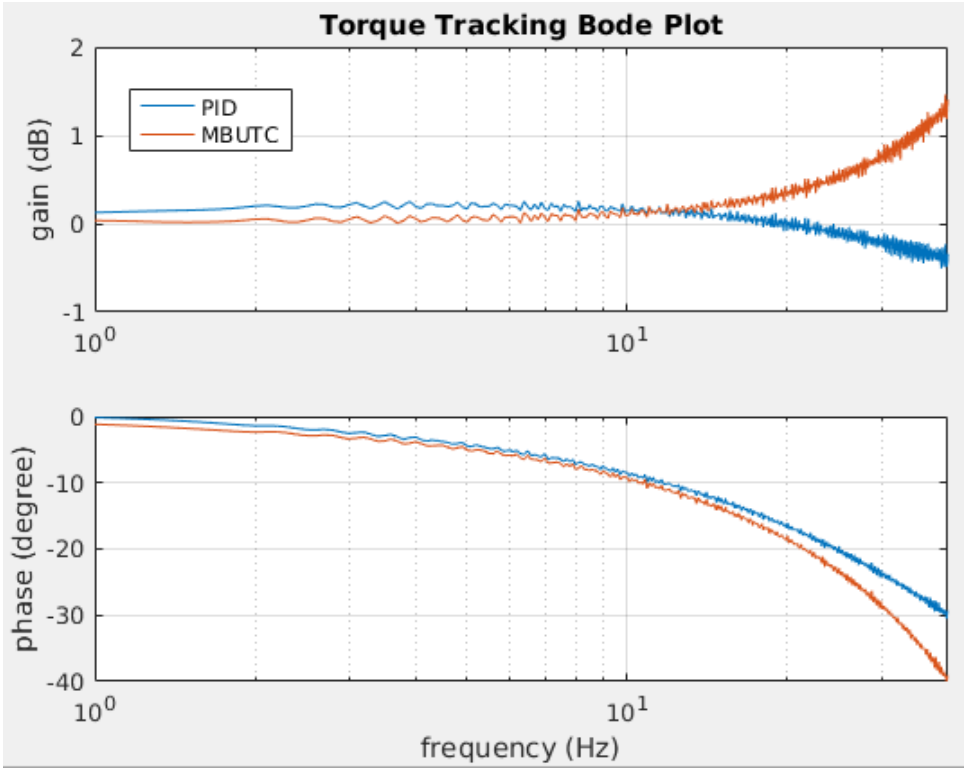

Figure 3.5: Torque Tracking Frequency Response with Joint Blocked

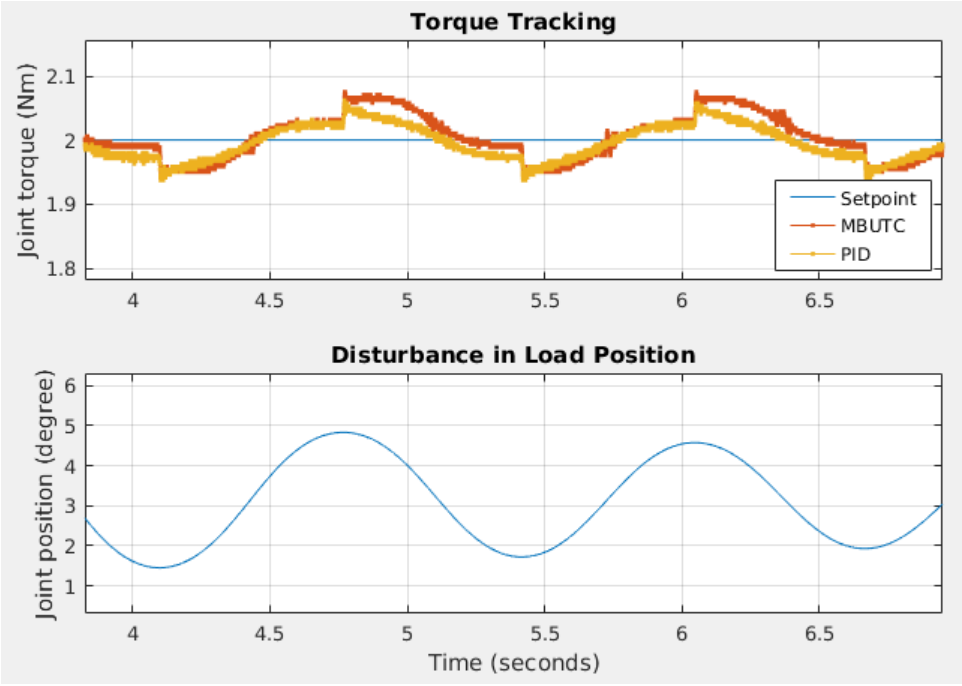

Figure 3.6: Step Response with Pendulum Load

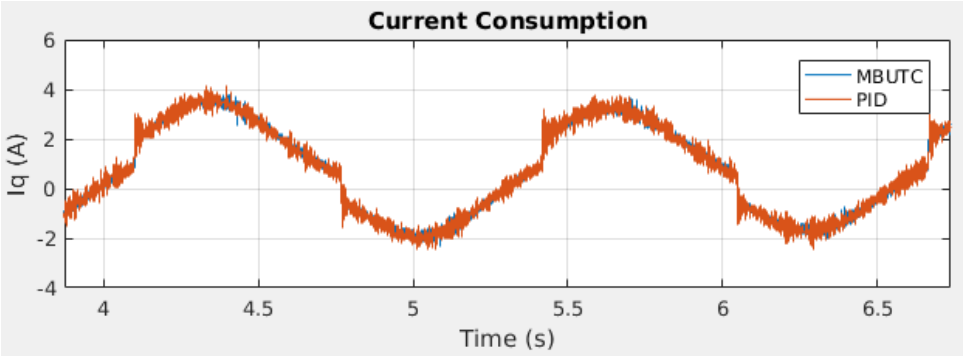

Figure 3.7: Current Consumption during Step Response with Pendulum Load

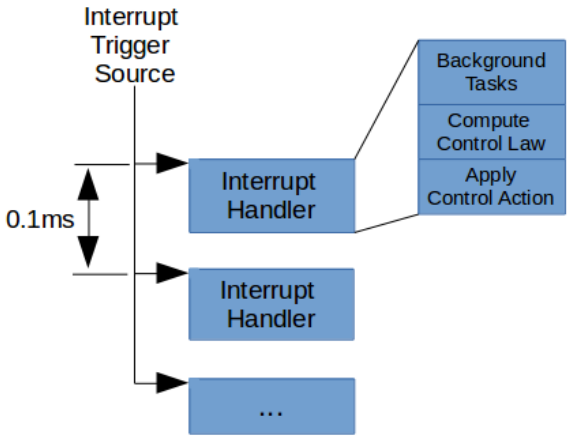

Figure 3.8: Execution Timing of the Firmware

can then be executed from the MCU in the ANYdrive. The ANYdrive is equipped with a ARM Cortex-A9 MCU, which executes the firmware on a strictly real-time basis. In order not to violate the real-time constraints of the firmware, the control algorithm is optimized for computational efficiency, and a unique task scheduling scheme is developed to effectively utilize the computational power of the MCU.
The firmware of the ANYdrive has a *bare-metal* implementation, which means there is no RTOS in place to handle task scheduling. The functions that compute the control laws are called from a interrupt handler function, which is assigned to a interrupt that is triggered every 0.1ms. A number of background tasks, such as communication, message passing, and sensor sampling, also take place during this 0.1ms time slice (Figure 3.8). As a result, the computation of the control law, along with the background tasks, should complete within 0.1ms. The real-time constraint of the firmware will be violated otherwise, and the next execution cycle will be delayed, resulting in latency in control action application.
The computation of the control law is first analyzed, and the computation of sparse matrix multiplications are manually optimized. When symmetric matrices are present, the symmetry is exploited to conserve computation power.
The control law computation of the MBUTC, when implemented as a C function with manually optimized computations and compiled with compiler optimization option enabled, takes approximately 0.1ms to execute. Together with the background tasks, it results in the violation of the 0.1ms real time constraint of the firmware. Simulation verification has shown the adequacy of updating the con-

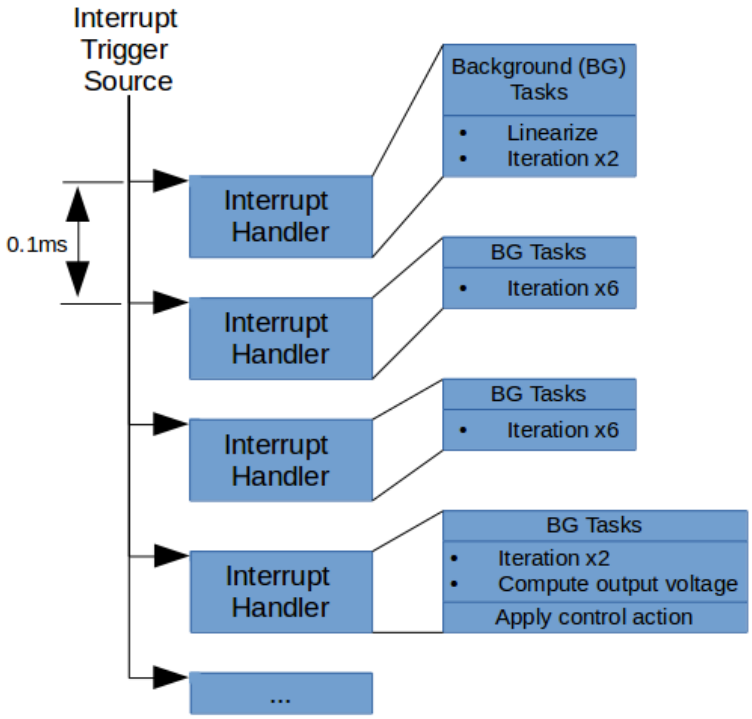

Figure 3.9: Hard Task Scheduling Scheme for Control Law Computation

troller at a frequency of only 2500Hz, leaving behind approximately 0.4ms for the computation of the control law. However, the absence of a RTOS makes it difficult to utilize computational power across multiple 0.1ms time slices. In order to solve this problem, a hard task scheduling scheme is developed for the MBUTC, which enables the distribution of the computation across multiple time slices.
First, 3 consecutive and separable steps of the computation of the control law are identified:

- Linearization of the model
- Dynamic programming iterations to solve the LQR problem
- Computation of the output voltage

The iteration to solve the LQR problem is computed 14 times, which can also be conducted separately.
Second, the computation is broken up into 4 segments, each of which computes on part of the control law. The segments are then executed in 4 consecutive 0.1ms time slices. A counter is used to keep track of the order of execution of the segments. The counter is increased by 1 every time slice, and rolls back to 0 once it reaches 4. This scheme is illustrated in Figure 3.9.
The distribution of computation is done manually to ensure that the execution time of each segment is well below 0.1ms.
With optimized sparse matrix multiplication, the hard task scheduling scheme and compiler optimization, the MBUTC algorithm can be computed in the firmware of the ANYdrive without violating the real-time constrains, and output voltage is updated every 0.4ms (i.e. at 2500Hz).

Chapter 4

Parameter Identification

The identification of parameters of the ANYdrive model is discussed in this chapter. Focus will be placed on the identification of the parameters of the mechanical system. Electrical system parameters are taken from the data sheet of the PMSM, except for the torque constant, which will also be identified experimentally.
Two experiments are conducted. The first one retrieves the response of the ANYdrive to a chirp torque with the joint blocked, which enables the identification of multiple parameters using a modified linear regression method. The second one is the so-called break-away test (see [9]), which determines the value of maximum static friction.

4.1 Parameters of the Mechanical System

This section elaborates the identification of the following parameters:

- Viscous damping coefficient, PMSM (b_m)
- Torque constant of the PMSM (K_τ)
- Inertia of the PMSM together with the gearbox (J_m)

To identify the parameters, the joint of the ANYdrive is first blocked. A chirp excitation signal is applied as the desired i_q of the PMSM. The FOC is in charge of tracking this setpoint. A sinusoidal torque waveform with increasing frequency is generated by the PMSM as a result. Meanwhile, the position (ϕ_m) and velocity ($\dot{\phi}_m$) measured by the encoder of the motor are recorded. Typical waveforms of the test are shown in Figure 4.1.
Using the data collected from the chirp excitation test, parameters of the ANYdrive are identified using a modified linear regression method. First, a gray-box model with known structure and unknown parameters is constructed. this model is derived from the full model of the ANYdrive, while the relatively irrelevant electrical dynamics of the PMSM is dropped out. This is a valid simplification because the FOC is used to control i_q and i_d. If the closed loop bandwidth of the FOC is significantly higher than the maximum frequency component of the chirp, the PMSM can be approximated using a perfect torque source:

$$\tau_m = K_\tau i_q \tag{4.1}$$

The gray-box model for parameter identification is defined as follows:

$$\begin{bmatrix} \dot{\phi}_m \\ \ddot{\phi}_m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{J_m {n_g}^2 \eta} & -\frac{c_m}{J_m} \end{bmatrix} \begin{bmatrix} \phi_m \\ \dot{\phi}_m \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_\tau}{J_m} \end{bmatrix} i_q + \begin{bmatrix} 0 \\ \frac{-F_s}{J_m} \tanh(\frac{2.09}{\omega_{bk}} \dot{\phi}_m) \end{bmatrix} \tag{4.2}$$

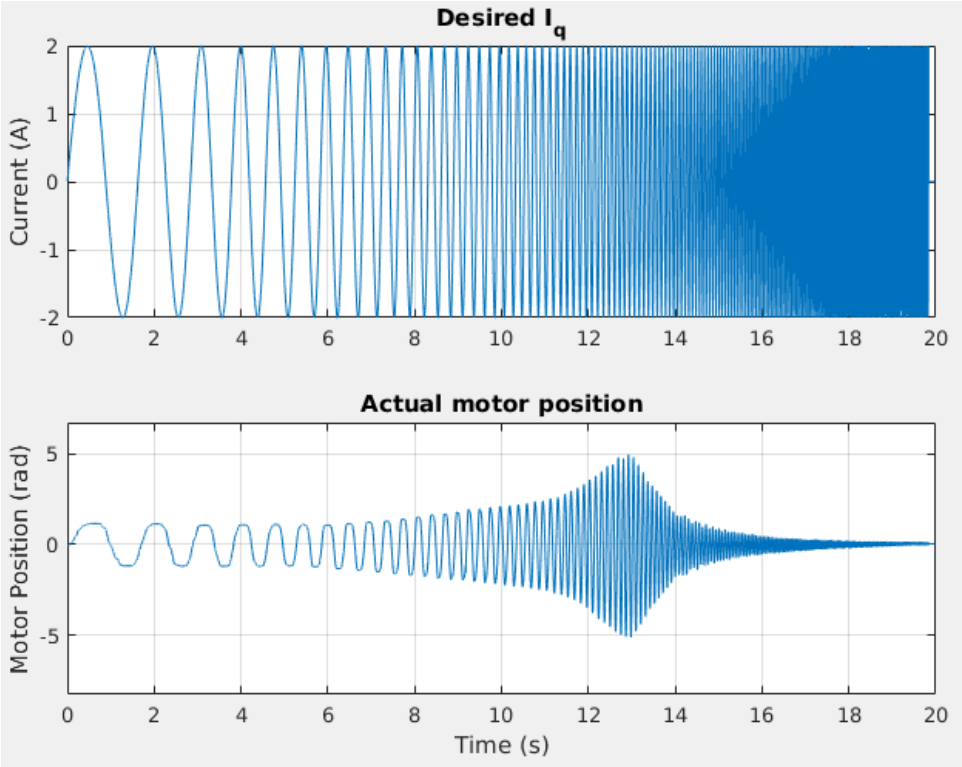

Figure 4.1: Chirp Excitation Test Typical Waveform

The model is then discretized, given timestep dt, using first order approximation:

$$\begin{bmatrix}\phi_m \\ \dot{\phi}_m\end{bmatrix}_{k+1} = \begin{bmatrix}a_{11} & a_{12} \\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}\phi_m \\ \dot{\phi}_m\end{bmatrix}_k + \begin{bmatrix}b_1 \\ b_2\end{bmatrix} i_q + \begin{bmatrix}c_1 \\ c_2\end{bmatrix} \tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}_m) \tag{4.3}$$

where

$$\mathbf{A_{id}} = \begin{bmatrix}a_{11} & a_{12} \\ a_{21} & a_{22}\end{bmatrix} \simeq \begin{bmatrix}1 & dt \\ -\frac{k_s}{J_m {n_g}^2 \eta}dt & 1 - \frac{c_m}{J_m}dt\end{bmatrix} \tag{4.4}$$

$$\mathbf{B_{id}} = \begin{bmatrix}b_1 \\ b_2\end{bmatrix} \simeq \begin{bmatrix}0 \\ \frac{K_\tau}{J_m}dt\end{bmatrix} \tag{4.5}$$

The parameters c_1 and c_2 are not part of the parameters to be identified. Instead they are used to compensate for the effect of the nonlinear friction, which is modeled by a hyperbolic tangent function. The value of ω_{bk} should be the same as that of the full model. It takes advantage of the fact that the nonlinear friction term is linear in terms of $\tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}_m)$, which enables us to treat the nonlinear friction as part of the linear regression problem. The actual value of maximum static friction, however, is identified in a separate experiment later on.

The linear regression problem is now constructed. The regressor matrix is defined as:

$$\mathbf{\Phi}(k) = \begin{bmatrix}\phi_m(k) & \dot{\phi}_m(k) & i_q(k) & \tanh(\frac{2.09}{\omega_{bk}}\dot{\phi}_m(k))\end{bmatrix} \tag{4.6}$$

The parameter matrix is defined as:

$$\mathbf{\Theta} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix} \tag{4.7}$$

The prediction matrix is defined as:

$$\mathbf{Y}(k) = \begin{bmatrix} \phi_m(k+1) & \dot{\phi}_m(k+1) \end{bmatrix} \tag{4.8}$$

The parameter matrix can now be solved using the standard linear regression formula:

$$\mathbf{\Theta} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{Y} \tag{4.9}$$

This process yields direct estimation of $\mathbf{A_{id}}$ and $\mathbf{B_{id}}$. The physically meaningful parameters of the ANYdrive are then calculated using equations 4.4 and 4.5. In order to obtain the values of b_m, K_τ and J_m, it is further assumed that the spring constant, k_s, is known. From statistics of quality control tests of the ANYdrive, $k_s = 180 Nm/rad$ is used.
To further improve the quality of parameter estimation, the test is repeated 5 times, and the mean values of parameters identified from the repeated tests are used as the final estimation.
The following configurations are used for the test:

- Amplitude of chirp excitation: 2A
- Frequency range of the chirp excitation: 0.5Hz to 40Hz
- Sampling frequency: 1000Hz

The identified parameters are shown in Table 4.1

Table 4.1: Maximum Static Friction

Parameter	Symbol	Value
PMSM Viscoud Damping	b_m	$2.8 \times 10^{-4}\ Nms/rad$
PMSM Torque Constant	K_τ	$0.056\ Nm/A$
PMSM Effective Inertia	J_m	$2.6 \times 10^{-5}\ kgm^2$

The bode plot of the motor position response is shown in Figure 4.2.

4.2 Static Friction

Breakaway test is conducted to determine the breakaway torque of the motor and the joint, which provides a good approximation to the maximum static friction, F_s. The breakaway torque is defined as the torque required for the component of concern to start rotating. In a breakaway test, the Q-axis current of the motor, i_q, is ramped up at a predefined rate, starting from zero, while the D-axis current, i_d, is controlled to zero by the FOC. This generates a linearly increasing torque. The velocity of the motor and the joint are constantly monitored. If the velocity rises above a predefined threshold, a breakaway event is confirmed, and the corresponding i_q and joint torque are recorded. When breakaway events are confirmed for both the motor and the joint, the test is completed, and i_q is reset to zero. The typical current waveform during the test is shown in Figure 4.3.

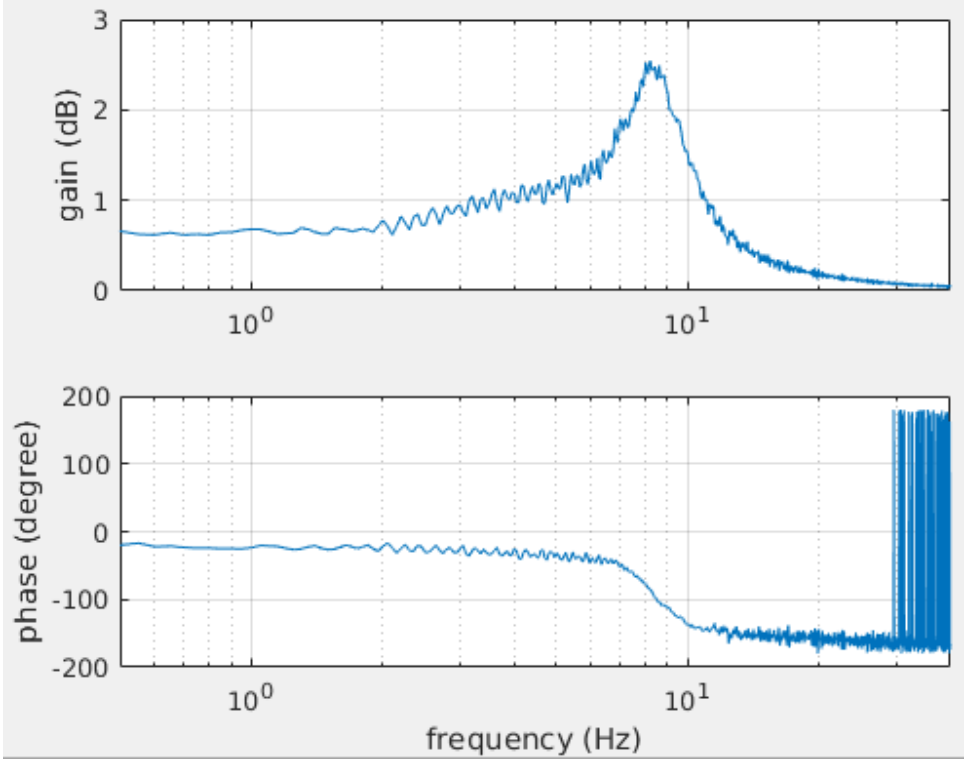

Figure 4.2: Bode Plot of Motor Position Response

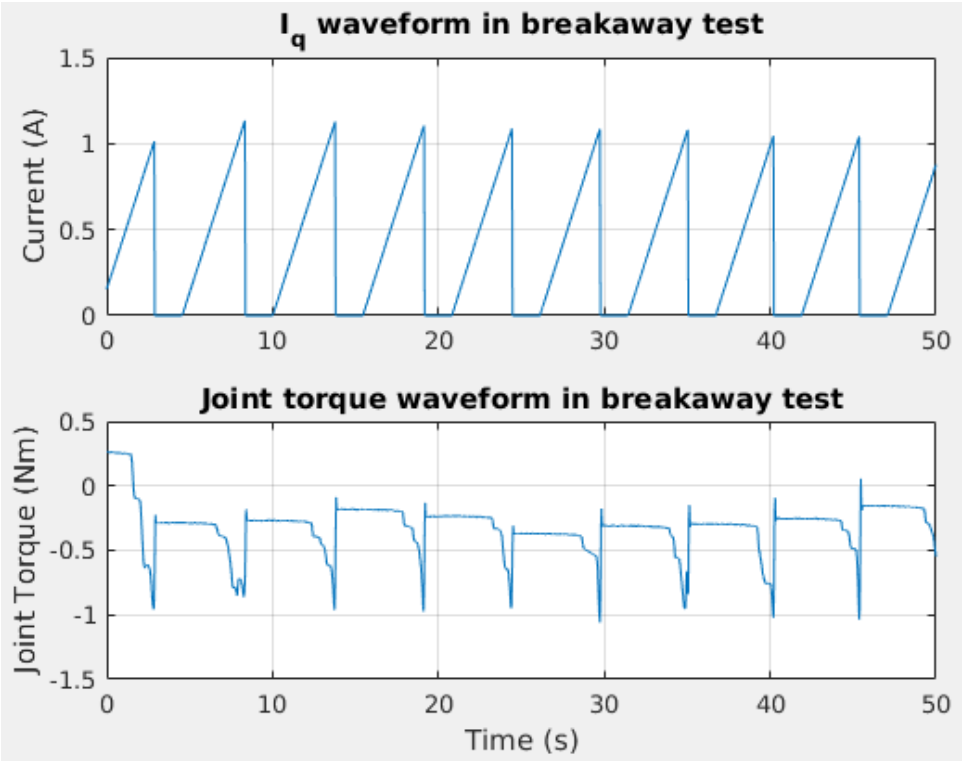

Figure 4.3: Breakaway Test Typical Waveform

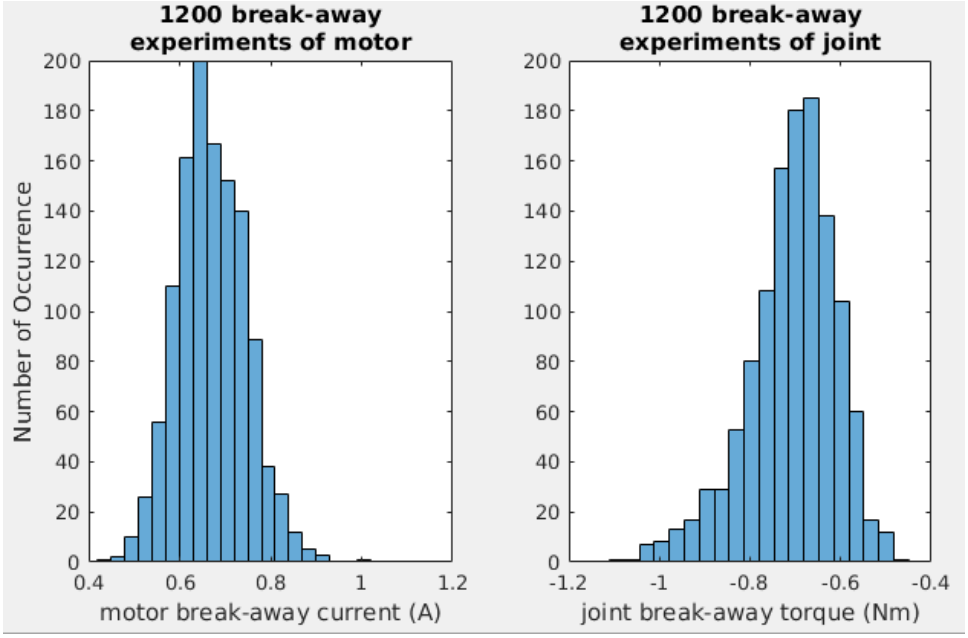

Figure 4.4: Breakaway Current and Torque Histogram

The breakaway test is repeated for a large number of times. Due to the fact that static friction can be position dependent, it is important to conduct the test at different joint positions that are evenly distributed in a full revolution of the joint. As mentioned in [9], every time a breakaway event occurs, the joint is moved forward by a small distance. Therefore, if the test is simply repeated many times, breakaway torque will eventually be sampled at roughly evenly distributed positions along a full revolution. In addition, in order to prevent noise in velocity measurement to be identified as a false breakaway event, it is required that the velocity be larger than the threshold for at least 2ms.
Prior to the start of the test, the ANYdrive is first "warmed up". The ANYdrive is rotated in both directions, for at least 3 minutes in each direction, through the application of a constant Q-axis current. The purpose is to increase the temperature of the drive, in particular the temperature of the lubricant in bearings, so the identified breakaway torques better approximates those under nominal operating conditions.
The following configurations are used for the breakaway test:

- i_q ramp-up rate: 0.3A/s
- Breakaway velocity threshold, PMSM: 1 RPM
- Breakaway velocity threshold, Joint: 1 RPM
- Number of repetitions: 1200

The histogram shown in Figure 4.4 shows the result of a sequence of 1200 breakaway tests. Both the i_q value at the breakaway of the PMSM and the joint torque at the breakaway of the joint are recorded.
Using the PMSM torque constant identified earlier, the static friction of the PMSM can be computed:

$$F_s = mean(I_{q-bk})K_\tau \tag{4.10}$$

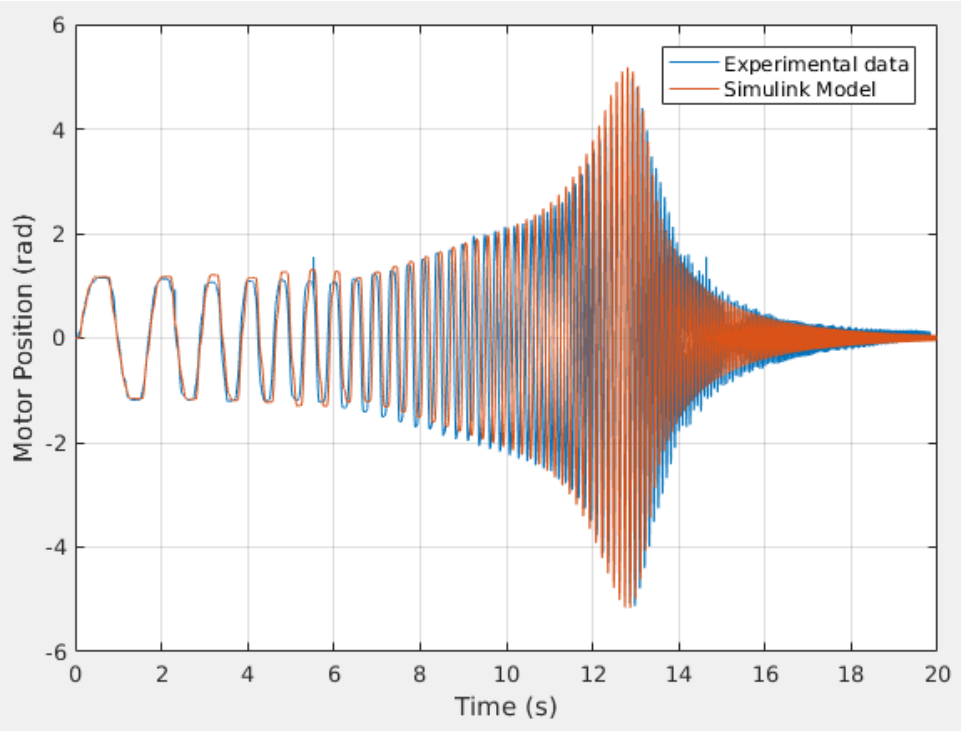

Figure 4.5: Time-domain Waveform of PMSM Position Response

The absolute value of joint torque at breakaway directly reflects the value of maximum static friction at joint.
The maximum static frictions of the PMSM and the joint, according to the test, are summarized in Table 4.2.

Table 4.2: Maximum Static Friction

	Static Friction	Standard Deviation
PMSM	0.037 Nm	0.076
Joint	0.71 Nm	0.098

4.3 Verification of the Full Model

After the identification of parameter, the full model of the ANYdrive derived in chapter 2 is verified against experimental results. The identified parameters are first introduced into the Simulink model of the ANYdrive. Next, a chirp excitation signal identical to the one used in section 4.1 is applied. The resulting motor position given by the simulation is compared to the actual motor position measured in a experiment.
The results are shown in Figures 4.5 and 4.6. The Simulink model gives results that qualitatively match with experimental data, which indicates that the model effectively captures the behavior of the real system. This proves the validity of the model as a virtual testbench for control algorithms.

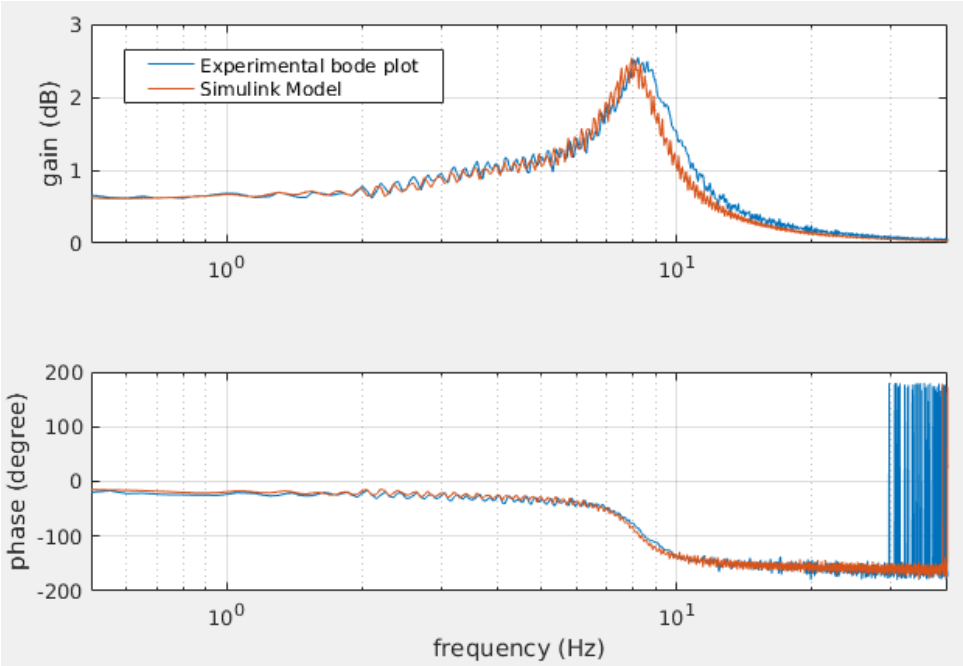

Figure 4.6: Bode Plot of PMSM Position Response

Chapter 5

Experimental Verification of the Unified Torque Controller

The performance of the MBUTC is examined experimentally in a few different test cases. The performance is compared with that of a PID controller, which is currently the default controller of the ANYdrive and serves as a benchmark. In this chapter, the setup of the experiments is discussed, and the experimental results are presented and analyzed.

5.1 Experiment Setup

Figure 5.1 shows the hardware setups of the experiments. In the first setup, the joint of the ANYdrive is blocked. In the second setup, a pendulum is attached to the joint as a load. The inertia of the pendulum is approximately 0.1 kgm^2. The exact inertia of the load does not play an important role, as the goal is to test the controller's ability to track torque regardless of joint load.
The ANYdrive is connected to the PC via a EtherCAT interface. On the PC side, dedicated ROS nodes handle communication between user applications and the ANYdrive. Figure 5.2 shows the GUI for the ANYdrive (ANYdrive Studio), and Figure 5.3 shows the GUI to send custom parameters to the MBUTC (Parameter Handler). Process data from the ANYdrive can also be plotted in real time using the Rqt Multiplot Plugin, which is shown in Figure 5.4.
The above mentioned GUIs are started using ROS launch:

```
roslaunch anydrive_test_rhlqr anydrive.launch
roslaunch anydrive_test_rhlqr aduc.launch
```

The procedure to set up an experiment is described as follows:

- Load the firmware into ANYdrive using CCS and the debugger.
- Launch the ANYdrive Studio, the Rqt Multiplot, and Parameter Handler.
- In the ANYdrive Studio ensure the [FSM State] is set to [Configure].
- In the Parameter Handler, enter the appropriate parameters for the MBUTC.
- In the Parameter Handler, set [en_mpc_comp] to one, and set [en_mpc_output] to zero.

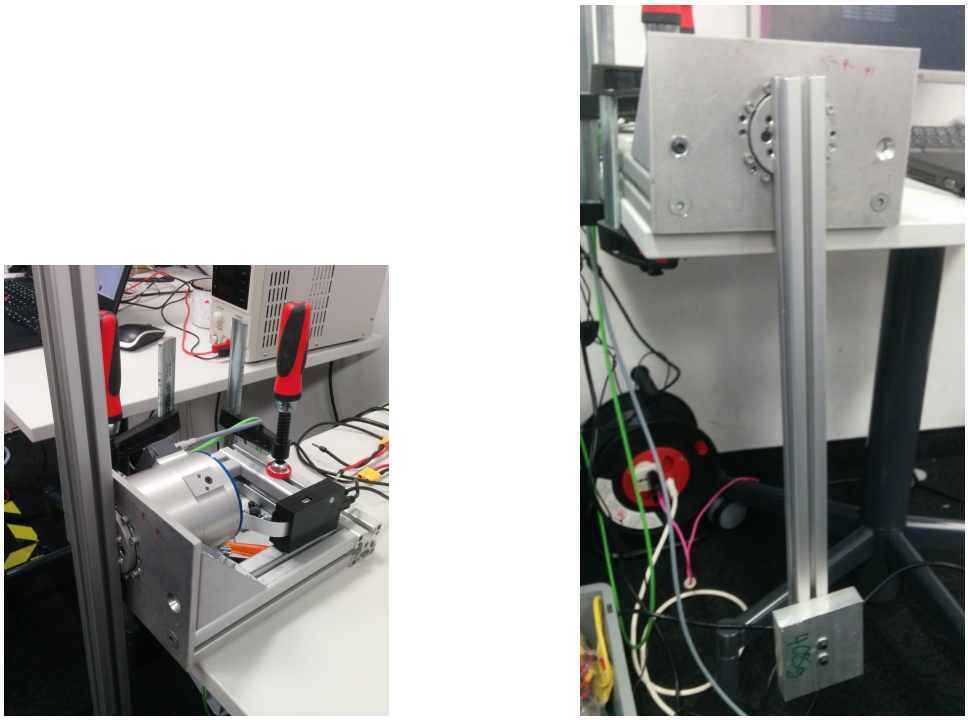

Figure 5.1: Hardware Setup, Blocked Joint (left) and Pendulum Load (right)

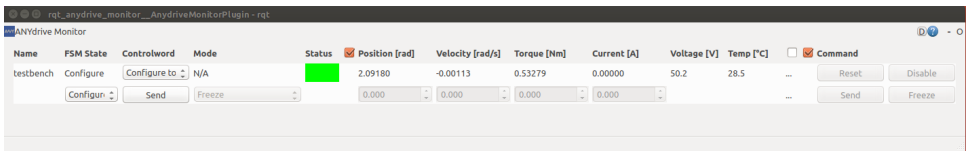

Figure 5.2: The ANYdrive Studio GUI

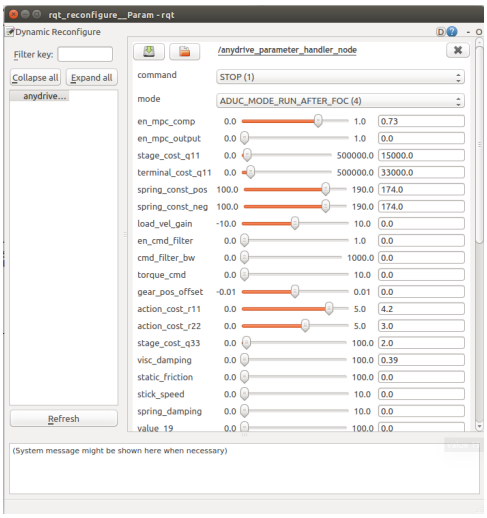

Figure 5.3: The Parameter Handler GUI

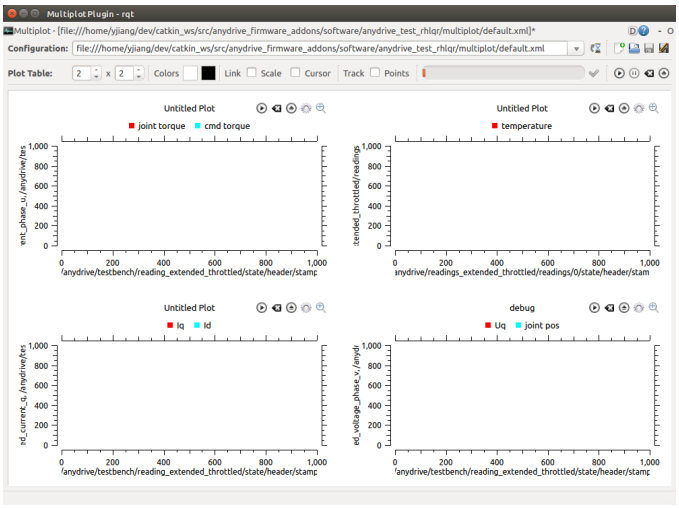

Figure 5.4: The Rqt Multiplot GUI

- Carry out safety check.
- In the ANYdrive Studio, set [FSM State] to [ControlOp].
- In the ANYdrive Studio, set [Mode] to [Current], and set desired current to zero.
- If ready to start, then in the Parameter Handler, set [Command] to [Start(0)].
- In the Parameter Handler, set [en_mpc_output] to one. The voltage computed by the MBUTC is then applied to the motor.
- To apply changes in parameters, first set [Command] to [Stop(1)], and then set it to [Start(0)] again.

5.2 Blocked Joint Step Response

The response of the MBUTC to a step torque command is tested with the joint blocked. The performance of the MBUTC is compared to that of a PIC controller in Figure 5.5. As can be seen, the MBUTC results a longer settling time and a larger overshoot than the PID controller, while both consumes simular current.
In addition, although less obvious in the figure, the MBUTC does not eliminate steady state tracking error. Instead, when the joint is slightly perturbed to make the actual torque deviate from the setpoint, the MBUTC does not generate motor motion to eliminate the perturbation.

5.3 Blocked Joint Chirp Response

With the joint blocked, a chirp signal is applied as torque command, and the closed-loop frequency response of the torque controller is measured. The following configurations are used:

- Amplitude of chirp excitation: 1.5 Nm
- Frequency range of the chirp excitation: 0.2Hz to 30Hz

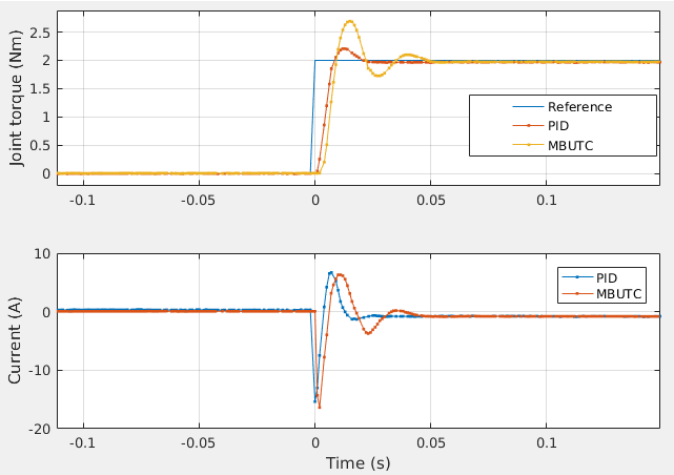

Figure 5.5: Step Response with Joint Blocked

- Sampling frequency: 1000Hz

The closed-loop torque tracking bode plot of the MBUTC is compared with that of the PID controller in Figure 5.6. The MBUTC has similar performance to the PID controller at low frequency range. However, the resonance of the MBUTC at higher frequencies is an undesirable behavior.
The instantenous power consumptions of the controllers are compared in Figure 5.7. Instantaneous power of the PMSM is defined as:

$$P = i_d u_d + i_q u_q \tag{5.1}$$

The MBUTC is shown to have a slightly lower power consumption. PID achieves an average power consumption of 0.87W during the test, while MBUTC manages with 0.84W.

5.4 Step Response with Pendulum Load

In this test, a step torque command is applied to the ANYdrive when a pendulum is attached to the joint as a load. The pendulum starts at its stable equilibrium position. When a torque is applied, the pendulum will move in the direction of the torque. Dynamic torque tracking capability of the controller can then be verified.
The performance of the MBUTC and the PID controller are compared in Figure 5.8, while the movement of the joint when MBUTC is used is shown in Figure 5.9. Despite relatively large joint motion, MBUTC successfully tracks torque command. However, similar to the case where the joint is blocked, the MBUTC produces a higher overshoot than the PID controller.

5.5 Analysis

Experiments conducted so far show that the MBUTC is able to achieve its basic functionalities. However, performance wise the PID controller is better in practice. The problems that currently exist with the MBUTC are summarized as follows:

- Unable to eliminate steady state error
- Unable to track small disturbances close to the setpoint

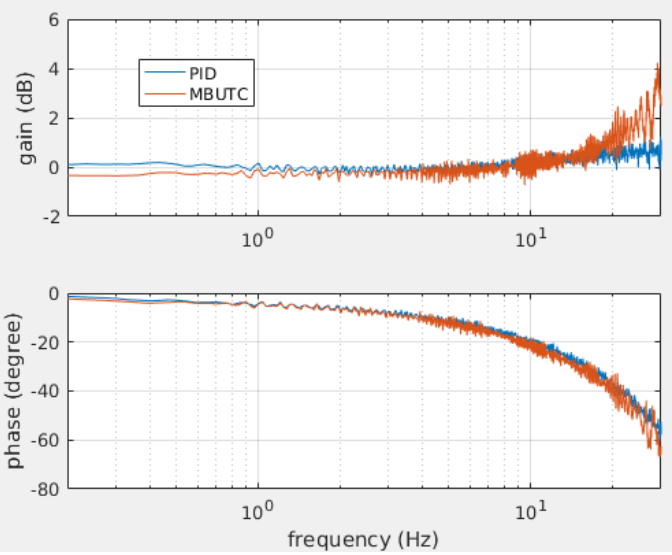

Figure 5.6: Closed-loop Frequency Response with Joint Blocked

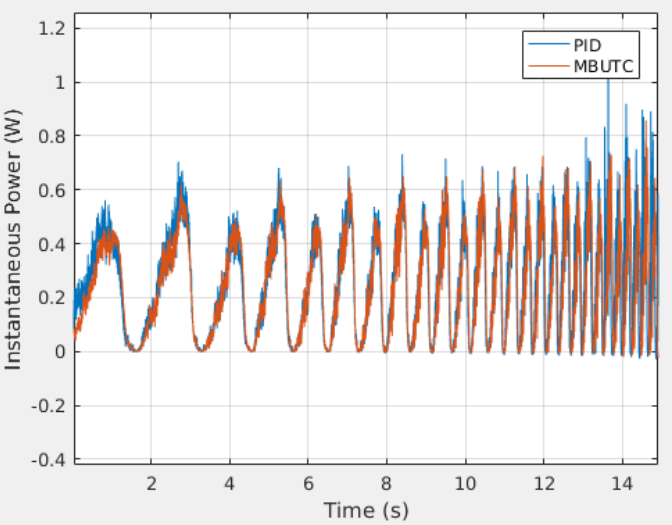

Figure 5.7: Power Consumption during the Frequency Response Test

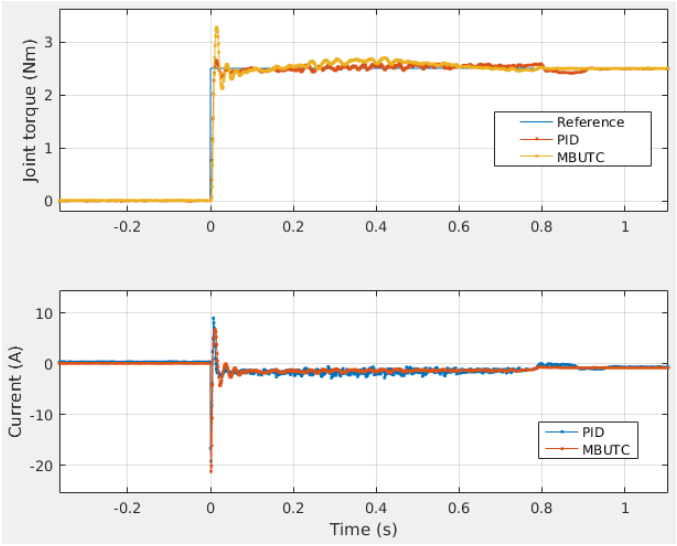

Figure 5.8: Step Response with a Pendulum as Load

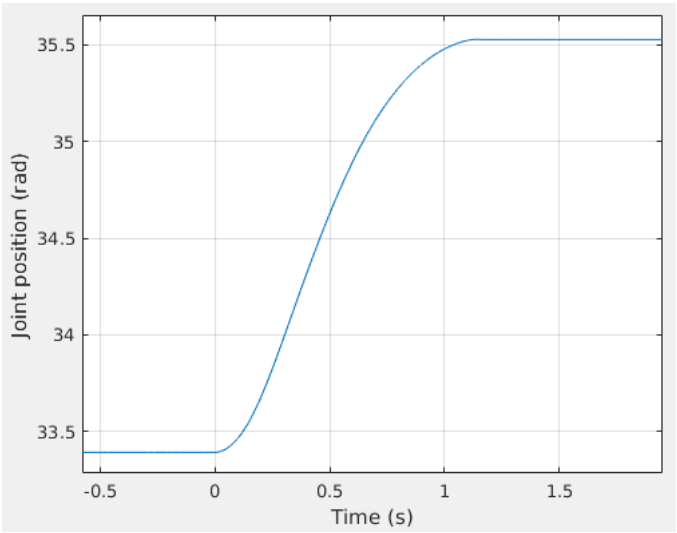

Figure 5.9: Joint Position During Step Response Test (when MBUTC is used)

- Relatively high overshoot
- When joint speed is high, torque tracking performance worsens.

The poor performance of the MBUTC in eliminating steady state error might be related to two facts. First, the MBUTC employs a finite horizon LQR algorith, for which the cost of applying higher voltage to eliminate steady state error could be higher than the cost of maintaining the current position. The formulation does not result in the continuous accumulation of cost when the target is not reached. Second, the the actual static friction effect is discontinuous, which is different from the model. In the real system, the torque must rise above a certain threshold for the PMSM to start moving. Meanwhile the model only applies a larger viscous damping when the speed is low. The sticking effect of the static friction is not accurately modeled.

Chapter 6

Conclusion

This chapter first gives a brief overview of the accomplishments in this work. Reflections on the outcomes and possibilities of future developments are then discussed.
The main objective of this work is to first develop a full model of the ANYdrive, and then to develop and test a novel model-based torque controller for the ANYdrive. The following results have been delivered:

- Derivation of the ANYdrive full model
- Implementation of the full model in Simulink and as a C module
- Derivation of the system model for controller design
- Formulation and verification of the MBUTC
- Optimization for execution with real-time constrains
- Parameter identification and verification of the full model against experiment.
- Experimental verification of the MBUTC

The full model of the ANYdrive developed in this work is proven to faithfully recreate the dynamics of the real system, which enables its use as a virtual testbench for control algorithms and/or as a component for larger scale simulations.
Simulation results show that the MBUTC is able to provide superior performance to the PIC controller in certain aspects. The feasibility of implementing this computationally expensive control algorithm in a real-time system is also verified. However, the MBUTC produces less optimal performance than the PID controller in experiments in the real system. Namely, it exhibits poorer torque tracking ability than the PID controller under disturbance, and it does not eliminates steady state error. A number of issues with the MBUTC could have contributed to this results:

- Finite horizon of the LQR
- Inaccurate modeling of the sticking effect at low speed
- Lack of integrator action to eliminate steady state error

To further improve the performance of the MBUTC, a possibility for future work is the introduction of a disturbance observer to the algorithm, which is likely to increase the performance of the MBUTC against disturbance. The short horizon of the MBUTC and the inaccurate modeling contribute together to the steady state error and relatively poor dynamic tracking performance. A well-tuned disturbance

observer observes these effects collectively as a disturbance torque, which possibly compensates for the short-sightedness of the MBUTC.
Simulation results show the superiority of the MBUTC in responding fast to set-points with smaller current consumption, which gives it certain advantages over the traditional PID controller. However, this property is yet to be demonstrated experimentally. Considering in addition the complexity of this algorithm, the MBUTC is yet to be qualified as a substitute for the PID controller, and further improvements are certainly needed.

Bibliography

[1] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995, pp. 3137–3181.

[2] H. Vallery, R. Ekkelenkamp, H. van der Kooij, and M. Buss, "Passive and Accurate Torque Control of Series Elastic Actuators," *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3534–3538, 2007.

[3] S. Emre, C. Gong, and Y. Haoyong, "An active disturbance rejection controller design for the robust position control of series elastic actuators," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 266–272, 2016.

[4] N. Paine and S. M. Joshua, "Actuator control for the nasa-jsc valkyrie humanoid robot: A decoupled dynamics approach for torque control of series elastic robots," *J Field Rob*, pp. 378–396, 2015.

[5] http://www.anybotics.com/anydrive/.

[6] D. Ocen, "Direct torque control of a permanent magnet synchronous motor," 2005.

[7] P. Brown and J. McPhee, "A continuous velocity-based friction model for dynamics and control with physically meaningful parameters," *Journal of Computational and Nonlinear Dynamics*, 2016.

[8] http://people.ee.ethz.ch/~cohysys/control.php.

[9] B. Armstrong-Hélouvry, *Control of Machines with Friction*. Springer US, 1991.

Appendix A

ANYdrive Poster

ANYbotics

ANYdrive robot joint

Integrated, robust, torque controllable

Fully integrated

ANYdrive consists of a powerful brush-less motor, custom spring, backlash-free gear, high-precision encoders, and efficient power electronics.

Absolute position sensing

Precise absolute encoders make repeated calibration of the joints unnecessary.

Programmable controller

Custom control algorithms can be implemented through the open API (coming soon).

Accurate position & torque control, impact robustness

The intergrated spring enables accurate torque tracking while protecting the gear from impacts.

High load bearing & hollow shaft

The robust design and hollow shaft allow for compact robot design and optimal cable routing.

Ingress-protection IP67

The ANYdrive is completely sealed against dust and water ingress.

ANYdrive is a complete robot joint: It enables building robots of any form with minimal complexity and maximal performance.

Example applications

Peak/nominal power	720 W / 240 W	**Absolute joint position**	18-bit, ‹0.025°
Nominal voltage	48 VDC	**Joint output torque resolution**	‹0.1 Nm
Peak/nominal torque	40 Nm / 15 Nm	**Torque control bandwith**	›60 Hz
Peak joint velocity	114 rpm (12 rad/s)	**Backlash**	±0.02°
Dimensions (L x D)	95 x 90 mm (w/o connector housing)	**Control modes**	Position, torque, impedance, velocity, or current control
Mass	1.0 kg	**Communication**	EtherCAT, ROS integration
Max. bending moment	120 Nm		
Hollow shaft diameter	17 mm		