

# Identification and Compensation of Parasitic Effects in Coreless Linear Motors

## PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een  
commissie aangewezen door het College voor  
Promoties, in het openbaar te verdedigen op  
maandag 1 oktober 2018 om 11:00 uur

door

Tuan Trong Nguyen

geboren te Hanoi, Vietnam

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr.ir. J.H. Blom  
1<sup>e</sup> promotor: prof.dr.ir. P.M.J. Van den Hof  
2<sup>e</sup> promotor: prof.dr.ir. H. Butler  
copromotor: dr. M. Lazar  
leden: prof.dr. M. Diehl (Albert-Ludwigs-Universität Freiburg)  
prof.dr. M. Gilson-Bagrel (Université de Lorraine)  
prof.dr. H. Nijmeijer  
prof.dr. E.A. Lomonova MSc

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

# Identification and Compensation of Parasitic Effects in Coreless Linear Motors

Tuan Trong Nguyen

Identification and Compensation of Parasitic Effects in Coreless Linear Motors  
by Tuan Trong Nguyen.

Eindhoven: Technische Universiteit Eindhoven, 2018. Proefschrift.

A catalogue record is available from the Eindhoven University of Technology Library  
ISBN: 978-90-386-4586-5

This thesis was prepared with the  $\LaTeX$  documentation system.

Cover design: Tuan Trong Nguyen and Bregje Jaspers.

Printed by: Gildeprint, Enschede, The Netherlands.

Copyright © 2018 by Tuan Trong Nguyen. All rights reserved.

# Summary

## **Identification and compensation of parasitic effects in coreless linear motors**

Coreless linear motors (CLMs) are widely used in industrial positioning systems. Future generations of industrial positioning systems are facing with increasing demand in precision and throughput. As a result, CLMs are required to operate at high acceleration while maintaining high accuracy. This is a challenging problem as high acceleration generally amplifies the parasitic effects and thus reduces the accuracy of the motors. Therefore, in order to achieve high acceleration and high accuracy simultaneously, compensation for parasitic effects in CLMs has to be addressed.

This thesis considers the problem of compensation for parasitic effects in CLMs from a control perspective. The thesis presents a comprehensive approach to accurate data-driven modeling and model-based optimal control and commutation of CLMs. The five main contributions of this thesis are summarized below.

Firstly, this thesis provides an analysis of the main parasitic effects present in a CLM, not only in the driving direction but also in non-driving directions. The derivation of first-principle models of the main parasitic effects is presented using available Fourier modeling technique, under the assumption that the exact motor parameters are known.

Secondly, to eliminate the need for exact knowledge of the physical parameters of the CLMs, a data-driven modeling method to identify the model of a CLM from measurement data is introduced. The main idea of the method is to fit the model structure obtained from first-principle modeling to the measurement data, thereby obtaining the model's parameters. This presents a challenging problem to closed-loop system identification, due to position-dependent static nonlinearities, which is approached using the instrumental variable identification framework with bias correction.

The third contribution is a new optimal commutation method. The commutation is formulated as an optimization problem which delivers the desired driving force while minimizing the power loss in the coils and parasitic forces in non-driving directions. Fast methods for solving the commutation optimization problem are

developed and analyzed.

The fourth contribution is the design and implementation of a nonlinear model predictive control (MPC) algorithm for CLMs. The position controller and commutation are combined into one single nonlinear MPC controller which is capable of delivering excellent tracking performance while guaranteeing that physical constraints are satisfied. The advantages of the proposed nonlinear MPC controller compared to linear MPC and commutation is analyzed.

The last contribution is a fast optimization solver for solving MPC and optimal commutation which is capable of dealing with the stringent timing constraints of real-time CLMs. The solver is a variant of sequential quadratic programming which allows an arbitrary positive (semi-) definite Hessian approximation. It is proven that the solver has guaranteed convergence without the need for a good Hessian approximation, thereby reducing the online computational demand and allowing real-time application on CLMs.

The theoretical results are complemented by experimental tests of the developed methods for identification, control and commutation of CLMs, using a high-tech industrial CLM which is used for long-stroke positioning of reticle stage in ASML lithography machines. The experimental CLM prototype was developed in collaboration with ASML

# Contents

<b>Summary</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Coreless linear motors and applications . . . . .	1
1.2 Parasitic effects in coreless linear motors . . . . .	3
1.3 Literature survey on modeling and control of coreless linear motors . . . . .	3
1.3.1 Modeling of CLMs . . . . .	4
1.3.2 Control of CLMs . . . . .	5
1.4 Research questions . . . . .	7
1.5 Thesis outline . . . . .	9
1.6 Publications . . . . .	11
<b>2 First-principle modeling of CLMs in multiple degrees of freedom</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Background knowledge . . . . .	15
2.2.1 Coreless linear motors topology and operating principle . . . . .	15
2.2.2 Mathematical operators . . . . .	16
2.2.3 Magnetostatic Maxwell equations . . . . .	17
2.3 Parasitic forces in linear motors . . . . .	17
2.3.1 Lorentz force . . . . .	18
2.3.2 Reluctance force . . . . .	18
2.3.3 Drag force . . . . .	18
2.4 Fourier modeling of Lorentz force in a CLM . . . . .	19
2.4.1 Division in regions . . . . .	19
2.4.2 Source term description . . . . .	19
2.4.3 Semianalytical solution . . . . .	20
2.4.4 Boundary conditions . . . . .	23
2.4.5 Force calculation . . . . .	24
2.5 Fourier modeling of reluctance force in a CLM . . . . .	25

2.5.1	Division in regions . . . . .	25
2.5.2	Source term description . . . . .	26
2.5.3	Semianalytical solution . . . . .	27
2.5.4	Boundary conditions . . . . .	28
2.5.5	Force calculation . . . . .	29
2.6	Model validation . . . . .	30
2.7	Model structure . . . . .	33
2.7.1	Linear dynamics . . . . .	33
2.7.2	Static nonlinearity . . . . .	34
2.8	Conclusions . . . . .	36
<b>3</b>	<b>Data-driven modeling of coreless linear motors</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Identification in the driving direction . . . . .	40
3.2.1	Problem formulation . . . . .	40
3.2.2	Instrumental variable method . . . . .	43
3.2.3	Predictor models . . . . .	45
3.2.4	Algorithm . . . . .	50
3.2.5	Simulation results . . . . .	51
3.3	Identification in the non-driving directions . . . . .	55
3.3.1	Problem formulation . . . . .	55
3.3.2	Instrumental variable method . . . . .	57
3.3.3	Predictor model . . . . .	58
3.3.4	Algorithm . . . . .	60
3.3.5	Simulation results . . . . .	61
3.4	Conclusions . . . . .	63
<b>4</b>	<b>Optimal commutation with nonlinear constraints</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Classical commutation . . . . .	67
4.2.1	Three-phase frame . . . . .	67
4.2.2	$dq0$ frame . . . . .	68
4.3	Optimal commutation . . . . .	70
4.3.1	Optimal commutation problem formulation . . . . .	71
4.3.2	Look-up table solution . . . . .	72
4.3.3	Analytical solutions . . . . .	73
4.3.4	Fast numerical optimization solution . . . . .	76
4.4	Simulation results . . . . .	79
4.4.1	Resulting forces and torque . . . . .	80
4.4.2	Dissipated power . . . . .	81
4.4.3	Computational time . . . . .	81
4.4.4	Summary . . . . .	82
4.5	Conclusions . . . . .	83



---

<b>5</b>	<b>Model predictive control of linear motors</b>	<b>85</b>
5.1	Introduction . . . . .	85
5.2	Linear model predictive control and optimal commutation . . . . .	87
5.2.1	Cascaded LMPC and optimal commutation . . . . .	87
5.2.2	Combined LMPC and optimal commutation . . . . .	90
5.3	Nonlinear model predictive control . . . . .	91
5.4	Simulation results . . . . .	93
5.4.1	Simulation settings . . . . .	93
5.4.2	Results . . . . .	95
5.4.3	Analysis . . . . .	97
5.5	Conclusions . . . . .	99
<b>6</b>	<b>An SQP method with arbitrary Hessian approximation</b>	<b>101</b>
6.1	Introduction . . . . .	101
6.1.1	Notation . . . . .	103
6.2	Nonlinear programming problem . . . . .	103
6.3	Basic sequential quadratic programming . . . . .	104
6.4	SQP with arbitrary Hessian approximation . . . . .	106
6.5	Local convergence . . . . .	108
6.6	Global convergence . . . . .	113
6.7	Treatment of inequality constraints . . . . .	117
6.8	Numerical examples . . . . .	118
6.8.1	A constrained nonlinear least squares problem . . . . .	119
6.8.2	Nonlinear model predictive control of linear motors . . . . .	121
6.9	Conclusions . . . . .	122
<b>7</b>	<b>Experimental validation</b>	<b>125</b>
7.1	Introduction . . . . .	125
7.2	Experimental setup . . . . .	125
7.2.1	System overview . . . . .	125
7.2.2	Mechanical design . . . . .	127
7.2.3	Hardware . . . . .	129
7.2.4	Software . . . . .	130
7.3	Data-driven modeling experiments . . . . .	131
7.3.1	Friction identification . . . . .	131
7.3.2	Identification in the driving direction . . . . .	131
7.3.3	Identification in the non-driving directions . . . . .	136
7.4	Optimal commutation experiments . . . . .	139
7.5	Conclusions . . . . .	142
<b>8</b>	<b>Conclusions and recommendations</b>	<b>145</b>
8.1	Conclusions . . . . .	145
8.2	Future work and recommendations . . . . .	148

<b>Bibliography</b>	<b>151</b>
<b>List of Abbreviations</b>	<b>161</b>
<b>Acknowledgments</b>	<b>163</b>
<b>Curriculum Vitae</b>	<b>165</b>

# Chapter 1

## Introduction

### 1.1 Coreless linear motors and applications

Permanent magnet linear motors are electrical machines that generate direct linear motion [45]. The history of linear motors can be traced back to the work of Charles Wheatstone in 1840s. After that, several patents for trains driven by linear motors have been granted. The first full-size working model was developed by Professor Eric Laithwaite of Imperial College in London in the late 1940s [42].

Linear motors are capable of producing direct thrust force and therefore have many advantages over classical rotary-to-linear devices which consist of rotary motors and mechanical transmission elements such as lead or ball screws, lead screws, timing belts, racks and pinions. Linear motors offer superior accuracy and faster response compared to their rotary-to-linear counterparts. Their thrust force capability is higher since it is not limited by the tensile strength of the belts like in the rotary-to-linear devices. Furthermore, as there is no mechanical wear, linear motors have excellent reliability and very long lifetime with almost no maintenance.

The two main types of permanent magnet linear motors are ironcore linear motors (ILMs) and coreless linear motors (CLMs). The two types have different performance characteristics and therefore are suitable for different types of applications. An ILM is composed of a magnet array and coil windings which are inserted in a slotted lamination stack made of iron. The presence of the iron core significantly increases the thrust force density as it focuses the magnetic field created by the coil windings. However, the iron core also results in undesirable effects such as cogging force, eddy current and a strong attractive force between the iron core and the magnets. Unlike ILMs, CLMs have lower thrust force density due to the absence of the iron core, but on the other hands do not suffer from cogging force or eddy currents. As a result, CLMs are more suitable for applications that require smooth and accurate movements. This thesis mainly focuses on CLMs, but the developed modeling and control methods in principle can also be applied to ILMs.

A picture and a cross-sectional view of a CLM are shown in Figure 1.1 and Fig-

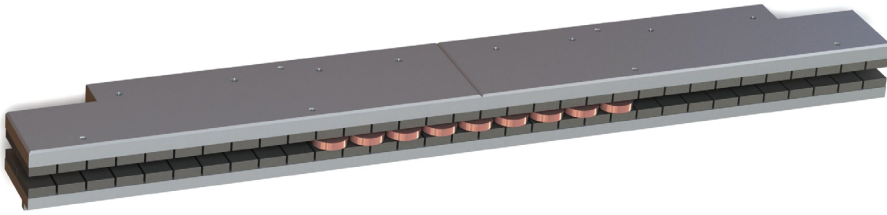


Figure 1.1: A coreless linear motor.

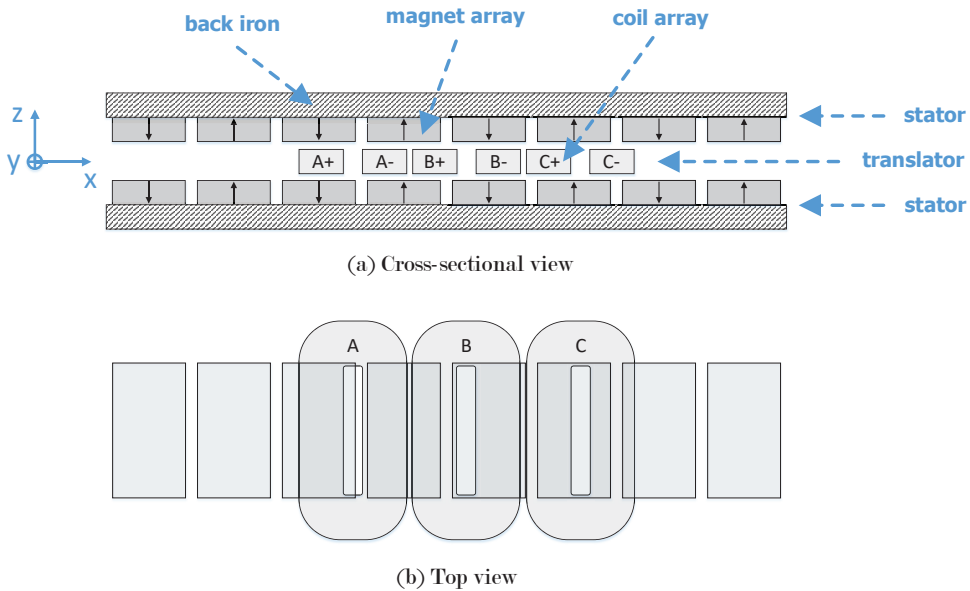


Figure 1.2: Cross-sectional view of a coreless linear motor.

ure 1.2, respectively. A CLM contains a stationary part called the stator and a moving part called the translator. The stator consists of two permanent magnet arrays mounted on two iron plates. The translator contains one or multiple sets of three-phase coils placed in the center of the air gap between the two magnet arrays. When there is electrical current flowing in the coils, the interaction between magnetic field and the current carrying coils produces an electromagnetic force that actuates the motor. The motor is actuated in the  $x$ -direction, which is called the driving direction, and in the ideal case produces no force in other directions.

CLMs are widely used in industrial high-precision positioning systems. They are used in machine tools such as feed axes in milling, turning or grinding machines. Another example is pick and place machines that require high accuracy and high speed. Linear motors are also used in scanners that need smooth and precise move-

ments. Lithography machines for semiconductor industry also make use of linear motors for fast and high precision positioning. In 2004, the North American market for linear motors reached the size of US\$ 95 million and the European linear motor market reached the size of US\$ 114 million [31]. The worldwide linear motor system market was estimated at about US\$ 400 million in 2007 [45].

## 1.2 Parasitic effects in coreless linear motors

Future generations of industrial positioning systems are facing increasing demands in precision and throughput. As a result, linear motors are required to operate at higher acceleration and higher accuracy. This is a challenging problem due to the presence of parasitic effects in linear motors, as high acceleration generally amplifies the parasitic effects and thus reduces the accuracy of the motors. Therefore, in order to operate at high acceleration while still maintaining high accuracy, it is necessary to identify and compensate for the parasitic effects in linear motors.

The main causes of parasitic effects in CLMs are manufacturing tolerances. Theoretically, CLMs are highly accurate by design. However, in practice it is infeasible to manufacture a CLM that matches the exact design parameters. There are usually many manufacturing tolerances in a real CLM which cause various parasitic forces and torques [46]. For instance, the variation of coil dimensions and the variation of magnet properties and dimensions result in parasitic Lorentz forces and torques in multiple directions. The misalignment of the coil array within the air gap causes parasitic Lorentz forces and torques in multiple directions, and also parasitic reluctance forces and torques in the non-driving directions.

These parasitic forces and torques are undesirable as they decrease the performance of a CLM. The parasitic forces in the driving direction reduce the position tracking performance of the motor. The parasitic forces and torques in non-driving directions generate disturbances to other components of the system, thus reducing the overall system performance. Therefore, these parasitic effects make it challenging to achieve fast and accurate control of CLMs. These parasitic effects are generally neglected in classical control of CLMs as will be discussed in Section 1.3. To address the future challenges in industrial positioning systems, a systematic approach to identify and compensate for parasitic effects in linear motors is required.

## 1.3 Literature survey on modeling and control of coreless linear motors

In this section we review the existing approaches for modeling and control of CLMs and discuss their advantages and limitations.

### 1.3.1 Modeling of CLMs

A linear motor can be modeled as two separate parts, namely the electromagnetic part and the motion dynamics part, as depicted in Figure 1.3. The electromagnetic part converts the input currents in the coils to electromagnetic forces. The resulting forces act on the motion dynamics and generate motion. The motion dynamics part is a linear dynamical system, while the electromagnetic part is a static nonlinearity.

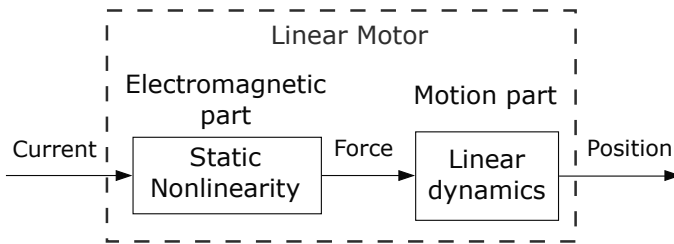


Figure 1.3: Block diagram of a linear motor model.

The motion dynamics part represents a mass moving in the  $x$ -direction, supported by a linear bearing which induces friction on the movement. It is therefore generally modeled as a mass-damper linear dynamical system [77, 98, 110]. The electromagnetic part represents the relation between the currents in the coils and the resulting electromagnetic forces, which can be considered to be static [30, 91, 92, 105]. Despite being static, the relation between the currents and the forces is complicated due to the presence of various parasitic effects caused by manufacturing tolerances such as misalignment of the coils in the air gap, variation of the coil dimensions, variation of the magnets properties and dimensions [46]. Conventionally, for control purpose, the electromagnetic part is modeled simply as a gain in the driving direction, which is known as the motor constant. The force in the driving direction is modeled as the product of the motor constant and the peak input current, while the forces and torques in the non-driving directions are generally neglected [56, 57, 77, 97, 98, 100]. This simple model does not take into account any parasitic effects in the motor. Therefore, it is not a good candidate for model-based control in high-precision applications, as the performance of model-based control is highly dependent on the quality of the model. In order to capture the parasitic effects, a more accurate model is required.

There are well-developed first-principle methods to model the electromagnetic behaviors with high accuracy such as the finite element method (FEM), the Fourier modeling method and the surface charge modeling method, which are usually used for motor design and analysis [4, 38, 48, 63, 104]. In [46], the surface charge model is used to investigate the effects of manufacturing tolerances in a CLM in three degrees of freedom (DOFs). However, these modeling methods are usually too complex for control purposes due to the large computational load which is difficult to implement in real-time systems. Furthermore, these models require knowledge of the physical parameters of the motor, including the manufacturing tolerances, which typically

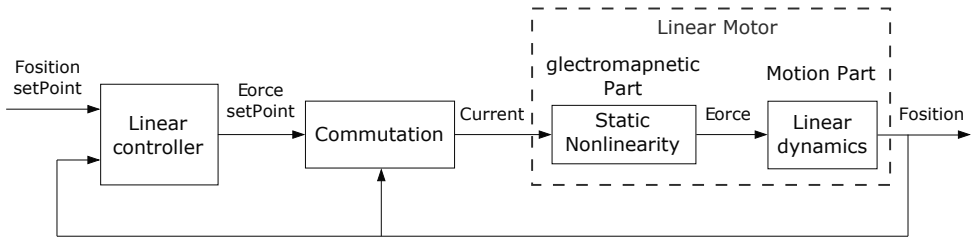


Figure 1.4: Classical control scheme for linear motors.

are not known exactly.

Another modeling approach is data-driven modeling, also known as system identification. In this approach, the model of a linear motor is obtained by fitting a model structure to measurement data. In the literature, the existing works on identification of linear motors have considered only the driving direction. To the best of the author's knowledge, there has been no research on identification of linear motors in the non-driving directions. In the driving direction, identification of linear motors is typically formulated as identification of the model of the force ripple as function of position, velocity and currents. The force ripple model is usually written as a sum of basis functions where the coefficients are to be estimated by fitting the model to measurement data. Some of the research works only consider position-dependent and velocity-dependent force ripple [62, 113]. Several methods to identify the current-dependent force ripple has been developed in [7, 99, 114]. However, the contribution of the current in each coil to the force ripple is not addressed therein. In [88, 89], a method to identify the force function of each coil is proposed, but this is limited to linear motors with only one set of three-phase coils. Furthermore, the above-mentioned works in general require measurements of the driving force, which are usually not available in real applications. In addition, the effect of the output measurement noise on the parameter estimation has not been addressed.

### 1.3.2 Control of CLMs

The standard position control scheme of a linear motor is shown in Figure 1.4. Due to the special structure of the linear motor model which consists of a linear dynamical system preceded by a static nonlinearity, the standard linear motor controller typically consists of two sub-controllers: a commutation algorithm and a linear position controller. The commutation algorithm aims to invert the static nonlinearity, thus removing it from the control problem. Linear control techniques can then be applied to the remaining linear dynamics.

The inversion of the static nonlinearity is known in the electrical machines literature as commutation [3, 89, 92, 106]. It is also known by different names in different fields. In systems and control literature, it can be considered as a special static case of feedback linearization, or exact linearization [1, 78, 93]. Authors in the aerospace

area name it control allocation [51,76,101]. These different names refer to the same problem, which is calculating the inversion of the nonlinearity, in order to remove the nonlinearity from the control problem. Classical commutation of linear motors makes use of simple three-phase sinusoidal current waveforms [45]. This approach does not take into account the parasitic effects in the electromagnetic nonlinearity. Therefore, the nonlinearity is not perfectly inverted, which results in force ripples and hence reduces the tracking performance of the position control loop.

A more accurate commutation method is to find the exact inverse of the electromagnetic nonlinearity model. This method is applicable when the number of independent current inputs is greater than or equal to the number of DOFs that need to be controlled. In a conventional CLM, generally a single three-phase amplifier is used to power one or multiple sets of three-phase coils. Consequently, the number of independent inputs is two, since the coils are connected in star-configuration. This is sufficient for classical control of CLMs because normally only one DOF is controlled, while the other DOFs are constrained by the guiding system. The force and torque in these DOFs are generally neglected. In order to control three DOFs of a CLM, it is necessary to have at least two sets of three-phase coils powered independently by two power amplifiers. In this case, the number of independent current inputs is greater than the number of DOFs that need to be controlled and the CLM is called over-actuated. The inversion problem consequently has an infinite number of solutions. The extra degrees of freedom can be used to minimize the power losses in the coils. As a result, the commutation problem becomes an optimization problem. This optimization problem can be solved using numerical optimization [2,3,61,64,65,87,92,94], which is in general computationally expensive. In the ideal case, the only force component in a CLM is the Lorentz force. Consequently, at one fixed position of the translator, the relation between the current vector and the force vector is linear. As a result, the commutation optimization problem can be solved analytically by eliminating equality constraints [24,85,86,88,89], by using Lagrange multipliers [8,9,111], or by using the minimum 2-norm generalized inverse [91,92,106].

However, the commutation problem is more complicated in nonideal CLMs where the coil array is not exactly in the center of the air gap. In addition to parasitic Lorentz forces and torques, there are also parasitic reluctance forces and torques in the non-driving directions [46]. Since the reluctance forces and torques are quadratic functions of the current vector, the commutation problem becomes a quadratic optimization problem with quadratic equality constraints. In general, it is difficult to find an analytical solution and hence numerical methods are necessary for solving the commutation problem. Newton's method has been used to solve commutation problems in different types of electrical machines in [64,81], without real-time experimental validation. Theoretically, Newton's method can also be applied to solve the commutation problem for parasitic forces compensation in CLMs. However, solving an optimization problem numerically using Newton's method requires a high computational load, which makes it challenging for real-time implementation in CLMs with high sampling rates. Therefore, more computationally efficient



methods are needed for real-time implementation of optimal commutation methods.

For linear position control of linear motors, simple proportional-integral-derivative (PID) feedback and feedforward controllers are used in practice. The tracking performance is mostly determined by the feedforward controller. The feedback controller is mainly responsible for disturbance rejection. This classical approach does not take into account the physical constraints of the system, such as current limit and thrust force limit. This will be a drawback for future high-precision positioning systems where linear motors have to operate near their physical limit to increase efficiency and throughput.

Furthermore, classical control and commutation of CLMs generally do not take into account the non-driving directions, resulting in parasitic forces in non-driving directions. These parasitic forces become higher when the motor moves at higher acceleration and hence can seriously impact system dynamics.

## 1.4 Research questions

To address the limitation of classical control approaches, in this thesis we aim to develop a systematic approach to identify and compensate for parasitic effects in CLMs in both driving and non-driving directions. The main research objective of the thesis is formulated as follows:

Research objective: To identify and compensate for main parasitic effects in coreless linear motors in multiple degrees of freedom.

We will address this main research objective step-by-step by investigating a series of research questions. Considering the control scheme of a linear motor, we see three main research problems as illustrated in Figure 1.5: modeling, commutation and control. In this thesis, we formulate six research questions, in which the first two consider the modeling problem, the third one deals with the commutation problem, the fourth one addresses the control problem, the fifth one considers the real-time implementation issue of commutation and control of linear motors, and the last one is on the experimental validation of the methods developed in the thesis.

First and foremost, it is necessary to understand the physical behavior of a non-ideal CLM, not only in the driving direction, but also the non-driving directions. The parasitic effects caused by different manufacturing tolerances in a nonideal CLM have been investigated in [46] by Monte-Carlo simulations using a three dimensional surface charge model. However, the surface charge model generally has to be solved numerically, which is not convenient for identification and control purposes. To address this problem, our first research objective is to find closed form expressions of the main parasitic effects in a CLM. The first research question is thus formulated as follows:

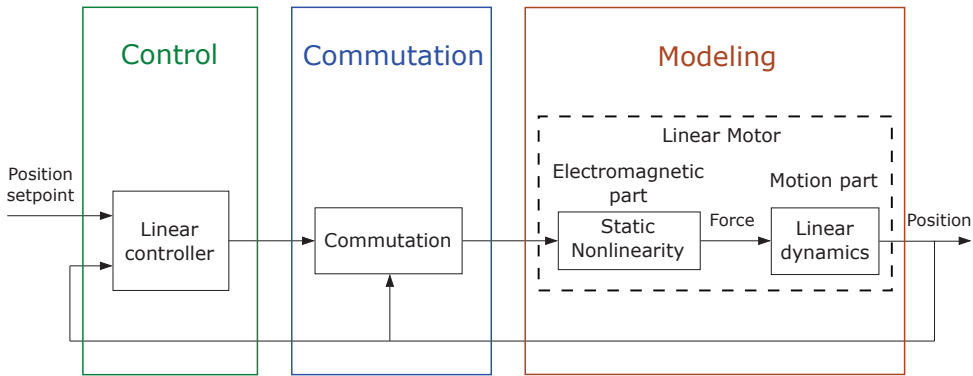


Figure 1.5: Research problems.

Q1: How to represent the main parasitic forces in a CLM mathematically in closed form expressions?

By solving research question Q1, we obtain a first-principle model structure of the physical behavior of a linear motor. However, to achieve high accuracy, first-principle models require knowledge of the physical parameters of the linear motor, including all the manufacturing tolerances, such as the size of each magnet and its remanent magnetization, the size of each coil, the misalignment of the coil within the air gap, etc. In practice, these informations are generally not available. An alternative modeling approach is to obtain the model of the motor from measurement data. This technique is known as data-driven modeling, or system identification. Our objective is to perform experiments on a linear motor, measure the input and output data and then fit the model structure obtained from first-principle modeling to the measurement data, thereby estimating the model's parameters. The second research question is formulated as follows:

Q2: How to identify the parameters of the physical model structure of a CLM from measurement data?

Having an accurate model of the linear motor, the next research objective is to utilize the model to develop a compensation method for the parasitic effects. Since the main source of parasitic effects in a linear motor is the electromagnetic part, an accurate commutation method can cancel out these effects. Classical commutation is based on a simplified model of the motor and thus cannot compensate for parasitic effects. Our goal is to use the data-driven model, which also captures the behavior of the parasitic effects, to derive an advanced commutation technique which can compensate for these effects. The third research question is formulated as follows:

Q3: How to derive an advanced commutation technique which can compensate for the main parasitic effects in a CLM based on its data-driven model?

The next research question considers the position controller of a CLM. As discussed in the previous section, the controller of a CLM is conventionally divided into two sub-controllers: a linear position controller and a commutation algorithm. A clear advantage of this approach is its simplicity. However, this is not the optimal setting, since the combination of the optimal solutions of the two sub-controllers may not yield the optimal solution for the whole controller. As an example, when there is a limit on the input currents for safety reasons, the linear position controller is not aware of this limit due to the separation. Consequently, it can give a force setpoint that is unachievable for the commutation problem due to the current limit, causing the commutation problem to be infeasible. To address this problem, our goal is to design a single nonlinear position controller which receives position measurements and directly calculates the necessary currents in the coils. The fourth research question is formulated as follows:

Q4: How to design a single nonlinear controller for a CLM instead of using a separate linear position controller and commutation?

The next research question addresses the real-time implementation issue of the developed control techniques. The developed optimal control and commutation algorithms require solving optimization problems numerically, which is in general computationally demanding. This makes it difficult for real-time implementation in CLMs with fast sampling rates where the time for computation is limited. To address this problem, our research objective is to develop a computationally efficient optimization algorithm which can be implemented in real-time. The fifth research question is formulated as follows:

Q5: How to design a fast optimization solver such that the developed MPC and commutation algorithms can be implemented in real-time?

Finally, it is of high importance to validate the developed identification, commutation and control methods in a real CLM setup. Our last research question is formulated as follows:

Q6: Can we validate the developed identification, commutation and control methods in experiments with a real CLM setup?

The formulated research questions will be investigated in the next chapters of this thesis.

## 1.5 Thesis outline

This thesis is divided into 8 chapters, including the introduction in Chapter 1 and conclusions in Chapter 8. The contents of the other chapters are as follows.

Chapter 2 addresses research questions Q1. First, the main types of parasitic forces resulting from manufacturing tolerances in a CLM are discussed. First-principle models of these types of forces are then derived using the Fourier modeling method. The resulting models are analytical and therefore are convenient for control purposes which require fast computation of the models. Based on the resulting models, a general model structure of a CLM is formulated, which will be used for identification purpose in Chapter 3.

Chapter 3 addresses research questions Q2. Based on the first-principle model structure derived in Chapter 2, an instrumental variable (IV) method for closed-loop identification of CLMs is developed. We analyze the bias of the estimate obtained by using the IV method together with the nonlinear autoregressive exogenous (NARX) predictor and show that it is negligible in many applications. Furthermore, when the probability distribution of the output measurement noise is known, a bias-corrected estimate is introduced and is proven to be consistent. A simulation example is shown for demonstration purposes. Some results presented in this chapter have been published in [72].

Chapter 4 addresses research question Q3 by analyzing an optimal commutation problem formulation which can compensate for parasitic effects in CLMs. The commutation problem is formulated as an optimization problem which minimizes the dissipated power in the coils, subject to the constraints that the desired forces and torques are obtained. As the constraints are quadratic, it is difficult to find an analytical solution of the optimization problem and hence numerical methods are necessary for solving the commutation problem. However, numerical optimization methods are computationally demanding and therefore are difficult to implement in systems with fast sampling rates like a CLM, which can be up to 20 kHz. To address this problem, we derive two sub-optimal analytical solutions for two special cases when the CLM has only one or two sets of three-phase coils. The solutions are analytical and thus are more computationally efficient, but still close to the optimal one. A look-up table solution which is very fast for online computation is also discussed. Furthermore, a computationally efficient optimization algorithm for commutation is developed, which will be discussed in more details in Chapter 6. Some of the results in this chapter have been published in [67, 68, 73].

Chapter 5 considers research question Q4. This chapter presents the design and implementation of a nonlinear model predictive control (MPC) algorithm for CLMs. The position controller and commutation are combined into one single nonlinear MPC controller which is capable of delivering excellent tracking performance while guaranteeing that physical constraints are satisfied. The advantages of the proposed nonlinear MPC control scheme compared to the cascaded control scheme which consists of linear MPC and commutation is analyzed. Some results presented in this chapter have been published in [71].

Chapter 6 addresses research question Q5 by developing a sequential quad-ratic programming (SQP) algorithm with arbitrary positive (semi-) definite Hessian approximations. The proposed algorithm is proven to guarantee local convergence for SQP even with poor Hessian approximation. As a result, any simple positive (semi-)

definite Hessian approximations, even the identity matrix, can be used as the Hessian approximation. Consequently, the computational load is reduced, especially for large-scale problems like MPC. The effectiveness of the developed algorithm is demonstrated in numerical examples. Parts of the results presented in this chapter have been published in [69, 70].

Chapter 7 considers research question Q6. To answer this question, an experimental CLM setup is designed and constructed in collaboration with ASML. The setup consists of a real industrial linear motor, power amplifiers and an encoder provided by ASML. Additional force sensors are added in order to measure the forces and torques produced by the motor in driving and non-driving directions. A dSPACE MicroLabBox development system is employed to control the system. The developed modeling, commutation and control methods are validated on the setup. The experimental results are presented and analyzed.

## 1.6 Publications

Preliminary versions of the results presented in this thesis have been published in the articles listed below.

- Chapter 3 contains results that have been presented in:
  - T. T. Nguyen, M. Lazar, H. Butler and P. M. J. Van den Hof, “An instrumental variable method for closed-loop identification of coreless linear motors,” in IEEE American Control Conference (ACC), Milwaukee, WI, 2018.
- Chapter 4 contains results that have been presented in:
  - T. T. Nguyen, M. Lazar and H. Butler, “Cancellation of normal parasitic forces in coreless linear motors,” in IEEE 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, 2015.
  - T. T. Nguyen, H. Butler and M. Lazar, “An analytical commutation law for parasitic forces and torques compensation in coreless linear motors,” in IEEE European Control Conference (ECC), Aalborg, 2016.
  - T. T. Nguyen, M. Lazar, H. Butler, “A computationally efficient commutation algorithm for parasitic forces and torques compensation in ironless linear motors,” in P. Hubbard (Ed.), 7th IFAC Symposium on Mechatronic Systems, 2016.
- Chapter 5 contains results that have been presented in:
  - T. T. Nguyen, M. Lazar and H. Butler, “Nonlinear model predictive control of ironless linear motors,” in IEEE 2nd Conference on Control Technology and Applications (CCTA), Copenhagen, 2018.
- Chapter 6 contains results that have been presented in:

T. T. Nguyen, M. Lazar and H. Butler, “A Hessian-free algorithm for solving quadratic optimization problems with nonlinear equality constraints,” in IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, 2016.

T. T. Nguyen, M. Lazar and H. Butler, “A method to guarantee local convergence for sequential quadratic programming with poor Hessian approximation,” in IEEE 56th Conference on Decision and Control (CDC), Melbourne, 2017.

T. T. Nguyen, M. Lazar and H. Butler, “A sequential quadratic programming method with arbitrary (semi-) positive definite Hessian approximation,” in preparation for journal submission, 2018.

# Chapter 2

## First-principle modeling of coreless linear motors in multiple degrees of freedom

### 2.1 Introduction

CLMs are known for their extremely high precision compared to ironcore linear motors or rotary-to-linear devices. An ideal CLM produces only driving force. There is no force in other directions due to its symmetrical stator design. The ripple in the driving force is very low since there is no cogging force. Thus, CLMs are widely used in precision positioning systems which require smooth motion and extremely high accuracy.

However, in a real CLM, there are deviations from the design parameters such as misalignment of the translator within the air gap, variation of coil dimensions and variation of magnet properties and dimensions. These manufacturing tolerances result in parasitic forces and torques in a CLM. The parasitic Lorentz forces and torques caused by various manufacturing tolerances have been analyzed in [46] by Monte-Carlo simulation using a three dimensional surface charge model. However, the parasitic reluctance forces and torques are neglected therein.

In this chapter, we discuss all the main parasitic force components in a CLM. In spite of various manufacturing tolerances, there are mainly three force components in a CLM, which are Lorentz force, reluctance force and drag force. Furthermore, we aim to derive the mathematical models of the main force components using first-principle modeling methods. The resulting first-principle model structures will be used for identification purpose in Chapter 3. Therefore, it is of interest to have first-principle models which are analytical and have simple model structures. The two most popular analytical methods for high-accuracy modeling of linear and planar actuators are the surface charge method and the Fourier method [19, 48].

The surface charge modeling method provides an analytical solution of the magnetic field in free space [19]. In this method, a cuboidal magnet is modeled as two surface charges. The magnetic field of an array of permanent magnets is then calculated as the sum of the contributions of the individual magnets [48]. Although the solution of the magnetic field is analytical, the solutions of the Lorentz force and torque generally require numerical integration. Closed form expressions of the force and torque can be obtained when a straight current carrying volume is parallel to the sides of a magnet [90]. However, these expressions are complicated and therefore not favorable to identification and control purposes.

The Fourier modeling method, also called harmonic modeling method, is a semi-analytical modeling technique based on Fourier analysis [38,41,115]. In this method the source term distribution is written in terms of Fourier series. The geometry of the actuator is divided into separate regions. The resulting magnetic field solutions in each region are obtained by solving Maxwell's equations. Then the force and torque can also be obtained analytically. As a result, the Fourier modeling method is attractive for identification and control purposes and therefore will be employed in this chapter. The main disadvantage of the Fourier model is that it assumes periodical structure of the motor, which is not the case for a non-ideal CLM. However, this drawback can be fixed simply by taking the whole motor length as the base Fourier period.

To validate the resulting Fourier model, we will compare its output to the output of a finite element method (FEM) model. The FEM is a numerical technique for obtaining approximate solutions of boundary value problems [50]. In this method, the domain is discretized into a finite number of small elements in which the fields can be approximated by simple equations. All the elemental equations are then assembled into a global system of equations. Finally, the solution is obtained by solving the system of equations. The FEM is highly accurate and is applicable to complex geometric shapes. In this chapter, we consider the FEM model as the "true motor", due to its high accuracy. The Fourier model will be validated by comparing its output to the output of the FEM model.

Furthermore, based on the analysis of the force components and the insights provided by the Fourier model, a general model structure of CLMs is formulated. This general model structure will be useful for identification and model-based controller design in the later chapters.

This chapter is organized as follows. Section 2.2 provides background knowledge on coreless linear motor and Maxwell's equation. Section 2.3 discusses the main force components present in a CLM. The Fourier model of the Lorentz force component is derived in Section 2.4. The Fourier model of the reluctance force component is derived in Section 2.5. In Section 2.6, the Fourier model is validated by comparing its outputs to the outputs of a FEM model. In Section 2.7, a general model structure of CLMs is formulated. The conclusions are summarized in Section 2.8.



## 2.2 Background knowledge

### 2.2.1 Coreless linear motors topology and operating principle

A cross-sectional view with dimensions of a basic coreless linear motor is shown in Figure 2.1. The stator consists of two permanent magnet arrays mounted on two iron plates. The translator contains one or multiple sets of three-phase coils placed in between the two magnet arrays. The notation for the dimensions are summarized in Table 2.1.

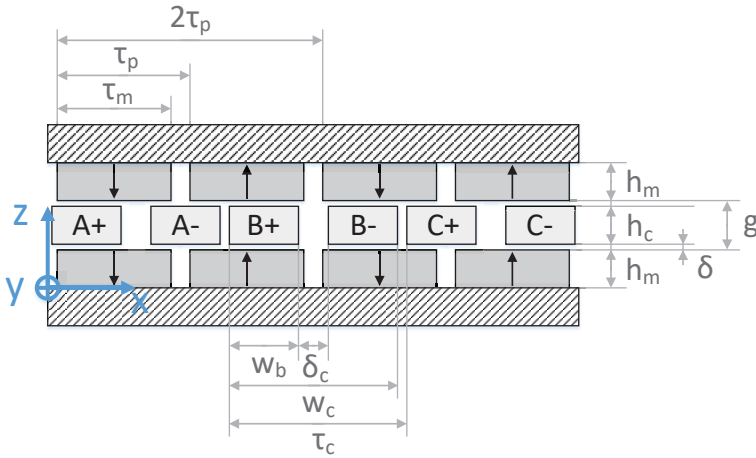


Figure 2.1: Cross-sectional view of a coreless linear motor.

Table 2.1: CLM parameters

<i>Symbol</i>	<i>Parameter</i>	<i>Unit</i>
$\tau_p$	Magnet pole pitch	mm
$\tau_c$	Coil pitch	mm
$\delta$	Air gap	mm
$w_m$	Magnet width	mm
$h_m$	Magnet height	mm
$w_c$	Coil width	mm
$\delta_c$	Coil eye width	mm
$h_c$	Coil height	mm
$w_b$	Coil bundle width	mm
$D$	Motor depth	mm

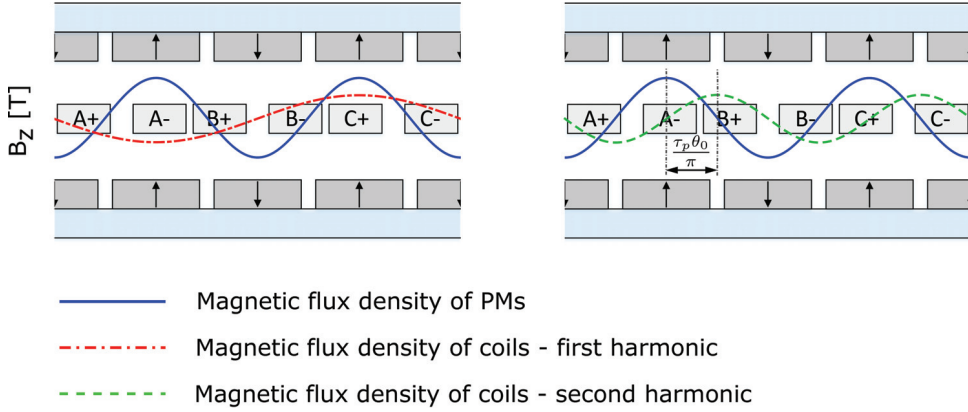


Figure 2.2: Main magnetic field components in a CLM.

CLMs are synchronous electrical machines. The operating principle of CLMs is similar to that of rotating synchronous motors. The force in a CLM with moving coils is produced due to the interaction of magnetic fields produced by the permanent magnets (PMs) and currents in the coils. The PMs generate a magnetic field which is fixed in space. The currents in the coils produce a magnetic field with double frequency in which the first harmonic component is moving and the second harmonic component is fixed. The first harmonic component causes eddy current losses and does not produce any force. The second harmonic component produces force as it attempts to align with the magnetic field of the PMs. The maximum force is produced when the phase shift  $\theta_0$  between the magnetic field of the PMs and the second harmonic component of the magnetic field of the coils is 90 degrees [60]. An illustration of the main magnetic field components in a CLM is shown in Figure 2.2.

### 2.2.2 Mathematical operators

Let us consider a three dimensional Cartesian coordinate system and let  $\vec{e}_x$ ,  $\vec{e}_y$ , and  $\vec{e}_z$  be the unit vectors for the  $x$ -,  $y$ -, and  $z$ -axes, respectively. The gradient operator is defined as follows

$$\nabla \vec{f} = \frac{\partial f}{\partial x} \vec{e}_x + \frac{\partial f}{\partial y} \vec{e}_y + \frac{\partial f}{\partial z} \vec{e}_z. \quad (2.1)$$

Let  $\vec{f} = f_x \vec{e}_x + f_y \vec{e}_y + f_z \vec{e}_z$ . The divergence operator is defined as follows

$$\nabla \cdot \vec{f} = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}. \quad (2.2)$$

The curl operator, or rotation operator, is defined as follows

$$\nabla \times \vec{f} = \left( \frac{\partial f_z}{\partial y} - \frac{\partial f_y}{\partial z} \right) \vec{e}_x + \left( \frac{\partial f_x}{\partial z} - \frac{\partial f_z}{\partial x} \right) \vec{e}_y + \left( \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y} \right) \vec{e}_z. \quad (2.3)$$

### 2.2.3 Magnetostatic Maxwell equations

The magnetostatic Maxwell equations are ([29]):

$$\nabla \times \vec{H} = \vec{J}, \quad (2.4a)$$

$$\nabla \cdot \vec{B} = 0, \quad (2.4b)$$

where  $H$  (A/m) is the magnetic field strength,  $B$  (T) is the magnetic flux density and  $J$  (A/m<sup>2</sup>) is the current density.

The magnetostatic Maxwell equations can be solved directly for the fields. However, it is often more convenient to obtain the fields using potential functions [29]. Since the divergence of a curl is zero, equation (2.4b) implies

$$\vec{B} = \nabla \times \vec{A}. \quad (2.5)$$

The vector  $\vec{A}$  is called the magnetic vector potential. By introducing  $\vec{A}$ , we solve one second order partial differential equation instead of two first order ones.

The magnetic flux density can be written as [38]

$$\vec{B} = \mu_0 \mu_r \vec{H} + \mu_0 \vec{M}_0, \quad (2.6)$$

where  $\mu_0$  is the permeability of free space,  $\mu_r$  is the relative permeability of the magnetic material and  $\vec{M}_0$  (A/m) is the magnetization vector which is given by

$$\vec{M}_0 = \frac{\vec{B}_{\text{rem}}}{\mu_0}, \quad (2.7)$$

where  $\vec{B}_{\text{rem}}$  is the remanent magnetization of the magnets. Due to (2.4a), (2.5) and (2.6), the magnetostatic Maxwell equations are reduced into a Poisson equation

$$\nabla^2 \vec{A} = -\mu_0 (\nabla \times \vec{M}_0) - \mu \vec{J}, \quad (2.8)$$

where  $\mu = \mu_0 \mu_r$ . In regions with no magnetic source, the Poisson equation is reduced to a Laplace equation

$$\nabla^2 \vec{A} = 0. \quad (2.9)$$

The Poisson and Laplace equations will be used to calculate the magnetic field solution in Sections 2.4 and 2.5.

## 2.3 Parasitic forces in linear motors

In spite of various manufacturing tolerances, there are only three main types of forces in a CLM, which are the Lorentz force, the reluctance force and the drag force. These three types of forces are discussed in this section. We only consider the force in the  $x$ - and  $z$ -directions and torque about the  $y$ -axis, as the forces and torques in other directions are negligible [46].

### 2.3.1 Lorentz force

The Lorentz force is the force exerted on a current-carrying wire in a magnetic field [29]. Linear motors are a type of Lorentz actuators. The Lorentz force is the dominant type of force in linear motors and is used to actuate the motor. The Lorentz force is linear with the input currents and can be written as follows

$$w_{\text{Lor}} = K(x)u, \quad (2.10)$$

where  $u \in \mathbb{R}^{n_u}$  is the input current vector,  $K(x) \in \mathbb{R}^{3 \times n_u}$  is the force constant vector which is position dependent, and  $w_{\text{Lor}}$  is the Lorentz wrench vector defined as follows

$$w_{\text{Lor}} = [F_{x\text{Lor}} \quad F_{z\text{Lor}} \quad T_{y\text{Lor}}]^\top. \quad (2.11)$$

If the motor is ideal, the total Lorentz force is equal to the nominal force and there will be no parasitic force. However, in practice, there are manufacturing tolerances in CLMs which cause parasitic Lorentz forces in multiple directions. For instance, the coils may not be placed exactly in the center of the air gap, some magnets may have different remanent magnetization than the designed value, the magnets and coils may vary in shape and size. The resulting total Lorentz force is therefore different from the designed nominal force. The difference is defined as the parasitic Lorentz force.

### 2.3.2 Reluctance force

The attractive forces between the coil array and the two back iron plates are reluctance forces [77]. If the coil is placed exactly in the center of the air gap then the two reluctance forces cancel each other. However, if the coil array is misaligned, the reluctance force between the coil array and the closer iron plate becomes larger, resulting in nonzero total reluctance force. It is obvious that the reluctance force only act in the  $z$ -direction.

The reluctance force is quadratic in the input currents and can be written as follows

$$F_{z\text{rel}} = u^\top G u, \quad (2.12)$$

where  $G \in \mathbb{R}^{n_u \times n_u}$  is a constant matrix. It will be shown in Section 2.5 that the reluctance force is independent of the  $x$ -position of the translator.

### 2.3.3 Drag force

Drag force is the force acting in the opposite direction of the movement of the translator. In CLMs, drag force is mainly caused by friction. The friction force consists of Coulomb friction, viscous friction and Stribeck friction [110]. The friction force can be modeled using the LuGre model as follows

$$F_{\text{fric}} = \left( F_c + (F_s - F_c)e^{-(v/v_s)^2} + dv \right) \text{sign}(v), \quad (2.13)$$

where  $F_c$  is the minimum level of Coulomb friction,  $F_s$  is the level of static friction,  $v$  is the speed,  $v_s$  is the lubricant parameter determined by empirical experiment, and  $d$  is the viscous frictional coefficient [6, 98, 112].

## 2.4 Fourier modeling of Lorentz force in a CLM

Fourier modeling is a semi-analytical modeling technique based on Fourier analysis [38, 41, 115]. In this method, the source term distribution is written in terms of Fourier series. The geometry of the actuator is divided into separate regions. The resulting magnetic field solutions in each region are obtained by solving Laplace and Poisson equations. The unknown coefficients are determined from a set of boundary conditions. The Fourier modeling method is popular for design and analysis of linear and planar actuators [4, 20, 39, 47, 80].

The resulting Fourier model is semi-analytical because it contains an infinite number of harmonics. In practice, only a finite number of harmonics can be taken into account. The number of harmonics is chosen dependent on the required accuracy of the model. As a result, the Fourier model is analytical and therefore can be calculated faster than a FEM model and a surface charge model. Fourier modeling is thus attractive for control and fast simulation. Fourier modeling is usually used for periodical structures. However, it can still be used to model a non-ideal CLM, which is non-periodical, simply by taking the whole motor length as one Fourier period, at the price of increasing the number of harmonics, which increases the computational cost.

In this section, we employ the Fourier modeling technique to calculate the Lorentz force acting on the coil array of a CLM. We follow the modeling procedure described in [38].

### 2.4.1 Division in regions

Firstly, the CLM is divided into three regions as shown in Figure 2.3. Regions I and III are the magnetized regions. Region II is the source-free region.

The fundamental frequency of a region is defined as:

$$\omega_n = \frac{n\pi}{\tau}, \quad (2.14)$$

where  $n$  is the harmonic number. The width of the region is  $2\tau$ . For ease of presentation,  $\tau$  is chosen equal to  $\tau_p$  in this section. However,  $\tau$  can also be chosen as the whole motor length in order to capture the manufacturing tolerances.

### 2.4.2 Source term description

In the CLM topology, permanent magnets are used to generate a magnetic field. The permanent magnets are magnetized in the normal direction as shown in Figure 2.4.

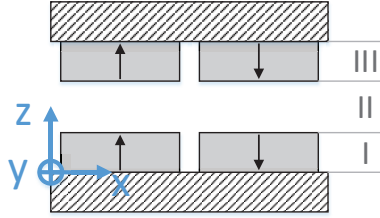


Figure 2.3: Magnetic source regions.

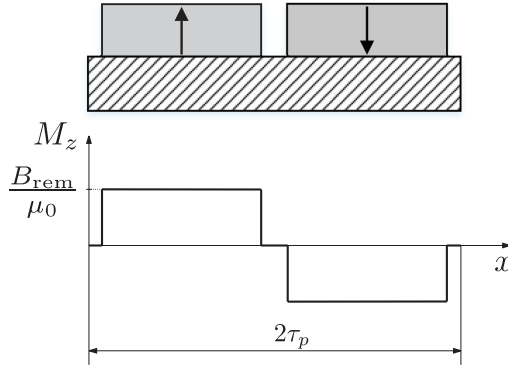


Figure 2.4: Magnetization of permanent magnets.

As a result, only the  $z$ -component of the magnetization vector  $\vec{M}_0$  is non-zero. In region I, the magnetization of the magnets in the  $z$ -direction can be described by a Fourier series

$$M_z^I(x) = \sum_{n=1}^{\infty} (M_{zcn}^I \cos(\omega_n x) + M_{zsn}^I \sin(\omega_n x)), \quad (2.15)$$

where

$$M_{zsn}^I = \frac{1}{\tau_p} \int_0^{2\tau_p} M_z^I(x) \sin(\omega_n x) dx = \frac{4B_{rem}}{\mu_0 n \pi} \sin\left(\frac{n\pi}{2}\right) \sin\left(\frac{n\pi\tau_m}{2\tau_p}\right), \quad (2.16)$$

$$M_{zcn}^I = \frac{1}{\tau_p} \int_0^{2\tau_p} M_z^I(x) \cos(\omega_n x) dx = 0. \quad (2.17)$$

A similar source term description applies to region III. There is no magnetic source in region II.

### 2.4.3 Semianalytical solution

Since the source terms are described by Fourier series, the solution of the magnetic vector potential is also written in terms of Fourier components. The field equations

are given by the Laplace equation in region II and Poisson equation in region I and III. These equations are solved for each region by separation of variables. Since the magnetization vector  $\vec{M}_0$  has only a non-zero component in the  $z$ -direction, the magnetic vector potential  $\vec{A}$  only has a non-zero component in the  $y$ -direction which is denoted by  $A_y$ .

### Region II (air)

As there is no magnetic source in region II, the field equation is given by the Laplace equation

$$\nabla^2 \vec{A}^{\text{II}} = 0. \quad (2.18)$$

The  $y$ -component  $A_y^{\text{II}}$  of the magnetic vector potential can be written as the sum of harmonic components as follows

$$A_y^{\text{II}} = \sum_{n=1}^{\infty} A_{yn}^{\text{II}}, \quad (2.19)$$

where  $A_{yn}^{\text{II}}$  denotes the  $n^{\text{th}}$  harmonic components of  $A_y^{\text{II}}$ . The Laplace equation is solved by separation of variables [17]. Assume that the solution has the following form

$$A_{yn}^{\text{II}}(x, z) = X_n^{\text{II}}(x) Z_n^{\text{II}}(z). \quad (2.20)$$

Since the magnetization is described using a Fourier series as function of  $x$ , the function for the  $x$ -direction  $X_n^{\text{II}}(x)$  also consists of sine and cosine functions as follows

$$X_n^{\text{II}}(x) = a_{xn}^{\text{II}} \cos(\omega_n x) + b_{xn}^{\text{II}} \sin(\omega_n x), \quad (2.21)$$

For the Laplace equation (2.18) to be satisfied, the function for the  $z$ -direction  $Z_n^{\text{II}}(z)$  is given by [40]

$$Z_n^{\text{II}}(z) = a_{zn}^{\text{II}} e^{\omega_n z} + b_{zn}^{\text{II}} e^{-\omega_n z}. \quad (2.22)$$

Here, due to the choice of the coordinate system, the magnetization has only sine terms. Consequently, the magnetic vector potential only has cosine terms. Therefore, the solution has the form

$$A_{yn}^{\text{II}} = \frac{1}{\omega_n} (a_{xn}^{\text{II}} a_{zn}^{\text{II}} e^{\omega_n z} + a_{xn}^{\text{II}} b_{zn}^{\text{II}} e^{-\omega_n z}) \cos(\omega_n x), \quad (2.23)$$

which is then simplified to

$$A_{yn}^{\text{II}} = \frac{1}{\omega_n} (a_n^{\text{II}} e^{\omega_n z} + b_n^{\text{II}} e^{-\omega_n z}) \cos(\omega_n x). \quad (2.24)$$

From equation (2.5), the  $x$ -component and the  $z$ -component of the magnetic flux density are given by

$$B_x = \frac{\partial A_y}{\partial z}, \quad (2.25)$$

$$B_z = -\frac{\partial A_y}{\partial x}. \quad (2.26)$$

As a result, it follows from (2.24) that the  $n^{\text{th}}$  harmonic components of  $B_x^{\text{II}}$  and  $B_z^{\text{II}}$  are given by

$$B_{xn}^{\text{II}} = (a_n^{\text{II}} e^{\omega_n z} - b_n^{\text{II}} e^{-\omega_n z}) \cos(\omega_n x), \quad (2.27)$$

$$B_{zn}^{\text{II}} = (a_n^{\text{II}} e^{\omega_n z} + b_n^{\text{II}} e^{-\omega_n z}) \sin(\omega_n x), \quad (2.28)$$

Here, the coefficients  $a_n^{\text{II}}$  and  $b_n^{\text{II}}$  are unknown and to be determined by boundary conditions.

### Region I and III (Magnet array)

Regions I and III contain permanent magnet arrays. Consequently, the field equations in these regions are given by the Poisson equation. Let us first consider region I. We have the Poisson equation

$$\nabla^2 \vec{A}^{\text{I}} = -\mu_0 (\nabla \times \vec{M}_0). \quad (2.29)$$

The Poisson equation is solved using separation of variables. The solution of each harmonic component of  $A_y^{\text{I}}$  has the form

$$A_{yn}^{\text{I}}(x, z) = X_n^{\text{I}}(x) Z_n^{\text{I}}(z) + P_n^{\text{I}}(x, z) \quad (2.30)$$

where  $X_n^{\text{I}}(x) Z_n^{\text{I}}(z)$  is the homogeneous solution and  $P_n^{\text{I}}(x, z)$  is the particular solution. The homogeneous solution has the same form as (2.24). Since the righthand side of (2.29) is independent of  $z$ , we choose a particular solution depending only on  $x$ , i.e.  $P_n^{\text{I}}(x, z) = P_n^{\text{I}}(x)$ . Therefore, it follows from the Poisson equation (2.29) that

$$\frac{d^2 P_n^{\text{I}}(x)}{dx^2} = -\mu_0 \frac{dM_{zn}^{\text{I}}(x)}{dx} = -\mu_0 \omega_n M_{zsn}^{\text{I}} \cos(\omega_n x). \quad (2.31)$$

By twice integrating the above equation we obtain

$$P_n^{\text{I}}(x) = \mu_0 \frac{1}{\omega_n} M_{zsn}^{\text{I}} \cos(\omega_n x). \quad (2.32)$$

Combining the homogeneous solution and the particular solution we have

$$A_{yn}^{\text{I}} = \frac{1}{\omega_n} (a_n^{\text{I}} e^{\omega_n z} + b_n^{\text{I}} e^{-\omega_n z} + \mu_0 M_{zsn}^{\text{I}}) \cos(\omega_n x). \quad (2.33)$$



The magnetic flux density can then be calculated using formulas (2.25) and (2.26). The results are as follows

$$B_{xn}^I = (a_n^I e^{\omega_n z} - b_n^I e^{-\omega_n z}) \cos(\omega_n x), \quad (2.34)$$

$$B_{zn}^I = (a_n^I e^{\omega_n z} + b_n^I e^{-\omega_n z} + \mu_0 M_{zsn}^I) \sin(\omega_n x). \quad (2.35)$$

Here, the coefficients  $a_n^I$  and  $b_n^I$  are unknown and to be determined by boundary conditions. A similar description of the magnetic flux density applies to region III.

### Summary

In summary, the magnetic flux densities in all regions are given by

$$B_x^I = \sum_{n=1}^{\infty} (a_n^I e^{\omega_n z} - b_n^I e^{-\omega_n z}) \cos(\omega_n x) \quad (2.36a)$$

$$B_z^I = \sum_{n=1}^{\infty} (a_n^I e^{\omega_n z} + b_n^I e^{-\omega_n z} + \mu_0 M_{zsn}^I) \sin(\omega_n x) \quad (2.36b)$$

$$B_x^{II} = \sum_{n=1}^{\infty} (a_n^{II} e^{\omega_n z} - b_n^{II} e^{-\omega_n z}) \cos(\omega_n x) \quad (2.36c)$$

$$B_z^{II} = \sum_{n=1}^{\infty} (a_n^{II} e^{\omega_n z} + b_n^{II} e^{-\omega_n z}) \sin(\omega_n x) \quad (2.36d)$$

$$B_x^{III} = \sum_{n=1}^{\infty} (a_n^{III} e^{\omega_n z} - b_n^{III} e^{-\omega_n z}) \cos(\omega_n x) \quad (2.36e)$$

$$B_z^{III} = \sum_{n=1}^{\infty} (a_n^{III} e^{\omega_n z} + b_n^{III} e^{-\omega_n z} + \mu_0 M_{zsn}^{III}) \sin(\omega_n x) \quad (2.36f)$$

### 2.4.4 Boundary conditions

In general, the boundary conditions are [38]:

- Between two regions:
  - $H$  tangential is continuous at an interface if the current density at the surface is zero.
  - $B$  normal is continuous across an interface.
- At interface of infinitely permeable iron:
  - $H$  tangential is zero.
- At infinity:
  - $A$ ,  $B$ ,  $H$  are all zero.

For the considered CLM topology, the boundary conditions (BCs) are:

- BC1:  $H_x^I \Big|_{z=0} = 0$ .
- BC2:  $B_z^I \Big|_{z=h_m} = B_z^{II} \Big|_{z=h_m}$ .
- BC3:  $H_x^I \Big|_{z=h_m} = H_x^{II} \Big|_{z=h_m}$ .
- BC4:  $B_z^{II} \Big|_{z=g+h_m} = B_z^{III} \Big|_{z=g+h_m}$ .
- BC5:  $H_x^{II} \Big|_{z=g+h_m} = H_x^{III} \Big|_{z=g+h_m}$ .
- BC6:  $H_x^{III} \Big|_{z=g+2h_m} = 0$ .

These boundary conditions lead to a set of linear equations:

$$\text{BC1: } a_n^I - b_n^I = 0 \quad (2.37a)$$

$$\text{BC2: } a_n^I e^{\omega_n h_m} + b_n^I e^{-\omega_n h_m} - a_n^{II} e^{\omega_n h_m} - b_n^{II} e^{-\omega_n h_m} = -\mu_0 M_{zsn}^I \quad (2.37b)$$

$$\text{BC3: } a_n^I e^{\omega_n h_m} - b_n^I e^{-\omega_n h_m} - \mu_r a_n^{II} e^{\omega_n h_m} + \mu_r b_n^{II} e^{-\omega_n h_m} = 0 \quad (2.37c)$$

$$\text{BC4: } a_n^{II} e^{\omega_n(g+h_m)} + b_n^{II} e^{-\omega_n(g+h_m)} - a_n^{III} e^{\omega_n(g+h_m)} - b_n^{III} e^{-\omega_n(g+h_m)} = \mu_0 M_{zsn}^{III} \quad (2.37d)$$

$$\text{BC5: } \mu_r a_n^{II} e^{\omega_n(g+h_m)} - \mu_r b_n^{II} e^{-\omega_n(g+h_m)} - a_n^{III} e^{\omega_n(g+h_m)} + b_n^{III} e^{-\omega_n(g+h_m)} = 0 \quad (2.37e)$$

$$\text{BC6: } a_n^{III} e^{\omega_n(g+2h_m)} - b_n^{III} e^{-\omega_n(g+2h_m)} = 0 \quad (2.37f)$$

The unknown coefficients  $a_n^k$  and  $b_n^k$ , with  $k = I, II, III$ , are then obtained by solving the set of equations (2.37).

## 2.4.5 Force calculation

Let us consider a single coil with a current density  $\vec{j}$  in the magnetic field in the air gap, which is region II. The only non-zero component of the current density is the component in the  $y$ -direction  $J_y$ . The electromagnetic force acting on the coil is given by Lorentz's formula [29]

$$\vec{F}_{\text{Lor}} = \int_S \vec{j} \times \vec{B}^{II} dS = \int_S J_y B_z^{II} dS \vec{e}_x - \int_S J_y B_x^{II} dS \vec{e}_z, \quad (2.38)$$

where  $S$  is the cross-sectional area of the coil. It follows that

$$\begin{aligned} F_{x\text{Lor}} &= \int_S J_y B_z^{II} dS, \\ F_{z\text{Lor}} &= - \int_S J_y B_x^{II} dS. \end{aligned} \quad (2.39)$$

Given the description of the magnetic flux density in (2.36), the force  $F_x$  and  $F_z$  are calculated as follows

$$\begin{aligned} F_{x\text{Lor}} &= \int_S J_y B_z^{\text{II}} dS \\ &= \frac{J_y D \tau_p^2}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} (a_n^{\text{II}} e^{\omega_n z_2} - b_n^{\text{II}} e^{-\omega_n z_2} - a_n^{\text{II}} e^{\omega_n z_1} + b_n^{\text{II}} e^{-\omega_n z_1}) \\ &\quad (\cos(\omega_n x_1) - \cos(\omega_n x_2) - \cos(\omega_n x_3) + \cos(\omega_n x_4)), \end{aligned} \quad (2.40)$$

$$\begin{aligned} F_{z\text{Lor}} &= - \int_S J_y B_x^{\text{II}} dS \\ &= \frac{J_y D \tau_p^2}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} (a_n^{\text{II}} e^{\omega_n z_2} + b_n^{\text{II}} e^{-\omega_n z_2} - a_n^{\text{II}} e^{\omega_n z_1} - b_n^{\text{II}} e^{-\omega_n z_1}) \\ &\quad (\sin(\omega_n x_1) - \sin(\omega_n x_2) - \sin(\omega_n x_3) + \sin(\omega_n x_4)), \end{aligned} \quad (2.41)$$

where the limits of integration are:

$$x_1 = x, \quad (2.42)$$

$$x_2 = x + w_c, \quad (2.43)$$

$$x_3 = x + w_c + \delta_c, \quad (2.44)$$

$$x_4 = x + w_c + \delta_c + w_c, \quad (2.45)$$

and

$$z_1 = h_m + \delta, \quad (2.46)$$

$$z_2 = h_m + \delta + h_c. \quad (2.47)$$

The total Lorentz force acting on the translator is then calculated as the sum of the Lorentz forces acting on each coil. It can be seen that the Lorentz force is dependent on the current density and the position of the translator. At a fixed position of the translator, the Lorentz force is linear with the current density.

## 2.5 Fourier modeling of reluctance force in a CLM

In this section we apply the Fourier modeling method to model the reluctance force in a CLM. We follow the same modeling procedure as described in the previous section. The main difference compared to the previous section is that the magnetic sources in this case are the current-carrying coils.

### 2.5.1 Division in regions

The geometry is divided into source and source-free regions as shown in Figure 2.5. Region II is the current-carrying region. Regions I and III are the source-free regions.

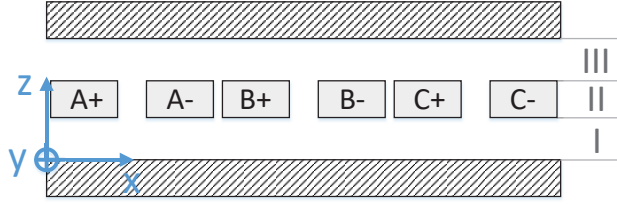


Figure 2.5: Coil regions.

The fundamental frequency of a region is defined as

$$\omega_n = \frac{n\pi}{\tau}, \quad (2.48)$$

where  $n$  is the harmonic number. The width of the region is  $2\tau$ . In this section, the width of the region is chosen equal to the length of the coils array, which is  $4\tau_p$ .

### 2.5.2 Source term description

In the reluctance force case, the magnetic sources are the current-carrying coils. Similar to the Lorentz case, we write the magnetic source term, which is the current density, as a Fourier series

$$J_y(x) = J_{y0} + \sum_{n=1}^{\infty} (J_{ycn} \cos(\omega_n x) + J_{ysn} \sin(\omega_n x)), \quad (2.49)$$

where

$$J_{y0} = \frac{1}{4\tau_p} \int_0^{4\tau_p} J_y(x) dx = 0, \quad (2.50)$$

$$\begin{aligned} J_{ycn} &= \frac{1}{2\tau_p} \int_0^{4\tau_p} J_y(x) \cos(\omega_n x) dx \\ &= \frac{1}{n\pi} \left( J_A \sin\left(\frac{n\pi w_c}{2\tau_p}\right) - J_A \sin(0) \right. \\ &\quad - J_A \sin\left(\frac{2}{3}n\pi\right) + J_A \sin\left(\frac{n\pi}{2\tau_p}\left(\frac{4}{3}\tau_p - w_c\right)\right) \\ &\quad + J_B \sin\left(\frac{n\pi}{2\tau_p}\left(\frac{4}{3}\tau_p + w_c\right)\right) - J_B \sin\left(\frac{2}{3}n\pi\right) \\ &\quad - J_B \sin\left(\frac{4}{3}n\pi\right) + J_B \sin\left(\frac{n\pi}{2\tau_p}\left(\frac{8}{3}\tau_p - w_c\right)\right) \\ &\quad + J_C \sin\left(\frac{n\pi}{2\tau_p}\left(\frac{8}{3}\tau_p + w_c\right)\right) - J_C \sin\left(\frac{4}{3}n\pi\right) \\ &\quad \left. - J_C \sin(2n\pi) + J_C \sin\left(\frac{n\pi}{2\tau_p}(4\tau_p - w_c)\right) \right), \end{aligned} \quad (2.51)$$

$$\begin{aligned}
J_{y sn} &= \frac{1}{2\tau_p} \int_0^{4\tau_p} J_y(x) \sin(\omega_n x) dx \\
&= \frac{1}{n\pi} \left( -J_A \cos\left(\frac{n\pi w_c}{2\tau_p}\right) + J_A \cos(0) \right. \\
&\quad + J_A \cos\left(\frac{2}{3}n\pi\right) - J_A \cos\left(\frac{n\pi}{2\tau_p}\left(\frac{4}{3}\tau_p - w_c\right)\right) \\
&\quad - J_B \cos\left(\frac{n\pi}{2\tau_p}\left(\frac{4}{3}\tau_p + w_c\right)\right) + J_B \cos\left(\frac{2}{3}n\pi\right) \\
&\quad + J_B \cos\left(\frac{4}{3}n\pi\right) - J_B \cos\left(\frac{n\pi}{2\tau_p}\left(\frac{8}{3}\tau_p - w_c\right)\right) \\
&\quad - J_C \cos\left(\frac{n\pi}{2\tau_p}\left(\frac{8}{3}\tau_p + w_c\right)\right) + J_C \cos\left(\frac{4}{3}n\pi\right) \\
&\quad \left. + J_C \cos(2n\pi) - J_C \cos\left(\frac{n\pi}{2\tau_p}(4\tau_p - w_c)\right) \right). \tag{2.52}
\end{aligned}$$

Here,  $J_A$ ,  $J_B$  and  $J_C$  are the current densities in the coils A, B and C, respectively.

### 2.5.3 Semianalytical solution

The solutions for the magnetic vector potentials are obtained by solving the Laplace equation in region I and III and Poisson equation in region II. These equations are solved by separation of variables. Since the current density vector  $\vec{j}$  has only a non-zero component in the  $y$ -direction, it follows that the magnetic vector potential  $\vec{A}$  only has a non-zero component in the  $y$ -direction which is denoted by  $A_y$ .

#### Region I and III: air

The field equation in region I is given by the Laplace equation

$$\nabla^2 \vec{A}^I = 0. \tag{2.53}$$

The Laplace equation is solved by separation of variables similar to Section 2.4.3. The resulting magnetic vector potential is

$$A_{yn}^I = \frac{1}{\omega_n} (a_n^I e^{\omega_n z} + b_n^I e^{-\omega_n z}) \cos(\omega_n x) + \frac{1}{\omega_n} (c_n^I e^{\omega_n z} + d_n^I e^{-\omega_n z}) \sin(\omega_n x). \tag{2.54}$$

The flux density can then be calculated using formulas (2.25) and (2.26). The results are as follows

$$B_{xn}^I = (a_n^I e^{\omega_n z} - b_n^I e^{-\omega_n z}) \cos(\omega_n x) + (c_n^I e^{\omega_n z} - d_n^I e^{-\omega_n z}) \sin(\omega_n x) \tag{2.55}$$

$$B_{zn}^I = (a_n^I e^{\omega_n z} + b_n^I e^{-\omega_n z}) \sin(\omega_n x) - (c_n^I e^{\omega_n z} + d_n^I e^{-\omega_n z}) \cos(\omega_n x) \tag{2.56}$$

A similar description of the magnetic flux density applies to region III.

### Region II: Coil array

The field equation in region II is given by the Poisson equation

$$\nabla^2 \vec{A}^{\text{II}} = -\mu \vec{J}. \quad (2.57)$$

The Poisson equation is solved by separation of variables similar to Section 2.4.3. The resulting magnetic vector potential is

$$\begin{aligned} A_{yn}^{\text{II}} = & \frac{1}{\omega_n} \left( a_n^{\text{II}} e^{\omega_n z} + b_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ycn} \right) \cos(\omega_n x) \\ & + \frac{1}{\omega_n} \left( c_n^{\text{II}} e^{\omega_n z} + d_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ysn} \right) \sin(\omega_n x). \end{aligned} \quad (2.58)$$

The flux density can then be calculated using formulas (2.25) and (2.26). The results are as follows

$$B_{xn}^{\text{II}} = \left( a_n^{\text{II}} e^{\omega_n z} - b_n^{\text{II}} e^{-\omega_n z} \right) \cos(\omega_n x) + \left( c_n^{\text{II}} e^{\omega_n z} - d_n^{\text{II}} e^{-\omega_n z} \right) \sin(\omega_n x), \quad (2.59)$$

$$\begin{aligned} B_{zn}^{\text{II}} = & \left( a_n^{\text{II}} e^{\omega_n z} + b_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ycn} \right) \sin(\omega_n x) \\ & - \left( c_n^{\text{II}} e^{\omega_n z} + d_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ysn} \right) \cos(\omega_n x). \end{aligned} \quad (2.60)$$

### Summary

In summary, the magnetic flux densities in all regions are given by

$$B_{xn}^{\text{I}} = \left( a_n^{\text{I}} e^{\omega_n z} - b_n^{\text{I}} e^{-\omega_n z} \right) \cos(\omega_n x) + \left( c_n^{\text{I}} e^{\omega_n z} - d_n^{\text{I}} e^{-\omega_n z} \right) \sin(\omega_n x), \quad (2.61a)$$

$$B_{zn}^{\text{I}} = \left( a_n^{\text{I}} e^{\omega_n z} + b_n^{\text{I}} e^{-\omega_n z} \right) \sin(\omega_n x) - \left( c_n^{\text{I}} e^{\omega_n z} + d_n^{\text{I}} e^{-\omega_n z} \right) \cos(\omega_n x), \quad (2.61b)$$

$$B_{xn}^{\text{II}} = \left( a_n^{\text{II}} e^{\omega_n z} - b_n^{\text{II}} e^{-\omega_n z} \right) \cos(\omega_n x) + \left( c_n^{\text{II}} e^{\omega_n z} - d_n^{\text{II}} e^{-\omega_n z} \right) \sin(\omega_n x), \quad (2.61c)$$

$$\begin{aligned} B_{zn}^{\text{II}} = & \left( a_n^{\text{II}} e^{\omega_n z} + b_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ycn} \right) \sin(\omega_n x) \\ & - \left( c_n^{\text{II}} e^{\omega_n z} + d_n^{\text{II}} e^{-\omega_n z} + \frac{\mu_0}{\omega_n} J_{ysn} \right) \cos(\omega_n x), \end{aligned} \quad (2.61d)$$

$$B_{xn}^{\text{III}} = \left( a_n^{\text{III}} e^{\omega_n z} - b_n^{\text{III}} e^{-\omega_n z} \right) \cos(\omega_n x) + \left( c_n^{\text{III}} e^{\omega_n z} - d_n^{\text{III}} e^{-\omega_n z} \right) \sin(\omega_n x), \quad (2.61e)$$

$$B_{zn}^{\text{III}} = \left( a_n^{\text{III}} e^{\omega_n z} + b_n^{\text{III}} e^{-\omega_n z} \right) \sin(\omega_n x) - \left( c_n^{\text{III}} e^{\omega_n z} + d_n^{\text{III}} e^{-\omega_n z} \right) \cos(\omega_n x). \quad (2.61f)$$

### 2.5.4 Boundary conditions

The boundary conditions are similar to Section 2.4.4. For the considered CLM topology, the boundary conditions are

- BC1:  $H_x^{\text{I}} \Big|_{z=0} = 0$ .
- BC2:  $B_z^{\text{I}} \Big|_{z=z_1} = B_z^{\text{II}} \Big|_{z=z_1}$ ,

- BC3:  $H_x^I \Big|_{z=z_I} = H_x^{II} \Big|_{z=z_I}$ ,
- BC4:  $B_z^{II} \Big|_{z=z_{II}} = B_z^{III} \Big|_{z=z_{II}}$ ,
- BC5:  $H_x^{II} \Big|_{z=z_{II}} = H_x^{III} \Big|_{z=z_{II}}$ ,
- BC6:  $H_x^{III} \Big|_{z=z_{III}} = 0$ ,

where

$$z_I = h_m + \delta, \quad (2.62)$$

$$z_{II} = h_m + \delta + h_c, \quad (2.63)$$

$$z_{III} = 2h_m + \delta + h_c. \quad (2.64)$$

These boundary conditions lead to a set of linear equations

$$\text{BC1: } a_n^I - b_n^I = 0, \quad (2.65a)$$

$$c_n^I - d_n^I = 0, \quad (2.65b)$$

$$\text{BC2: } a_n^I e^{\omega_n z_I} + b_n^I e^{-\omega_n z_I} - a_n^{II} e^{\omega_n z_I} - b_n^{II} e^{-\omega_n z_I} = \frac{\mu_0}{\omega_n} J_{ycn}, \quad (2.65c)$$

$$c_n^I e^{\omega_n z_I} + d_n^I e^{-\omega_n z_I} - c_n^{II} e^{\omega_n z_I} - d_n^{II} e^{-\omega_n z_I} = \frac{\mu_0}{\omega_n} J_{ysn}, \quad (2.65d)$$

$$\text{BC3: } a_n^I e^{\omega_n z_I} - b_n^I e^{-\omega_n z_I} - a_n^{II} e^{\omega_n z_I} + b_n^{II} e^{-\omega_n z_I} = 0, \quad (2.65e)$$

$$c_n^I e^{\omega_n z_I} - d_n^I e^{-\omega_n z_I} - c_n^{II} e^{\omega_n z_I} + d_n^{II} e^{-\omega_n z_I} = 0, \quad (2.65f)$$

$$\text{BC4: } a_n^{II} e^{\omega_n z_{II}} + b_n^{II} e^{-\omega_n z_{II}} - a_n^{III} e^{\omega_n z_{II}} - b_n^{III} e^{-\omega_n z_{II}} = -\frac{\mu_0}{\omega_n} J_{ycn}, \quad (2.65g)$$

$$c_n^{II} e^{\omega_n z_{II}} + d_n^{II} e^{-\omega_n z_{II}} - c_n^{III} e^{\omega_n z_{II}} - d_n^{III} e^{-\omega_n z_{II}} = -\frac{\mu_0}{\omega_n} J_{ysn}, \quad (2.65h)$$

$$\text{BC5: } a_n^{II} e^{\omega_n z_{II}} - b_n^{II} e^{-\omega_n z_{II}} - a_n^{III} e^{\omega_n z_{II}} + b_n^{III} e^{-\omega_n z_{II}} = 0, \quad (2.65i)$$

$$c_n^{II} e^{\omega_n z_{II}} - d_n^{II} e^{-\omega_n z_{II}} - c_n^{III} e^{\omega_n z_{II}} + d_n^{III} e^{-\omega_n z_{II}} = 0, \quad (2.65j)$$

$$\text{BC6: } a_n^{III} e^{\omega_n z_{III}} - b_n^{III} e^{-\omega_n z_{III}} = 0, \quad (2.65k)$$

$$c_n^{III} e^{\omega_n z_{III}} - d_n^{III} e^{-\omega_n z_{III}} = 0. \quad (2.65l)$$

The unknown coefficients  $a_n^k$ ,  $b_n^k$ ,  $c_n^k$  and  $d_n^k$ , with  $k = I, II, III$ , are then obtained by solving the set of equations (2.65).

## 2.5.5 Force calculation

Reluctance forces are the attractive forces between the coil array and the two back iron plates. The reluctance forces are only present in the  $z$ -direction. The reluctance force between the coil and the lower back iron plate in Figure 2.5 can be calculated

by applying Maxwell stress tensor to a plane surface  $S$  in the air gap at  $z_S = h_m + \delta$  as follows [80]

$$\begin{aligned} F_{z\text{rel}}^I &= \frac{D}{2\mu_0} \oint_S ((B_z^I)^2 - (B_x^I)^2) dS \\ &= \frac{4\tau_p D}{\mu_0} \sum_{n=1}^{\infty} (a_n^I b_n^I + c_n^I d_n^I). \end{aligned} \quad (2.66)$$

The reluctance force between the coil and the upper back iron plate  $F_{z\text{rel}}^{\text{III}}$  is calculated in the same way. The total reluctance force is then calculated as the sum of the two reluctance forces. It can be seen from (2.65) and (2.66) that the reluctance force is dependent on the current density  $J$  and the position of the translator in the  $z$ -direction, but it is not dependent on the position in the  $x$ -direction.

## 2.6 Model validation

In this section, the Fourier modeling procedure described in the previous sections is applied to an example motor. The resulting Fourier model is compared to a FEM model for validation. The FEM is a numerical technique for solving partial differential equations. The mathematical model is divided into small components of simple geometry, which are called finite elements. The response of the mathematical model is obtained by assembling the responses of all elements. The FEM is highly accurate and is applicable to complex geometric shapes. Therefore, in this chapter, we validate the Fourier model by comparing its output to the output of the FEM model.

A FEM model of an example CLM is built using Cobham Opera Simulation software as shown in Figure 2.6. The motor parameters are summarized in Table 2.2. The example motor has two sets of three-phase coils next to each other. Some artificial manufacturing tolerances are added to make the motor nonideal. Firstly, the coils are shifted 1 mm out of the center of the air gap in the  $z$ -direction. This displacement causes not only parasitic Lorentz forces and torques, but also reluctance forces and torques, since the motor geometry becomes asymmetric. Secondly, some magnets have higher or lower remanent magnetization as depicted in Figure 2.6. The variation of remanent magnetization causes parasitic Lorentz forces and torques in the motor.

For Fourier modeling, the whole motor length of 780 mm is taken as the base Fourier period. The number of harmonics is chosen as  $n_F = 50$ . The manufacturing tolerances of the FEM model are also included in the Fourier model. We simulate



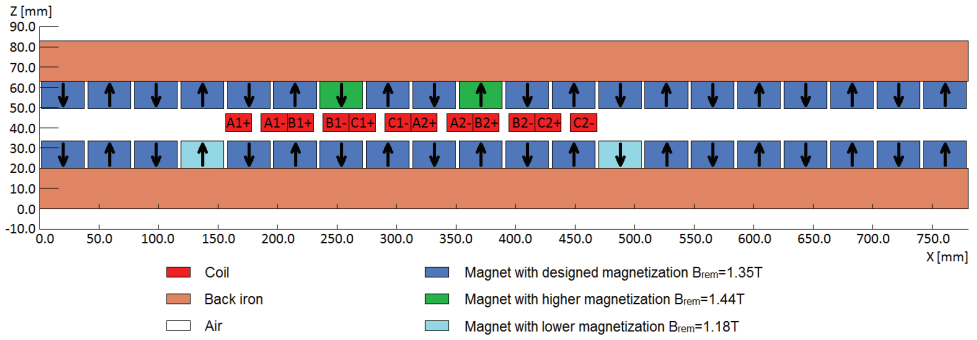


Figure 2.6: FEM model of the example coreless linear motor.

Table 2.2: Geometric parameters and material properties of the example motor.

Parameter	Symbol	Value	Unit
Magnet pole pitch	$\tau_p$	39	mm
Coil pitch	$\tau_c$	52	mm
Air gap	$g$	3.45	mm
Magnet width	$w_m$	36	mm
Magnet height	$h_m$	13.6	mm
Coil width	$w_c$	52	mm
Coil height	$h_c$	9	mm
Coil leg width	$w_b$	22	mm
Number of turns per coil	$N_t$	352	
Motor depth	$D$	90	mm
Remanent magnetization	$B_{rem}$	1.35	T
Motor constant	$k_m$	129.6	N/A
Magnet relative permeability	$\mu_r$	1.033	
Number of coil sets	$N_c$	2	

the Fourier model and the FEM model with the following input currents

$$u_j^A = \frac{F_x^*}{k_m} \cos\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right), \quad (2.67a)$$

$$u_j^B = \frac{F_x^*}{k_m} \cos\left(\frac{\pi}{\tau_p}x\right), \quad (2.67b)$$

$$u_j^C = \frac{F_x^*}{k_m} \cos\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right), \quad (2.67c)$$

where  $j \in \{1, 2\}$  is the index of the coil set,  $F_x^*$  is the reference driving force and  $k_m$  is the motor force constant. In this example motor we have  $k_m = 129.6$  N/A. The reference force is chosen as  $F_x^* = 1000$  N. The simulation is performed along the  $x$ -direction, in the range between 156 mm and 312 mm.

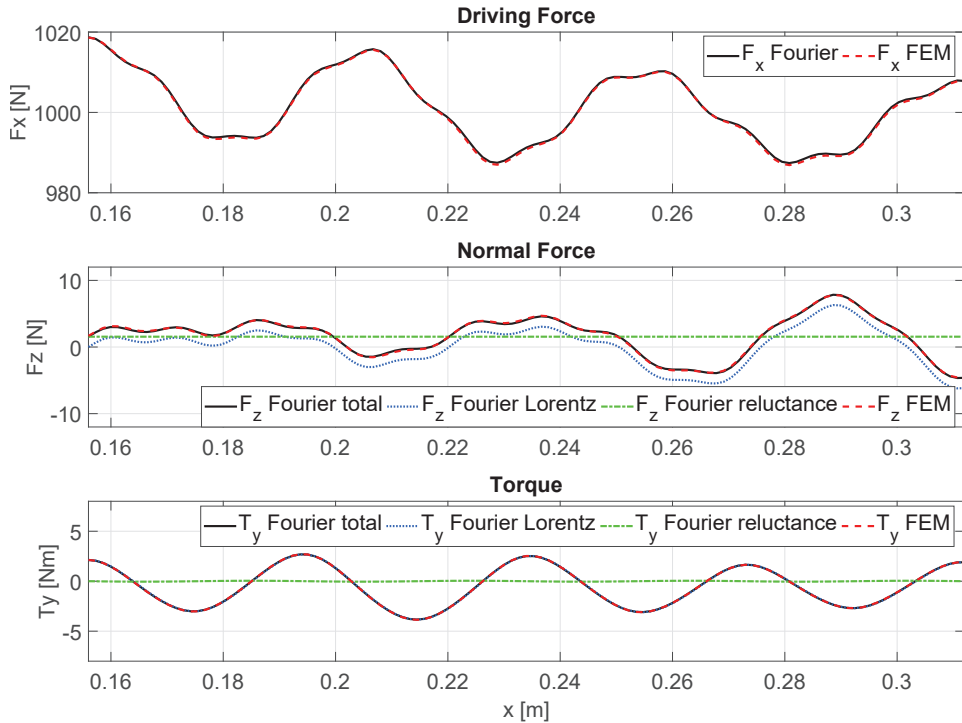


Figure 2.7: Forces and torque produced by the Fourier model and the FEM model.

The resulting forces and torque of the Fourier model and the FEM model are plotted in Figure 2.7. It can be seen that the outputs of the Fourier model and FEM model are closely matched. The root mean square (rms) values of the errors of  $F_x$ ,  $F_z$  and  $T_y$  are 0.31 N, 0.09 N and 0.01 Nm, respectively.

The simulation results have demonstrated that the Fourier modeling technique is able to accurately capture the parasitic forces and torques in a nonideal CLM. Furthermore, the Fourier model is analytical and has simple structure, which is attractive for model-based control purpose.

However, the Fourier model requires knowledge of the exact manufacturing tolerances of the motor. For instance, to model the example motor in Figure 2.6, we need to know the exact  $z$ -displacement of the coil array and the exact remanent magnetization of each magnet. In practice, it is generally difficult to obtain these informations. This problem can be addressed by using data-driven modeling methods based on the model structure of the Fourier model. This topic will be discussed in Chapter 3.

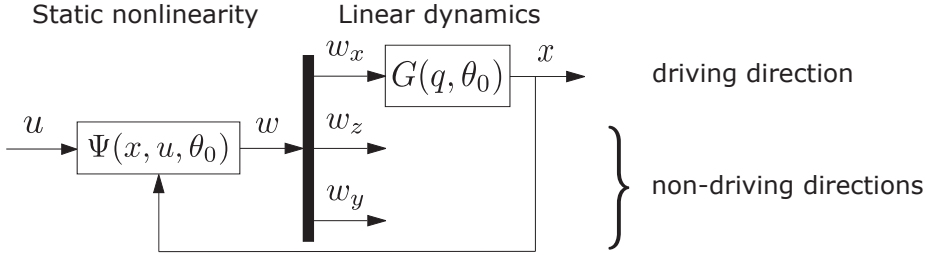


Figure 2.8: Block diagram of a CLM model .

## 2.7 Model structure

Based on the analysis in the previous sections, we can now formulate a general model structure of a CLM, for both the driving and non-driving directions. The overall model structure of a CLM is depicted in Figure 2.8. The description of the signals in Figure 2.8 is listed below.

- $u \in \mathbb{R}^{2N_c}$  is the input current vector. Here,  $N_c$  is the number of coil sets.
- $w_x \in \mathbb{R}$  is the force in the  $x$ -direction, which is denoted as  $F_x$  in the previous sections.
- $w_z \in \mathbb{R}$  is the force in the  $z$ -direction, which is denoted as  $F_z$  in the previous sections.
- $w_y \in \mathbb{R}$  is the torque around the  $y$ -direction, which is denoted as  $T_y$  in the previous sections.
- $x \in \mathbb{R}$  is the position of the translator.

It can be seen that a CLM is a nonlinear multiple-input single-output (MISO) system. In the driving direction, a CLM can be modeled as a linear dynamical system preceded by a static nonlinearity. In the non-driving directions, there is no motion dynamics due to the high-stiffness bearing. Therefore, the CLM can be modeled just as a static nonlinearity.

As analyzed in Section 2.3, the forces produced by a CLM are dependent on the input currents, the position in the driving direction, and the velocity in the driving direction. The input-dependent and position-dependent forces are captured in the static nonlinearity part, while the velocity-dependent force is captured in the linear dynamics part.

### 2.7.1 Linear dynamics

The linear dynamics part captures the motion dynamics of the CLM in the driving direction. Its input is the driving force  $w_x$  and its output is the position  $x$  of the

translator in the driving direction. The velocity-dependent drag force in the driving direction can be modeled as a damping coefficient  $d$  in this linear dynamics part. Let  $m$  be the mass of the translator, then the linear dynamics in the driving direction can be written as

$$G(s) = \frac{1}{ms^2 + ds}. \quad (2.68)$$

This continuous-time transfer function can be discretized with a sampling period  $T_s$  using zero-order hold. The resulting discrete-time transfer function is given by

$$G(q, \theta) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}, \quad (2.69)$$

where  $q^{-1}$  is the delay operator with  $q^{-i}x(t) = x(t - i)$  and

$$\begin{aligned} a_1 &= -1 - e^{-\frac{d}{m}T_s}, & a_2 &= e^{-\frac{d}{m}T_s}, \\ b_1 &= \frac{m}{d^2} \left( \frac{d}{m}T_s - 1 + e^{-\frac{d}{m}T_s} \right), & b_2 &= \frac{m}{d^2} \left( 1 - e^{-\frac{d}{m}T_s} - \frac{d}{m}T_s e^{-\frac{d}{m}T_s} \right). \end{aligned}$$

Here,  $\theta \in \mathbb{R}^{n_\theta}$  denotes the vector of all the parameters to be identified, not only in the linear dynamics part but also in the static nonlinearity part.

## 2.7.2 Static nonlinearity

The static nonlinearity part describes the relation between the input currents and the forces and torques produced by the motor. The force components which are position- and current-dependent are captured in this static nonlinearity part. For ease of presentation, we will only describe the model structure of the static nonlinearity in the driving direction. The same model structure applies to the non-driving directions.

Given the first-principle model structures of these force components in Section 2.3, the static nonlinearity in the driving direction can be represented by the following form

$$w_x = \Psi_x(x, u, \theta) = \Phi_x(x, \theta)\Lambda_x(u). \quad (2.70)$$

Here, the functions  $\Lambda_x(u)$  and  $\Phi_x(x, \theta)$  represent the current-dependent and position-dependent behavior of the force components, respectively.

In general, the function  $\Lambda(u)$  is a column vector in  $\mathbb{R}^{n_\Lambda}$  which can contain the following three components

$$\Lambda(u) = \begin{bmatrix} \Lambda^{\text{Lor}}(u) \\ \Lambda^{\text{rel}}(u) \\ \Lambda^{\text{pos}} \end{bmatrix}. \quad (2.71)$$

This function describes the dependency of the force components on the input  $u$ .

- Lorentz force: in the Lorentz force case, the force is linear with the input current. Therefore, the current dependent function  $\Lambda_{\text{Lor}}(u)$  has the following form

$$\Lambda^{\text{Lor}}(u) = [u_1 \quad u_2 \quad \dots \quad u_{n_u}]^\top. \quad (2.72)$$

- Reluctance force: the reluctance forces are represented by quadratic functions and all cross terms as follows

$$\Lambda^{\text{rel}}(u) = [u_1 u_1 \quad u_1 u_2 \quad \dots \quad u_2 u_2 \quad u_2 u_3 \quad \dots \quad u_{n_u} u_{n_u}]^\top. \quad (2.73)$$

- Position-dependent only force: the function  $\Lambda_{\text{pos}}$  is used to represent the force component which is position-dependent but does not depend on the input  $u$ . The function  $\Lambda^{\text{pos}}$  can thus be represented by the unit function

$$\Lambda^{\text{pos}} = 1. \quad (2.74)$$

Although this type of force component is not present in coreless linear motors, it can be used to describe the position-dependent cogging force in ironcore linear motors.

The force function  $\Phi_x(x, \theta)$  is a position-dependent row vector

$$\Phi_x(x, \theta) = [\Phi_{x,1}(x, \theta) \quad \dots \quad \Phi_{x,n_\Lambda}(x, \theta)]. \quad (2.75)$$

It represents the position-dependent behavior of the force components. As analyzed in the previous sections, each force function  $\Phi_{x,l}(x, \theta)$ , with  $l = 1, \dots, n_\Lambda$ , can be described by a Fourier series

$$\Phi_{x,l}(x, \theta) = f_l + \sum_{n=1}^{n_F} (c_{l,n} \cos(\omega_n x) + d_{l,n} \sin(\omega_n x)), \quad l = 1, \dots, n_\Lambda, \quad (2.76)$$

where  $n_F$  is the number of Fourier harmonics;  $n_\Lambda$  is the number of elements in  $\Lambda_x(u)$ ;  $f_l$ ,  $c_{l,n}$  and  $d_{l,n}$  are the Fourier coefficients. Here, we denote  $\omega_n = n\omega_1$ , where  $\omega_1$  is the fundamental frequency of the Fourier series. It should be noted that the reluctance force is not dependent on the  $x$ -position as analyzed in Section 2.5. Therefore, we have  $c_{l,n}^{\text{rel}} = 0$  and  $d_{l,n}^{\text{rel}} = 0$ , for all  $n = 1, \dots, n_F$ .

The static nonlinearity in the non-driving directions are modeled in the same way. However, there is a difference in the types of forces present in each direction. In the driving direction, the main force component is the Lorentz force. In the non-driving directions, both Lorentz force and reluctance force are present. As a result, the function  $\Lambda(u)$  in each direction can be written as follows

$$\Lambda_x(u) = \Lambda^{\text{Lor}}(u), \quad \Lambda_z(u) = \Lambda_y(u) = \begin{bmatrix} \Lambda^{\text{Lor}}(u) \\ \Lambda^{\text{rel}}(u) \end{bmatrix}. \quad (2.77)$$

In ironcore linear motors where cogging force is present,  $\Lambda_x(u)$  also contains  $\Lambda^{\text{pos}}$ .

## 2.8 Conclusions

In this section we have analyzed the three main force components in a nonideal CLM, which are the Lorentz force, the reluctance force and the drag force. First-principle modeling of Lorentz force and reluctance force using Fourier modeling method has been presented. In order to capture the manufacturing tolerances, the whole motor length is considered as the base Fourier period. The resulting Fourier models have been validated by comparing their outputs to the outputs of FEM models. The results are closely matched. This demonstrates the ability of the Fourier modeling technique to accurately model the parasitic forces and torques in a non-ideal CLM. Furthermore, the Fourier model is analytical and has simple structure, which is attractive for model-based control purpose.

Based on the analysis of the force components and the insights provided by the Fourier model, a general model structure of CLMs has been formulated. This general model structure will be utilized for identification and model-base controller design purposes in the next chapters.

# Chapter 3

## Data-driven modeling of coreless linear motors

### 3.1 Introduction

First-principle modeling provides an insight into the physical behavior of a linear motor. There are first-principle modeling methods to model a linear motor with high accuracy, such as finite element method, surface charge modeling method and Fourier modeling method. However, first-principle modeling methods require knowledge of the exact physical parameters of the motor, including all manufacturing tolerances such as the variation in size and magnetization of each magnet, the variation in size of each coil, the misalignment of the coil within the air gap, etc. Unfortunately, the knowledge of the manufacturing tolerances in a linear motor is generally not available in practice. To address this problem, instead of calculating the parameters of the first-principle model by using the exact geometric parameters and material properties as described in Chapter 2, we aim to identify them from measurement data, thereby eliminating the need to know all the manufacturing tolerances in the linear motor.

In the literature, the existing research works on identification of linear motors have considered only the driving direction. To the best of the author's knowledge, there has been no research on identification of linear motors in the non-driving directions. In the driving direction, identification of linear motors is typically formulated as identification of the model of the force ripple as function of position, velocity and currents. The force ripple model is usually written as a sum of basis functions where the coefficients are to be estimated by fitting the model to measurement

---

Part of the content of this chapter has been published in:

- T. T. Nguyen, M. Lazar, H. Butler and P. M. J. Van den Hof, "An instrumental variable method for closed-loop identification of coreless linear motors," in IEEE American Control Conference (ACC), Milwaukee, WI, 2018.

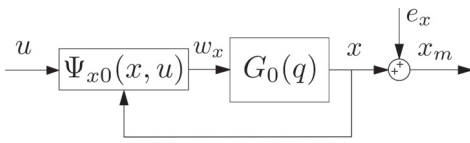


Figure 3.1: Linear motor model in the driving direction.

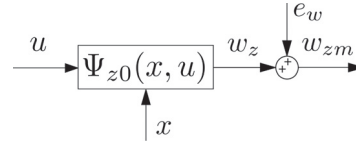


Figure 3.2: Linear motor model in the  $z$ -direction.

data. Some of the research works only consider position-dependent and velocity-dependent force ripple [62, 113]. Several methods to identify the current-dependent force ripple has been developed in [7, 99, 114]. However, the contribution of the current in each coil to the force ripple is not addressed therein. In [88, 89], a method to identify the force function of each coil is proposed, but this is limited to linear motors with only one set of three-phase coils. Furthermore, the above-mentioned works in general require measurements of the driving force, which are usually not available in real applications. In addition, the effect of the output measurement noise on the parameter estimation has not been addressed.

In this chapter, we consider the identification of linear motors problem in both driving and non-driving directions. Based on the analysis in Chapter 2, the models of a CLM in the driving and non-driving directions can be represented by block structures as shown in Figure 3.1 and Figure 3.2, respectively.

In the driving direction, a CLM can be modeled as a multiple-input single-output (MISO) nonlinear dynamical system with electrical currents as inputs and the position of the translator in the driving direction as output. The nonlinear system consists of a linear dynamical system preceded by a static nonlinearity which is nonlinearly dependent both on the input currents and on the noise-free position output as depicted in Figure 3.1. The linear dynamical system captures the motion dynamics of the translator. It describes the dynamical relation between the driving force and the position in the driving direction. The static nonlinearity describes the position-dependent relation between the currents in the coils and the resulting forces. It should be noted that this model structure is different from the well known Hammerstein model structure. The static nonlinear part of this model structure is nonlinear not only in the input but also in the noise-free output, while the static nonlinear part of the Hammerstein model structure is nonlinear only in the input [5, 25]. The identification methods available for the Hammerstein model structure are therefore not directly applicable to the CLM model.

In this chapter we aim to develop an identification method which can provide a consistent estimate of the model parameters of a CLM in the driving direction. Our goal is to identify the parameters based on the input currents measurements and output position measurements. The driving force measurement is in general not available. There are two main challenges in this research objective.

Firstly, the CLM must always operate in closed-loop for safety reasons. As a result, the input is correlated with the measurement noise. Consequently, the simple



linear least squares (LS) identification method results in a biased estimate [28, 34, 96]. In this chapter, we address this challenge by using the instrumental variable (IV) identification framework, which has been proven to provide a consistent estimate for closed-loop measurement data [35].

The second challenge is the nonlinear dependency of the static part of the CLM model on the unknown noise-free output. The nonlinear dependency on the unknown noise-free output makes it difficult to find an appropriate predictor model for IV identification. The simple nonlinear autoregressive exogenous (NARX) model structure suffers from unrealistic noise assumptions, which leads to a biased estimate in the presence of output measurement noise [66]. On the other hand, the more realistic nonlinear output error (NOE) model structure can provide a consistent estimate, but it is nonlinear in the parameters and requires the global solution of a nonconvex optimization problem, which is difficult to find. In this chapter, we introduce a new linear-in-the-parameter predictor model, which is a modification of the NARX model. It will be proven that the IV method using the introduced predictor model provides a consistent estimate. The method only requires the analytical solution of a simple generalized linear least squares problem. In addition, we provide an analysis of the bias obtained by using the NARX predictor model. It is shown that in many applications where the output measurement noise is small compared to the magnet pole pitch of the CLM, the simple NARX predictor model, which does not require knowledge of the statistical properties of the output measurement noise, can provide an estimate that is very close to the true parameter. A numerical example is presented for demonstration.

Regarding practical implementation, the new identification method is easy to implement compared to other identification methods for linear motors, e.g. [7, 88, 89]. It is able to identify the force functions of all the coils together with the motion dynamics of the translator from a single experiment. In addition, there is no need to apply a constant load on the motor.

In this chapter we also consider the identification problem of CLMs in non-driving directions. Unlike the driving direction which has an encoder for position measurement, there is no measurement equipment in the non-driving directions in conventional linear motors. This makes it almost impossible to identify the model of the non-driving directions, as there is no information available in these directions. Adding encoders is not practical as the motor is fixed in these directions due to the high-stiffness bearing. However, the motor can still generate forces in these directions. Therefore, our solution is to add force sensors in the non-driving directions. The CLM model in the non-driving directions can then be represented by a position-dependent static nonlinearity, with currents as input and force as output, as shown in Figure 3.2. It should be noted that this static nonlinearity is nonlinearly dependent on the noise-free position output in the driving directions. Our objective is to identify the parameters of this static nonlinear model based on the currents measurements, force measurements and position measurements. Although the model is static, the identification problem is not trivial due to the nonlinear dependency of the model on the unknown noise-free position  $x$ . The IV identification method

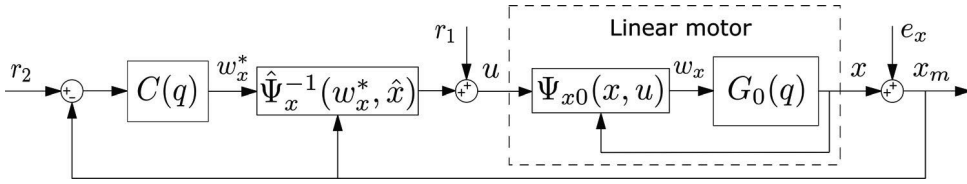


Figure 3.3: A linear motor control loop.

developed for the driving direction can be applied to solve this problem with a few modifications. It will be proven that the resulting estimate is consistent. The effectiveness of the method will be demonstrated in a numerical example.

The remainder of this chapter is organized as follows. An instrumental variable method for identification of driving direction is introduced in Section 3.2. Section 3.3 presents a method for identification of the linear motor model in the non-driving directions. The conclusions are summarized in Section 3.4.

## 3.2 Identification in the driving direction

### 3.2.1 Problem formulation

#### System description

The standard control loop of a CLM is shown in Fig. 3.3. The controller consists of two parts: a commutation block  $\hat{\Psi}_x^{-1}(w_x^*, \hat{x})$ , which aims to invert the static nonlinearity, and a linear controller  $C(q)$  which controls the linear dynamics. The description of the signals is listed below:

- $u(t) \in \mathbb{R}^{n_u}$ : the input of the system, which is the vector of the currents in the coils. The number of independent coils in the translator is  $n_u$ .
- $x(t) \in \mathbb{R}$ : the noise-free output of the system, which is the position of the translator in the driving direction.
- $x_m(t) \in \mathbb{R}$ : the noise-corrupted output, which is the position measurement obtained from the encoder.
- $e_x(t) \in \mathbb{R}$ : the output measurement noise. It is assumed that  $e_x(t)$  is a zero-mean white noise with a symmetric probability distribution.
- $w_x(t) \in \mathbb{R}$ : an unmeasurable internal signal, which is the motor force in the driving direction.
- $\hat{x}(t) \in \mathbb{R}$ : an estimation of the position for commutation purpose.
- $r_2(t) \in \mathbb{R}$ : the output reference.

- $r_1(t) \in \mathbb{R}^{n_u}$ : the additional input excitation signal.

The input signal  $u$  and the output measurement  $x_m$  are known. The signals  $w_x$  and  $x$  cannot be measured. The position estimation  $\hat{x}(t)$  is chosen as the output of the predictor (3.7).

Mathematically, the data generating system is given by the following relations

$$\mathcal{S} : \begin{cases} x_m(t) = x(t) + e_x(t), \\ x(t) = G_0(q)w_x(t), \\ w_x(t) = \Psi_{x0}(x(t), u(t)), \\ u(t) = r_1(t) + \hat{\Psi}_x^{-1}(w_x^*(t), \hat{x}(t)), \\ w_x^*(t) = C(q)(r_2(t) - x_m(t)). \end{cases} \quad (3.1)$$

The input-output relation of the linear motor is given by

$$x_m(t) = G_0(q)\Psi_{x0}(x(t), u(t)) + e_x(t). \quad (3.2)$$

Here,  $t \in \mathbb{N}$  denotes the time instance and  $q^{-1}$  is the delay operator with  $q^{-i}x(t) = x(t-i)$ . The controller  $C(q)$  is chosen such that there is no direct feedthrough from  $x_m(t)$  to  $u(t)$ , which implies that  $u(t)$  and  $e_x(t)$  are uncorrelated. It follows that  $x(t)$  and  $e_x(t)$  are also uncorrelated. It is assumed that the controller  $C(q)$  is known.

### Parameterized model structure

The parameterized model structure of the linear motor is chosen based on the analysis in Section 2.7. The linear dynamics  $G_0(q)$  is described by a discrete-time linear transfer function

$$G_0(q) = G(q, \theta_0) = \frac{\sum_{k=1}^{n_b} b_k^0 q^{-k}}{1 + \sum_{j=1}^{n_a} a_j^0 q^{-j}}, \quad (3.3)$$

where  $n_a$  and  $n_b$  are the orders of the denominator and numerator, respectively. The noise-free output can then be written as

$$x(t) = - \sum_{j=1}^{n_a} a_j^0 x(t-j) + \sum_{k=1}^{n_b} b_k^0 w_x(t-k). \quad (3.4)$$

As analyzed in Section 2.7.2, equation (2.70), the static nonlinearity  $\Psi_{x0}(x(t), u(t))$  is parameterized as follows

$$\begin{aligned} w_x(t) &= \Psi_{x0}(x(t), u(t)) \\ &= \Psi_x(x(t), u(t), \theta_0) \\ &= \sum_{l=1}^{n_\Lambda} \left( \left( f_l^0 + \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t)) + \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t)) \right) \Lambda_l(u(t)) \right), \end{aligned} \quad (3.5)$$

where the function  $\Lambda(u)$  is given by (2.71);  $n_\Lambda$  is the number of elements in  $\Lambda(u)$ ;  $n_F$  is the number of harmonics in the Fourier model. Here,  $\theta_0$  denotes the true

parameter vector of the system. Substituting (3.4) and (3.5) into system (3.1), the input-output relation of the linear motor can be written in the following form

$$\begin{aligned}
x_m(t) &= - \sum_{j=1}^{n_a} a_j^0 x(t-j) + \sum_{k=1}^{n_b} b_k^0 w_x(t-k) + e_x(t) \\
&= - \sum_{j=1}^{n_a} a_j^0 x(t-j) + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} f_l^0 \Lambda_l(u(t-k)) \\
&\quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t-k)) \Lambda_l(u(t-k)) \\
&\quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t-k)) \Lambda_l(u(t-k)) + e_x(t). \tag{3.6}
\end{aligned}$$

Here,  $a_j^0$ ,  $b_k^0$ ,  $f_l^0$ ,  $c_{l,n}^0$  and  $d_{l,n}^0$  are the true parameters of the system. It is noted that (3.6) is a NOE model structure which is nonlinear in both the input  $u$  and the noise-free output  $x$ . Furthermore, the model (3.6) is bilinear in the parameters due to the products  $b_k^0 f_l^0$ ,  $b_k^0 c_{l,n}^0$  and  $b_k^0 d_{l,n}^0$ .

### Problem statement

The identification problem is formulated as follows: given  $N$  samples of input-output measurements  $\{u(t), x_m(t)\}_{t=1}^N$ , identify the unknown parameters

$$\begin{aligned}
a &= [a_1 \dots a_{n_a}]^\top, \quad b = [b_1 \dots b_{n_b}]^\top, \quad f = [f_1 \dots f_{n_\Lambda}]^\top, \\
c &= [c_{1,1} \dots c_{n_\Lambda, n_F}]^\top, \quad d = [d_{1,1} \dots d_{n_\Lambda, n_F}]^\top.
\end{aligned}$$

The internal signal  $w(t)$  is unavailable.

The considered identification problem has two main challenges. Firstly, the identification experiments must be performed in closed-loop for safety reasons. As a result, the input is correlated with the measurement noise. It should be noted that the input  $u(t)$  is uncorrelated with  $e_x(t)$  as there is no direct feed-through in the controller, but it is correlated with the measurement noise at previous time instants  $e_x(t-1)$ ,  $e_x(t-2)$ , etc. Consequently, the simple linear LS identification method results in a biased estimate [28, 34, 96]. This challenge will be addressed using the IV identification framework, which has been proven to provide a consistent estimate for closed-loop measurement data [35, 96].

The second challenge is the nonlinear dependency of the static nonlinear part of the CLM model on the unknown noise-free output, as described in (3.6). The model structure (3.6) is different from the well known Hammerstein model structure [5, 25, 54, 107] in that its static part is nonlinearly dependent both on the inputs and on the noise-free output, while the static part of a Hammerstein model is nonlinearly dependent only on the inputs. The identification methods available for Hammerstein systems are therefore not applicable. The nonlinear dependency of the model structure (3.1) on the unknown noise-free output makes it challenging to find an appropriate predictor for IV identification.

Using the simple NARX predictor model structure, which will be described in (3.14), the noise-corrupted output enters the predictor nonlinearly, which can result in a biased estimate. On the other hand, the more realistic NOE predictor model is nonlinear in the parameters and thus requires solving a nonconvex optimization problem, which is generally difficult.

In this chapter, we employ the IV method, together with a new linear-in-the-parameter predictor, which is a modification of the NARX model. We will prove that the IV method together with the new predictor provide a consistent estimate of the parameters. The method only requires the analytical solution of a generalized linear least squares problem, which is simple to compute.

### 3.2.2 Instrumental variable method

#### The IV framework

Consider a linear-in-the-parameter predictor of the form

$$\hat{x}(t, \theta) = \varphi^\top(t) \theta, \quad (3.7)$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector,  $\varphi(t) \in \mathbb{R}^{n_\theta}$  is the regression vector and  $\hat{x}(t, \theta) \in \mathbb{R}$  is the predicted output. It should be noted that the NOE model (3.6) is nonlinear-in-the-parameter and thus cannot be written in the linear form (3.7). Therefore, in order to apply the IV framework, we have to employ other predictor models which are linear in the parameters as will be described in Section 3.2.3.

The parameter vector  $\theta$  can be estimated using the simple LS solution

$$\hat{\theta}_{\text{LS}} = \left( \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \varphi(t) x_m(t) \right), \quad (3.8)$$

where  $N$  is the data length. However, for closed-loop data in which the input is correlated with the output measurement noise, the LS estimate is biased [28, 34, 96]. As an alternative, we can use the IV estimate, which is the generalized version of the LS estimate. The IV estimate is given by

$$\hat{\theta}_{\text{IV}} = \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) x_m(t) \right), \quad (3.9)$$

where  $\zeta(t) \in \mathbb{R}^{n_\theta}$  is the instrumental vector [35]. The selection of the instrumental vector  $\zeta(t)$  will be discussed later in this section. It can be seen that the IV estimate is the analytical solution of a generalized linear least squares problem and is therefore attractive from a computational perspective.

The IV estimate is said to be consistent if  $\hat{\theta}_{\text{IV}} \rightarrow \theta_0$  with probability 1 as  $N \rightarrow$

$\infty$  [35]. Let us analyze the consistency of the IV estimate. We have

$$\begin{aligned}
\hat{\theta}_{IV} - \theta_0 &= \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) x_m(t) \right) \\
&\quad - \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right) \theta_0 \\
&= \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) (x_m(t) - \varphi(t)^\top \theta_0) \right) \\
&= \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) (x_m(t) - \hat{x}(t, \theta_0)) \right) \tag{3.10}
\end{aligned}$$

As a result, the IV estimate is consistent if the following two conditions are satisfied

$$\bar{\mathbb{E}}[\zeta(t) \varphi(t)^\top] \text{ is nonsingular,} \tag{3.11}$$

$$\bar{\mathbb{E}}[\zeta(t) (x_m(t) - \hat{x}(t, \theta_0))] = 0. \tag{3.12}$$

Here, the notation  $\bar{\mathbb{E}}[\cdot] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \mathbb{E}[\cdot]$  is adopted from the prediction error framework [58]. Condition (3.11) is satisfied if the system is sufficiently excited and  $\zeta(t)$  is well correlated with  $\varphi(t)$ . To satisfy condition (3.12), the instrumental vector  $\zeta(t)$  must be uncorrelated with the measurement noise and the predictor  $\hat{x}(t, \theta)$  should be selected appropriately. The selection of the predictor will be discussed in Section 3.2.3.

### Instrumental vector selection

The instrumental vector  $\zeta(t)$  should be selected such that it is uncorrelated with the output measurement noise  $e_x(t-k)$ , with  $k = 1, \dots, n_b$ , and well correlated with  $\varphi(t)$ . In this chapter, we choose the instrumental vector as the noise-free version of  $\varphi(t)$ , obtained by simulating the noise-free model as shown in Figure 3.4. The noise-free model can be described as follows

$$\mathcal{S} : \begin{cases} \hat{x}(t) = G(q, \hat{\theta}) \hat{w}_x(t), \\ \hat{w}_x(t) = \Psi_x(\hat{x}(t), \hat{u}(t), \hat{\theta}), \\ \hat{u}(t) = r_1(t) + \Psi_x^{-1}(\hat{w}_x^*(t), \hat{x}(t), \hat{\theta}), \\ \hat{w}_x^*(t) = C(q)(r_2(t) - \hat{x}(t)). \end{cases} \tag{3.13}$$

Here,  $\hat{\theta}$  is an estimate of  $\theta_0$ , which can be obtained by first-principle modeling using nominal physical parameters provided by the manufacturer. The signal  $r_1(t)$  and  $r_2(t)$  here are the same as those used in the real experiment, in order to make  $\zeta(t)$  and  $\varphi(t)$  well correlated. It is obvious that  $\zeta(t)$  is uncorrelated with  $e_x(t-k)$ , for all  $k = 1, \dots, n_b$ .

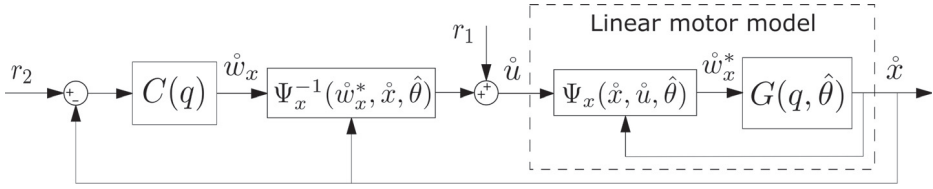


Figure 3.4: The noise-free model for generating instrumental vector.

### 3.2.3 Predictor models

In this section we discuss the selection of the predictor for the IV method. This is a challenging problem due to the nonlinear dependency of the system on the unknown noise-free output. We will show that the simple NARX model results in a biased estimate, although the bias is negligible in many applications. A bias-corrected predictor is then introduced and consistency of the resulting IV estimate is proven.

#### NARX predictor

Let us consider the NARX predictor given by

$$\begin{aligned}
 \hat{x}_{\text{NARX}}(t, \theta) &= - \sum_{j=1}^{n_a} a_j x_m(t-j) + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} f_l \Lambda_l(u(t-k)) \\
 &\quad + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t-k)) \Lambda_l(u(t-k)) \\
 &\quad + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t-k)) \Lambda_l(u(t-k)) \\
 &= \varphi_{\text{NARX}}^\top(t) \theta,
 \end{aligned} \tag{3.14}$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector

$$\begin{aligned}
 \theta &= [a_1 \ \dots \ a_{n_a} \ b_1 f_{11} \ \dots \ b_{n_b} f_{n_\Lambda} \\
 &\quad b_1 c_{1,1} \ \dots \ b_{n_b} c_{n_\Lambda, n_F} \ b_1 d_{1,1} \ \dots \ b_{n_b} d_{n_\Lambda, n_F}]^\top,
 \end{aligned} \tag{3.15}$$

and  $\varphi_{\text{NARX}}(t) \in \mathbb{R}^{n_\theta}$  is the NARX regression vector given by

$$\begin{aligned}
 \varphi_{\text{NARX}}(t) &= [-x_m(t-1) \ \dots \ -x_m(t-n_a) \\
 &\quad \Lambda_1(u(t-1)) \ \dots \ \Lambda_{n_\Lambda}(u(t-n_b)) \\
 &\quad \cos(\omega_1 x_m(t-1)) \Lambda_1(u(t-1)) \ \dots \ \cos(\omega_{n_F} x_m(t-k)) \Lambda_{n_\Lambda}(u(t-n_b)) \\
 &\quad \sin(\omega_1 x_m(t-1)) \Lambda_1(u(t-1)) \ \dots \ \sin(\omega_{n_F} x_m(t-k)) \Lambda_{n_\Lambda}(u(t-n_b))]^\top.
 \end{aligned} \tag{3.16}$$

We have  $n_\theta = n_a + n_b n_\Lambda + 2n_b n_\Lambda n_F$ .

It should be noted that the NARX model (3.14) is written in a linear-in-the-parameter form by using the overparameterization technique, which transforms a bilinear-in-the-parameter model to a linear-in-the-parameter model by replacing every crossproduct of parameters with new independent parameters [5]. When the new parameters have been identified, the original parameters can be obtained by performing singular value decomposition as explained in [5].

In addition, it is important to note that this parameterization is not unique. Any set of parameter vectors  $\tilde{a} = a$ ,  $\tilde{b} = \beta b$ ,  $\tilde{f} = \beta^{-1} f$ ,  $\tilde{c} = \beta^{-1} c$  and  $\tilde{d} = \beta^{-1} d$ , where  $\beta$  is a nonzero scalar, provides identical input-output relation as the one in (3.14). To have a unique parameterization, a common approach is to fix the first element of  $b$  to a constant [5, 54, 74].

In what follows we will analyze the consistency of the IV estimate obtained by using the NARX predictor. For this purpose we will calculate the term in condition (3.12). By substituting  $x_m(t) = x(t) + e_x(t)$  and using trigonometric identities we have

$$\cos(\omega_n x_m(t)) = \cos(\omega_n x(t)) \cos(\omega_n e_x(t)) - \sin(\omega_n x(t)) \sin(\omega_n e_x(t)), \quad (3.17)$$

$$\sin(\omega_n x_m(t)) = \sin(\omega_n x(t)) \cos(\omega_n e_x(t)) + \cos(\omega_n x(t)) \sin(\omega_n e_x(t)). \quad (3.18)$$

Consequently, subtracting  $\hat{x}_{\text{NARX}}(t, \theta_0)$  from  $x_m(t)$  results in

$$\begin{aligned} & x_m(t) - \hat{x}_{\text{NARX}}(t, \theta_0) \\ &= \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t-k)) \Lambda_l(u(t-k)) [1 - \cos(\omega_n e_x(t-k))] \\ & \quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t-k)) \Lambda_l(u(t-k)) [1 - \cos(\omega_n e_x(t-k))] \\ & \quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \sin(\omega_n x(t-k)) \Lambda_l(u(t-k)) \sin(\omega_n e_x(t-k)) \\ & \quad - \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \cos(\omega_n x(t-k)) \Lambda_l(u(t-k)) \sin(\omega_n e_x(t-k)) \\ & \quad + \sum_{j=1}^{n_a} a_j^0 e_x(t-j) + e_x(t). \end{aligned} \quad (3.19)$$

To analyze condition (3.12), let us calculate the expected values of  $\cos(\omega_n e_x(t-k))$  and  $\sin(\omega_n e_x(t-k))$ . The characteristic function of  $e_x(t)$  is defined as [95]

$$\phi_{e_x}(\alpha) := \mathbb{E}[e^{i\alpha e_x(t)}], \text{ where } \alpha \in \mathbb{R}. \quad (3.20)$$

We have the following proposition.

**Proposition 3.2.1.** *If  $e_x(t)$  is a zero-mean white noise with a symmetric probability distribution then*

$$\mathbb{E}[\cos(\omega_n e_x(t))] = \phi_{e_x}(\omega_n), \quad \mathbb{E}[\sin(\omega_n e_x(t))] = 0. \quad (3.21)$$



*Proof.* Using Euler's formula we have

$$\phi_{e_x}(\omega_n) = \mathbb{E}[\cos(\omega_n e_x(t))] + i\mathbb{E}[\sin(\omega_n e_x(t))]. \quad (3.22)$$

Since  $e_x(t)$  is a zero-mean white noise with a symmetric probability distribution, the characteristic function  $\phi_{e_x}(\omega_n)$  is real-valued [95]. By equating the real part and the imaginary part of the two sides of equation (3.22), the proposition is proven.  $\square$

Now let us substitute (3.19) into the term in condition (3.12). We note that  $e_x(t-k)$  is uncorrelated with  $x(t-k)$ ,  $u(t-k)$  and  $\zeta(t)$ , for all  $k = 1, \dots, n_b$ . As a result, using the fact that

$$\mathbb{E}[\chi_1 \chi_2] = \mathbb{E}[\chi_1] \mathbb{E}[\chi_2] \quad (3.23)$$

if  $\chi_1$  and  $\chi_2$  are independent variables, and using the result of Proposition 3.2.1, it follows that

$$\begin{aligned} & \bar{\mathbb{E}}[\zeta(t)(x_m(t) - \hat{x}_{\text{NARX}}(t, \theta_0))] \\ &= \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \cos(\omega_n x(t-k)) \Lambda_l(u(t-k))] (1 - \phi_{e_x}(\omega_n))) \\ & \quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \sin(\omega_n x(t-k)) \Lambda_l(u(t-k))] (1 - \phi_{e_x}(\omega_n))) \\ & \neq 0. \end{aligned} \quad (3.24)$$

Therefore, condition (3.12) is not satisfied. We conclude that the IV method using the NARX predictor does not satisfy the condition for consistency.

### Bias-corrected predictor

In this section we introduce a simple bias-correction for the NARX model. Let us define the bias-correction factors

$$\rho_n = \frac{1}{\mathbb{E}[\cos(\omega_n e_x(t))]} = \frac{1}{\phi_{e_x}(\omega_n)}, \quad n = 1, \dots, n_F. \quad (3.25)$$

Assume that  $\phi_{e_x}(\omega_n)$  is known, we introduce the following bias-corrected predictor

$$\begin{aligned} \hat{x}_{\text{bc}}(t, \theta) &= - \sum_{j=1}^{n_a} a_j x_m(t-j) + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} f_l \Lambda_l(u(t-k)) \\ & \quad + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t-k)) \Lambda_l(u(t-k)) \rho_n \\ & \quad + \sum_{k=1}^{n_b} b_k \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t-k)) \Lambda_l(u(t-k)) \rho_n \\ &= \varphi_{\text{bc}}^\top(t) \theta, \end{aligned} \quad (3.26)$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector defined in (3.15) and  $\varphi_{bc}(t) \in \mathbb{R}^{n_\theta}$  is the bias-corrected regression vector given by

$$\begin{aligned} & \varphi_{bc}^\top(t) \\ &= [-x_m(t-1) \quad \dots \quad -x_m(t-n_a) \\ & \quad \Lambda_1(u(t-1)) \quad \dots \quad \Lambda_{n_\Lambda}(u(t-n_b)) \\ & \quad \cos(\omega_{n_F} x_m(t-1)) \Lambda_1(u(t-1)) \rho_1 \quad \dots \quad \cos(\omega_{n_F} x_m(t-n_b)) \Lambda_{n_\Lambda}(u(t-n_b)) \rho_{n_F} \\ & \quad \sin(\omega_1 x_m(t-1)) \Lambda_1(u(t-1)) \rho_1 \quad \dots \quad \sin(\omega_{n_F} x_m(t-n_b)) \Lambda_{n_\Lambda}(u(t-n_b)) \rho_{n_F}]. \end{aligned} \quad (3.27)$$

It can be seen that the proposed bias-corrected predictor preserves the linear-in-the-parameter property. We will now show that the IV method using the bias-corrected predictor results in a consistent estimate.

**Theorem 3.2.2.** *The IV estimate (3.9), with  $\zeta(t)$  and  $\varphi(t)$  satisfy the following conditions*

- condition (3.11) holds,
- $\zeta(t)$  is uncorrelated with  $e_x(t-k)$ , for all  $k = 1, \dots, n_b$ ,
- $\varphi(t)$  is chosen equal to  $\varphi_{bc}(t)$  given by (3.27),

is consistent.

*Proof.* Let us consider condition (3.12). Due to (3.17) and (3.18), subtracting  $\hat{x}_{bc}(t, \theta_0)$  in (3.26) from  $x_m(t)$  in (3.6) results in

$$\begin{aligned} & x_m(t) - \hat{x}_{bc}(t, \theta_0) \\ &= \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t-k)) \Lambda_l(u(t-k)) [1 - \cos(\omega_n e_x(t-k)) \rho_n] \\ & \quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t-k)) \Lambda_l(u(t-k)) [1 - \cos(\omega_n e_x(t-k)) \rho_n] \\ & \quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \sin(\omega_n x(t-k)) \Lambda_l(u(t-k)) \sin(\omega_n e_x(t-k)) \rho_n \\ & \quad - \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \cos(\omega_n x(t-k)) \Lambda_l(u(t-k)) \sin(\omega_n e_x(t-k)) \rho_n \\ & \quad + \sum_{j=1}^{n_a} a_j^0 e_x(t-j) + e_x(t). \end{aligned} \quad (3.28)$$

Note that for all  $k = 1, \dots, n_b$ , we have that  $e_x(t-k)$  is uncorrelated with  $x(t-k)$ ,  $u(t-k)$  and  $\zeta(t)$ . Therefore, by substituting (3.28) into the term in condition (3.12)

and using (3.23) and Proposition 3.2.1 we have

$$\begin{aligned}
& \bar{\mathbb{E}}[\zeta(t)(x_m(t) - \hat{x}_{bc}(t, \theta_0))] \\
&= \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \cos(\omega_n x(t-k)) \Lambda_l(u(t-k))] (1 - \phi_{e_x}(\omega_n) \rho_n)) \\
&\quad + \sum_{k=1}^{n_b} b_k^0 \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \sin(\omega_n x(t-k)) \Lambda_l(u(t-k))] (1 - \phi_{e_x}(\omega_n) \rho_n)) \\
&= 0.
\end{aligned} \tag{3.29}$$

Therefore, condition (3.12) is satisfied. Given that condition (3.11) is also satisfied, the IV estimate using the bias-corrected predictor (3.26) is consistent.  $\square$

It is noted that the bias-corrected predictor requires knowledge of the probability distribution of the output measurement noise. For example, the two most common types of probability distribution of measurement noise are the normal (or Gaussian) distribution and the uniform distribution.

- If the measurement noise is normally distributed with zero mean and variance  $\sigma^2$ , which is common in practice, then the bias-correction factors are

$$\rho_n^{\text{normal}} = \frac{1}{\phi_{e_x}^{\text{normal}}(\omega_n)} = e^{\frac{\omega_n^2 \sigma^2}{2}}. \tag{3.30}$$

- If the measurement noise is uniformly distributed on the interval  $[-\eta, \eta]$  then the bias-correction factors are

$$\rho_n^{\text{uniform}} = \frac{1}{\phi_{e_x}^{\text{uniform}}(\omega_n)} = \frac{\omega_n \eta}{\sin(\omega_n \eta)}. \tag{3.31}$$

However, it will be shown in later in this section that in many applications, it is possible to obtain an estimate that is very close to the true parameter just by using the simple NARX predictor, which does not require knowledge of the probability distribution of the output measurement noise.

In practice, the measurement noise may not be a zero-mean white noise with a symmetric probability distribution as assumed. This noise model mismatch can lead to inconsistency of the estimation. This is a well known problem for ARX-type predictors which requires further research [58].

### Relation between the NARX IV estimate and the bias-corrected IV estimate

Comparing the NARX regression vector  $\varphi_{\text{NARX}}(t)$  in (3.16) and the bias-corrected regression vector  $\varphi_{bc}(t)$  in (3.27) we have

$$\varphi_{bc}(t) = \Omega \varphi_{\text{NARX}}(t), \tag{3.32}$$

where  $\Omega \in \mathbb{R}^{n_\theta \times n_\theta}$  is the diagonal bias-correction matrix

$$\Omega = \text{diag}(1, \dots, 1, \rho_1, \dots, \rho_{n_F}, \rho_1, \dots, \rho_{n_F}). \quad (3.33)$$

From (3.9) and (3.32), it follows that

$$\hat{\theta}_{\text{IV}}^{\text{bc}} = \Omega^{-1} \hat{\theta}_{\text{IV}}^{\text{NARX}}. \quad (3.34)$$

Therefore, the bias-corrected IV estimate can be obtained simply by multiplying the NARX IV estimate by  $\Omega^{-1}$ .

### Analysis of the bias in NARX IV

Let us calculate the bias of the IV estimate obtained using the NARX predictor. From Theorem 3.2.2, we know that  $\hat{\theta}_{\text{IV}}^{\text{bc}} \rightarrow \theta_0$  with probability 1 as  $N \rightarrow \infty$ . Consequently, it follows from (3.34) that

$$\hat{\theta}_{\text{IV}}^{\text{NARX}} \rightarrow \Omega \theta_0 \text{ with probability 1 as } N \rightarrow \infty. \quad (3.35)$$

Therefore, as  $N \rightarrow \infty$ , the bias is

$$\hat{\theta}_{\text{IV}}^{\text{NARX}} - \theta_0 \rightarrow (\Omega - I)\theta_0 = \Gamma\theta_0, \quad (3.36)$$

where  $I$  is the identity matrix and

$$\Gamma = \text{diag}(0, \dots, 0, \rho_1 - 1, \dots, \rho_{n_F} - 1, \rho_1 - 1, \dots, \rho_{n_F} - 1). \quad (3.37)$$

It is observed from (3.30) and (3.31) that if the variance of the output measurement noise is small compared to the base Fourier period then the factors  $\rho_n$  are very close to 1. For example, if the measurement noise is white Gaussian noise with variance  $\sigma = 0.01\tau_p$ , where  $\tau_p$  is the magnet pole pitch of the CLM, and the Fourier period is  $2\tau_p$ , then  $\rho_1^{\text{normal}} - 1 = 5 \times 10^{-4}$ . In high-precision applications, the magnet pole pitch is usually in the mm range, while the measurement noise is usually in the  $\mu\text{m}$  range, making  $(\rho_n - 1)$  very small. Consequently, the bias is negligible and we can thus safely use the NARX model as the predictor model. The knowledge of the probability distribution of the measurement noise is therefore not required in this case.

### 3.2.4 Algorithm

The main steps of the IV algorithm for identification of CLMs in the driving direction are summarized below.

**Algorithm 3.2.3** (IV algorithm for driving direction).

1. Perform experiment and collect  $N$  samples of input-output measurement data  $\{u(t), x_m(t)\}_{t=1}^N$ .

2. Construct the regression vector  $\varphi(t)$  using (3.16) (NARX) or (3.27) (bias-corrected).
3. Perform simulation with the noise-free model (3.13), using the same signals  $r_1(t)$  and  $r_2(t)$  as used in the experiment, then construct the instrumental vector  $\zeta(t)$  as the noise-free version of the regression vector  $\varphi(t)$ .
4. Calculate the IV estimate using (3.9).

### 3.2.5 Simulation results

In this section, a numerical example is presented to verify the performance of the proposed IV method. Assume that we have a CLM with three-phase coils A, B, C as shown in Fig. 1.2. The coils are connected in star configuration, which implies that the sum of the three currents is zero. Therefore, we actually only have two control inputs  $u_A$  and  $u_B$ .

In the driving direction, the Lorentz force is the main force component in the static nonlinearity. It follows that

$$\Lambda(u) = \Lambda^{\text{Lor}}(u) = \begin{bmatrix} u_A \\ u_B \end{bmatrix}. \quad (3.38)$$

In the ideal case, the force function  $\Phi_{xA}(x)$  and  $\Phi_{xB}(x)$  only contains the first order harmonics  $\omega_1 = \frac{\pi}{\tau_p}$ , where  $\tau_p = 0.04\text{m}$  is the magnet pole pitch. In reality, however, there are also other harmonic components due to manufacturing tolerances. In this example, it is assumed that there is a higher-order harmonic component  $\omega_2 = 2\omega_1$ . Consequently,  $\Phi_x(x)$  can be written as

$$\Phi_x(x) = [\Phi_{xA}(x) \quad \Phi_{xB}(x)] = \begin{bmatrix} \sum_{n=1}^2 (c_{A,n} \cos(\omega_n x) + d_{A,n} \sin(\omega_n x)) \\ \sum_{n=1}^2 (c_{B,n} \cos(\omega_n x) + d_{B,n} \sin(\omega_n x)) \end{bmatrix}^\top. \quad (3.39)$$

Here,  $\Phi_{xA}(x), \Phi_{xB}(x) \in \mathbb{R}$  and  $\Phi_x(x) \in \mathbb{R}^{1 \times 2}$ . As a result, the force produced by the motor in the driving direction can be written as

$$w_x = \begin{bmatrix} \sum_{n=1}^2 (c_{A,n} \cos(\omega_n x) + d_{A,n} \sin(\omega_n x)) \\ \sum_{n=1}^2 (c_{B,n} \cos(\omega_n x) + d_{B,n} \sin(\omega_n x)) \end{bmatrix}^\top \begin{bmatrix} u_A \\ u_B \end{bmatrix}. \quad (3.40)$$

The linear dynamical part is a mass-damper system which has the following discrete-time transfer function

$$G(q, \theta) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}. \quad (3.41)$$

The sampling frequency of the system is  $F_s = 10$  kHz. We fix  $b_1 = 1 \times 10^{-7}$ . The true parameters of the system are given in Table 3.1. The reference signal  $r_2(t)$  is generated as a consecutive sequence of random third-order motion profiles in the

range [0m,0.4m]. The input excitation signal is chosen as  $r_1(t) = \sum_{n=1}^{100} p_n \sin(t + \psi_n)$ , where  $p_n$  and  $\psi_n$  are random amplitudes and phase shifts. The output measurement noise is a zero-mean white Gaussian noise with standard deviation  $\sigma_{e_x} = 5 \times 10^{-6}$  m. The signal to noise ratio is 90 dB.

The system parameters are estimated from closed-loop data of length  $N = 2 \times 10^6$ . A Monte-Carlo simulation of 150 runs is performed. The results are summarized in Table 3.1. It is observed that there is no significant difference between the NARX IV method and the bias-corrected IV method, as the noise variance is small. Both methods give unbiased estimates. However, in spite of the large number of data, the standard deviations of  $b$  and the Fourier coefficients  $c$  and  $d$  are still quite large. How to improve the statistical efficiency of the method needs further research.

Table 3.1: Mean and standard deviation of 150 estimated models

Parameter	True value	NARX IV	Bias-corrected IV
$a_1$	-1.9950	-1.9950 ± 0.0001	-1.9950 ± 0.0001
$a_2$	0.9950	0.9950 ± 0.0001	0.9950 ± 0.0001
$\bar{b}_2$	0.9983	1.0230 ± 0.1881	1.0230 ± 0.1881
$c_{A,1}$	0	-0.0032 ± 0.2716	-0.0032 ± 0.2716
$c_{A,2}$	-0.6988	-0.6965 ± 0.1340	-0.6965 ± 0.1340
$c_{B,1}$	-9.0781	-9.0421 ± 0.8498	-9.0421 ± 0.8498
$c_{B,2}$	-0.2745	-0.2767 ± 0.1126	-0.2767 ± 0.1126
$d_{A,1}$	7.8619	7.8475 ± 0.7323	7.8475 ± 0.7323
$d_{A,2}$	-0.3694	-0.3503 ± 0.1082	-0.3503 ± 0.1082
$d_{B,1}$	-4.5391	-4.5189 ± 0.5322	-4.5189 ± 0.5322
$d_{B,2}$	0.4592	0.4598 ± 0.1288	0.4598 ± 0.1288

Note:  $\bar{b}_2 = b_2 \times 10^7$ .

The Bode diagrams of the linear dynamics of true system and the estimated models are plotted in Figure 3.5. It can be seen that the estimated models are very close to the true system. The Bode diagrams of the 150 estimated models obtained using the bias-corrected IV method are plotted in Figure 3.6.

Figure 3.7 shows the force produced by the true system model and the estimated model when three-phase sinusoidal current waveforms are applied to the coils:

$$u_A = u_p \cos\left(\omega_1 x + \frac{2\pi}{3}\right), \quad u_B = u_p \cos(\omega_1 x), \quad (3.42)$$

where  $u_p = 6.4$  A is the amplitude of the current. The resulting forces are closely matched. The maximum force error between the true system and the estimated model is about 0.46%. The forces produced by the 150 estimated models obtained using the bias-corrected IV method are plotted in Figure 3.8. It is observed from Figure 3.6 and Figure 3.8 that the variance of the estimated models is high.

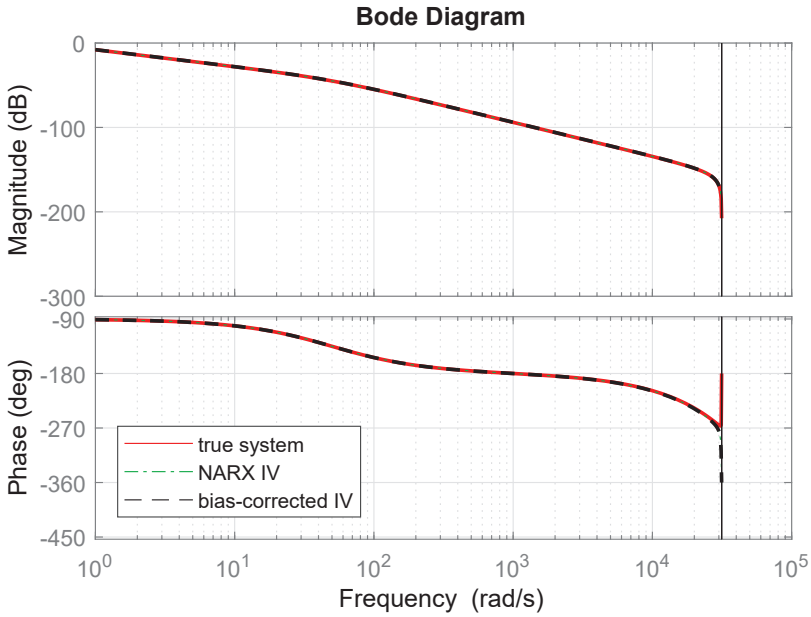


Figure 3.5: Bode diagrams of the true system and the estimated model.

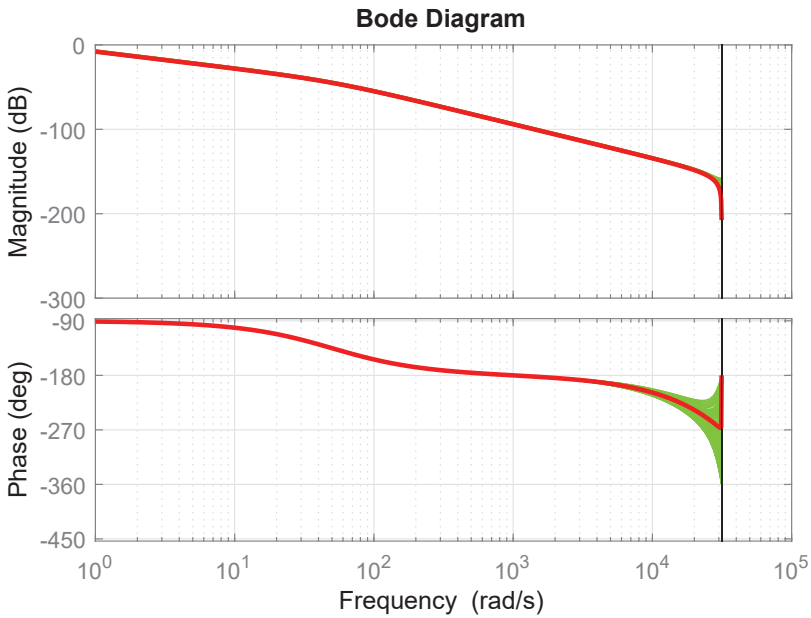


Figure 3.6: Bode diagrams of the true system (red) and the 150 estimated models obtained using the bias-corrected IV method (green).

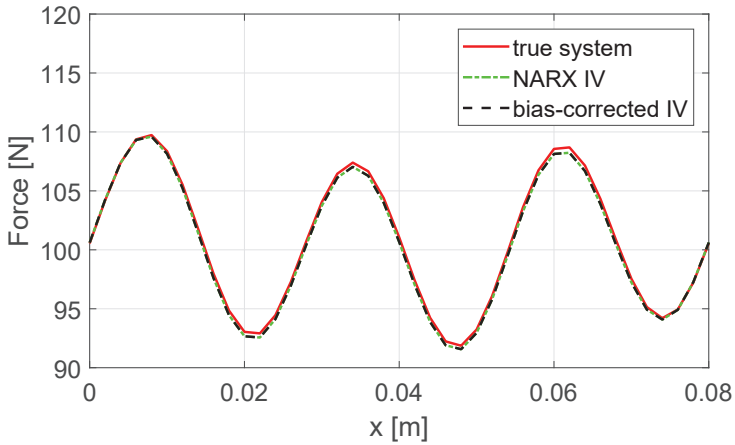


Figure 3.7: Forces produced by the true system and the estimated model.

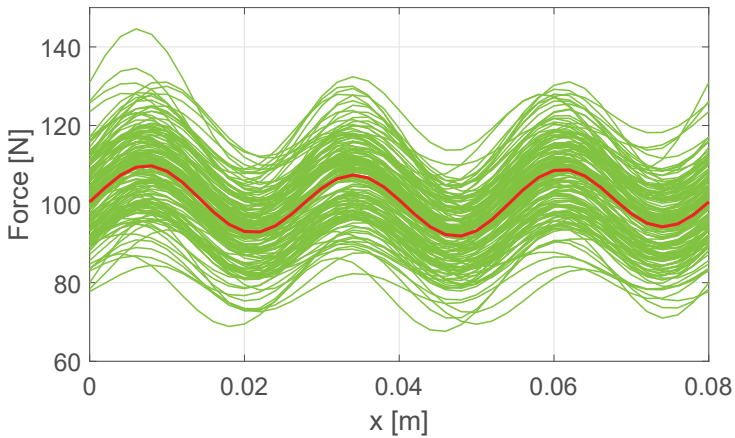


Figure 3.8: Forces produced by the true system (red) and the 150 estimated model obtained using the bias-corrected IV method (green).

The LS estimation (3.8) is also tested but the resulting estimate is very far from the true parameter, due to the correlation between the measurement noise and the inputs, and therefore is not shown here.

## Discussion

It is observed that the statistical efficiency of the method is low. In spite of the high signal to noise ratio and the large amount of data, the variances of  $b$ ,  $c$  and  $d$  are high. In this example, we have to simulate with low measurement noise variance because when we increase the noise variance, it takes a very large amount of data to get a decent estimate of  $b$ ,  $c$  and  $d$ .



A possible cause of this low statistical efficiency is the huge difference in the range of  $a$  compared to the range of  $bc$  and  $bd$ . The absolute values of  $b_k c_{l,n}$  and  $b_k d_{l,n}$  are about  $10^7$  times smaller than those of  $a_j$ . Consequently, a small relative error in the estimate of  $a$  can cause a huge relative error in the estimates of  $bc$  and  $bd$ . This makes the estimates of  $b_k c_{l,n}$  and  $b_k d_{l,n}$  very sensitive to noise and disturbance. This problem causes difficulty in estimating the parameters of a real CLM as will be discussed in Chapter 7. In our experimental setup, the situation is even worse because the mass of the real CLM therein is higher than the mass of the linear motor in this example. This causes the absolute values of  $bc$  and  $bd$  to be even smaller, about  $10^9$  times smaller than the absolute values of  $a$ . One alternative solution for this problem will be discussed in Chapter 7.

### 3.3 Identification in the non-driving directions

In order to identify the model of a CLM in the non-driving directions, it is necessary to have measurement information in these directions. As the motor hardly moves in these directions, using encoders or other position sensors is not practical. However, the motor can still generate forces in these directions. Therefore, our solution is to add force sensors in the non-driving directions. Our aim is to identify the relation between the input currents and the force produced in non-driving directions. Although there is no dynamics involved, the identification problem is not trivial due to the nonlinear dependency of the model on the unknown noise-free position  $x$  in the driving direction. The IV identification method developed in Section 3.2 for the driving direction can be applied to solve this problem with a few modifications as described below.

For ease of presentation, we present the method only for the  $z$ -direction. The same method applies to other non-driving directions.

#### 3.3.1 Problem formulation

##### System description

The control loop of a CLM is depicted in Figure 3.9. In addition to the signals in the driving direction described in Section 3.2.1, the signals in the  $z$ -direction are listed below:

- $w_z(t) \in \mathbb{R}$ : the noise-free output force in the  $z$ -direction.
- $w_{zm}(t) \in \mathbb{R}$ : the noise-corrupted output force measurement obtained from the force sensor.
- $e_w(t) \in \mathbb{R}$ : the output force measurement noise. It is assumed that  $e_w(t)$  is a zero-mean white noise.

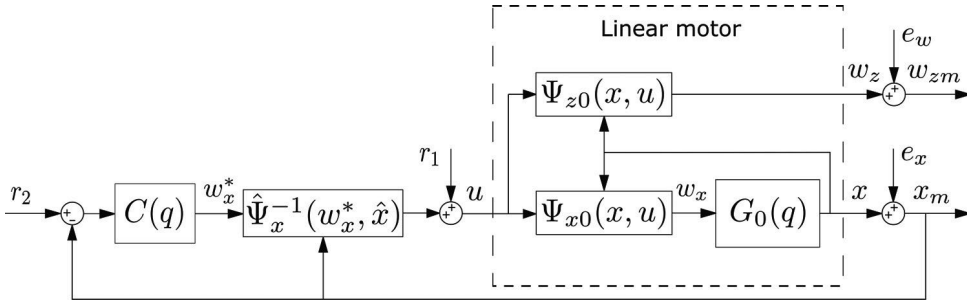


Figure 3.9: A linear motor control loop.

The data generating system is the same as in (3.1) with the addition of the relations in the  $z$ -direction

$$\mathcal{S} : \begin{cases} x_m(t) = x(t) + e_x(t), \\ w_{zm}(t) = w_z(t) + e_w(t), \\ x(t) = G_0(q)w_x(t), \\ w_x(t) = \Psi_{x0}(x(t), u(t)), \\ w_z(t) = \Psi_{z0}(x(t), u(t)), \\ u(t) = r_1(t) + \hat{\Psi}_x^{-1}(w_x^*(t), \hat{x}(t)), \\ w_x^*(t) = C(q)(r_2(t) - x_m(t)). \end{cases} \quad (3.43)$$

The input-output relation of the linear motor in the  $z$ -direction is given by

$$w_{zm}(t) = \Psi_{z0}(x(t), u(t)) + e_w(t). \quad (3.44)$$

### Parameterized model structure

The model structure is chosen based on the analysis in Section 2.7. The static non-linearity  $\Psi_{z0}(x(t), u(t))$  is parameterized as follows

$$\begin{aligned} w_z(t) &= \Psi_{z0}(x(t), u(t)) \\ &= \Psi_z(x(t), u(t), \theta_0) \\ &= \sum_{l=1}^{n_\Lambda} \left( f_l^0 + \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t)) + \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t)) \right) \Lambda_l(u(t)), \end{aligned} \quad (3.45)$$

where  $n_\Lambda$  is the number of elements in  $\Lambda(u)$  and  $n_F$  is the number of harmonics in the Fourier model;  $\theta_0 \in \mathbb{R}^{n_\theta}$  denotes the true parameter vector;  $f_l^0, c_{l,n}^0, d_{l,n}^0 \in \mathbb{R}$  denote the true parameters of the system. Substituting (3.45) into (3.44), the input-output

relation of the linear motor in the  $z$ -direction can be written in the following form

$$w_{zm}(t) = \sum_{l=1}^{n_A} f_l^0 \Lambda_l(u(t)) + \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t)) \Lambda_l(u(t)) \\ + \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t)) \Lambda_l(u(t)) + e_w(t). \quad (3.46)$$

The model structure (3.46) is nonlinear in the noise-free position  $x$  and the input  $u$ . In addition, it is observed that the model structure (3.46) is linear in the parameters.

### Problem statement

The identification problem is formulated as follows: given  $N$  samples of input-output measurements  $\{u(t), x_m(t), w_{zm}(t)\}_{t=1}^N$ , identify the unknown parameters

$$f = [f_1 \dots f_{n_A}]^\top, \quad c = [c_{1,1} \dots c_{n_A, n_F}]^\top, \quad d = [d_{1,1} \dots d_{n_A, n_F}]^\top.$$

Although the model structure (3.46) is static and linear in the parameters, the identification problem is not trivial due to the nonlinear dependency of the model on the unknown noise-free position  $x$  in the driving direction. The IV identification method developed in Section 3.2 for the driving direction can be applied to solve this problem with a few modifications as will be described in the next sections.

### 3.3.2 Instrumental variable method

Consider a linear-in-the-parameter predictor of the form

$$\hat{w}_z(t, \theta) = \varphi^\top(t) \theta, \quad (3.47)$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector,  $\varphi(t) \in \mathbb{R}^{n_\theta}$  is the regression vector and  $\hat{w}_z(t, \theta) \in \mathbb{R}$  is the predicted output.

Similar to Section 3.2, we employ the IV method for identification of the CLM model in the non-driving directions. The IV estimate is given by

$$\hat{\theta}_{\text{IV}} = \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) \varphi(t)^\top \right)^{-1} \left( \frac{1}{N} \sum_{t=1}^N \zeta(t) w_z(t) \right). \quad (3.48)$$

Using similar analysis as in (3.10), it follows that the IV estimate is consistent if the following two conditions are satisfied

$$\bar{\mathbb{E}}[\zeta(t) \varphi(t)^\top] \text{ is nonsingular}, \quad (3.49)$$

$$\bar{\mathbb{E}}[\zeta(t) (w_{zm}(t) - \hat{w}_z(t, \theta_0))] = 0. \quad (3.50)$$

Condition (3.49) is satisfied if the system is sufficiently excited and  $\zeta(t)$  is well correlated with  $\varphi(t)$ . To satisfy condition (3.50), the instrumental vector  $\zeta(t)$  must be uncorrelated with the measurement noise and the predictor  $\hat{w}_z(t, \theta)$  should be selected appropriately. The instrumental vector  $\zeta(t)$  can be chosen as the noise-free version of  $\varphi(t)$ , obtained by simulating the noise-free model similar to Section 3.2.2. The selection of the predictor will be discussed in the next section.

### 3.3.3 Predictor model

In order to apply the IV framework, a linear-in-the-parameter predictor is required. In this section, we analyze the consistency property of the NARX IV predictor and introduce a bias-corrected predictor, which will be proven to provide a consistent estimate.

#### NARX predictor

Let us consider the NARX predictor given by

$$\begin{aligned}\hat{w}_{z\text{NARX}}(t, \theta) &= \sum_{l=1}^{n_\Lambda} f_l \Lambda_l(u(t)) + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t)) \Lambda_l(u(t)) \\ &\quad + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t)) \Lambda_l(u(t)) \\ &= \varphi_{\text{NARX}}^\top(t) \theta,\end{aligned}\tag{3.51}$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector

$$\theta = [f_1 \ \dots \ f_{n_\Lambda} \ c_{1,1} \ \dots \ c_{n_\Lambda, n_F} \ d_{1,1} \ \dots \ d_{n_\Lambda, n_F}]^\top,\tag{3.52}$$

and  $\varphi_{\text{NARX}}(t) \in \mathbb{R}^{n_\theta}$  is the NARX regression vector given by

$$\begin{aligned}\varphi_{\text{NARX}}(t) &= [\Lambda_1(u(t)) \ \dots \ \Lambda_{n_\Lambda}(u(t)) \\ &\quad \cos(\omega_1 x_m(t)) \Lambda_1(u(t)) \ \dots \ \cos(\omega_{n_F} x_m(t)) \Lambda_{n_\Lambda}(u(t)) \\ &\quad \sin(\omega_1 x_m(t)) \Lambda_1(u(t)) \ \dots \ \sin(\omega_{n_F} x_m(t)) \Lambda_{n_\Lambda}(u(t))]^\top.\end{aligned}\tag{3.53}$$

We have  $n_\theta = n_\Lambda(1 + 2n_F)$ .

Similar to Section 3.2.3, it can be proven that the NARX predictor (3.51) results in a biased estimate. Due to (3.17) and (3.18), subtracting (3.51) from (3.46) results in

$$\begin{aligned}w_{zm}(t) - \hat{w}_{zbc}(t, \theta_0) &= \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t)) \Lambda_l(u(t)) [1 - \cos(\omega_n e_x(t))] \\ &\quad + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t)) \Lambda_l(u(t)) [1 - \cos(\omega_n e_x(t))] \\ &\quad + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \sin(\omega_n x(t)) \Lambda_l(u(t)) \sin(\omega_n e_x(t)) \rho_n \\ &\quad - \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \cos(\omega_n x(t)) \Lambda_l(u(t)) \sin(\omega_n e_x(t)) \rho_n \\ &\quad + e_w(t).\end{aligned}\tag{3.54}$$

We note that  $e_x(t)$  is uncorrelated with  $x(t)$ ,  $u(t)$  and  $\zeta(t)$ . Furthermore,  $e_x(t)$  and  $e_w(t)$  are white and zero-mean. Therefore, by substituting (3.54) into condition

(3.50) and using (3.23) and Proposition 3.2.1 we have

$$\begin{aligned}
& \bar{\mathbb{E}}[\zeta(t)(w_{zm}(t) - \hat{w}_{zbc}(t, \theta_0))] \\
&= \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n}^0 \left( \bar{\mathbb{E}}[\zeta(t) \cos(\omega_n x(t)) \Lambda_l(u(t))] (1 - \phi_{e_x}(\omega_n)) \right) \\
&\quad + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n}^0 \left( \bar{\mathbb{E}}[\zeta(t) \sin(\omega_n x(t)) \Lambda_l(u(t))] (1 - \phi_{e_x}(\omega_n)) \right) \\
&\neq 0.
\end{aligned} \tag{3.55}$$

Therefore, condition (3.50) is not satisfied. We conclude that the IV method using the NARX predictor results in a biased estimate. However, it can be proved in a same way as in Section 3.2.3 that the bias can be negligible when the variance of the position measurement noise is small compared to the magnet pole pitch of the CLM. When this is not the case, the bias can be eliminated by using the bias-corrected predictor introduced below.

### Bias-corrected predictor

In order to eliminate the bias, similar to Section 3.2.3, the bias-correction factors  $\rho_n$  defined in (3.25) are employed, resulting in the following bias-corrected predictor

$$\begin{aligned}
\hat{w}_{zbc}(t, \theta) &= \sum_{l=1}^{n_\Lambda} f_l \Lambda_l(u(t)) + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t)) \Lambda_l(u(t)) \rho_n \\
&\quad + \sum_{l=1}^{n_\Lambda} \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t)) \Lambda_l(u(t)) \rho_n \\
&= \varphi_{bc}^\top(t) \theta,
\end{aligned} \tag{3.56}$$

where  $\theta \in \mathbb{R}^{n_\theta}$  is the parameter vector defined in (3.52) and  $\varphi_{bc}(t) \in \mathbb{R}^{n_\theta}$  is the bias-corrected regression vector given by

$$\begin{aligned}
\varphi_{bc}(t) &= [\Lambda_1(u(t)) \quad \dots \quad \Lambda_{n_\Lambda}(u(t)) \\
&\quad \cos(\omega_1 x_m(t)) \Lambda_1(u(t)) \rho_1 \quad \dots \quad \cos(\omega_{n_F} x_m(t)) \Lambda_{n_\Lambda}(u(t)) \rho_{n_F} \\
&\quad \sin(\omega_1 x_m(t)) \Lambda_1(u(t)) \rho_1 \quad \dots \quad \sin(\omega_{n_F} x_m(t)) \Lambda_{n_\Lambda}(u(t)) \rho_{n_F}]^\top.
\end{aligned} \tag{3.57}$$

In what follows we will show that the IV method using the proposed predictor results in a consistent estimate.

**Theorem 3.3.1.** *The IV estimate (3.48), with  $\zeta(t)$  and  $\varphi(t)$  satisfy the following conditions*

- condition (3.49) holds,
- $\zeta(t)$  is uncorrelated with  $e_x(t)$ ,
- $\varphi(t)$  is chosen equal to  $\varphi_{bc}(t)$  given by (3.57),

is consistent.

*Proof.* Due to (3.17) and (3.18), subtracting (3.56) from (3.46) results in

$$\begin{aligned}
w_{zm}(t) - \hat{w}_{zbc}(t, \theta_0) &= \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} c_{l,n}^0 \cos(\omega_n x(t)) \Lambda_l(u(t)) [1 - \cos(\omega_n e_x(t)) \rho_n] \\
&\quad + \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} d_{l,n}^0 \sin(\omega_n x(t)) \Lambda_l(u(t)) [1 - \cos(\omega_n e_x(t)) \rho_n] \\
&\quad + \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} c_{l,n}^0 \sin(\omega_n x(t)) \Lambda_l(u(t)) \sin(\omega_n e_x(t)) \rho_n \\
&\quad - \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} d_{l,n}^0 \cos(\omega_n x(t)) \Lambda_l(u(t)) \sin(\omega_n e_x(t)) \rho_n \\
&\quad + e_w(t).
\end{aligned} \tag{3.58}$$

We note that  $e_x(t)$  is uncorrelated with  $x(t)$ ,  $u(t)$  and  $\zeta(t)$ . Furthermore,  $e_x(t)$  and  $e_w(t)$  are white and zero-mean. Therefore, by substituting (3.58) into condition (3.50) and using (3.23) and Proposition 3.2.1 we have

$$\begin{aligned}
&\bar{\mathbb{E}}[\zeta(t)(w_{zm}(t) - \hat{w}_{zbc}(t, \theta_0))] \\
&= \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} c_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \cos(\omega_n x(t)) \Lambda_l(u(t))] (1 - \phi_{e_x}(\omega_n) \rho_n)) \\
&\quad + \sum_{l=1}^{n_A} \sum_{n=1}^{n_F} d_{l,n}^0 (\bar{\mathbb{E}}[\zeta(t) \sin(\omega_n x(t)) \Lambda_l(u(t))] (1 - \phi_{e_x}(\omega_n) \rho_n)) \\
&= 0.
\end{aligned} \tag{3.59}$$

Therefore, condition (3.50) is satisfied. Given that condition (3.49) is also satisfied, the IV estimate using the bias-corrected predictor (3.56) is consistent.  $\square$

### 3.3.4 Algorithm

The main steps of the IV algorithm for identification of CLMs in the non-driving directions are summarized below.

**Algorithm 3.3.2** (IV algorithm for non-driving directions).

1. Perform experiment and collect  $N$  samples of input-output measurement data  $\{u(t), w_{zm}(t)\}_{t=1}^N$ .
2. Construct the regression vector  $\varphi(t)$  using (3.53) (NARX) or (3.57) (bias-corrected).
3. Perform simulation with the noise-free model, using the same signals  $r_1(t)$  and  $r_2(t)$  as used in the experiment, then construct the instrumental vector  $\zeta(t)$  as the noise-free version of the regression vector  $\varphi(t)$ .
4. Calculate the IV estimate using (3.48).

### 3.3.5 Simulation results

In this section, a numerical example is presented to verify the performance of the proposed IV method in the non-driving directions. Similar to Section 3.2.5, we assume that the CLM has three-phase coils A, B, C as shown in Fig. 1.2. We actually only have two control inputs  $u_A$  and  $u_B$  since the sum of the three currents is zero due to the star connection.

The main force components in the  $z$ -direction are the Lorentz force and the reluctance force. It follows that

$$\Lambda(u) = \begin{bmatrix} \Lambda^{\text{Lor}}(u) \\ \Lambda^{\text{rel}}(u) \end{bmatrix} = [u_A \quad u_B \quad u_A^2 \quad u_B^2 \quad u_A u_B]^\top. \quad (3.60)$$

The force function  $\Phi_z(x)$  also consists of the Lorentz and the reluctance parts

$$\Phi_z(x) = [\Phi_z^{\text{Lor}}(x) \quad \Phi_z^{\text{rel}}]. \quad (3.61)$$

Similar to Section 3.2.5, we assume that the Lorentz force functions  $\Phi_{zA}(x)$  and  $\Phi_{zB}(x)$  contain the first order harmonic component  $\omega_1 = \frac{\pi}{\tau_p}$  and a higher-order harmonic component  $\omega_2 = 2\omega_1$ , where  $\tau_p = 0.04\text{m}$  is the magnet pole pitch. As a result,  $\Phi_{z\text{Lor}}(x)$  can be written as

$$\Phi_z^{\text{Lor}}(x) = [\Phi_{zA}(x) \quad \Phi_{zB}(x)] = \begin{bmatrix} \sum_{n=1}^2 (c_{A,n} \cos(\omega_n x) + d_{A,n} \sin(\omega_n x)) \\ \sum_{n=1}^2 (c_{B,n} \cos(\omega_n x) + d_{B,n} \sin(\omega_n x)) \end{bmatrix}^\top. \quad (3.62)$$

Here,  $\Phi_{zA}(x), \Phi_{zB}(x) \in \mathbb{R}$  and  $\Phi_z^{\text{Lor}}(x) \in \mathbb{R}^{1 \times 2}$ . As analyzed in Chapter 2, the reluctance forces are not dependent on the position of the translator in the driving direction. As a result, the reluctance force function can be represented as follows

$$\Phi_z^{\text{rel}} = [\Phi_{zAA} \quad \Phi_{zBB} \quad \Phi_{zAB}] = [f_{AA} \quad f_{BB} \quad f_{AB}]. \quad (3.63)$$

Therefore, the total force produced by the motor in the  $z$ -direction can be written as

$$w_z = \begin{bmatrix} \sum_{n=1}^2 (c_{A,n} \cos(\omega_n x) + d_{A,n} \sin(\omega_n x)) \\ \sum_{n=1}^2 (c_{B,n} \cos(\omega_n x) + d_{B,n} \sin(\omega_n x)) \\ f_{AA} \\ f_{BB} \\ f_{AB} \end{bmatrix}^\top \begin{bmatrix} u_A \\ u_B \\ u_A^2 \\ u_B^2 \\ u_A u_B \end{bmatrix}. \quad (3.64)$$

The simulation settings are similar to Section 3.2.5. The sampling frequency of the system is  $F_s = 10$  kHz. The reference signal  $r_2(t)$  is generated as a consecutive sequence of random third-order motion profiles in the range  $[0\text{m}, 0.08\text{m}]$ . The input excitation signal is chosen as  $r_1(t) = \sum_{n=1}^{100} p_n \sin(t + \psi_n)$ , where  $p_n$  and  $\psi_n$  are random amplitudes and phase shifts. The force output measurement noise is a zero-mean white Gaussian noise with standard deviation  $\sigma_{e_w} = 0.01$  N. The position output

Table 3.2: Mean and standard deviation of 100 estimated models

Parameter	True value	LS	NARX IV	Bias-corrected IV
$c_{A,1}$	0.8660	$0.6796 \pm 0.0441$	$1.1763 \pm 0.0246$	$0.8641 \pm 0.0181$
$c_{A,2}$	-0.4100	$-0.1242 \pm 0.0378$	$-1.4055 \pm 0.0959$	$-0.4093 \pm 0.0279$
$c_{B,1}$	0.1250	$0.1903 \pm 0.0598$	$0.1726 \pm 0.0299$	$0.1268 \pm 0.0220$
$c_{B,2}$	0.3050	$0.0648 \pm 0.0392$	$1.0454 \pm 0.0784$	$0.3044 \pm 0.0228$
$d_{A,1}$	0.4330	$0.4476 \pm 0.0533$	$0.5903 \pm 0.0249$	$0.4337 \pm 0.0183$
$d_{A,2}$	0.4150	$0.0917 \pm 0.0334$	$1.4239 \pm 0.0838$	$0.4147 \pm 0.0244$
$d_{B,1}$	0.7500	$0.5253 \pm 0.0498$	$1.0214 \pm 0.0324$	$0.7504 \pm 0.0238$
$d_{B,2}$	-0.2600	$-0.0919 \pm 0.0340$	$-0.8916 \pm 0.0813$	$-0.2596 \pm 0.0237$
$f_{AA}$	0.0570	$0.0577 \pm 0.0024$	$0.0580 \pm 0.0056$	$0.0580 \pm 0.0056$
$f_{BB}$	0.0570	$0.0560 \pm 0.0028$	$0.0562 \pm 0.0065$	$0.0562 \pm 0.0065$
$f_{AB}$	0.0570	$0.0574 \pm 0.0039$	$0.0568 \pm 0.0091$	$0.0568 \pm 0.0091$

measurement noise is a zero-mean white Gaussian noise with standard deviation  $\sigma_{e_x} = 0.01$  m. The signal to noise ratio is 27 dB.

The exact knowledge of the noise characteristic is used for the bias-corrected IV estimation. It should be noted that here we use higher measurement noise level compared to Section 3.2.5, in order to demonstrate the effectiveness of the bias correction scheme. It will be shown that with this high noise variance, the bias of the NARX IV estimate is clearly noticeable, while the bias-corrected IV method is able to eliminate the bias.

The system parameters are estimated from closed-loop data of length  $N = 100000$ . A Monte-Carlo simulation of 100 runs is performed. The LS, the NARX IV and the bias-corrected IV methods are tested. The results are summarized in Table 3.2. It is observed that the LS method and the NARX IV method result in biased estimates. On the other hand, the bias-corrected IV method provides a consistent estimate. This result demonstrates the effectiveness of the new bias-corrected IV method.

Fig. 3.10 shows the forces in the  $z$ -direction produced by the true system model and the estimated models when three-phase sinusoidal current waveforms are applied to the inputs:

$$u_A = u_p \cos\left(\omega_1 x + \frac{2\pi}{3}\right), \quad u_B = u_p \cos(\omega_1 x), \quad (3.65)$$

where  $u_p = 6.4$  A is the amplitude of the current. Compared to the true system, the maximum force error of the bias-corrected IV estimated model is about 0.17%, while the maximum force errors of the LS estimated model and the NARX IV estimated model are much higher, about 7.48% and 21.95% respectively. The forces in the  $z$ -direction produced by the 100 estimated models obtained using the bias-corrected IV method are plotted in Figure 3.11.



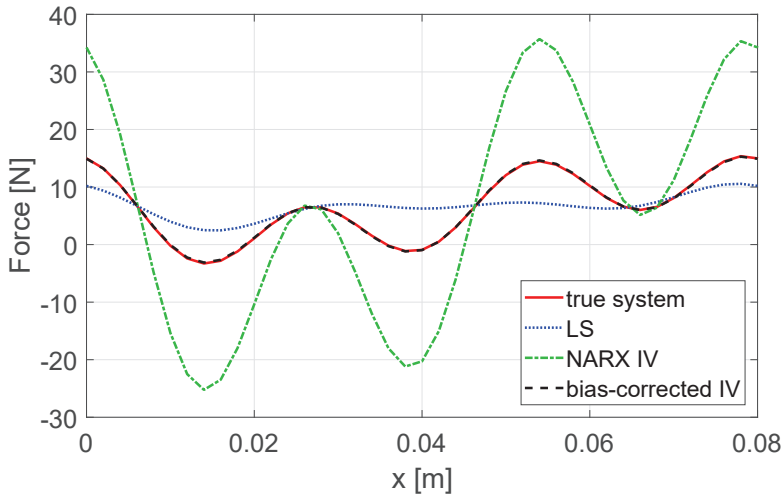


Figure 3.10: Forces in the  $z$ -direction produced by the true system and the estimated models.

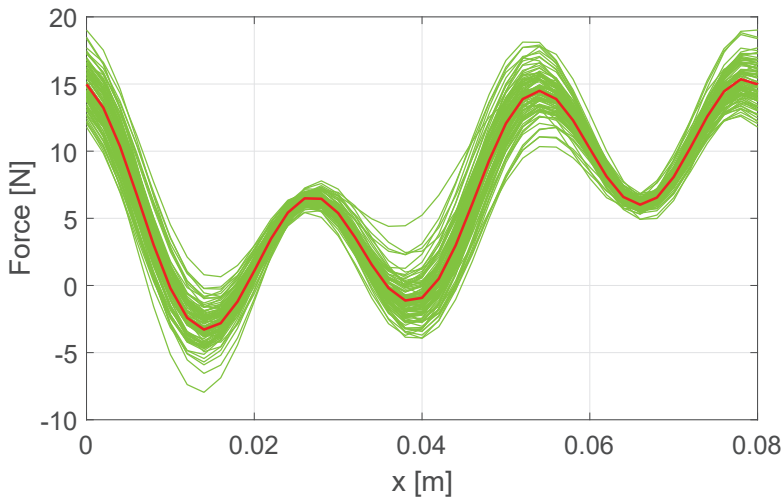


Figure 3.11: Forces in the  $z$ -direction produced by the true system (red) and the 100 estimated models obtained using the bias-corrected IV method (green).

## 3.4 Conclusions

In this chapter we have discussed the closed-loop identification problem of CLMs, both in the driving and non-driving directions. There are two main challenges in this identification problem. Firstly, due to the closed-loop data, the input is correlated with the measurement noise, causing the simple LS estimate to be biased. Secondly, the static nonlinearity of the CLM model is nonlinearly dependent on the unknown

noise-free position output, which makes it difficult to find an appropriate predictor for identification.

To address the problem caused by closed-loop data, we employ the IV identification framework. Furthermore, we have proved that the IV framework using the simple NARX predictor results in a biased estimate, due to the nonlinear dependency of the model on the unknown noise-free position output. However, the bias has been shown to be small if the position measurement noise is small compared to the magnet pole pitch. When this does not hold, we have developed a bias-corrected IV method which has been proved to deliver a consistent estimate. This method requires knowledge of the probability distribution of the measurement noise. The effectiveness of the new bias-corrected IV method has been demonstrated in numerical examples. The resulting identified model will be used for commutation and control in Chapters 4 and 5.

It is observed in simulation that the IV method for the driving direction in Section 3.2, which aims to estimate the parameters of the static nonlinearity and the linear dynamical system simultaneously, has low statistical efficiency, due to the large difference in the values of the parameters in the parameter vector. Consequently, the estimate is very sensitive to noise and disturbance. This problem causes difficulty for identification of a real CLM as will be discussed further in Chapter 7.

# Chapter 4

## Optimal commutation with nonlinear constraints

### 4.1 Introduction

A linear motor controller typically consists of a linear position controller and a commutation algorithm. This chapter focuses on the commutation problem. The purpose of a commutation algorithm is to invert the static nonlinearity between currents and forces in a linear motor, thereby removing it from the control problem. More specifically, the commutation algorithm calculates the required input currents in the coils such that the desired forces and torques are obtained.

As commutation is the inverse of the static nonlinear model, the performance of a commutation algorithm is determined by the accuracy of the model. Classical commutation is derived from a simplified model of linear motors. The force function of each coil is modeled as a pure sine wave as function of position. Consequently, the resulting input currents are three-phase sinusoidal waveforms [45]. This approach does not take into account the parasitic effects in the electromagnetic nonlinearity. Therefore, the nonlinearity is not perfectly inverted, which results in force ripples and hence reduces the tracking performance of the position control loop. Moreover, the non-driving directions are neglected, resulting in parasitic forces and torques in

---

Parts of the content of this chapter have been published in:

- T. T. Nguyen, M. Lazar, H. Butler, “A computationally efficient commutation algorithm for parasitic forces and torques compensation in ironless linear motors,” in P. Hubbard (Ed.), 7th IFAC Symposium on Mechatronic Systems, 2016.
- T. T. Nguyen, H. Butler and M. Lazar, “An analytical commutation law for parasitic forces and torques compensation in coreless linear motors,” in IEEE European Control Conference (ECC), Aalborg, 2016.
- T. T. Nguyen, M. Lazar and H. Butler, “Cancellation of normal parasitic forces in coreless linear motors,” in IEEE 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, 2015.

these directions.

To address this problem, a more accurate model of the static nonlinearity is needed. An accurate model can be obtained from first-principle modeling as discussed in Chapter 2, or data-driven modeling as discussed in Chapter 3. The commutation problem can then be formulated as finding the exact inverse of the accurate nonlinearity model. The inverse of the nonlinearity model can be found by solving a set of equations. When a linear motor is over-actuated, the set of equations is underdetermined and thus has an infinite number of solutions. The extra degrees of freedom can be used to minimize the power losses in the coils. As a result, the commutation problem becomes an optimization problem. This optimization problem can be solved using numerical optimization [2, 3, 61, 64, 65, 87, 92, 94], which is in general computationally expensive. In the case when the relation between the force vector and the current vector at a fixed position is linear, this optimization problem can be solved analytically by eliminating equality constraints [24, 85, 86, 88, 89], by using Lagrange multipliers [8, 9, 111], or by using the minimum 2-norm generalized inverse [92, 106]. These analytical commutation methods can be applied to ideal CLMs, since the driving force is linear with the currents in the coils at a fixed position.

However, in nonideal CLMs where the coils are not exactly in the center of the air gap, the parasitic forces and torques are quadratic functions of the current vector due to the presence of reluctance forces [46]. The commutation problem becomes a quadratic optimization problem with quadratic equality constraints. In general, it is difficult to find an analytical solution and hence numerical methods are necessary for solving the commutation problem. Newton's method has been used to solve commutation problems in different types of electrical machines in [64, 81], without real-time experimental validation. Theoretically, Newton's method can also be applied to solve the commutation problem for parasitic forces compensation in CLMs. However, this method requires a high computational load and therefore it is difficult to apply it to real machines with high sampling rate. More computationally efficient methods are needed for real-time implementation of optimal commutation.

In this chapter we develop computationally efficient methods for solving the optimal commutation problem in CLMs. The simplest solution is to solve the optimal commutation problem offline and store the solutions in a look-up table for online interpolation. This approach is simple and fast, but requires a large amount of memory in order to store a smooth solution, especially for applications which require control in multiple degrees of freedom [49]. As an alternative, sub-optimal analytical solutions are developed for cases when the number of coils in the translator is low. The analytical solutions can be computed very fast and still are close to the optimal one. Furthermore, in this chapter we also develop a fast optimization algorithm which is well-suited to the commutation problem. The performance of the developed commutation methods is verified by simulation with a FEM model.

The remainder of this chapter is organized as follows. The classical approach to commutation is described in Section 4.2. In Section 4.3 we present the optimal commutation formulation and develop computationally efficient methods for solving it.

A simulation example with a FEM model is shown in Section 4.4 to demonstrate the effectiveness of the developed commutation methods. The conclusions are summarized in Section 4.5.

## 4.2 Classical commutation

In this section we discuss the classical commutation technique in three-phase  $ABC$ -frame and  $dq0$ -frame.

### 4.2.1 Three-phase frame

The classical approach to commutation using three-phase sinusoidal currents is depicted in Figure 4.2. This commutation technique is based on a simplified model of the linear motor in the driving direction and neglects the non-driving directions.

In classical commutation, it is assumed that the permanent magnet (PM) flux linked by each coil is a sinusoidal function of the position  $x$  and has a spatial frequency equal to  $\pi/\tau_p$

$$\lambda_{A,pm} = \hat{\lambda}_{pm} \sin\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right), \quad (4.1a)$$

$$\lambda_{B,pm} = \hat{\lambda}_{pm} \sin\left(\frac{\pi}{\tau_p}x\right), \quad (4.1b)$$

$$\lambda_{C,pm} = \hat{\lambda}_{pm} \sin\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right), \quad (4.1c)$$

where  $\hat{\lambda}_{pm}$  is the per-phase amplitude and  $\tau_p$  is the magnet pole pitch as depicted earlier in Figure 2.1 [79]. The driving force exerted on each coil is given by

$$w_{xA} = u_A \frac{\partial \lambda_{A,pm}}{\partial x} = u_A \frac{\pi}{\tau_p} \hat{\lambda}_{pm} \cos\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right), \quad (4.2a)$$

$$w_{xB} = u_B \frac{\partial \lambda_{B,pm}}{\partial x} = u_B \frac{\pi}{\tau_p} \hat{\lambda}_{pm} \cos\left(\frac{\pi}{\tau_p}x\right), \quad (4.2b)$$

$$w_{xC} = u_C \frac{\partial \lambda_{C,pm}}{\partial x} = u_C \frac{\pi}{\tau_p} \hat{\lambda}_{pm} \cos\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right). \quad (4.2c)$$

The balanced three-phase currents are supplied to the coils

$$u_A = u_p \sin\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3} + \theta_0\right), \quad (4.3a)$$

$$u_B = u_p \sin\left(\frac{\pi}{\tau_p}x + \theta_0\right), \quad (4.3b)$$

$$u_C = u_p \sin\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3} + \theta_0\right), \quad (4.3c)$$

where  $u_p$  is the peak current and  $\theta_0$  is the commutation angle. By substituting (4.3) into (4.2) and taking the sum of the components in (4.2), the total driving force is obtained as

$$w_x = w_{xA} + w_{xB} + w_{xC} = \underbrace{\frac{3\pi}{2\tau_p} \hat{\lambda}_{pm}}_{k_m} u_p \sin(\theta_0), \quad (4.4)$$

where  $k_m$  is the motor force constant. It is observed from the above equation that the maximum force is obtained when  $\theta_0 = \pm\pi/2$  rad. The sign of  $\theta_0$  determines the direction of the force. In order to obtain a maximum positive driving force,  $\theta_0$  is chosen equal to  $\pi/2$  rad. The current amplitude is calculated from the desired driving force  $w_x^*$  as follows

$$u_p = \frac{w_x^*}{k_m}. \quad (4.5)$$

The resulting currents supplied to the coils are

$$u_A = \frac{w_x^*}{k_m} \sin\left(\frac{\pi}{\tau_p} x + \frac{2\pi}{3} + \frac{\pi}{2}\right), \quad (4.6a)$$

$$u_B = \frac{w_x^*}{k_m} \sin\left(\frac{\pi}{\tau_p} x + \frac{\pi}{2}\right), \quad (4.6b)$$

$$u_C = \frac{w_x^*}{k_m} \sin\left(\frac{\pi}{\tau_p} x - \frac{2\pi}{3} + \frac{\pi}{2}\right). \quad (4.6c)$$

For demonstration, Figure 4.1 shows the force and torque produced by a CLM using classical commutation with  $u_p = 8$  A and  $\theta_0 = \pi/2$  rad. The parameters of the CLM are given in Table 2.2. The coil array is shifted 1 mm out of the center of the air gap in the  $z$ -direction. The force and torque are obtained by simulation using the Fourier model derived in Chapter 2.

## 4.2.2 $dq0$ frame

Assuming that the three-phase system is balanced, the analysis of a CLM can be simplified by transforming the three-phase quantities to a  $dq0$ -reference frame with Park's transformation [79]. The  $dq0$ -reference frame is fixed to the magnet array. The  $d$ -axis is aligned with the PM flux and the  $q$ -axis is perpendicular to the  $d$ -axis. The transformation from the three-phase quantities to the  $dq0$ -reference frame is given by

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = T_{dq0} \begin{bmatrix} f_A \\ f_B \\ f_C \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin\left(\frac{\pi}{\tau_p} x + \frac{2\pi}{3}\right) & \sin\left(\frac{\pi}{\tau_p} x\right) & \sin\left(\frac{\pi}{\tau_p} x - \frac{2\pi}{3}\right) \\ \cos\left(\frac{\pi}{\tau_p} x + \frac{2\pi}{3}\right) & \cos\left(\frac{\pi}{\tau_p} x\right) & \cos\left(\frac{\pi}{\tau_p} x - \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_A \\ f_B \\ f_C \end{bmatrix}. \quad (4.7)$$

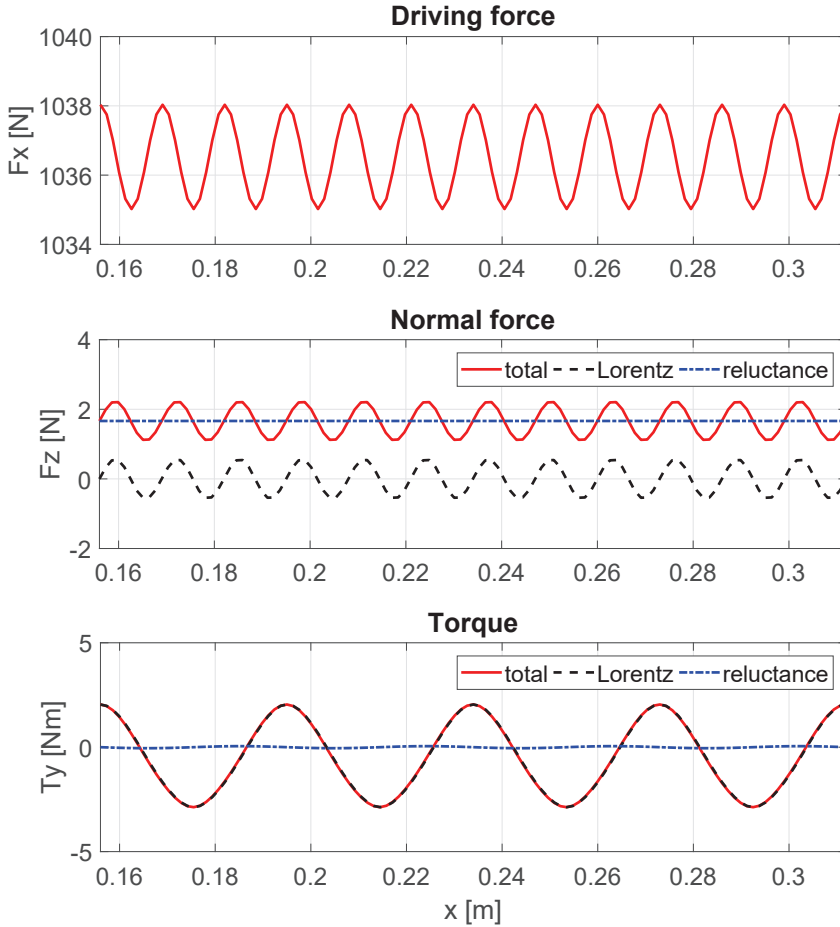


Figure 4.1: Forces and torque produced by the example CLM.

The inverse transformation is given by

$$\begin{bmatrix} f_A \\ f_B \\ f_C \end{bmatrix} = T_{dq0}^{-1} \begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = \begin{bmatrix} \sin\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right) & \cos\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right) & 1 \\ \sin\left(\frac{\pi}{\tau_p}x\right) & \cos\left(\frac{\pi}{\tau_p}x\right) & 1 \\ \sin\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right) & \cos\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right) & 1 \end{bmatrix} \begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix}. \quad (4.8)$$

The driving force in (4.4) can be rewritten as

$$w_x = \frac{\pi}{\tau_p} \hat{\lambda}_{pm} \begin{bmatrix} \cos\left(\frac{\pi}{\tau_p}x + \frac{2\pi}{3}\right) & \cos\left(\frac{\pi}{\tau_p}x\right) & \cos\left(\frac{\pi}{\tau_p}x - \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} u_A \\ u_B \\ u_C \end{bmatrix}. \quad (4.9)$$

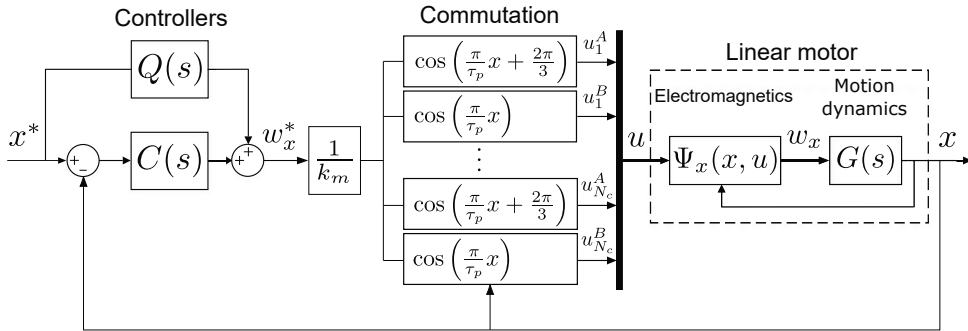


Figure 4.2: Classical commutation scheme.

Applying the transformation (4.8) yields

$$w_x = \frac{3\pi}{2\tau_p} \hat{\lambda}_{pm} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_d \\ u_q \\ u_0 \end{bmatrix} = \frac{3\pi}{2\tau_p} \hat{\lambda}_{pm} u_q = k_m u_q. \quad (4.10)$$

This equation shows that the driving force can be controlled via the  $q$ -axis current as they are directly proportional. By following similar calculation, the force in the  $z$ -direction is given by

$$w_z = \frac{3}{2} \frac{\partial \lambda_{pm}}{\partial z} u_d. \quad (4.11)$$

It is therefore possible to control the force in the  $z$ -direction via the  $d$ -axis current. It is noted that when the motor is ideal and the coils are exactly in the center of the air gap then we have  $\frac{\partial \lambda_{pm}}{\partial z} = 0$ . As a result, force production in the  $z$ -direction is not possible.

In summary, the classical commutation method is based on a simplified model of the CLM which does not take into account the higher-order harmonics and the parasitic effects. Consequently, the classical commutation method does not provide the exact desired driving force. Furthermore, classical commutation generally neglects the force and torque in the non-driving directions. Although the normal force  $w_z$  can be controlled via the  $d$ -axis current, it should be noted that the  $dq0$ -transformation is also based on a simplified model of the CLM and is only correct for balanced three-phase systems, which is usually not the case for nonideal CLMs. Therefore, classical commutation generally is unable to attenuate the parasitic force and torque in the non-driving directions.

### 4.3 Optimal commutation

In this section, we formulate the optimal commutation problem which is capable of eliminating parasitic forces and torques in the non-driving directions. Furthermore,



computationally efficient methods are developed for solving the formulated optimal commutation problem. As discussed in Chapter 2, we only consider the forces in the  $x$ - and  $z$ -directions and torque about the  $y$ -axis, as the forces and torques in other directions are negligible [46].

### 4.3.1 Optimal commutation problem formulation

The purpose of the commutation algorithm is to calculate the required input currents  $u$  such that the desired forces and torque are obtained. Mathematically, the commutation algorithm has to find a current vector  $u \in \mathbb{R}^{n_u}$  that satisfies

$$\Psi(x, u) = w^*. \quad (4.12)$$

Here,  $w^*$  is the desired wrench vector

$$w^* = \begin{bmatrix} F_x^* & F_z^* & T_y^* \end{bmatrix}^\top, \quad (4.13)$$

where  $F_x^*$ ,  $F_z^*$  and  $T_y^*$  denote the desired forces and torque. In the context of CLMs, it is desired to have  $F_z^* = 0$  and  $T_y^* = 0$  since they are acting in non-driving directions.

When a CLM is over-actuated, equation (4.12) can have an infinite number of solutions. As a result, we have the freedom to choose the solution which is beneficial for the application. An attractive solution is the one that minimizes the dissipated power in the coils. The commutation problem can be reformulated as

**Problem 4.3.1** (Optimal commutation).

$$\begin{aligned} & \min_u \quad u^\top u \\ & \text{subject to} \quad \Psi(x, u) = w^*. \end{aligned}$$

In an ideal CLM, the main force component is the Lorentz force which is linear in  $u$ . As a result, the static nonlinearity part can be written as

$$w = \Psi(x, u) = K(x)u, \quad (4.14)$$

where  $K(x) \in \mathbb{R}^{3 \times n_u}$  is a position-dependent matrix. In this case, the optimal commutation problem 4.3.1 has a closed-form solution

$$u = K(x)^\dagger w^*, \quad (4.15)$$

where  $K(x)^\dagger$  denotes the minimum 2-norm generalized inverse, or the right inverse, of  $K(x)$  [84]

$$K(x)^\dagger = K(x)^\top (K(x)K(x)^\top)^{-1}. \quad (4.16)$$

In the non-ideal case when the translator is not exactly in the center of the air gap, there are reluctance forces in the non-driving directions. As a result, the static

nonlinearity part can be written as

$$w = \Psi(x, u) = \begin{bmatrix} K_x(x)u \\ K_z(x)u + u^\top G_z u \\ K_y(x)u + u^\top G_y u \end{bmatrix}, \quad (4.17)$$

where  $K_x(x)$ ,  $K_z(x)$ ,  $K_y(x) \in \mathbb{R}^{3 \times 1}$  are position-dependent matrices, and  $G_z$ ,  $G_y \in \mathbb{R}^{n_u \times n_u}$  are constant matrices, since the reluctance force is not dependent on the  $x$ -position as analyzed in Chapter 2. It can be seen that the optimal commutation problem 4.3.1 has nonlinear equality constraints in this case. As a result, the problem does not have a closed-form solution and generally a numerical optimization method is required to search for the solution.

However, numerical optimization methods require high computational demand and therefore are difficult to apply to real-time setups. In what follows we will present alternative methods which are computationally efficient for solving the optimal commutation problem.

### 4.3.2 Look-up table solution

A look-up table is a very simple but efficient solution. In this approach, the ranges of  $x$  and  $w^*$ , from the minimum to the maximum values, are gridded. The optimal commutation problem 4.3.1 is solved offline at the grid points. The resulting solutions are stored in a look-up table. The online solution can then be obtained by interpolation between the stored values. An example of a look-up table is shown in Table 4.1.

Table 4.1: A one-dimensional look-up table.

	$F_x^*[1]$	$F_x^*[2]$	...	$F_x^*[n_{F_x}]$
$x[1]$	$u[1, 1]$	$u[1, 2]$	...	$u[1, n_{F_x}]$
$x[2]$	$u[2, 1]$	$u[2, 2]$	...	$u[2, n_{F_x}]$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$x[n_x]$	$u[n_x, 1]$	$u[n_x, 2]$	...	$u[n_x, n_{F_x}]$

The look-up table solution is computationally efficient for online use as it only requires interpolation between pre-computed values. However, in order to have a smooth solution, a large number of grid points is required, which costs a large amount of memory [49]. In addition, solving the optimal commutation problem 4.3.1 offline for all the grid points is a time-consuming task. Furthermore, for applications which requires control in multiple degrees of freedom (DOFs), a multi-dimensional look-up table is needed, causing the number of grid points and the required amount of memory to grow exponentially. Due to these reasons, a look-up table is not an attractive solution for applications which require control in multiple DOFs.

### 4.3.3 Analytical solutions

In this section we present analytical solutions for some special cases of the commutation optimization problem where the number of coils in the translator is low, which is quite common in practice.

#### One set of three-phase coils

Let us consider the case when the motor has only one set of three-phase coils connected in star configuration. In this case, the sum of the three currents must be zero. As a result, we have only two independent control inputs  $u_A$  and  $u_B$ , which are the currents in coil  $A$  and coil  $B$ , respectively. Consequently, it is not possible to control the forces and torque in three DOFs. A possible solution is to neglect the parasitic torque and only compensate for the parasitic forces in the  $x$ - and  $z$ -directions. As a result, the commutation problem becomes a set of two equations with two variables

$$K_x(x)u = F_x^*, \quad (4.18)$$

$$K_z(x)u + u^\top G_z u = 0, \quad (4.19)$$

with  $u = [u_A \quad u_B]^\top$ . The solution of this set of equations can be calculated analytically. Let us denote

$$K_x = [K_{xA} \quad K_{xB}], \quad (4.20)$$

$$K_z = [K_{zA} \quad K_{zB}], \quad (4.21)$$

$$G_z = \begin{bmatrix} G_{zAA} & G_{zAB} \\ G_{zBA} & G_{zBB} \end{bmatrix}. \quad (4.22)$$

Here, the dependency of the matrices  $K_x$  and  $K_z$  on  $x$  is omitted for brevity. It follows from (4.18) that

$$u_B = \frac{F_x^* - K_{xA}u_A}{K_{xB}}. \quad (4.23)$$

Substituting (4.23) into (4.19) results in a quadratic equation in  $u_A$

$$a_2 u_A^2 + a_1 u_A + a_0 = 0, \quad (4.24)$$

where

$$a_2 = G_{zAA} - \frac{K_{xA}}{K_{xB}}(G_{AB} + G_{BA}) + \frac{K_{xA}^2}{K_{xB}^2}G_{BB}, \quad (4.25a)$$

$$a_1 = -2\frac{K_{xA}}{K_{xB}^2}G_{BB} + \frac{1}{K_{xB}}(G_{AB} + G_{BA})F_x^* + K_{zA} - \frac{K_{xA}}{K_{xB}}K_{zB}, \quad (4.25b)$$

$$a_0 = \frac{K_{zB}}{K_{xB}}F_x^* + \frac{1}{K_{xB}^2}G_{BB}F_x^{*2}. \quad (4.25c)$$

The solutions for  $u_A$  are:

$$u_A = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2a_0}}{2a_2}. \quad (4.26)$$

The solution with smaller value of  $u^\top u$  is then selected in order to minimize the power losses in the coils.

The presented commutation algorithm is fully analytical. The computational demand is thus low and is promising for real-time implementation.

### Two sets of three-phase coils

If the motor has two sets of three-phase coils, there are four independent control inputs  $u_{A1}$ ,  $u_{B1}$ ,  $u_{A2}$  and  $u_{B2}$ . As a result, we have enough degrees of freedom to control the forces and torque in three DOFs. The commutation problem has to find an input vector  $u$  such that

$$K_x u = F_x^*, \quad (4.27)$$

$$u^\top G_z u + K_z u = 0, \quad (4.28)$$

$$K_y u = 0, \quad (4.29)$$

with  $u = [u_{A1} \ u_{B1} \ u_{A2} \ u_{B2}]^\top$ . Here, the reluctance part in the torque around the  $y$ -axis is neglected since it is generally very small as can be seen in Section 2.6. As there are four variables and three equations, we have one extra degree of freedom. This extra degree of freedom can be used to minimize the power losses in the coils as discussed in Section 4.3.1. However, this approach leads to an optimization problem which requires numerical optimization methods. The high computational load of numerical optimization methods makes it difficult to implement them in systems with a fast sampling rate.

To avoid solving an optimization problem numerically, we develop a sub-optimal analytical solution for the commutation problem. It is observed that the commutation problem in this case has four variables and three constraints, in which one is quadratic and two are linear. An analytical solution will be possible if we add one more suitable linear constraint, making the number of variables equal to the number of constraints. The question is how to find a suitable additional linear constraint. In what follows we will present the derivation of a suitable additional constraint.

Let  $F_{x1}$  and  $F_{x2}$  be the driving forces produced by coil set 1 and coil set 2, respectively. We have

$$F_x = F_{x1} + F_{x2} = K_{x1} u_1 + K_{x2} u_2. \quad (4.30)$$

We know that the driving force  $F_x$  is a linear function of the current vector  $u$ . Instead of minimizing  $u^\top u$ , which is equal to:

$$u^\top u = u_1^\top u_1 + u_2^\top u_2, \quad (4.31)$$

we can try to minimize  $F_{x1}^2 + F_{x2}^2$ , which is equal to:

$$\begin{aligned} F_{x1}^2 + F_{x2}^2 &= u_1^\top (K_{x1}^\top K_{x1}) u_1 + u_2^\top (K_{x2}^\top K_{x2}) u_2 \\ &= u^\top Q u, \end{aligned} \quad (4.32)$$

where

$$Q = \begin{bmatrix} K_{x1}^\top K_{x1} & 0_{2 \times 2} \\ 0_{2 \times 2} & K_{x2}^\top K_{x2} \end{bmatrix} \quad (4.33)$$

is a positive definite matrix. Here,  $0_{2 \times 2}$  denotes a zero matrix of dimension  $[2 \times 2]$ . In short, instead of minimizing  $u^\top u$ , we minimize  $u^\top Q u$ , where  $Q$  is a positive definite matrix defined by (4.33).

Besides, the constraint (4.27) is equivalent to

$$K_{x1} u_1 + K_{x2} u_2 = F_x^*, \quad (4.34)$$

or

$$F_{x1} + F_{x2} = F_x^*. \quad (4.35)$$

Consequently, we can write:

$$F_{x1} = \alpha F_x^*, \quad (4.36)$$

$$F_{x2} = (1 - \alpha) F_x^*, \quad (4.37)$$

where  $0 \leq \alpha \leq 1$ . The new cost function to be minimized is:

$$J(\alpha) = (\alpha F_x^*)^2 + ((1 - \alpha) F_x^*)^2 \quad (4.38)$$

$$= (2\alpha^2 - 2\alpha + 1) F_x^{*2}. \quad (4.39)$$

Taking the derivative with respect to  $\alpha$ , we have the optimality condition:

$$\frac{\partial J(\alpha)}{\partial \alpha} = 2(2\alpha - 1) F_x^{*2} = 0, \quad (4.40)$$

which has the solution  $\alpha = 0.5$ . This means the driving forces produced by the two sets of coils are equal. Therefore, the new optimality condition can be written as a linear constraint:

$$F_{x1} = F_{x2}, \quad (4.41)$$

or equivalently

$$K_{x1} u_1 - K_{x2} u_2 = 0. \quad (4.42)$$

The commutation problem now has four equations (4.27), (4.28), (4.29) and (4.42). From the three linear equations, the current vector  $u$  can be written as a function of one current, e.g.  $u_{A1}$ :

$$u = V u_{A1} + W, \quad (4.43)$$

where  $V, W \in \mathbb{R}^{4 \times 1}$ . Substituting (4.43) into (4.28), we have a quadratic equation:

$$a_2 u_{A1}^2 + a_1 u_{A1} + a_0 = 0, \quad (4.44)$$

where

$$a_2 = V^\top G_z V, \quad (4.45)$$

$$a_1 = K_z V + V^\top G_z W + W^\top G_z V, \quad (4.46)$$

$$a_0 = K_z W + W^\top G_z W. \quad (4.47)$$

The solutions of (4.44) are

$$u_{A1} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2 a_0}}{2a_2}. \quad (4.48)$$

The current vector is then computed using (4.43). The solution with smaller value of  $u^\top Q u$  is selected out of the two possible solutions.

The proposed solution is not the optimal solution of Problem 4.3.1 since it minimizes  $u^\top Q u$  instead of  $u^\top u$ . However, it provides a good trade-off between optimality and computational efficiency, as will be demonstrated in Section 4.4.

### Three or more sets of three-phase coils

For CLMs with three or more sets of three-phase coils, it is also possible to find analytical sub-optimal solutions by adding suitable constraints such that the number of equations is equal to the number of variables. How to choose additional constraints is dependent on specific applications and will not be discussed further.

#### 4.3.4 Fast numerical optimization solution

In general, a numerical optimization algorithm is required to find the optimal solution of the commutation problem 4.3.1. In this section we develop an optimization algorithm which is computationally efficient and well-suited to the commutation problem. Let us define

$$g(u) = \Psi(x, u) - w^*, \quad (4.49)$$

where  $u \in \mathbb{R}^n$  and  $g(u) \in \mathbb{R}^m$ , with  $n$  denoting the number of variables and  $m$  denoting the number of constraints. Here, we omit the dependency of  $g$  on  $x$  since  $x$  does not change in one sampling period and thus can be considered as a constant in the optimization problem. Problem 4.3.1 is rewritten as

$$\begin{aligned} \min_u \quad & u^\top u \\ \text{subject to} \quad & g(u) = 0. \end{aligned} \quad (4.50)$$

Problem (4.50) is a quadratic optimization problem with nonlinear equality constraints. In what follows we first review the standard approach to solve this class of optimization problems using Lagrange multipliers and Newton's method. Then our new optimization algorithm is presented.

### Newton's method

A common approach to solve the optimization problem (4.50) is to solve the set of optimality conditions, which are known as the Karush–Kuhn–Tucker (KKT) conditions [53], using Newton's method. This approach has been used to solve the commutation problem in other types of actuators in [64, 81].

First, we define the Lagrange function of the optimization problem (4.50) as follows

$$\mathcal{L}(u, \lambda) = u^\top u + \lambda^\top g(u), \quad (4.51)$$

where  $\lambda \in \mathbb{R}^m$  is the vector of Lagrange multipliers. The KKT optimality conditions of Problem (4.50) are

$$\nabla_{[u, \lambda]^\top} \mathcal{L}(u, \lambda) = \mathbf{0}_{(n+m) \times 1}, \quad (4.52)$$

with

$$\nabla_{[u, \lambda]^\top} \mathcal{L}(u, \lambda) = \begin{bmatrix} 2u + \nabla_u g(u) \lambda \\ g(u) \end{bmatrix}. \quad (4.53)$$

The system of equations (4.52) is solved iteratively using Newton's method. Let  $u_k$  and  $\lambda_k$  be the estimated values of the optimal solution of  $u$  and  $\lambda$  at any iteration  $k \in \mathbb{N}$ . Then the new estimated values are computed as follows:

$$\begin{bmatrix} u_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \lambda_k \end{bmatrix} - \left( \nabla_{[u_k, \lambda_k]^\top}^2 \mathcal{L}(u_k, \lambda_k) \right)^{-1} \nabla_{[u_k, \lambda_k]^\top} \mathcal{L}(u_k, \lambda_k), \quad (4.54)$$

where  $\nabla_{[u_k, \lambda_k]^\top}^2 \mathcal{L}(u_k, \lambda_k)$  is the Hessian matrix of  $\mathcal{L}(u_k, \lambda_k)$ :

$$\nabla_{[u, \lambda]^\top}^2 \mathcal{L}(u, \lambda) = \begin{bmatrix} \nabla_u^2 \mathcal{L}(u, \lambda) & \nabla_u g(u) \\ \nabla_u g(u)^\top & \mathbf{0}_{m \times m} \end{bmatrix}. \quad (4.55)$$

The iterative process (4.54) stops when a predefined accuracy  $\epsilon$  is reached, i.e.

$$\|g(u_k)\|_2 \leq \epsilon, \quad (4.56)$$

where  $\epsilon$  is a small positive value.

This algorithm converges locally to a local minimum of Problem (4.50) and the convergence rate is quadratic. The proof of convergence of the Newton's method can be found in [52, 83]. However, this method introduces additional variables, i.e. Lagrange multipliers, which increases the complexity of the problem. In addition, the need to evaluate and store the Hessian matrix increases the computation time and the amount of memory needed.

### Proposed algorithm

In this section, we develop a new algorithm which does not introduce additional optimization variables and does not require the Hessian matrix. Instead of formulating the optimality conditions (4.52), we directly solve the set of constraints in problem (4.50). The proposed method to solve the constraints of problem (4.50) is inspired by the classical Newton's method. The original idea of the iterative Newton's algorithm is to linearize  $g(u)$  around the current estimate  $u_k$  at iteration  $k$ :

$$g(u) \approx \tilde{g}(u) := g(u_k) + \nabla g(u_k)^\top (u - u_k). \quad (4.57)$$

Then the new estimate  $u_{k+1}$  of the solution has to satisfy

$$\tilde{g}(u_{k+1}) = 0, \quad (4.58)$$

which is equivalent to

$$\nabla g(u_k)^\top u_{k+1} = \nabla g(u_k)^\top u_k - g(u_k). \quad (4.59)$$

Since the system of equations (4.58) is underdetermined, i.e.  $m < n$ , it has an infinite number of solutions. A feasible solution of (4.58) is

$$u_{k+1} = u_k - \nabla g(u_k)^\dagger g(u_k), \quad k = 0, 1, \dots, \quad (4.60)$$

where  $\nabla g(u_k)^\dagger$  is the minimum 2-norm generalized inverse, or the right inverse of  $\nabla g(u_k)^\top$  [84]:

$$\nabla g(u_k)^\dagger = \nabla g(u_k) (\nabla g(u_k)^\top \nabla g(u_k))^{-1}. \quad (4.61)$$

The update (4.60) is known as the generalized Newton's method for underdetermined systems of equations, which provides a feasible solution of (4.58). However, instead of just computing a feasible solution, we can compute the minimum 2-norm solution of (4.58), which is the solution that has smallest value of  $u_{k+1}^\top u_{k+1}$ , i.e.

$$u_{k+1} = \nabla g(u_k)^\dagger (\nabla g(u_k)^\top u_k - g(u_k)), \quad (4.62)$$

or, equivalently

$$u_{k+1} = \nabla g(u_k)^\dagger \nabla g(u_k)^\top u_k - \nabla g(u_k)^\dagger g(u_k). \quad (4.63)$$

The iteration (4.63) is identical to the generalized Gauss-Newton (GGN) method for nonlinear constrained least squares problems which was originally proposed and analyzed in [13].

The generalized Newton's method (4.60) only provides a feasible solution of the set of constraints in problem (4.50), but its local convergence has been proven in [10, 55]. On the other hand, the iteration (4.63) provides the minimum 2-norm solution of the set of constraints in problem (4.50), but it only converges if the



residual function of the nonlinear constrained least squares problem is small at the solution [21]. To solve this issue of (4.63), we use a new iteration, which is the interpolation between iteration (4.60) and iteration (4.63), i.e.

$$u_{k+1} = \alpha u_k + (1 - \alpha) \nabla g(u_k)^\dagger \nabla g(u_k)^\top u_k - \nabla g(u_k)^\dagger g(u_k), \quad (4.64)$$

where  $0 < \alpha < 1$ . It can be proved that the proposed iteration (4.64) inherits the local convergence property of (4.60) and the optimality property of (4.63). The proof will be presented in Chapter 6.

The presented algorithm has a linear convergence rate, while the convergence rate of the Newton's method is proven to be quadratic. Therefore, the proposed algorithm may need more iterations to converge. However, each proposed iteration has lower complexity than each Newton's iteration. Since the complexity of inverting a matrix in  $\mathbb{R}^u$  is  $\mathcal{O}(u^3)$ , the complexity of multiplying a matrix in  $\mathbb{R}^{u \times v}$  and a matrix in  $\mathbb{R}^{v \times w}$  is  $\mathcal{O}(uvw)$  [36], it can easily be verified that the complexity of one proposed iteration, including the computation of the generalized inverse, is  $\mathcal{O}(n^2 m)$ . This is lower than the complexity of one Newton's iteration, which is  $\mathcal{O}((n + m)^3)$ .

In summary, the proposed algorithm converges to the same optimal solution as the Newton's method (4.54), but without the need to introduce additional optimization variables and to evaluate the Hessian matrix. As a result, the proposed algorithm is computationally efficient and therefore is beneficial for the commutation problem in real-time implementation, since the sampling rate of a CLM is usually very high and the time for computation is limited.

## 4.4 Simulation results

In this section, simulation results of the developed commutation methods are presented for demonstration purpose. The simulation is performed with the FEM model shown in Figure 2.6. The FEM model is created using Cobham Opera-2D simulation software. The parameters of the FEM model are summarized in Table 2.2. The example motor has two sets of three-phase coils next to each other. Some artificial manufacturing tolerances are added to make the motor nonideal. Firstly, the coils are shifted 1 mm out of the center of the air gap in the  $z$ -direction. This displacement causes not only parasitic Lorentz forces and torques, but also reluctance forces and torques, since the motor geometry becomes asymmetric. Secondly, some magnets have higher or lower remanent magnetization as depicted in Figure 2.6. The variation of remanent magnetization causes parasitic Lorentz forces and torques in the motor.

We simulate the classical commutation algorithm described in Section 4.2, the analytical commutation algorithm in Section 4.3.3 and the fast optimization commutation algorithm in Section 4.3.4. The Fourier model obtained from Section 2.6 is used for the analytical commutation and fast optimization commutation.

For the fast optimization algorithm, the final result of the commutation algorithm at each position is used as the "warm-start" initial estimate  $u_0$  for the next position.

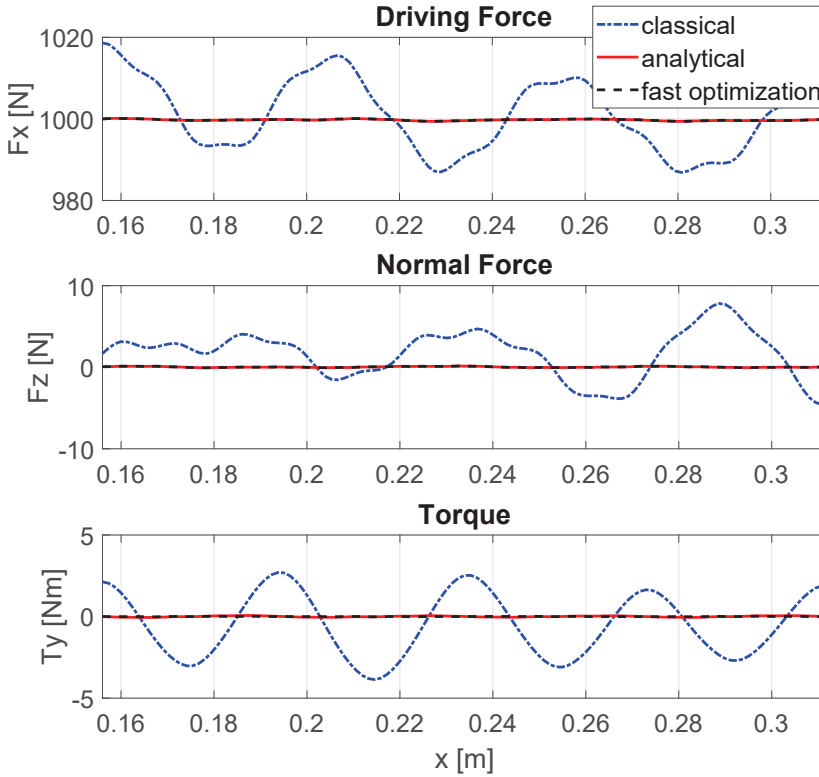


Figure 4.3: Resulting forces and torque.

The interpolation coefficient is chosen as  $\alpha = 0.01$ . The predefined accuracy is  $\epsilon = 10^{-6}$ . The Newton's method is also simulated for comparison.

The simulation is performed along the  $x$ -direction, in the range between 156 mm and 312 mm. The reference driving force is  $F_x^* = 1000$  N. The simulation results are presented and discussed below.

#### 4.4.1 Resulting forces and torque

The resulting forces and torque of the considered commutation methods are shown in Figure 4.3. The results of the classical commutation described in Section 4.2 are also plotted for comparison. The root mean square (rms) values of the forces and torque error are summarized in Table 4.2.

It is clearly seen that the classical commutation results in high ripples in the driving force and nonzero normal force and torque. On the other hand, with the new commutation methods, almost the exact reference forces and torque are obtained. The forces and torque errors are small as summarized in Table 4.2.

The results of the developed optimization algorithm are the same as those of

Table 4.2: Forces and torque error

	Classical solution	Analytical solution	Fast optimization solution
Fx error rms (N)	8.8687	0.3058	0.3020
Fz error rms (N)	3.3745	0.0655	0.0660
Ty error rms (Nm)	1.9922	0.0336	0.0079

the Newton's method and the commercial solvers Knitro [16], IPOPT [109] and Matlab's solver *fmincon*. This verifies the local optimality property of the proposed optimization algorithm.

#### 4.4.2 Dissipated power

The current waveforms of the classical solution, the analytical solution and the fast optimization solution are shown in Figure 4.4. It is observed that, in contrast to the classical commutation method which uses perfect sinusoidal current waveforms, the new commutation methods use distorted sinusoidal waveforms, in order to compensate for the imperfection of the motor.

The values of  $u^T u$ , which is proportional with the dissipated power in the coils, are plotted in Figure 4.5. The dissipated power of the proposed commutation methods is in general higher than that of the classical commutation. The fast optimization method provides the solution which can compensate for parasitic forces and torques with minimum dissipated power. The dissipated power of the analytical solution is a little higher than that of the fast optimization solution, as the analytical solution is only a sub-optimal solution which does not guarantee minimum dissipated power. However, it can be seen that the difference in dissipated power of the two methods is small.

#### 4.4.3 Computational time

The simulation is performed on a 3.6 GHz computer. The average computation time of the analytical solution is 0.16 ms. The new optimization algorithm converges after 2 or 3 iterations. The average computation time for commutation of the new optimization algorithm is 0.22 ms. It can be seen that the computation time of the proposed methods are higher than that of the classical commutation, which is about 0.09 ms. However, they are still low compared to other numerical optimization methods. Using Newton's method, it also takes 2 to 3 iterations to converge and the average computation time is 0.35 ms. The average computation times using the solvers Knitro, IPOPT and *fmincon* are much higher, about 11.4 ms, 56.3 ms and 6.5 ms respectively. It is noted that the solvers also make use of "warm-start" initial estimate.

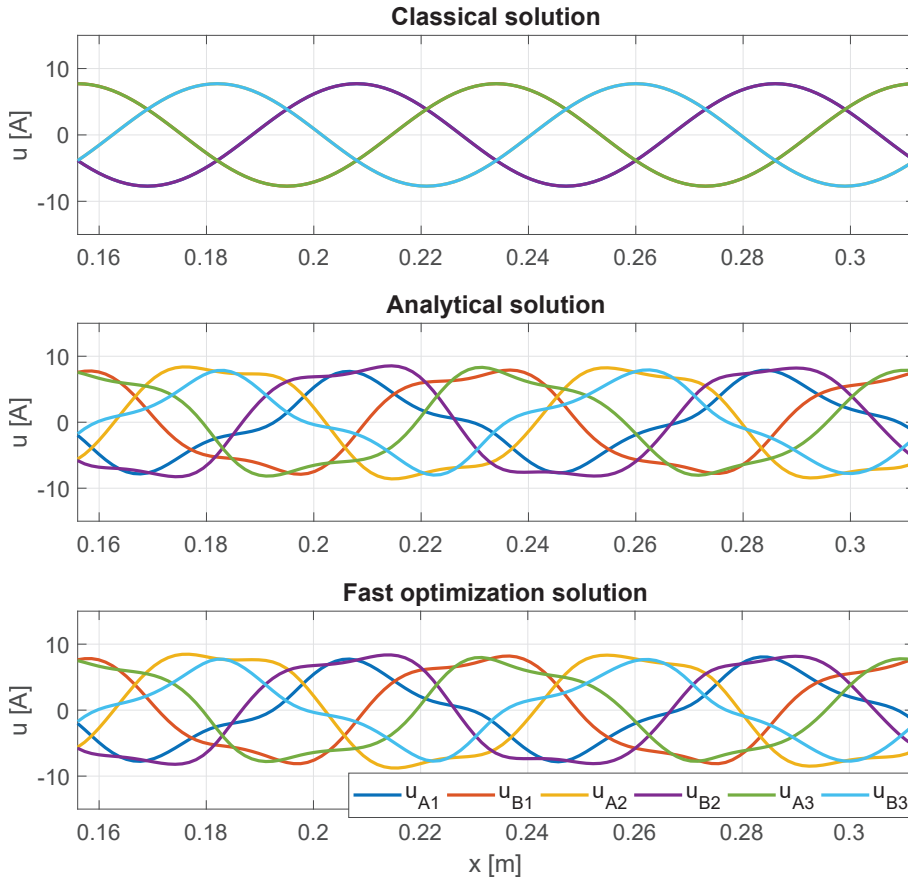


Figure 4.4: Current waveforms of the classical and new commutation methods.

It should be noted that here the developed algorithms are programmed in Matlab code, which generally run slower than C code. By converting Matlab code to C code, the algorithms run much faster and we are able to implement them on a real linear motor setup with a fast sampling rate of 10 kHz as will be shown in Chapter 7.

#### 4.4.4 Summary

In summary, compared to the classical commutation, the developed commutation methods are able to compensate for parasitic forces and torques, at the price of increased dissipated power and computation time. Compared to other numerical optimization methods and solvers such as Newton's method, Knitro solver, IPOPT solver and *fmincon* solver, the proposed fast optimization solution provides the same local optimal solution, but with lower computation time. The analytical solution is a sub-optimal solution of the commutation problem which does not guarantee mini-

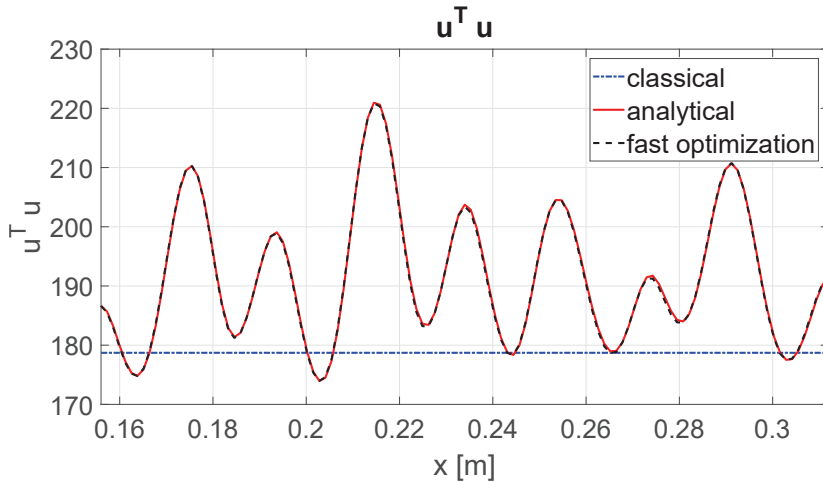


Figure 4.5: The values of  $u^T u$ .

imum power losses, but requires less computation time than numerical optimization methods. Its dissipated power is a little higher than that of the fast optimization solution.

## 4.5 Conclusions

In this chapter we have formulated the commutation problem as an optimization problem which minimizes the power losses in the coils, subject to the constraints that the desired forces and torques are obtained. Compared to classical commutation, the optimal commutation method is capable of compensating for parasitic forces and torques in driving and non-driving directions. However, solving the optimal commutation problem generally requires numerical optimization methods which are computationally expensive. To address this problem, we have developed computationally efficient methods for solving the optimal commutation problem. A look-up table is a simple but efficient method, but on the other hand requires a large amount of memory and is not suitable for applications which requires control in multiple DOFs. As an alternative, sub-optimal analytical solutions have been developed for cases when the number of coils in the translator is low. Furthermore, we also develop a new optimization algorithm which is more computationally efficient than the classical Newton's method. The effectiveness of the developed commutation methods has been verified by simulation with a FEM model.



# Chapter 5

## Model predictive control of linear motors

### 5.1 Introduction

A standard CLM control scheme typically consists of two sub-controllers: a commutation algorithm and a linear position controller. The commutation algorithm aims to invert the position-dependent static nonlinearity between forces and currents, thereby removing it from the control problem. The remaining linear motion dynamics is then controlled by a linear position controller which may include feedforward.

In practice, a classical approach is used for commutation and linear position control of linear motors. Classical commutation makes use of three-phase sinusoidal current waveforms as functions of position, which results in force ripples in both driving and non-driving directions as discussed in Chapter 4 of this thesis. The linear position controller typically consists of a proportional-integral-derivative (PID) feedback controller and a feedforward controller. This classical position controller has several drawbacks. Firstly, when there are physical constraints on the system, such as current limit and force limit, the classical control approach cannot guarantee constraint satisfaction. This will be a drawback for future high-precision positioning systems where linear motors have to operate near their physical limit to increase efficiency and throughput. Secondly, the performance of the classical controllers is mostly determined by the feedforward controller. However, the feedforward control input is calculated based on the knowledge of the model and then implemented in an open-loop manner. Consequently, model mismatches and disturbances can result in reduced performance. Thirdly, for control in multiple degrees of freedom (DOFs),

---

Part of the content of this chapter has been published in:

- T. T. Nguyen, M. Lazar and H. Butler, “Nonlinear model predictive control of ironless linear motors,” in IEEE 2nd Conference on Control Technology and Applications (CCTA), Copenhagen, 2018.

designing and tuning of classical controller can be a cumbersome task, as we have to decouple the system and then design single-output single-input (SISO) feedback and feedforward controllers for every DOF [30, 105].

There are alternative approaches which can overcome the limitations of the classical commutation and position controller. Classical commutation can be replaced by optimal commutation (OC) as discussed in Chapter 4. The OC is formulated as an optimization problem which calculates the currents in the coils such that the desired forces are obtained and the power loss is minimized [64, 92]. The OC is capable of eliminating force ripples in driving and non-driving directions as demonstrated in Chapter 4. The classical PID and feedforward approach can be replaced by linear model predictive control (LMPC), which is known for its high performance and ability to guarantee constraint satisfaction [18, 26, 43, 110].

Replacing both the classical position controller and classical commutation, we obtain a cascaded control scheme which consists of a LMPC controller and an OC algorithm. This scheme can lead to excellent performance of linear motors as demonstrated in our previous work [68] and also in this chapter. However, it requires solving two separate optimization problems, one for LMPC and one for OC. This is not the optimal setting, since the combination of the optimal solutions of the two sub-controllers may not yield the optimal solution for the whole controller. Furthermore, as the LMPC problem cannot take into account the limit of the currents but only the limit of the forces, it can give a force setpoint that is unachievable, causing the OC problem to be infeasible.

To overcome this limitation, our goal is to replace the traditional cascaded two-level control scheme with a single nonlinear position controller which receives position measurements and directly calculates the necessary currents in the coils. Model predictive control (MPC) is an excellent tool for this purpose thanks to its ability to deliver optimal performance while guaranteeing constraint satisfaction.

In this chapter we present two MPC schemes for linear motors. The first one is the Combined LMPC and OC scheme, in which the LMPC problem and the OC problem are combined into a single optimization problem. In this scheme only the linear dynamics is used for prediction. There is no prediction for the static nonlinearity part. The second scheme is the nonlinear model predictive controller (NMPC) scheme, where the full nonlinear model is used for prediction. In the two presented MPC schemes, we only need to solve a single optimization problem every sampling time. Furthermore, the proposed schemes can guarantee constraint satisfaction and are capable of compensating for parasitic forces and torques in driving and non-driving directions. The effectiveness of the proposed MPC schemes is demonstrated in a simulation example.

The remainder of this chapter is organized as follows. Section 5.2 discusses the Cascaded LMPC and optimal commutation scheme and present a Combined LMPC and optimal commutation scheme. The NMPC scheme for linear motors is proposed in Section 5.3. The simulation results are presented in Section 5.4. The conclusions are summarized in Section 5.5.



## 5.2 Linear model predictive control and optimal commutation

In order to optimize the performance and to guarantee constraint satisfaction, we replace the classical position controller and classical commutation with LMPC and OC (LMPC+OC). Compared to the classical controllers, LMPC can guarantee constraint satisfaction and OC can eliminate force ripples due to parasitic effects. In this section we discuss the Cascaded LMPC+OC scheme and its limitation. The Combined LMPC+OC scheme is then introduced to overcome the limitation.

### 5.2.1 Cascaded LMPC and optimal commutation

The Cascaded LMPC+OC scheme is obtained simply by replacing the classical position controller and classical commutation with a LMPC controller and an OC algorithm. This control scheme was first presented in our previous work [68]. The Cascaded LMPC+OC scheme is shown in Figure 5.1.

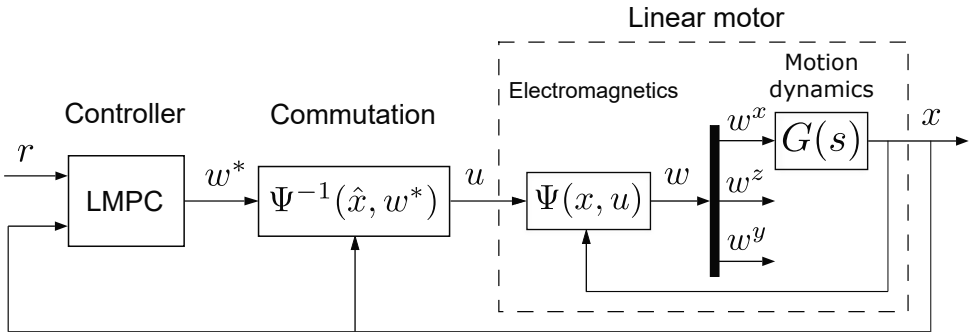


Figure 5.1: Cascaded LMPC+OC scheme.

In this section we present the formulation of the LMPC and the OC problems. First, let us define the state of the linear system as

$$\mathbf{x} = \begin{bmatrix} x \\ v \end{bmatrix}, \quad (5.1)$$

where  $v$  is the speed of the motor in the driving direction  $x$ . The linear motion dynamics in the driving direction  $x$  can be written in state-space form as follows

$$\mathbf{x}^+ = A\mathbf{x} + Bw^x, \quad (5.2)$$

where  $\mathbf{x} \in \mathbb{R}^2$ ,  $w^x \in \mathbb{R}$ ,  $A \in \mathbb{R}^{2 \times 2}$  and  $B \in \mathbb{R}^2$ . Here,  $\mathbf{x}^+$  denotes the state at the next time sample.

Let  $r(t)$  denotes the reference trajectory that the output position  $x(t)$  has to track. It is assumed that  $r(t)$  is given in advance. In what follows we present the offline

calculation of the reference state, reference force and reference input currents which are necessary for the LMPC and OC problem formulation. The reference state  $\mathbf{x}^r$  is defined as

$$\mathbf{x}^r = \begin{bmatrix} r \\ v^r \end{bmatrix}, \quad (5.3)$$

where  $v^r$  is the reference velocity. Given a reference trajectory  $r(t)$ , the reference velocity  $v^r(t)$  can be calculated offline noting that  $v^r$  is the derivative of  $r$ . This relation can be written in discrete-time transfer function as follows

$$r(t+1) = \frac{T_s q^{-1}}{1 - q^{-1}} v^r(t+1), \quad (5.4)$$

where  $T_s$  is the sampling time and  $q^{-1}$  is the delay operator with  $q^{-i} r(t) = r(t-i)$ . It follows that

$$v^r(t) = \frac{1}{T_s} (r(t+1) - r(t)). \quad (5.5)$$

The reference force  $w^{xr}$  is defined as

$$w^{xr} = m a^r, \quad (5.6)$$

where  $m$  is the mass of the translator and  $a^r$  is the reference acceleration, which can be calculated offline in the same way that we calculate  $v^r$

$$a^r(t) = \frac{1}{T_s} (v^r(t+1) - v^r(t)). \quad (5.7)$$

From the reference driving force  $w^{xr}(t)$ , we can calculate the reference input current  $u^r(t)$  as the solution of the offline optimal commutation problem

$$\begin{aligned} & \min_{u^r(t)} u^r(t)^\top u^r(t) \\ & \text{subject to } \Psi_x(r(t), u^r(t)) = w^{rx}(t), \\ & \Psi_z(r(t), u^r(t)) = 0, \\ & \Psi_y(r(t), u^r(t)) = 0, \\ & u_{\min} \leq u^r(t) \leq u_{\max}. \end{aligned} \quad (5.8)$$

Here,  $\Psi_x(x, u)$ ,  $\Psi_z(x, u)$  and  $\Psi_y(x, u)$  represent the relation between the current vector  $u$  and the driving force  $w^x$ , the normal force  $w^z$  and the torque  $w^y$  around the  $y$ -axis at a position  $x$ , respectively. It should be noted that in the context of CLMs, the reference force in the  $z$ -direction and the desired torque around the  $y$ -axis are always zero. The offline optimal commutation problem (5.8) searches for a current vector  $u^r(t)$  which satisfies the current limit such that the reference wrench is obtained and the power loss is minimized.

In the Cascaded LMPC+OC scheme, the LMPC problem and the OC problem are formulated and solved independently. The LMPC problem is formulated as follows

**Problem 5.2.1** (LMPC).

$$\begin{aligned} \min_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_N, \\ w_0^x, \dots, w_{N-1}^x}} \quad & (\mathbf{x}_N - \mathbf{x}_N^r)^\top P (\mathbf{x}_N - \mathbf{x}_N^r) + \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{x}_i^r)^\top Q (\mathbf{x}_i - \mathbf{x}_i^r) \\ & + \sum_{i=0}^{N-1} (w_i^x - w_i^{xr})^\top R_w (w_i^x - w_i^{xr}) \end{aligned}$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + Bw_i^x, & \forall i = 0, \dots, N-1, \\ \mathbf{x}_{\min} &\leq \mathbf{x}_i \leq \mathbf{x}_{\max}, & \forall i = 1, \dots, N, \\ w_{\min}^x &\leq w_i^x \leq w_{\max}^x, & \forall i = 0, \dots, N-1. \end{aligned}$$

Here,  $N \in \mathbb{N}$  is the prediction horizon,  $Q \in \mathbb{R}^{2N \times 2N}$  is the state cost,  $P \in \mathbb{R}^{2N \times 2N}$  is the terminal cost and  $R_w \in \mathbb{R}^{N \times N}$  is the control cost. Notice that the reference force  $w^{xr}$  as calculated in (5.6) is actually the classical feedforward force. In the LMPC problem we minimize the difference between the total force and the feedforward force since the feedforward force results in zero tracking error given that the system model is known exactly and there is no disturbance in the system.

Solving the LMPC problem 5.2.1, we get the optimal sequence of driving force  $\{w_0^x, \dots, w_{N-1}^x\}$ . The first element  $w_0^x$  of the sequence is used as the input for the commutation block. The optimal commutation problem is formulated as follows

**Problem 5.2.2** (Optimal commutation).

$$\min_{u_0} (u_0 - u_0^r)^\top (u_0 - u_0^r)$$

subject to

$$\begin{aligned} \Psi_x(x_0, u_0) &= w_0^x, \\ \Psi_z(x_0, u_0) &= 0, \\ \Psi_y(x_0, u_0) &= 0, \\ u_{\min} &\leq u_0 \leq u_{\max}. \end{aligned}$$

The optimal commutation problem 5.2.2 searches for a current vector  $u_0$  which satisfies the current limit such that the desired forces and torques in all directions are obtained. Furthermore, we minimize the difference between the input current  $u$  and the reference input current  $u^r$ , since  $u^r$  as calculated in problem (5.8) is the input with minimum power loss that produces the reference feedforward force  $w^{xr}$ , which results in zero tracking error given that the system model is known exactly and there is no disturbance in the system. The resulting solution of the OC problem 5.2.2 is then applied to the linear motor.

This control scheme requires solving two optimization problems, corresponding to two sub-controllers. This is not the optimal setting, as the LMPC problem is not aware of the physical limit of the current  $u_0$ . Consequently, it may be infeasible

for the commutation problem to obtain the force setpoint  $w_0^x$  given by the LMPC solution. Furthermore, this scheme requires solving two optimization problems separately. The number of optimization variables and constraints of each problem is summarized in Table 5.1.

Table 5.1: Number of variables and constraints of the Cascaded LMPC+OC scheme.

	LMPC	OC	Total
Number of variables	$(n_x + n_w)N$	$n_u$	$(n_x + n_w)N + n_u$
Number of equality constraints	$n_x N$	3	$n_x N + 3$
Number of inequality constraints	$(n_x + n_w)N$	$n_u$	$(n_x + n_w)N + n_u$

## 5.2.2 Combined LMPC and optimal commutation

To avoid the drawbacks of solving two separate optimization problems, we introduce the Combined LMPC+OC scheme which combines the LMPC problem 5.2.1 and the optimal commutation problem 5.2.2 into a single optimization problem as shown in Figure 5.2. The resulting optimization problem is formulated as follows

**Problem 5.2.3** (Combined LMPC+OC).

$$\begin{aligned} \min_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_N, \\ w_1, \dots, w_{N-1}, \\ u_0}} \quad & (\mathbf{x}_N - \mathbf{x}_N^*)^\top P (\mathbf{x}_N - \mathbf{x}_N^*) + \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{x}_i^*)^\top Q (\mathbf{x}_i - \mathbf{x}_i^*) \\ & + \sum_{i=1}^{N-1} (w_i^x - w_i^{xr})^\top R_w (w_i^x - w_i^{xr}) + (u_0 - u_0^r)^\top (u_0 - u_0^r) \end{aligned}$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + B w_i^x, & \forall i = 0, \dots, N-1, \\ \Psi_x(x_0, u_0) &= w_0^x, \\ \Psi_z(x_0, u_0) &= 0, \\ \Psi_y(x_0, u_0) &= 0, \\ \mathbf{x}_{\min} &\leq \mathbf{x}_i \leq \mathbf{x}_{\max}, & \forall i = 1, \dots, N, \\ w_{\min}^x &\leq w_i^x \leq w_{\max}^x, & \forall i = 0, \dots, N-1, \\ u_{\min} &\leq u_0 \leq u_{\max}. \end{aligned}$$

The resulting solution  $u_0$  is then applied to the linear motor. Compared to the Cascaded LMPC+OC scheme, the Combined LMPC+OC scheme only requires solving a single optimization problem. Furthermore, the scheme is able to guarantee constraint satisfaction and compensate for parasitic forces.

It is observed that in this scheme only the linear dynamics is used for prediction. There is no prediction for the static nonlinearity part. Consequently, the future

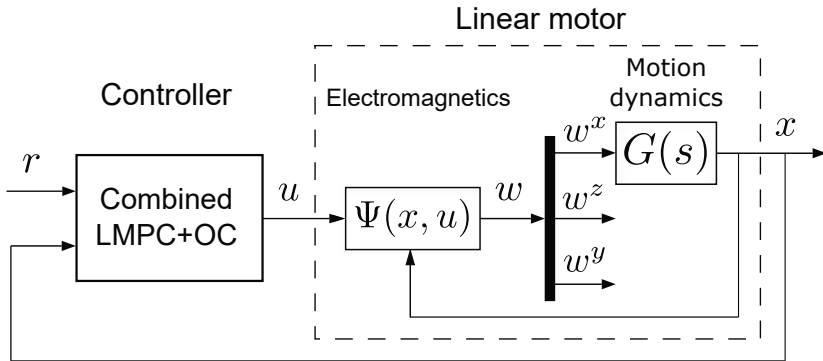


Figure 5.2: Combined LMPC+OC scheme.

error caused by the static nonlinearity is neglected, which may result in reduced performance. This limitation can be overcome by using the full nonlinear model for prediction as proposed in the next section.

### 5.3 Nonlinear model predictive control

In this section the NMPC formulation for linear motor is presented. The NMPC scheme is shown in Figure 5.3. This scheme also combines the MPC controller and optimal commutation into a single optimization problem. But compared to the Combined LMPC+OC scheme, the NMPC scheme uses the full nonlinear model of the linear motor for prediction. The resulting NMPC optimization problem is formulated as follows

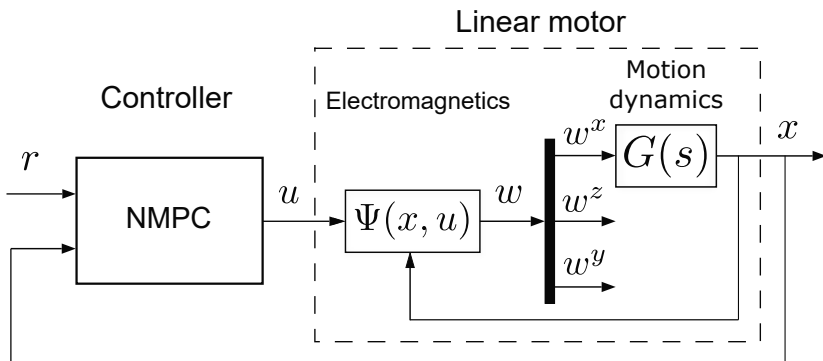


Figure 5.3: NMPC scheme.

**Problem 5.3.1** (NMPC).

$$\min_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_N, \\ u_0, u_1, \dots, u_{N-1}}} (\mathbf{x}_N - \mathbf{x}_N^*)^\top P (\mathbf{x}_N - \mathbf{x}_N^*) + \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{x}_i^*)^\top Q (\mathbf{x}_i - \mathbf{x}_i^*) \\ + \sum_{i=0}^{N-1} (u_i - u_i^r)^\top (u_i - u_i^r)$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + Bw_i^x, & \forall i = 0, \dots, N-1, \\ \Psi_x(x_i, u_i) &= w_i^x, & \forall i = 0, \dots, N-1, \\ \Psi_z(x_i, u_i) &= 0, & \forall i = 0, \dots, N-1, \\ \Psi_y(x_i, u_i) &= 0, & \forall i = 0, \dots, N-1, \\ \mathbf{x}_{\min} &\leq \mathbf{x}_i \leq \mathbf{x}_{\max}, & \forall i = 1, \dots, N, \\ w_{\min}^x &\leq w_i^x \leq w_{\max}^x, & \forall i = 0, \dots, N-1, \\ u_{\min} &\leq u_i \leq u_{\max}, & \forall i = 0, \dots, N-1. \end{aligned}$$

Solving the NMPC problem we obtain the optimal sequence of control inputs  $\{u_0, \dots, u_{N-1}\}$ . The first element  $u_0$  is then applied to the linear motor.

Similar to the Combined LMPC+OC scheme, the NMPC scheme is able to guarantee constraint satisfaction and compensate for parasitic forces. Compared to the Combined LMPC+OC scheme, the NMPC scheme can result in better performance as the future error caused by the static nonlinearity is also predicted and minimized. However, its computational load is higher as the number of variables and constraints increases, especially when the prediction horizon is high. The number of variables and constraints in the NMPC scheme is summarized in Table 5.2.

Table 5.2: Number of variables and constraints of the MPC schemes.

	Cascaded LMPC+OC	Combined LMPC+OC	NMPC
$N_{\text{var}}$	$(n_x + n_w)N + n_u$	$n_x N + n_w(N-1) + n_u$	$(n_x + n_u)N$
$N_{\text{eq}}$	$n_x N + 3$	$n_x N + 3$	$(n_x + 3)N$
$N_{\text{ineq}}$	$(n_x + n_w)N + n_u$	$n_x N + n_w(N-1) + n_u$	$(n_x + n_w + n_u)N$

A trade-off between performance and computational load can be made by defining a commutation horizon  $N_{\text{ch}}$ . While the prediction of the linear dynamics is performed for the whole prediction horizon  $N$ , the prediction of the static nonlinearity part is only performed up to the commutation horizon  $N_{\text{ch}}$ . The resulting control problem is formulated as follows

**Problem 5.3.2** (NMPC with commutation horizon).

$$\begin{aligned} \min_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_N, \\ u_0, u_1, \dots, u_{N_{\text{ch}}-1}, \\ w_{N_{\text{ch}}}, \dots, w_{N-1}}} & (\mathbf{x}_N - \mathbf{x}_N^*)^\top P (\mathbf{x}_N - \mathbf{x}_N^*) + \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{x}_i^*)^\top Q (\mathbf{x}_i - \mathbf{x}_i^*) \\ & + \sum_{i=0}^{N_{\text{ch}}-1} (u_i - u_i^r)^\top (u_i - u_i^r) + \sum_{i=N_{\text{ch}}}^{N-1} (w_i^x - w_i^{xr})^\top R_w (w_i^x - w_i^{xr}) \end{aligned}$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + Bw_i^x, & \forall i = 0, \dots, N-1, \\ \Psi_x(x_i, u_i) &= w_i^x, & \forall i = 0, \dots, N_{\text{ch}}-1, \\ \Psi_z(x_i, u_i) &= 0, & \forall i = 0, \dots, N_{\text{ch}}-1, \\ \Psi_y(x_i, u_i) &= 0, & \forall i = 0, \dots, N_{\text{ch}}-1, \\ \mathbf{x}_{\min} &\leq \mathbf{x}_i \leq \mathbf{x}_{\max}, & \forall i = 1, \dots, N, \\ w_{\min}^x &\leq w_i^x \leq w_{\max}^x, & \forall i = 0, \dots, N-1, \\ u_{\min} &\leq u_i \leq u_{\max}, & \forall i = 0, \dots, N-1. \end{aligned}$$

The higher the commutation horizon is, the better the performance is, but also the higher the computational load gets. If the commutation horizon  $N_{\text{ch}} = 1$ , we recover the Combined LMPC+OC scheme. If  $N_{\text{ch}} = N$ , we recover the NMPC scheme.

## 5.4 Simulation results

### 5.4.1 Simulation settings

In this section, a simulation example is presented for demonstration purpose. The example motor has two sets of three-phase coils. As a results, we have 4 independent inputs. The model of the static nonlinear part is given by

$$\begin{aligned} w^x &= \Psi_x(x, u) = K^x(x)u, \\ w^z &= \Psi_z(x, u) = K^z(x)u + u^\top G^z u, \\ w^t &= \Psi_t(x, u) = K^t(x)u + u^\top G^t u, \end{aligned}$$

The position-dependent matrices  $K^i(x)$ , with  $i = x, z, y$ , are  $[4 \times 1]$  matrices given by

$$K^i(x) = \begin{bmatrix} c_1^i \cos\left(\frac{\pi}{\tau_p} x\right) + d_1^i \sin\left(\frac{\pi}{\tau_p} x\right) \\ \vdots \\ c_4^i \cos\left(\frac{\pi}{\tau_p} x\right) + d_4^i \sin\left(\frac{\pi}{\tau_p} x\right) \end{bmatrix}^\top, \quad i = x, z, y, \quad (5.10)$$

where the parameters  $c$  and  $d$  are given in Table 5.3. The matrix  $G^z$  and  $G^y$  are  $[4 \times 4]$  matrices given by

$$G^z = \begin{bmatrix} 0.0128 & 0.0064 & 0.0045 & 0.0023 \\ 0.0064 & 0.0171 & 0.0023 & 0.0002 \\ 0.0045 & 0.0023 & 0.0128 & 0.0064 \\ 0.0023 & 0.0002 & 0.0064 & 0.0171 \end{bmatrix}, \quad (5.11)$$

$$G^y = \begin{bmatrix} -0.0100 & -0.0010 & 0 & 0 \\ -0.0010 & -0.0090 & 0 & 0 \\ 0 & 0 & 0.0100 & 0.0090 \\ 0 & 0 & 0.0090 & 0.0180 \end{bmatrix}. \quad (5.12)$$

Table 5.3: Parameters of the example linear motor.

Parameter	Value	Parameter	Value
$c_1^x$	0.7593	$d_1^x$	77.9009
$c_2^x$	66.5087	$d_2^x$	38.0571
$c_3^x$	-3.5733	$d_3^x$	77.8116
$c_4^x$	67.8933	$d_4^x$	38.2358
$c_1^z$	-0.8811	$d_1^z$	0.2941
$c_2^z$	-0.1817	$d_2^z$	0.2116
$c_3^z$	-1.0782	$d_3^z$	0.1265
$c_4^z$	-0.2469	$d_4^z$	1.0145
$c_1^y$	-0.8235	$d_1^y$	0.2795
$c_2^y$	-0.4406	$d_2^y$	0.8104
$c_3^y$	0.0068	$d_3^y$	0.1204
$c_4^y$	-0.1287	$d_4^y$	0.0260

The linear dynamical system in the  $x$ -direction is a mass-damper system:

$$G(s) = \frac{1}{ms^2 + ds}, \quad (5.13)$$

where  $m$  is the mass of the translator and  $d$  is the damping coefficient. In this example we used  $m = 2$  kg and  $d = 100$  kg/s. The driving force limit is  $\pm 5000$  N. The current limit is  $\pm 30$  A. The sampling rate of the control loop is 1 kHz.

In this example we simulate the step responses of four control schemes: classical scheme, Cascaded LMPC+OC scheme, Combined LMPC+OC scheme and NMPC scheme. The step reference is chosen since it is the most aggressive reference trajectory and hence it is a good test for performance and constraints handling ability. The control schemes are simulated in Matlab on a 2.4 GHz computer. The controller settings are chosen as follows.



### Classical controller

Instead of a simple PID controller, we use a higher-order linear controller for better performance. The linear feedback controller  $C(s)$  is designed using the loop-shaping technique and is given by

$$C(s) = \frac{320s^2 + 6912s + 2.388 \times 10^4}{3.029 \times 10^{-6}s^3 + 0.001658s^2 + 0.2315s}. \quad (5.14)$$

The classical commutation is implemented as described in Section 4.2.

### MPC controllers

In all the MPC schemes, the prediction horizon is  $N = 8$ . The weighting matrices are  $Q = \text{diag}(10^8, 1)$  and  $R_w = 10^{-4}$ . The terminal cost  $P$  is calculated as the solution of the discrete-time Lyapunov equation of the linear dynamics. Optimization problems are solved using the commercial solvers Knitro and IPOPT [16, 109]. The solver are interfaced to Matlab via YALMIP [59]. The solution of the MPC problem at a time sample is used as the "warm-start" initial point for the MPC problem next time sample.

## 5.4.2 Results

### Classical control

The simulation results of the classical control scheme are shown in Figure 5.4. It can be seen that there are high parasitic normal force and torque. The driving force and the currents violate the limits since there is no mechanism to guarantee constraint satisfaction. In practice, the currents can be saturated at the limit for safety. However, this approach lowers the performance of the motor and may even lead to instability.

The average computation time of the position controller is 0.06 ms. The average computation time of the classical commutation algorithm is 0.06 ms. In total, the computation time of the whole controller per sample is 0.12 ms.

### Cascaded LMPC+OC

During simulation, the optimal commutation problem becomes infeasible as the driving force setpoint given by the LMPC controller is unachievable due to the current limit. If we remove the limit on the currents then both the LMPC and optimal commutation problems are feasible. The simulation results are shown in Figure 5.5. It is observed that the parasitic normal force and the parasitic torque are compensated. However, the currents go beyond their limit.

The average computation time of the LMPC problem is 11.67 ms. The average computation time of the optimal commutation problem is 11.22 ms. In total, the average computation time of the whole controller per sample is 22.89 ms.

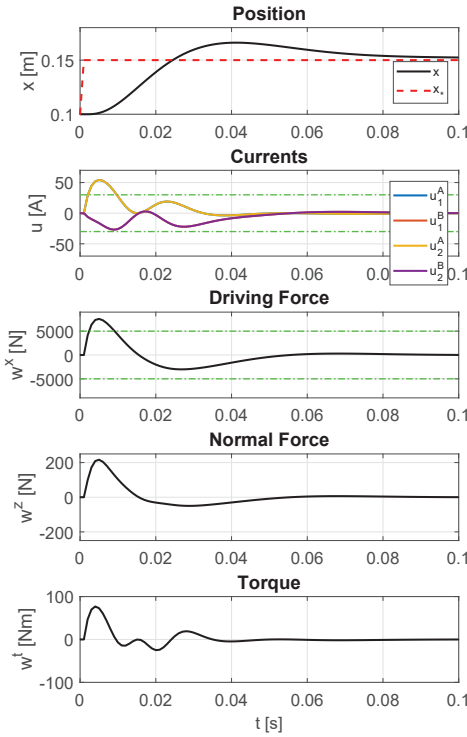


Figure 5.4: Simulation results of classical scheme.

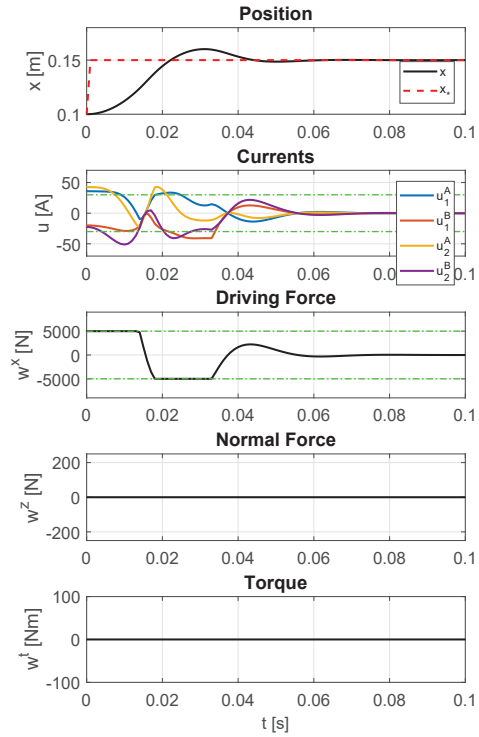


Figure 5.5: Simulation results of Cascaded LMPC+OC scheme.

### Combined LMPC+OC

The simulation results of the Combined LMPC+OC scheme are shown in Figure 5.6. Similar to the Cascaded LMPC+OC scheme, the parasitic force and torque in the non-driving directions are compensated. But in contrast to the Cascaded LMPC+OC scheme, the Combined LMPC+OC scheme is able to keep the input currents within their limits. However, the average computation time of of the Combined LMPC+OC problem is 66.45 ms, which is higher than that of the Cascaded LMPC+OC scheme.

### NMPC

The simulation results of the NMPC scheme are shown in Figure 5.7. It can be seen that the parasitic forces in non-driving directions are compensated, similar to the two LMPC+OC schemes. The driving force and the currents are kept within their limits. The average computation time of the NMPC problem is 93.21 ms, which is the highest of all schemes.

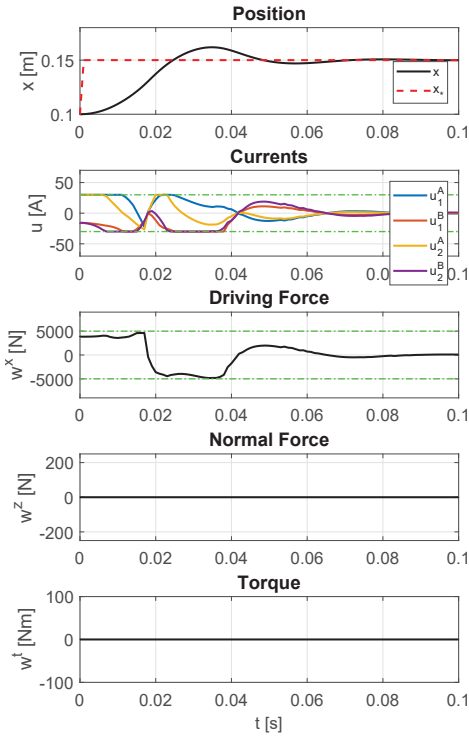


Figure 5.6: Simulation results of Combined LMPC+OC scheme.

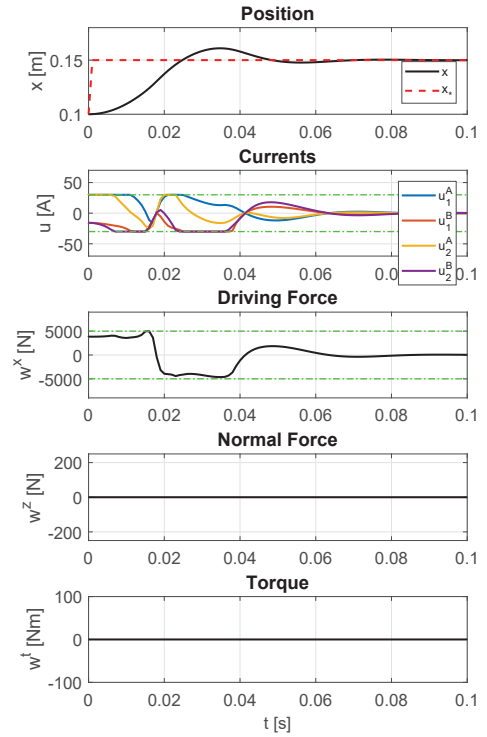


Figure 5.7: Simulation results of NMPC scheme.

### 5.4.3 Analysis

Figure 5.8 shows the step responses of all the control schemes. It is observed that the performance of the MPC schemes is more or less comparable. The Cascaded LMPC+OC scheme gives the fastest response, but it has to violate the current limit. The NMPC scheme gives a slightly faster response compared to the Combined LMPC+OC scheme, since it minimizes the predicted error caused by the static nonlinearity.

Table 5.4 summarizes the properties of all the control schemes. It is noted that we solve the MPC problems using two solvers Knitro and IPOPT. The best computation time is reported in the table. It is observed that the classical scheme requires the least computational load, while the NMPC scheme needs the highest computational effort.

Regarding parasitic effects, all the MPC schemes are able to compensate for parasitic forces and torques in non-driving directions, while the classical scheme cannot. Among all the schemes, only the Combined LMPC+OC and NMPC schemes are capable of guaranteeing constraint satisfaction. The classical control scheme does not have a constraints handling mechanism. Consequently, the constraints are violated

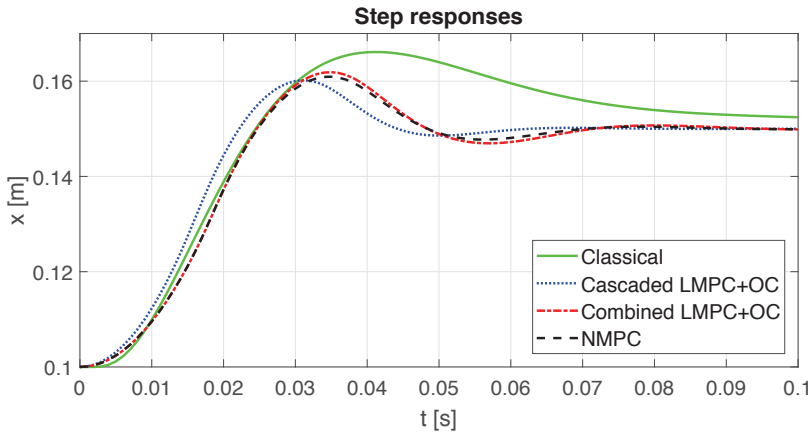


Figure 5.8: Step responses of the four control schemes.

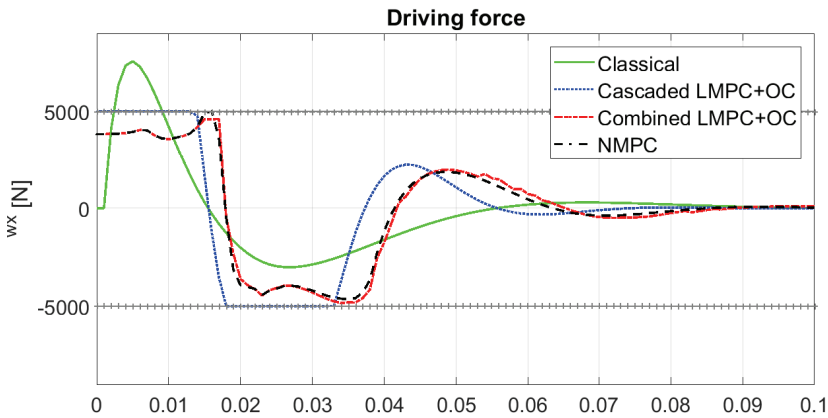


Figure 5.9: Driving forces of the four control schemes.

as shown in Figure 5.4. In the Cascaded LMPC+OC scheme, the LMPC controller is not aware of the current limit and may ask for a high force which is not feasible due to the current limit.

Figure 5.9 shows the comparison of the driving force in the four control schemes. It can be seen that in the Cascaded LMPC+OC scheme, at some points the LMPC controller asks for maximum driving force, which is unachievable at those points given the current limit. This leads to infeasibility of the optimal commutation problem. On the other hand, the Combined LMPC+OC scheme and NMPC scheme are well aware of the current limit and only asks for smaller force, making the problem feasible.

From the simulation results, it is observed that the Combined LMPC+OC is the best among the three MPC schemes. Similar to the NMPC scheme, it is able to compensate for parasitic force, guarantee constraint satisfaction. Although the NMPC scheme deliver slightly better performance, the Combined LMPC+OC scheme requires lower computation time, especially for high prediction horizon.

Table 5.4: Simulation results of the control schemes

	Classical	Cascaded LMPC+OC	Combined LMPC+OC	NMPC
Average computation time [ms]	0.12	22.89	66.45	93.21
Non-driving force compensation	No	Yes	Yes	Yes
Force constraints	No	Yes	Yes	Yes
Current constraints	No	No	Yes	Yes

## 5.5 Conclusions

In this chapter we analyze the drawbacks of the Cascaded LMPC+OC scheme. The Cascaded LMPC+OC scheme is obtained by simply replacing the classical controllers with LMPC and optimal commutation. This scheme requires solving two separate optimization problems, one for LMPC and one for optimal commutation. Due to the separation, the scheme cannot guarantee current constraint satisfaction since the LMPC problem is not aware of the current limit.

To address this problem, we introduce two new MPC schemes for linear motors, which are the Combined LMPC+OC scheme and the NMPC scheme. The introduced MPC schemes are able to compensate for parasitic forces and torques and also guarantee constraint satisfaction. The Combined LMPC+OC scheme combines the LMPC problem and optimal commutation problem into a single optimization problem. As a result, the constraint satisfaction can be guaranteed. However, the error caused by the static nonlinearity is not predicted and therefore not minimized. The NMPC scheme includes the full nonlinear model for prediction and therefore delivers better performance, at the price of higher computational effort. The effectiveness of the new MPC schemes has been demonstrated in a simulation example.

For high prediction horizon, the designed MPC schemes require a high computational effort. This makes it difficult to apply the MPC schemes to real CLMs with fast sampling rates where the time for computation is limited. In the next chapter we address this problem by developing a computationally efficient optimization algorithm.



# Chapter 6

## A sequential quadratic programming method with arbitrary Hessian approximation

### 6.1 Introduction

The optimal commutation and NMPC techniques developed in the previous chapters require solving nonlinear optimization problems online, which is in general computationally costly. This makes it challenging to apply these control techniques to a real linear motor where the sampling rate is high, typically in the kHz range, and the time for computation is thus limited. To address this problem, in this chapter we develop a computationally efficient optimization algorithm which can be considered as a variant of Sequential Quadratic Programming (SQP).

SQP is one of the most effective methods for solving nonlinear optimization problems. The main idea of SQP is to iteratively approximate the Nonlinear Programming (NLP) problem by a sequence of Quadratic Programming (QP) subproblems [15]. The QP subproblems should be constructed in a way that the resulting sequence of solutions converges to a local optimum of the NLP.

There are different ways to construct the QP subproblems. When the exact Hessian is used to construct the QP subproblems, local convergence with quadratic convergence rate is guaranteed. However, the true Hessian can be indefinite when far

---

Part of the content of this chapter has been published in:

- T. T. Nguyen, M. Lazar and H. Butler, “A method to guarantee local convergence for sequential quadratic programming with poor Hessian approximation,” in IEEE 56th Conference on Decision and Control (CDC), Melbourne, 2017.
- T. T. Nguyen, M. Lazar and H. Butler, “A Hessian-free algorithm for solving quadratic optimization problems with nonlinear equality constraints,” in IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, 2016.

from the solution. Consequently, the QP subproblems are non-convex and generally difficult to solve, since the objective may be unbounded below and there may be many local solutions [33]. Moreover, computing the exact Hessian is generally expensive, which makes SQP with exact Hessian difficult to apply to large-scale problems and real-time applications.

To overcome these drawbacks, positive (semi-) definite Hessian approximations are usually used in practice. SQP methods using Hessian approximations generally guarantee local convergence under some assumptions. Some SQP variants employ iterative updates scheme for the Hessian approximation to keep it close to the true Hessian. Broyden-Fletcher-Goldfarb-Shanno (BFGS) is one of the most popular update schemes of this type [27, 82]. The BFGS version of SQP guarantees super-linear convergence when the initial Hessian estimate is close enough to the true Hessian [15]. Another variant which is very popular for constrained nonlinear least squares problems is the generalized Gauss-Newton (GGN) method [13, 14]. GGN method converges locally only if the residual function is small at the solution [21]. Some other SQP variants belong to the class of sequential convex programming (SCP), or sequential convex quadratic programming (SCQP) methods, which exploit convexity in either the objective or the constraint functions to formulate convex QP subproblems [23, 108]. SCP methods also have local convergence under similar assumption of small residual function. However, these assumptions are not always satisfied in practice, resulting in poor Hessian approximation and thus no convergence is guaranteed.

In this chapter, we introduce a simple method to guarantee local convergence for SQP methods with poor Hessian approximations. In this method, the search direction is taken as the interpolation between the search direction provided by solving the QP subproblem and a feasible search direction. It will be proved that there exist suitable interpolation coefficients such that the resulting algorithm converges locally to a local optimum of the NLP with linear convergence rate. To guarantee convergence from any initial point, we employ the strategy for global convergence presented in [32], with a slight modification. Numerical examples are presented to demonstrate the effectiveness of the new optimization method.

The idea of interpolating an optimal search direction with a feasible search direction was first introduced in our previous work for quadratic optimization problems with nonlinear equality constraints [69]. The method presented in [69] has been applied effectively to a practical application in commutation of linear motors in Chapter 4. In the current chapter, the idea is extended to general nonlinear programming problems.

The remainder of this chapter is organized as follows. Section 6.2 presents the formulation and the optimality conditions of the generic NLP that is considered in this chapter. In Section 6.3, the basic SQP method for solving NLPs is discussed. In Section 6.4, we introduce our new variant of SQP and prove its optimality property. The local convergence property of the new method is proved in Section 6.5. In Section 6.6, a strategy for global convergence is presented. Section 6.7 discusses the treatment of inequality constraints. Two numerical examples are presented in



Section 6.8 to demonstrate the effectiveness of the new optimization method. The conclusions are summarized in Section 6.9.

### 6.1.1 Notation

Let  $\mathbb{N}$  denote the set of natural numbers,  $\mathbb{R}$  denote the set of real numbers. The notation  $\mathbb{R}_{[c_1, c_2]}$  denotes the set  $\{c \in \mathbb{R} : c_1 \leq c < c_2\}$ . Let  $\mathbb{R}^n$  denote the set of real column vectors of dimension  $n$ ,  $\mathbb{R}^{n \times m}$  denote the set of real  $n \times m$  matrices. For a vector  $x \in \mathbb{R}^n$ ,  $x_{[i]}$  denotes the  $i$ -th element of  $x$ . The notation  $0_{n \times m}$  denotes the  $n \times m$  zero matrix and  $I_n$  denotes the  $n \times n$  identity matrix. Let  $\|\cdot\|$  denote the 2-norm. The Nabla symbol  $\nabla$  denotes the gradient operator. For a vector  $x \in \mathbb{R}^n$  and a mapping  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$

$$\nabla_x \Phi(x) = \left[ \frac{\partial \Phi(x)}{\partial x_{[1]}} \quad \frac{\partial \Phi(x)}{\partial x_{[2]}} \quad \cdots \quad \frac{\partial \Phi(x)}{\partial x_{[n]}} \right]^\top.$$

Let  $\mathcal{B}(x_0, r)$  denote the open ball  $\{x \in \mathbb{R}^n : \|x - x_0\|_2 < r\}$ .

## 6.2 Nonlinear programming problem

In this chapter we consider the generic constrained nonlinear programming problem of the following form

**Problem 6.2.1 (NLP).**

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) \leq 0, \end{aligned}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^{m_g}$  and  $h : \mathbb{R}^n \rightarrow \mathbb{R}^{m_h}$  are twice continuously differentiable and possibly nonconvex. Here,  $n$  denotes the number of optimization variables,  $m_g$  denotes the number of equality constraints and  $m_h$  denotes the number of inequality constraints.

The KKT first-order optimality conditions of the NLP Problem 6.2.1 are [75]

$$\nabla f(x_*) + \nabla g(x_*)\mu_* + \nabla h(x_*)\nu_* = 0 \tag{6.1a}$$

$$g(x_*) = 0 \tag{6.1b}$$

$$h(x_*) \leq 0 \tag{6.1c}$$

$$\nu_{[i]*} \geq 0, \quad \forall i = 1, \dots, m_h \tag{6.1d}$$

$$\nu_{[i]*} h_{[i]}(x_*) = 0, \quad \forall i = 1, \dots, m_h, \tag{6.1e}$$

where  $\mu \in \mathbb{R}^{m_g}$  and  $\nu \in \mathbb{R}^{m_h}$  are the Lagrange multipliers vectors. Here,  $x_*$  denotes a local optimum of the NLP,  $\mu_*$  and  $\nu_*$  denotes its corresponding vectors of Lagrange multipliers.

For ease of presentation, in the next sections we only consider equality constraints. The treatment of inequality constraints will be discussed in Section 6.7. The NLP with equality constraints has the following form

**Problem 6.2.2** (NLP with equality constraints).

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0_{m \times 1}. \end{aligned}$$

The Lagrangian function of the problem is defined as

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^\top g(x). \quad (6.2)$$

where  $\lambda \in \mathbb{R}^m$  is the vector of Lagrange multipliers and  $m$  is the number of equality constraints. The KKT optimality condition of Problem 6.2.2 is [75]

$$\nabla \mathcal{L}(x_*, \lambda_*) = 0_{(n+m) \times 1}, \quad (6.3)$$

or equivalently

$$\nabla f(x_*) + \nabla g(x_*) \lambda_* = 0_{n \times 1}, \quad (6.4a)$$

$$g(x_*) = 0_{m \times 1}. \quad (6.4b)$$

The solution of the NLP is generally searched for in an iterative way. At a current iterate  $x_k$ , the next iterate is computed as

$$x_{k+1} = x_k + t_k \Delta x_k, \quad (6.5)$$

where  $\Delta x_k$  is the search direction and  $t_k \in \mathbb{R}_{(0,1)}$  is the stepsize, also known as the steplength. Different optimization methods have different ways to calculate the search direction such that the iteration eventually converges to a local minimum of the optimization problem. In the next section, we review the basic SQP method, which is one of the most powerful methods for solving nonlinear optimization problems. Furthermore, our new variant of SQP, which is designed to improve the computational efficiency of the method, is introduced and analyzed.

### 6.3 Basic sequential quadratic programming

In this section we review the basic SQP method for solving nonlinear optimization problems. In SQP method, the NLP is approximated by a sequence of QP subproblems. At an iteration  $k \in \mathbb{N}$ , the search direction  $\Delta x_k^{\text{QP}}$  is the solution of the following QP subproblem

$$\begin{aligned} \min_{\Delta x_k \in \mathbb{R}^n} \quad & \frac{1}{2} \Delta x_k^\top B_k \Delta x_k + \nabla f(x_k)^\top \Delta x_k \\ \text{s.t.} \quad & \nabla g(x_k)^\top \Delta x_k + g(x_k) = 0. \end{aligned} \quad (6.6)$$

Here,  $B_k \in \mathbb{R}^{n \times n}$  is either the exact Hessian of the Lagrangian  $\nabla \mathcal{L}(x_k, \lambda_k)$ , or a positive (semi-) definite approximation of the Hessian. Similar to [15], to guarantee that the QP subproblem has a unique solution, we assume that the matrices  $B_k$  satisfy the following conditions:

**Assumption 6.3.1.** *The matrices  $B_k$  are uniformly positive definite on the null spaces of the matrices  $\nabla g(x_k)^\top$ , i.e., there exists a  $\beta_1 > 0$  such that for each  $k$*

$$d^\top B_k d \geq \beta_1 \|d\|^2,$$

for all  $d \in \mathbb{R}^n$  which satisfy

$$\nabla g(x_k)^\top d = 0.$$

**Assumption 6.3.2.** *The sequence  $\{B_k\}$  is uniformly bounded, i.e., there exists a  $\beta_2 > 0$  such that for each  $k$*

$$\|B_k\| \leq \beta_2.$$

Furthermore, let us assume that the linear independence constraint qualification (LICQ) holds at each  $k$ .

**Assumption 6.3.3 (LICQ).** *For all  $k$ , the columns of  $\nabla g(x_k)$  are linearly independent.*

Given that the above assumptions hold, the QP subproblem can be solved as follows. The KKT optimality conditions of the QP subproblem (6.6) are

$$B_k \Delta x_k^{\text{QP}} + \nabla f(x_k) + \nabla g(x_k) \lambda_{k+1}^{\text{QP}} = 0_{n \times 1}, \quad (6.7a)$$

$$\nabla g(x_k)^\top \Delta x_k^{\text{QP}} + g(x_k) = 0_{m \times 1}, \quad (6.7b)$$

or equivalently

$$\begin{bmatrix} B_k & \nabla g(x_k) \\ \nabla g(x_k)^\top & 0_{m \times m} \end{bmatrix} \begin{bmatrix} \Delta x_k^{\text{QP}} \\ \lambda_{k+1}^{\text{QP}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \end{bmatrix}. \quad (6.8)$$

It should be noted that  $B_k$  is positive (semi-) definite and is not necessarily invertible, but the matrix  $\begin{bmatrix} B_k & \nabla g(x_k)^\top \\ \nabla g(x_k) & 0_{m \times m} \end{bmatrix}$  is invertible due to Assumptions 6.3.1 and 6.3.3 [11, Theorem 3.2]. Therefore, the KKT condition (6.8) has a unique solution

$$\begin{bmatrix} \Delta x_k^{\text{QP}} \\ \lambda_{k+1}^{\text{QP}} \end{bmatrix} = - \begin{bmatrix} B_k & \nabla g(x_k) \\ \nabla g(x_k)^\top & 0_{m \times m} \end{bmatrix}^{-1} \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \end{bmatrix}. \quad (6.9)$$

For brevity, let us denote

$$W_k := \begin{bmatrix} B_k & \nabla g(x_k) \\ \nabla g(x_k)^\top & 0_{m \times m} \end{bmatrix}.$$

In addition, let  $G_k$  denote the  $[n \times n]$  leading principal submatrix of  $W_k^{-1}$

$$G_k := [I_n \quad 0_{n \times m}] W_k^{-1} \begin{bmatrix} I_n \\ 0_{m \times n} \end{bmatrix}.$$

If  $B_k$  is the exact Hessian then the basic SQP method is equivalent to applying Newton's method to solve the KKT conditions (6.4), which guarantees quadratic local convergence rate [75, Chapter 18]. When an approximation is used instead, local convergence is guaranteed only when  $B_k$  is close enough to the true Hessian at the solution [15]. In [108], it is proved that the basic SQP method converges locally if and only if the Hessian approximation at the solution lies inside a certain region around the true Hessian at the solution.

## 6.4 Sequential quadratic programming with arbitrary Hessian approximation

The basic SQP method converges only if the Hessian approximation is close to the true Hessian. However, a good Hessian approximation can be expensive to compute. To address this problem, we introduce a variant of SQP which guarantees local convergence for arbitrary (semi-) positive definite Hessian approximation. In this method, the search iteration is chosen as the interpolation between an optimal search iteration, without local convergence guarantee, and a feasible search iteration with guaranteed local convergence.

The search direction  $\Delta x_k^{\text{QP}}$  can be viewed as an optimal direction which iteratively leads to the optimal solution of the NLP, if the iteration converges. However, local convergence is not guaranteed if  $B_k$  is a poor approximation of the true Hessian. To guarantee local convergence with poor Hessian approximation, we introduce a new search direction which is the interpolation between the optimal search direction  $\Delta x_k^{\text{QP}}$  and a feasible search direction  $\Delta x_k^{\text{f}}$ , i.e.

$$\Delta x_k = \alpha \Delta x_k^{\text{QP}} + (1 - \alpha) \Delta x_k^{\text{f}}, \quad (6.10)$$

where  $\alpha \in \mathbb{R}_{(0,1)}$  is the interpolation coefficient. The feasible search direction  $\Delta x_k^{\text{f}}$  only searches for a feasible solution of the set of constraints, but its local convergence is guaranteed. The idea of this new interpolated update is to combine the optimality property of the SQP update and the local convergence property of the feasible update.

The feasible search direction can be found as a solution of the linearized constraints

$$\nabla g(x_k)^\top \Delta x_k^{\text{f}} + g(x_k) = 0_{m \times 1}. \quad (6.11)$$

Since  $m < n$ , there is an infinite number of solutions for (6.11). We propose the following feasible search direction

$$\Delta x_k^{\text{f}} = -\nabla g(x_k)^\dagger g(x_k), \quad (6.12)$$

where  $\nabla g(x_k)^\dagger \in \mathbb{R}^{n \times m}$  is the Moore-Penrose generalized right inverse of  $\nabla g(x_k)^\top$  [84], i.e.

$$\nabla g(x_k)^\dagger := \nabla g(x_k)(\nabla g(x_k)^\top \nabla g(x_k))^{-1}.$$

Since  $\nabla g(x_k)^\top \nabla g(x_k)^\dagger = I_m$ , by multiplying  $\nabla g(x_k)^\top$  to the left of both sides of equation (6.12), it can be easily verified that  $\Delta x_k^f$  is a solution of (6.11). It has been proven that the feasible update (6.12) converges locally to a feasible solution of the constraints [55].

In summary, at a current iterate  $x_k$ , the next iterate is the interpolation between the SQP iterate and a feasible iterate. The resulting interpolated update is

$$\begin{aligned} x_{k+1} &= x_k + t_k \Delta x_k \\ &= x_k + t_k \left( \alpha \Delta x_k^{\text{QP}} + (1 - \alpha) \Delta x_k^f \right). \end{aligned} \quad (6.13)$$

In what follows we will prove the optimality property of the iteration (6.13).

**Theorem 6.4.1.** *Iteration (6.13) converges to a fixed point  $x_*$ , i.e.  $\|\Delta x_*\| = 0$ , if and only if  $x_*$  satisfies the KKT optimality conditions (6.4).*

*Proof.* First, we will prove that if  $\|\Delta x_*\| = 0$  then  $x_*$  satisfies the KKT optimality conditions (6.4). Since  $\Delta x_k^{\text{QP}}$  satisfies (6.7b) and  $\Delta x_k^f$  satisfies (6.11), it follows that

$$\nabla g(x_k)^\top \Delta x_k + g(x_k) = 0_{m \times 1}. \quad (6.14)$$

Therefore, if  $\|\Delta x_*\| = 0$  then we have

$$g(x_*) = 0_{m \times 1}. \quad (6.15)$$

Substituting (6.15) into (6.12) results in  $\Delta x_*^f = 0_{n \times 1}$ . Consequently, due to (6.10) we also have  $\Delta x_*^{\text{QP}} = 0_{n \times 1}$ . As a result, it follows from (6.7a) that

$$\nabla f(x_*) + \nabla g(x_*) \lambda_* = 0_{n \times 1}. \quad (6.16)$$

From (6.15) and (6.16), it can be concluded that  $x_*$  satisfies the KKT optimality conditions (6.4).

Now we will prove that if  $x_*$  satisfies the KKT optimality conditions (6.4) then  $\|\Delta x_*\| = 0$ . Substituting (6.4) into (6.7) yields

$$B_k \Delta x_*^{\text{QP}} = 0_{n \times 1}, \quad (6.17)$$

$$\nabla g(x_*)^\top \Delta x_*^{\text{QP}} = 0_{m \times 1}, \quad (6.18)$$

Due to Assumption 6.3.1, this only holds when  $\Delta x_*^{\text{QP}} = 0_{n \times 1}$ . Furthermore, substituting (6.4b) into (6.14) leads to  $\Delta x_*^f = 0_{n \times 1}$ . As  $\Delta x_*$  is an interpolation between  $\Delta x_*^{\text{QP}}$  and  $\Delta x_*^f$ , we have  $\|\Delta x_*\| = 0$ .  $\square$

## 6.5 Local convergence

An algorithm is locally convergent if it converges to a local minimum of the problem with unit stepsize, i.e.  $t_k = 1$ , given that the initial point is close enough to the local minimum. In this section we will prove the local convergence property of the search iteration (6.13) with unit stepsize, i.e.

$$x_{k+1} = x_k + \alpha \Delta x_k^{\text{QP}} + (1 - \alpha) \Delta x_k^{\text{f}}. \quad (6.19)$$

The following lemmas will be used in the proof.

**Lemma 6.5.1.** *Let  $\mathcal{D} \subseteq \mathbb{R}^n$  be a convex set in which  $g : \mathcal{D} \rightarrow \mathbb{R}^m$  is differentiable and  $\nabla g(x)$  is Lipschitz continuous for all  $x \in \mathcal{D}$ , i.e. there exists a  $\gamma > 0$  such that*

$$\|\nabla g(x) - \nabla g(y)\| \leq 2\gamma \|x - y\|, \quad \forall x, y \in \mathcal{D}. \quad (6.20)$$

Then

$$\|g(x) - g(y) - \nabla g(y)(x - y)\| \leq \gamma \|x - y\|^2, \quad \forall x, y \in \mathcal{D}. \quad (6.21)$$

A proof of 6.5.1 can be found in [55].

**Lemma 6.5.2.** *Let  $\mathcal{D}$  be a subset of  $\mathbb{R}^n$  where  $\nabla g(x_k)$  and  $\nabla g(x_k)^\dagger$  are well defined. For any  $x \in \mathcal{D}$ , it holds that*

$$\|I_n - \nabla g(x_k)^\dagger \nabla g(x_k)\| = 1. \quad (6.22)$$

*Proof.* Let us denote  $A := I_n - \nabla g(x_k)^\dagger \nabla g(x_k)$ . It can easily be verified that  $A$  is a Hermitian idempotent matrix, i.e.

$$A^\top = A, \quad (6.23)$$

$$AA = A. \quad (6.24)$$

Let  $\rho_i$  be an eigenvalue of  $A^\top A$ , then  $\rho_i$  satisfies

$$\begin{aligned} \rho_i x &= A^\top A x \\ &= A^\top A (A^\top A x) \\ &= A^\top A (\rho_i x) \\ &= \rho_i (A^\top A x) \\ &= \rho_i^2 x, \quad \forall x \in \mathcal{D}. \end{aligned} \quad (6.25)$$

This results in

$$\rho_i = \rho_i^2, \quad (6.26)$$

which means that  $\rho_i$  is equal to either 0 or 1. If  $A^\top A$  is a nonzero matrix then at least one eigenvalue is 1. The 2-norm of  $A$  is defined as the maximum singular value of  $A$ . This is equal to the square root of the maximum eigenvalue of  $A^\top A$ , which is equal to 1.  $\square$

In the following theorem we will prove the local convergence property of the iteration (6.19).

**Theorem 6.5.3.** *Let  $\mathcal{D} = \mathcal{B}(x_*, r)$ , where  $r > 0$ , be a neighborhood of  $x_*$  in which the following conditions hold*

1.  $f(x)$  and  $g(x)$  are Lipschitz continuous and continuously differentiable,
2.  $\nabla g(x)$  is Lipschitz continuous and bounded,
3.  $\nabla g(x)^\dagger$  is Lipschitz continuous and bounded,
4.  $W_k^{-1}$  is bounded for all  $k$ ,

Let the initialization  $x_0 \in \mathcal{D}$  be close enough to  $x_*$ . Then there exists a  $\alpha \in \mathbb{R}_{(0,1)}$  such that iteration (6.19) converges to  $x_*$ .

*Proof.* Let us consider two cases

- $g(x)$  is linear.
- $g(x)$  is nonlinear.

1. *Case 1:* in the first case when  $g(x)$  is linear, for any iterate  $x_k$  we can write

$$g(x) = g(x_k) + \nabla g(x_k)^\top (x - x_k). \quad (6.27)$$

It follows that

$$g(x_{k+1}) = g(x_k) + \nabla g(x_k)^\top (x_{k+1} - x_k) = 0_{m \times 1}, \quad (6.28)$$

where the last equality follows from (6.14). Therefore, we have that  $g(x_k) = 0_{m \times 1}$  for all  $k \geq 1$ . The interpolated update is then reduced to

$$x_{k+1} = x_k + \alpha \Delta x_k^{\text{QP}}, \quad \forall k \geq 1. \quad (6.29)$$

Substituting  $g(x_k) = 0_{m \times 1}$  into (6.9) we have

$$\Delta x_k^{\text{QP}} = -G_k \nabla f(x_k), \quad \forall k \geq 1. \quad (6.30)$$

We have that  $W_k$  is positive definite due to Assumption 6.3.1. It follows that  $G_k$  is positive definite, due to the facts that the inverse of a positive definite matrix is positive definite, and that every principal submatrix of a positive definite matrix is positive definite [12, Chapter 8]. As a result we have

$$\nabla f(x_k)^\top \Delta x_k^{\text{QP}} = -\nabla f(x_k)^\top G_k \nabla f(x_k) < 0, \quad \forall k \geq 1. \quad (6.31)$$

This shows that  $\Delta x_k^{\text{QP}}$  is a descent direction that leads to a decrease in the cost function  $f(x)$ . In addition, since  $g(x_k) = 0_{m \times 1}$  for all  $k \geq 1$ , we have that  $\Delta x_k^{\text{QP}}$  is

also a feasible direction. Therefore, there exists a stepsize  $\alpha \in \mathbb{R}_{(0,1)}$  such that the iteration (6.29) converges [75, Chapter 3].

2. *Case 2:* let us now consider the case when  $g(x)$  is nonlinear. Since the nonlinear constraints are solved by successive linearization (6.14), we can assume that the solution is reached asymptotically, i.e.  $g(x_k) \rightarrow 0$  as  $k \rightarrow \infty$  and  $g(x_k) \neq 0_{m \times 1}$  for all  $k < \infty$ .

Let us first analyze  $\Delta x^{\text{QP}}$ . Subtracting  $\begin{bmatrix} x_* \\ \lambda_* \end{bmatrix}$  to both sides of (6.9) results in

$$\begin{aligned}
& \begin{bmatrix} x_{k+1}^{\text{QP}} - x_* \\ \lambda_{k+1}^{\text{QP}} - \lambda_* \end{bmatrix} \\
&= \begin{bmatrix} x_k - x_* \\ -\lambda_* \end{bmatrix} - W_k^{-1} \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \end{bmatrix} \\
&= W_k^{-1} \left( W_k \begin{bmatrix} x_k - x_* \\ -\lambda_* \end{bmatrix} - \begin{bmatrix} \nabla f(x_k) \\ g(x_k) \end{bmatrix} \right) \\
&= W_k^{-1} \begin{bmatrix} B_k(x_k - x_*) - (\nabla f(x_k) + \nabla g(x_k)\lambda_*) \\ \nabla g(x_k)^\top (x_k - x_*) - g(x_k) \end{bmatrix} \\
&= W_k^{-1} \begin{bmatrix} B_k(x_k - x_*) - (\nabla f(x_k) + \nabla g(x_k)\lambda_* - \nabla f(x_*) - \nabla g(x_*)\lambda_*) \\ - (g(x_k) - g(x_*) - \nabla g(x_k)^\top (x_k - x_*)) \end{bmatrix} \\
&= W_k^{-1} \begin{bmatrix} (B_k - \nabla^2 f(x_*) - \sum_{i=1}^m \nabla^2 g_{[i]}(x_*)\lambda_{i*})(x_k - x_*) \\ - (g(x_k) - g(x_*) - \nabla g(x_k)^\top (x_k - x_*)) \end{bmatrix} + \begin{bmatrix} \mathcal{O}(\|x_k - x_*\|^2) \\ 0 \end{bmatrix} \\
&= W_k^{-1} \begin{bmatrix} (B_k - \nabla^2 \mathcal{L}_*)(x_k - x_*) \\ 0 \end{bmatrix} + \begin{bmatrix} \mathcal{O}(\|x_k - x_*\|^2) \\ \mathcal{O}(\|x_k - x_*\|^2) \end{bmatrix}. \tag{6.32}
\end{aligned}$$

Note that in the above derivation we use the following Taylor series expansion

$$\nabla f(x_k) - \nabla f(x_*) = \nabla^2 f(x_*)(x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2), \tag{6.33}$$

$$\nabla g(x_k)\lambda_* - \nabla g(x_*)\lambda_* = \sum_{i=1}^m \nabla^2 g_{[i]}(x_*)\lambda_{i*}(x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2). \tag{6.34}$$

It follows that

$$x_{k+1}^{\text{QP}} - x_* = G_k (B_k - \nabla^2 \mathcal{L}_*)(x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2). \tag{6.35}$$

Due to the condition that  $W_k^{-1}$  and  $B_k$  are bounded, there exists a constant  $N > 0$  such that

$$\|G_k (B_k - \nabla^2 \mathcal{L}_*)(x_k - x_*)\| \leq N \|x_k - x_*\|. \tag{6.36}$$

Now let us consider  $\Delta x^f$ . From (6.12) we have

$$\begin{aligned}
x_{k+1}^f - x_* &= x_k - x_* - \nabla g(x_k)^\dagger g(x_k) \\
&= x_k - x_* - \nabla g(x_k)^\dagger g(x_k) + \left( 0.5 \nabla g(x_*)^\dagger g(x_k) - 0.5 \nabla g(x_*)^\dagger g(x_k) \right) \\
&= x_k - x_* - 0.5 \nabla g(x_k)^\dagger g(x_k) - 0.5 \nabla g(x_*)^\dagger g(x_k) \\
&\quad - 0.5 (\nabla g(x_k)^\dagger - \nabla g(x_*)^\dagger) g(x_k). \tag{6.37}
\end{aligned}$$



Notice that  $g(x_k)$  can be written as

$$g(x_k) = g(x_k) - g(x_*) = \nabla g(x_k)^\top (x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2), \quad (6.38)$$

and also as

$$g(x_k) = g(x_k) - g(x_*) = \nabla g(x_*)^\top (x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2). \quad (6.39)$$

In addition, due to the Lipschitz continuity of  $\nabla g(x)^\dagger$ , we have

$$\nabla g(x_k)^\dagger - \nabla g(x_*)^\dagger = \mathcal{O}(\|x_k - x_*\|). \quad (6.40)$$

It follows that

$$0.5(\nabla g(x_k)^\dagger - \nabla g(x_*)^\dagger)g(x_k) = \mathcal{O}(\|x_k - x_*\|^2). \quad (6.41)$$

Equation (6.37) can then be rewritten as

$$\begin{aligned} x_{k+1}^f - x_* &= 0.5(I_n - \nabla g(x_k)^\dagger \nabla g(x_k)^\top)(x_k - x_*) \\ &\quad + 0.5(I_n - \nabla g(x_*)^\dagger \nabla g(x_*)^\top)(x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2). \end{aligned} \quad (6.42)$$

Here,  $(I_n - \nabla g(x_k)^\dagger \nabla g(x_k)^\top)$  and  $(I_n - \nabla g(x_*)^\dagger \nabla g(x_*)^\top)$  are the orthogonal projections onto the null space of  $\nabla g(x_k)^\top$  and  $\nabla g(x_*)^\top$ , respectively [12, Chapter 6]. Applying the result of Lemma 6.5.2 to (6.42), it follows that

$$\|0.5\left(I_n - \nabla g(x_k)^\dagger \nabla g(x_k)^\top + I_n - \nabla g(x_*)^\dagger \nabla g(x_*)^\top\right)(x_k - x_*)\| \leq \|x_k - x_*\|. \quad (6.43)$$

The equality holds if and only if  $(x_k - x_*)$  is in the null spaces of both  $\nabla g(x_k)$  and  $\nabla g(x_*)$ , i.e.

$$\Gamma(x_k) = \nabla g(x_k)^\top (x_k - x_*) = 0, \quad (6.44)$$

$$\Gamma(x_*) = \nabla g(x_*)^\top (x_k - x_*) = 0, \quad (6.45)$$

where  $\Gamma: \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as follows

$$\Gamma(x) = \nabla g(x)^\top (x_k - x_*).$$

Since  $\nabla g(x)^\top$  is Lipschitz continuous and bounded in  $\mathcal{D}$ , it follows that  $\Gamma(x)$  is also Lipschitz continuous in  $\mathcal{D}$ . Due to the Lipschitz continuity of  $\Gamma(x)$ , if  $\mathcal{D}$  is small enough then (6.44) and (6.45) both hold if and only if  $\Gamma(x) = 0$  for all  $x$  in the line segment between  $x_k$  and  $x_*$ . This means that  $g(x)$  is neither increasing nor decreasing in the line segment between  $x_k$  and  $x_*$ , i.e.  $g(x_k) = g(x_*) = 0_{m \times 1}$ , which contradicts the assumption that  $g(x_k) \neq 0$  for all  $k < \infty$ . Therefore, there exists a constant  $M \in \mathbb{R}_{(0,1)}$  such that

$$\|0.5(I_n - \nabla g(x_k)^\dagger \nabla g(x_k)^\top + I_n - \nabla g(x_*)^\dagger \nabla g(x_*)^\top)(x_k - x_*)\| \leq M\|x_k - x_*\|. \quad (6.46)$$

It follows from (6.35) and (6.42) that

$$\begin{aligned} x_{k+1} - x_* &= \alpha G_k (B_k - \nabla^2 \mathcal{L}_*) (x_k - x_*) \\ &\quad + (1 - \alpha) 0.5 (I - \nabla g(x_k)^\dagger \nabla g(x_k)^\top) (x_k - x_*) \\ &\quad + (1 - \alpha) 0.5 (I - \nabla g(x_*)^\dagger \nabla g(x_*)^\top) (x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2). \end{aligned} \quad (6.47)$$

From (6.47), (6.36) and (6.46) we have

$$\begin{aligned} \|x_{k+1} - x_*\| &\leq (\alpha N + (1 - \alpha)M) \|x_k - x_*\| + L \|x_k - x_*\|^2 \\ &\leq \underbrace{(\alpha N + M + L \|x_k - x_*\|)}_{=:K} \|x_k - x_*\|, \end{aligned} \quad (6.48)$$

$\underbrace{\hspace{10em}}_{=:Q}$

where  $L > 0$ . We have  $K < 1$  for any  $\alpha$  that satisfies

$$0 < \alpha < \min\left(\frac{1 - M}{N}, 1\right). \quad (6.49)$$

When  $K < 1$ , for a sufficiently close initial estimate such that

$$\|x_0 - x_*\| < \frac{1 - K}{L}, \quad (6.50)$$

we have  $Q < 1$  and the iteration converges at a linear rate to  $x_*$  due to the contraction mapping theorem.  $\square$

The proposed method can be applied to any positive (semi-) definite Hessian approximations which satisfy Assumption 6.3.1 and Assumption 6.3.2. Popular Hessian approximations such as BFGS, GGN, SCP generally satisfy these conditions. It is worth noting that the simple identity approximation  $B_k = I_n$  also satisfies the mentioned conditions.

The proposed method therefore can be useful in some of the following situations. When the exact Hessian is indefinite or is too expensive to compute and the search iteration using Hessian approximations fails to converge, the proposed method can be used to enforce convergence. For large-scale cases when even Hessian approximations are computationally costly, the simple identity Hessian approximation  $B_k = I_n$  can be used together with the proposed interpolation method. This results in the same search iteration as proposed in [102, 103], although the iteration and convergence therein have been derived in a different way. Furthermore, if the cost function is just the 2-norm  $f(x) = x^\top x$  and the identity Hessian approximation is used then the proposed algorithm recovers the algorithm in our previous work [69], which has been applied successfully to the commutation problem in Chapter 4. It should be noted, however, that the identity Hessian approximation may result in a slow convergence rate, as can be seen in the example in Section 6.8.

An alternative method to guarantee local convergence for SQP with arbitrary positive (semi-) definite Hessian approximation is to scale up the Hessian approximation such that it is greater than a certain lower bound given in [108] at the

solution. Our method follows a different strategy which uses the arbitrary Hessian approximation to generate an SQP search direction, and then interpolates this search direction with a feasible search direction.

## 6.6 Global convergence

An algorithm is globally convergent if it converges to a local optimum from any arbitrary initial point. SQP algorithms with an unit stepsize, i.e.  $t_k = 1$ , generally only guarantee local convergence, which means that the initial point must be close enough to a local minimum for the iteration to converge. In order to obtain global convergence, a common approach is to use an augmented Lagrangian merit function and a line search algorithm to monitor the progress of convergence.

In this chapter we employ the strategy for global convergence described in [32]. Similar to [32], we make the following assumptions.

**Assumption 6.6.1.** For all  $k \in \mathbb{N}$ ,  $x_k$  and  $x_k + \Delta x_k$  lie in a closed, bounded region  $\Omega \subset \mathbb{R}^n$ .

**Assumption 6.6.2.** The functions  $f(x)$ ,  $g(x)$  and their first derivatives are uniformly bounded in  $\Omega$ .

The augmented Lagrangian merit function is defined as follows

$$\mathcal{L}_{\text{aug}}(x, \lambda, \rho) := f(x) + \lambda^\top g(x) + \frac{1}{2} \rho g(x)^\top g(x), \quad (6.51)$$

where  $\rho \geq 0$  is a penalty parameter. At an iteration  $k$ , the new values of  $x$  and  $\lambda$  are defined by

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + t_k \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \end{bmatrix}. \quad (6.52)$$

Here, the search direction  $\Delta x_k$  is computed from (6.10) and  $\Delta \lambda_k$  is defined as

$$\Delta \lambda_k = \lambda_{k+1}^{\text{QP}} - \lambda_k, \quad (6.53)$$

where  $\lambda_{k+1}^{\text{QP}}$  is computed from (6.9). We introduce the following notation for brevity. Let  $\phi(t, \rho)$  denote the augmented Lagrangian merit function as a function of the stepsize  $t$

$$\phi(t, \rho) := \mathcal{L}_{\text{aug}}(x + t\Delta x, \lambda + t\Delta \lambda, \rho). \quad (6.54)$$

The notation  $\phi'(t, \rho)$  denotes the derivative of  $\phi(t, \rho)$  with respect to  $t$ .

The strategy for global convergence is similar to the one presented in [32], with just a slight modification. The strategy is described by the following algorithm.

**Algorithm 6.6.3** (Global convergence algorithm). At any iteration  $k$ , the following steps are performed:

1. Calculate  $\Delta x_k$  from (6.10) and  $\Delta \lambda_k$  from (6.53).
2. Find  $\rho_k > 0$  such that

$$\phi'(0, \rho_k) \leq -\gamma \Delta x_k^\top \Delta x_k, \quad (6.55)$$

where  $\gamma > 0$ .

3. Compute the stepsize  $t_k \in (0, 1)$  via line search that satisfies the Wolfe conditions [75]

$$\phi(t_k, \rho_k) - \phi(0, \rho_k) \leq \sigma_1 t_k \phi'(0, \rho_k), \quad (6.56)$$

$$\|\phi'(t_k, \rho_k)\| \leq -\sigma_2 \phi'(0, \rho_k), \quad (6.57)$$

where  $0 < \sigma_1 < \sigma_2 < 1$ .

4. Update  $x$  and  $\lambda$  using (6.52).

Condition (6.57) is typically omitted as it does not impede convergence of the algorithm in practice and this helps to reduce the computational load [44, 75, 102]. A stepsize  $t_k \in (0, 1)$  that satisfies condition (6.56) can be found using backtracking line search as follows [75].

**Algorithm 6.6.4** (Backtracking line search). Choose  $\sigma_1 \in \mathbb{R}_{(0,1)}$  and  $\varsigma \in \mathbb{R}_{(0,1)}$ .

Set  $t_k := 1$ ;

**repeat** until  $\phi(t_k, \rho_k) - \phi(0, \rho_k) \leq \sigma_1 t_k \phi'(0, \rho_k)$

$t_k := \varsigma t_k$ ;

**end (repeat)**

Compared to the strategy presented in [32], there is a slight modification. In [32], the penalty parameter  $\rho_k$  is determined by the following condition

$$\phi'(0, \rho_k) \leq -\frac{1}{2} \Delta x_k^\top B_k \Delta x_k. \quad (6.58)$$

This condition aims to guarantee that  $\phi'(0, \rho_k)$  is negative, which implies that  $\begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \end{bmatrix}$  is the descent direction of  $\phi(t, \rho_k)$  at  $(0, \rho_k)$  and hence there exists a stepsize  $t_k \in \mathbb{R}_{(0,1)}$  such that condition (6.56) is satisfied. However, in contrast to [32] in which  $B_k$  is assumed to be positive definite, we only assume positive (semi-) definiteness of  $B_k$  in Assumption 6.3.1. Consequently, the condition needs to be modified. We therefore employ condition (6.55), which guarantees that  $\phi'(0, \rho_k)$  is negative, for determining the penalty parameter  $\rho_k$ . In the following lemma we will prove the existence of a non-negative penalty parameter such that condition (6.55) holds.

**Lemma 6.6.5.** For any iteration  $k$ , there exists  $\hat{\rho}_k \geq 0$  such that

$$\phi'(0, \rho) \leq -\gamma \Delta x_k^\top \Delta x_k,$$

for all  $\rho \geq \hat{\rho}_k$ .

*Proof.* The gradient of the merit function with respect to  $x$  and  $\lambda$  is given by

$$\nabla \mathcal{L}_{\text{aug}}(x, \lambda) = \begin{bmatrix} \nabla f(x) + \nabla g(x)\lambda + \rho \nabla g(x)g(x) \\ g(x) \end{bmatrix}. \quad (6.59)$$

It follows that  $\phi'(0, \rho)$  at an iteration  $k$  is given by

$$\begin{aligned} \phi'(0, \rho) &= \nabla \mathcal{L}_{\text{aug}}(x_k, \lambda_k)^\top \begin{bmatrix} \Delta x_k \\ \Delta \lambda_k \end{bmatrix} \\ &= \nabla f(x_k)^\top \Delta x_k + \Delta x_k^\top \nabla g(x_k) \lambda_k + \rho g(x_k)^\top \nabla g(x_k)^\top \Delta x_k + g(x_k)^\top \Delta \lambda_k \\ &= \nabla f(x_k)^\top \Delta x_k - g(x_k)^\top \lambda_k - \rho g(x_k)^\top g(x_k) + g(x_k)^\top \Delta \lambda_k. \end{aligned} \quad (6.60)$$

Note that in the last equality we use the following relation which follows from (6.14)

$$g(x_k) = -\nabla g(x_k)^\top \Delta x_k. \quad (6.61)$$

Let us calculate the first term on the right hand side of (6.60). It follows from (6.7a) that

$$\nabla f(x_k) = -B_k \Delta x_k^{\text{QP}} - \nabla g(x_k) \lambda_{k+1}. \quad (6.62)$$

From (6.10) and (6.12) we obtain

$$\begin{aligned} \Delta x_k^{\text{QP}} &= \frac{1}{\alpha} \Delta x_k - \frac{1-\alpha}{\alpha} \Delta x_k^f \\ &= \frac{1}{\alpha} \Delta x_k + \frac{1-\alpha}{\alpha} \nabla g(x_k)^\dagger g(x_k) \\ &= \frac{1}{\alpha} \Delta x_k - \frac{1-\alpha}{\alpha} \nabla g(x_k)^\dagger \nabla g(x_k) \Delta x_k, \end{aligned} \quad (6.63)$$

where the last equality follows from (6.61). Substituting (6.63) into (6.62) yields

$$\nabla f(x_k) = -\frac{1}{\alpha} B_k \Delta x_k + \frac{1-\alpha}{\alpha} B_k \nabla g(x_k)^\dagger \nabla g(x_k) \Delta x_k - \nabla g(x_k) \lambda_{k+1}. \quad (6.64)$$

It follows that

$$\begin{aligned} \nabla f(x_k)^\top \Delta x_k &= -\frac{1}{\alpha} \Delta x_k^\top B_k \Delta x_k + \frac{1-\alpha}{\alpha} \Delta x_k^\top B_k \nabla g(x_k)^\dagger \nabla g(x_k) \Delta x_k \\ &\quad - \Delta x_k^\top \nabla g(x_k) \lambda_{k+1} \\ &= -\frac{1}{\alpha} \Delta x_k^\top B_k \Delta x_k + \frac{1-\alpha}{\alpha} \Delta x_k^\top B_k \nabla g(x_k)^\dagger \nabla g(x_k) \Delta x_k \\ &\quad + g(x_k)^\top \lambda_{k+1}. \end{aligned} \quad (6.65)$$

Substituting (6.65) into (6.60) results in

$$\begin{aligned} \phi'(0, \rho) &= -\frac{1}{\alpha} \Delta x_k^\top B_k \Delta x_k + \frac{1-\alpha}{\alpha} \Delta x_k^\top B_k \nabla g(x_k)^\dagger \nabla g(x_k) \Delta x_k \\ &\quad + 2g(x_k)^\top \Delta \lambda_k - \rho g(x_k)^\top g(x_k). \end{aligned} \quad (6.66)$$

Condition (6.55) is therefore equivalent to

$$-\frac{1}{2}\Delta x_k^\top V_k \Delta x_k + 2g(x_k)^\top \Delta \lambda_k \leq \rho g(x_k)^\top g(x_k), \quad (6.67)$$

where

$$V_k = B_k \left( \frac{1}{\alpha} I_n - \frac{1-\alpha}{\alpha} \nabla g(x_k)^\top \nabla g(x_k) \right) - \gamma I_n. \quad (6.68)$$

Due to Assumption 6.6.1 and Assumption 6.6.2, it can be verified that the left hand side of (6.67) is bounded. It is obvious that the lower bound  $\hat{\rho}_k$  can be taken as zero if

$$-\frac{1}{2}\Delta x_k^\top V_k \Delta x_k + 2g(x_k)^\top \Delta \lambda_k \leq 0. \quad (6.69)$$

When (6.69) does not hold, the following value of  $\hat{\rho}_k$  guarantees that (6.67) holds for all  $\rho \geq \hat{\rho}_k$

$$\hat{\rho}_k = \frac{\left\| -\frac{1}{2}\Delta x_k^\top V_k \Delta x_k + 2g(x_k)^\top \Delta \lambda_k \right\|}{\left\| g(x_k)^\top g(x_k) \right\|}. \quad (6.70)$$

In conclusion,  $\hat{\rho}_k$  can be taken as in (6.70) if (6.69) does not hold, and as zero otherwise.  $\square$

Similar to [32], the strategy for determining  $\rho$  is to retain the value at the previous iteration, and only increase if necessary to satisfy (6.55). The initial value  $\rho_0$  is chosen equal to zero. At an iteration  $k$ , the penalty parameter  $\rho_k$  is defined as follows

$$\rho_k := \begin{cases} \rho_{k-1} & \text{if } \phi'(0, \rho_{k-1}) \leq -\frac{1}{2}\Delta x_k^\top V_k \Delta x_k, \\ \max(\hat{\rho}_k, 2\rho_{k-1}) & \text{otherwise,} \end{cases} \quad (6.71)$$

where  $\hat{\rho}_k$  is defined as described in the proof of Lemma 6.6.5

$$\hat{\rho}_k := \begin{cases} \frac{\left\| -\frac{1}{2}\Delta x_k^\top V_k \Delta x_k + 2g(x_k)^\top \Delta \lambda_k \right\|}{\left\| g(x_k)^\top g(x_k) \right\|} & \text{if (6.69) does not hold,} \\ 0 & \text{otherwise.} \end{cases} \quad (6.72)$$

Due to this strategy for determining  $\rho$ , there are two possible cases, namely the bounded case and the unbounded case. In the bounded case, the value of  $\rho$  eventually becomes fixed at a certain value, while in the unbounded case,  $\rho$  will be tending to infinity. The next lemma is crucial for the proof of convergence in the unbounded case. The proof of the lemma can be found in [32].

**Lemma 6.6.6.** [32, Lemma 4.6] *Let  $k_l$ ,  $l = 0, 1, \dots$ , denote the indices of the subsequence of iterations when the penalty parameter increases. There exists a bounded constant  $M$  such that, for all  $l \in \mathbb{N}$ ,*

$$\rho_{k_l} \sum_{k=k_l}^{k_{l+1}-1} \|t_k \Delta x_k\|^2 < M. \quad (6.73)$$

The next lemma proves the existence of a lower bound on the stepsize defined by the line search in Step 3 of Algorithm 6.6.3.

**Lemma 6.6.7.** [32, Lemma 4.9] *The line search in Step 3 of Algorithm 6.6.3 defines a stepsize  $t_k$  such that*

$$\phi(t_k, \rho_k) - \phi(0, \rho_k) \leq \sigma_1 t_k \phi'(0, \rho_k)$$

and  $t_k \geq \bar{t}$ , where  $0 < \sigma_1 < 1$ , and  $\bar{t}$  is bounded away from zero and independent of  $k$ .

We are now ready to prove the global convergence property of Algorithm 6.6.3. The proof is similar to [32, Theorem 4.1].

**Theorem 6.6.8.** *Under Assumptions 6.3.1, 6.3.2, 6.3.3, 6.6.1 and 6.6.2, Algorithm 6.6.3 has the property that*

$$\lim_{k \rightarrow \infty} \|\Delta x_k\| = 0. \quad (6.74)$$

*Proof.* If  $\|\Delta x_k\| = 0$  for any finite  $k$ , the algorithm terminates and the theorem is true. Therefore, we assume that  $\|\Delta x_k\| \neq 0$  for any finite  $k$ .

Let us first consider the unbounded case when there is no upper bound on the penalty parameter  $\rho$ . While  $\rho$  tends to infinity, since there is a lower bound on  $t_k$  as stated in Lemma 6.6.7, the relation (6.73) implies that for any  $\delta > 0$ , there exists an iteration number  $\bar{k} \in \mathbb{N}$  such that  $\|\Delta x_k\| \leq \delta$  for all  $k \geq \bar{k}$ . This implies that (6.74) holds.

In the bounded case, there exists a value  $\bar{\rho}$  and an index  $\bar{k}$  such that  $\rho_k = \bar{\rho}$  for all  $k \geq \bar{k}$ . The proof is by contradiction. We assume that  $\|\Delta x_k\|$  does not tend to zero as  $k \rightarrow \infty$ . This implies that there exists  $\epsilon > 0$  and an index  $\hat{k} \geq \bar{k}$  such that  $\|\Delta x_k\| \geq \epsilon$  for all  $k \geq \hat{k}$ . Due to (6.55), (6.56) and Lemma 6.6.7, we therefore have

$$\phi(t_k) - \phi(0) \leq \sigma_1 t_k \phi'(0) \leq -\frac{1}{2} \rho \bar{t} \gamma \epsilon^2 < 0. \quad (6.75)$$

Therefore, every subsequent iteration must yield a strict decrease in the merit function. This implies that the merit function with  $\rho = \bar{\rho}$  must be unbounded below. This leads to a contradiction, due to Assumption 6.6.1 and Assumption 6.6.2. Therefore, (6.74) must hold.  $\square$

In summary, to guarantee global convergence, we employ the strategy from [32] with a slight modification in the determination of the penalty parameter of the merit function, in order to account for the positive (semi-) definiteness of the Hessian approximation.

## 6.7 Treatment of inequality constraints

The basic SQP method can be extended to solve the general nonlinear programming problem 6.2.1 which contains inequality constraints. The two main approaches

to handle inequality constraints are inequality-constrained QP (IQP) and equality-constrained QP (EQP) [37, 75].

In the IQP approach, the inequality constraints are also linearized, resulting in the following QP subproblem

$$\begin{aligned} \min_{\Delta x_k \in \mathbb{R}^n} \quad & \frac{1}{2} \Delta x_k^\top B_k \Delta x_k + \nabla f(x_k)^\top \Delta x_k \\ \text{s.t.} \quad & \nabla g(x_k)^\top \Delta x_k + g(x_k) = 0, \\ & \nabla h(x_k)^\top \Delta x_k + h(x_k) \leq 0. \end{aligned} \tag{6.76}$$

The QP subproblem (6.76) can be solved by using methods for solving QP such as interior-point method or active-set method. The set of active inequality constraints of the QP subproblem is the estimate of the active set of the nonlinear program problem at the solution. If the optimal active set is identified correctly in a finite number of iterations then the algorithm converges rapidly. On the other hand, the IQP approach can be computationally expensive due to the cost of solving inequality constrained QP subproblems.

In the EQP approach, a subset of inequality constraints is selected at each iteration to form the working set  $\mathcal{W}_k$ . The inequality constraints in the working set are treated as equality constraints while the other inequality constraints are ignored. The resulting QP subproblem which contains only equality constraints is given by

$$\begin{aligned} \min_{\Delta x_k \in \mathbb{R}^n} \quad & \frac{1}{2} \Delta x_k^\top B_k \Delta x_k + \nabla f(x_k)^\top \Delta x_k \\ \text{s.t.} \quad & \nabla g(x_k)^\top \Delta x_k + g(x_k) = 0, \\ & \nabla h_{[i]}(x_k)^\top \Delta x_k + h_{[i]}(x_k) = 0, \quad i \in \mathcal{W}_k. \end{aligned} \tag{6.77}$$

The solution of this equality constrained QP can be obtained by solving a system of linear equations similar to (6.8). The working set is updated at every iteration by adding and deleting constraints based on Lagrange multiplier estimates, or by solving an auxiliary subproblem [75]. The advantage of the EQP approach is that the cost to solve an equality constrained QP subproblem is less expensive than the cost to solve an inequality constrained QP subproblem, especially when considering large-scale problems.

The EQP approach will be employed for solving the NMPC of linear motor problem in the next section.

## 6.8 Numerical examples

In this section, we present two numerical examples to verify the performance of the new optimization algorithm. The first one is a simple constrained nonlinear least squares problem. The second one is the NMPC problem of linear motors which is described earlier in Chapter 5.



### 6.8.1 A constrained nonlinear least squares problem

Let us consider the following nonlinear optimization problem

**Problem 6.8.1.**

$$\begin{aligned} \min_{x \in \mathbb{R}^5} \quad & (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6 \\ \text{s. t.} \quad & x_1^2 x_4 + \sin(x_4 - x_5) - 6\sqrt{2} = 0, \\ & x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0. \end{aligned}$$

The initial estimate is  $x_0 = [2, 2, 2, 2, 2]^\top$  and  $\lambda_0 = [0, 0]^\top$ . This is a nonlinear equality constrained least squares problem with nonzero residual. In nonlinear constrained least squares problems, the cost function has the least squares form

$$f(x) = \frac{1}{2} \|R(x)\|^2,$$

where  $R: \mathbb{R}^n \rightarrow \mathbb{R}^p$ . A popular Hessian approximation for this type of problems is the GGN approximation [13, 14, 22]

$$B^{\text{GGN}}(x) = \nabla R(x)^\top \nabla R(x).$$

It is well known that the SQP method with GGN Hessian approximation, also known as the GGN method, converges locally if the residual function  $R(x)$  is small at the solution [21].

In this example, we test the exact Hessian SQP method (SQP-EH), the GGN method (SQP-GGN), the new interpolated method with GGN Hessian approximation (iSQP-GGN), and the new interpolated method with identity Hessian approximation  $B_k = I_n$  (iSQP-I). The optimization algorithms are programmed in Matlab and tested on a 3.6 GHz computer. The measure of convergence is the 2-norm of the KKT matrix  $\nabla \mathcal{L}(x_k, \lambda_k)$ , which is called the KKT residual. The optimization algorithms terminate when the KKT residual is less than  $10^{-6}$ .

The test results are as follows. The SQP-EH method converges quadratically as expected. The SQP-GGN method does not converge. The iSQP-GGN method converges linearly. This demonstrates that the proposed interpolation scheme can enforce convergence for the SQP method with GGN Hessian approximation. The iSQP-I method also converges linearly, but at a slower rate. This is expected since the GGN approximation is a better approximation than the identity matrix. The convergence rate of the methods are shown in Figure 6.1. The interpolation coefficients  $\alpha$  shown here are among the ones that result in fastest convergence rates for each method. The SQP-EH method, the proposed iSQP-GGN and iSQP-I methods converge to the same solution  $x^* = [2.0874, 2.0980, 1.2487, 1.7347, 0.5492]^\top$ .

The number of iterations and computation time are summarized in Table 6.1. It is observed that the SQP-EH method requires the least number of iterations, as it converges quadratically. The iSQP method needs a higher number of iterations, but the total computation time is lower, since it requires less computation per iteration.

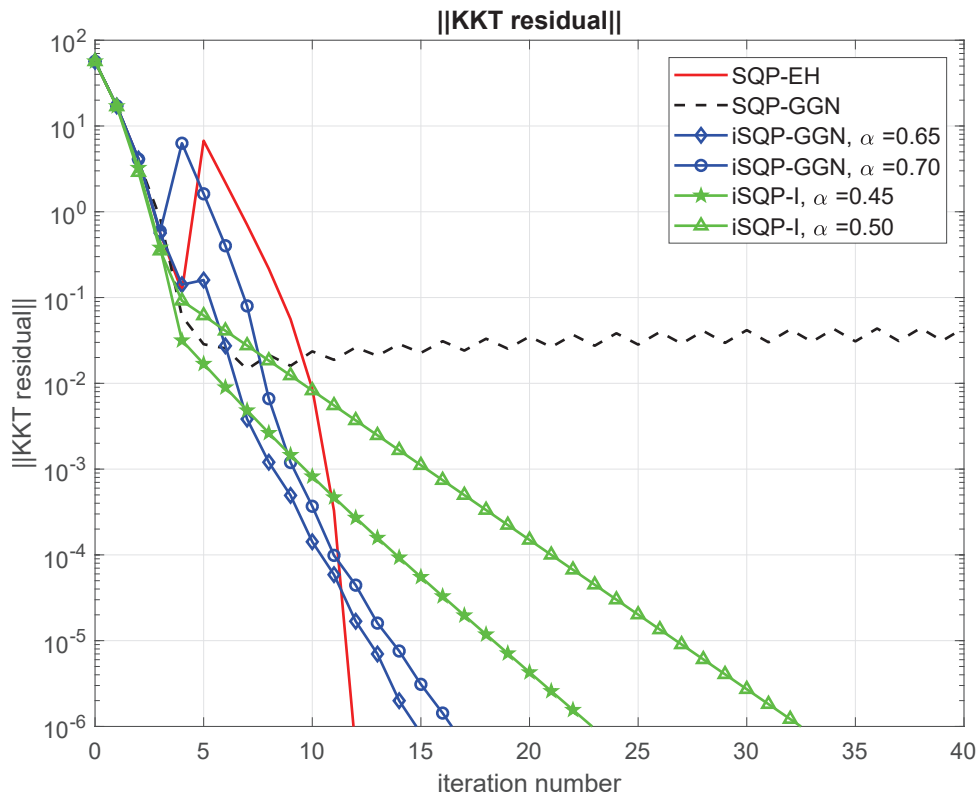


Figure 6.1: Convergence rate illustration.

Table 6.1: Computation time

Method	Number of iterations	Computation time [ms]
SQP-EH	13	19.2
iSQP-GGN, $\alpha = 0.65$	16	17.2
iSQP-GGN, $\alpha = 0.70$	18	17.5
iSQP-I, $\alpha = 0.45$	24	16.7
iSQP-I, $\alpha = 0.50$	34	17.4

This demonstrates that with a suitable choice of the interpolation coefficient  $\alpha$ , the new method can be more efficient than the SQP-EH method, especially in large-scale cases when computation of the exact Hessian can be very expensive.

The above initial estimate is close to the solution. In order to test the global convergence strategy described in Section 6.6, we select a further away initial estimate,

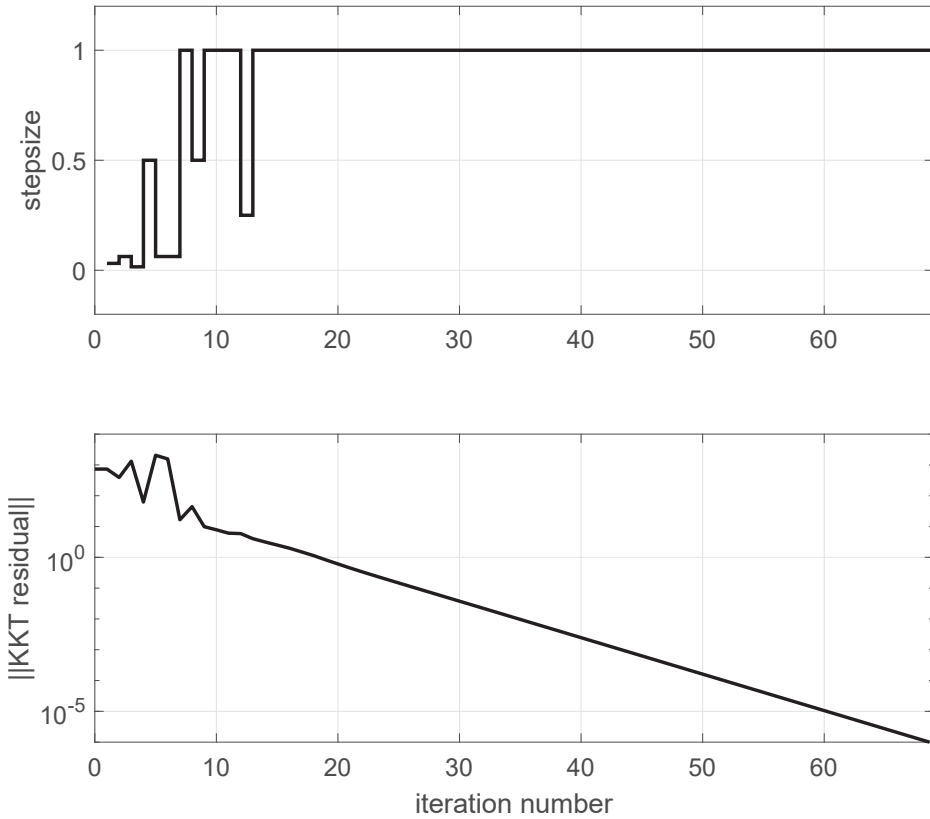


Figure 6.2: Stepsize and KKT residual of the iSQP-I method with  $\alpha = 0.25$ .

which is  $x_0 = [-2, -2, -2, -2, -2]^\top$  and  $\lambda_0 = [0, 0]^\top$ . The parameters for the global convergence algorithm are  $\sigma_1 = 0.1$ ,  $\zeta = 0.5$  and  $\gamma = 0.0001$ . With the new initial estimate, the iSQP-I method does not converge if we disable the global convergence strategy and keep the stepsize equal to 1. When the global convergence strategy is enabled then the method converges. The values of the stepsize and the KKT residual of the iSQP-I method with  $\alpha = 0.25$  are shown in Figure 6.2. It can be seen that the stepsize is generally smaller than 1 when the estimate is far from the solution, and recover to 1 when the estimate comes close to the solution.

### 6.8.2 Nonlinear model predictive control of linear motors

In this example we apply the new optimization algorithm to the NMPC problem of linear motors. As described in Chapter 5, the NMPC problem is formulated as follows

**Problem 6.8.2** (NMPC).

$$\begin{aligned} \min_{\substack{\mathbf{x}_1, \dots, \mathbf{x}_N, \\ u_0, u_1, \dots, u_{N-1}}} & (\mathbf{x}_N - \mathbf{x}_N^*)^\top P (\mathbf{x}_N - \mathbf{x}_N^*) + \sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{x}_i^*)^\top Q (\mathbf{x}_i - \mathbf{x}_i^*) \\ & + \sum_{i=0}^{N-1} (u_i - u_i^r)^\top (u_i - u_i^r) \end{aligned}$$

subject to

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + B\Psi_x(x_i, u_i), & \forall i = 0, \dots, N-1, \\ \Psi_z(x_i, u_i) &= 0, & \forall i = 0, \dots, N-1, \\ \Psi_y(x_i, u_i) &= 0, & \forall i = 0, \dots, N-1, \\ w_{\min}^x &\leq w_i^x \leq w_{\max}^x, & \forall i = 0, \dots, N-1, \\ u_{\min} &\leq u_i \leq u_{\max}, & \forall i = 0, \dots, N-1. \end{aligned}$$

In this example we use the same linear motor model as described in Section 5.4. The linear motor consists of 2 coil sets, which gives us 4 independent inputs. The mass of the translator is  $m = 20$  kg. The driving force limit is  $\pm 3000$  N. The current limit is  $\pm 25$  A. The sampling rate of the control loop is 1 kHz. The weighting matrix is  $Q = \text{diag}(10^8, 1)$ . The terminal cost  $P$  is calculated as the solution of the discrete-time Lyapunov equation of the linear dynamics. The step reference is used. The prediction horizon is  $N = 8$ . Consequently, Problem 6.8.2 has 48 optimization variables, 32 equality constraints and 80 inequality constraints. The solution of the MPC problem at a time sample is used as the "warm-start" initial point for the MPC problem next time sample. We employ an active set strategy to handle inequality constraints. The simulation is performed in Matlab on a 3.6 GHz computer.

Similar to the previous example, we test the SQP-EH, SQP-GGN and iSQP-I. In addition, we also test the commercial solvers Knitro and IPOPT for comparison [16, 109]. The NMPC simulation results are the same for all the tested methods. The results are shown in Figure 6.3. As discussed in Chapter 5, the NMPC controller is able to compensate for parasitic forces and torques while guaranteeing constraint satisfaction.

Table 6.2 summarizes the computation time of each method. It is observed that the SQP-GGN algorithm converges without the developed interpolation scheme and it is the fastest one. The SQP-EH method needs the highest computation time. Our iSQP-I algorithm is faster than the solver Knitro and is slightly slower than the solver IPOPT. It should be noted that our algorithm is programmed in Matlab code, which is generally slower than C code. By converting Matlab code to C code, the algorithm is expected to run faster and thereby achieving lower computation time.

## 6.9 Conclusions

In this chapter we have developed a new computationally efficient optimization method for solving nonlinear programming problems. The new method is a variant

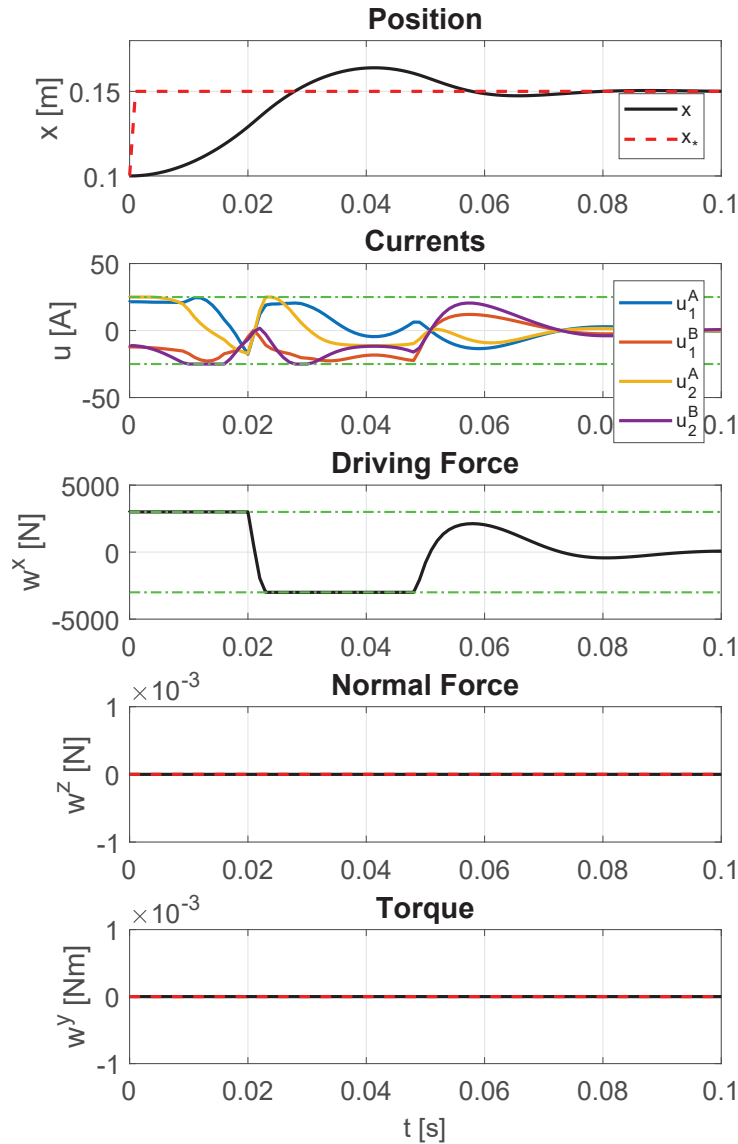


Figure 6.3: NMPC simulation results.

of SQP in which an arbitrary positive (semi-) definite Hessian approximation can be used. In this method, the search direction is taken as the interpolation between the search direction provided by solving the QP subproblem and a feasible search direction. It is proved that the algorithm converges locally at linear rate to a local optimum of the nonlinear programming problem. Furthermore, a strategy for global convergence is employed to guarantee convergence from any initial estimate. The

Table 6.2: NMPC computation times

Method	Average time [ms]	Best [ms]	Worst [ms]
SQP-EH	2118	985	54176
SQP-GGN	90	38	230
iSQP-I, $\alpha = 0.6$	99	63	241
Knitro	125	50	377
IPOPT	97	70	148

effectiveness of the method has been illustrated in two numerical examples. The method will be used to solve the optimal commutation problem in an experimental linear motor setup in the next chapter.

The new algorithm is implemented in Matlab code, which is generally slower than C code. By converting to C code, the computation time is expected to reduce significantly. We will implement this in future work. Furthermore, it is of interest to exploit the sparsity of the NMPC of linear motor problem to write a dedicated fast solver based on the algorithm developed in this chapter. Another interesting topic for future work is how to compute the interpolation coefficient such that the fastest speed of convergence is achieved.

# Chapter 7

## Experimental validation

### 7.1 Introduction

The modeling and control methods developed in the previous chapters have been demonstrated to deliver good performance in simulation. Now it is of our interest to validate the methods in a real CLM. We have designed and constructed an experimental CLM setup for this purpose. The setup consists of a real industrial linear motor, power amplifiers and encoder provided by ASML. Additional force sensors are added in order to measure the forces and torques produced by the motor in driving and non-driving directions. The setup is controlled by a dSPACE MicroLab-Box development system. Details of the design and construction of the experimental setup will be presented in this chapter.

The data-driven modeling method and the optimal commutation method are validated on the experimental setup. First, the measurement data collected from the experimental setup is used to identify a model of the linear motor using the data-driven modeling method presented in Chapter 3. The optimal commutation method presented in Chapter 4 is then validated on the setup. The experimental results will be presented and analyzed.

This chapter is organized as follows. The design and specifications of the setup are presented in Section 7.2. Section 7.3 describes the identification experiments and presents the identification results. Section 7.4 presents the results of the commutation experiments. The conclusions are summarized in Section 7.5.

### 7.2 Experimental setup

#### 7.2.1 System overview

The overview of the experimental setup is depicted in Figure 7.1. The linear motor contains three coil sets. Each set consists of three coils connected in star-configuration and is powered by a three-phase power amplifiers. The control soft-

ware is programmed on a dSPACE MicroLabBox development system. At every sampling time instance, the dSPACE MicroLabBox reads the measurement signals from the encoder, the power amplifiers and the force sensors, and then implements the control and commutation algorithms to calculate current setpoints and send them to the power amplifiers. The dSPACE MicroLabBox communicates with the power amplifiers through serial communication. The real-time applications can be accessed during run-time on a computer using the dSPACE ControlDesk software. A picture of the complete setup is shown in Figure 7.2.

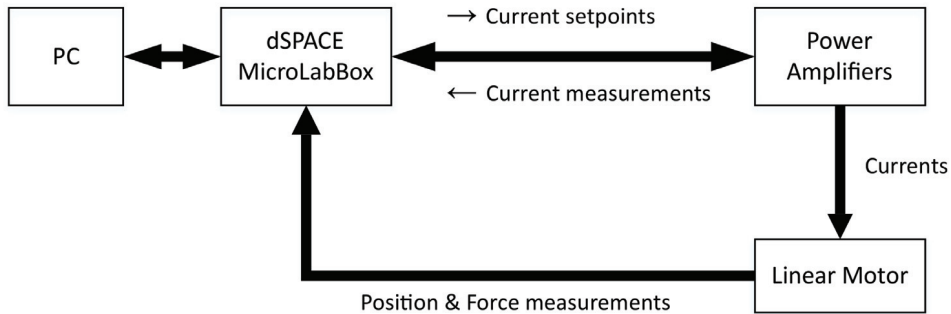


Figure 7.1: Overview of the experimental setup.

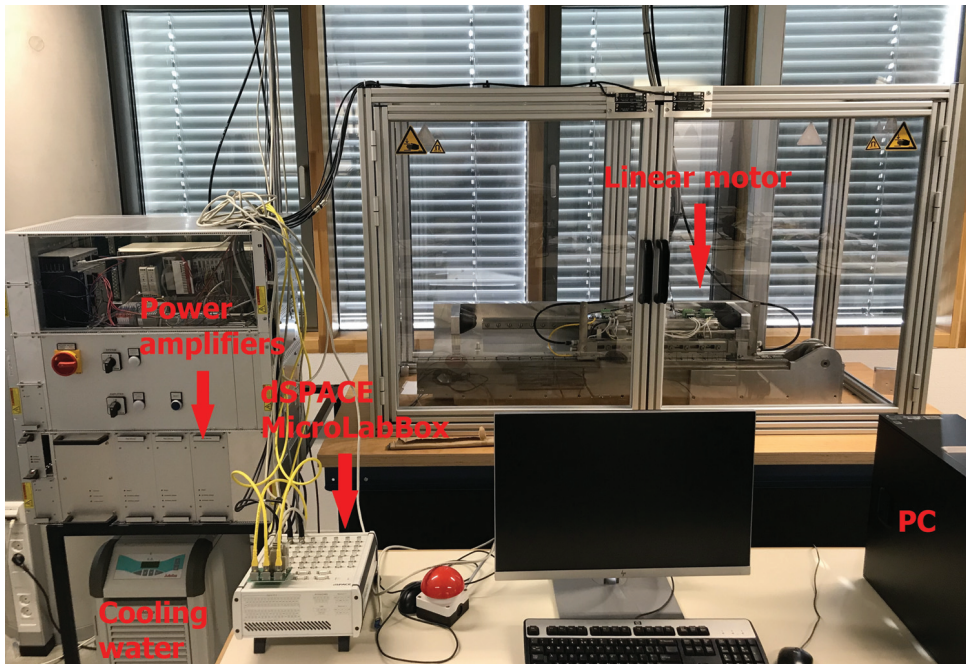


Figure 7.2: The experimental setup.



### 7.2.2 Mechanical design

The mechanical design of the setup is shown in Figure 7.3. The coil array is placed in the air gap between the two magnet arrays and is guided by a mechanical bearing. The force sensor in the  $x$ -direction is placed in between the coil array and the bearing. Two other force sensors are added to measure the force in the  $z$ -direction. An additional mass up to 50 kg can be connected to the coil via a rope and pulley system to generate a load force. The gravity is in the  $y$ -direction. Pictures of the real linear motor setup construction are shown in Figure 7.4.

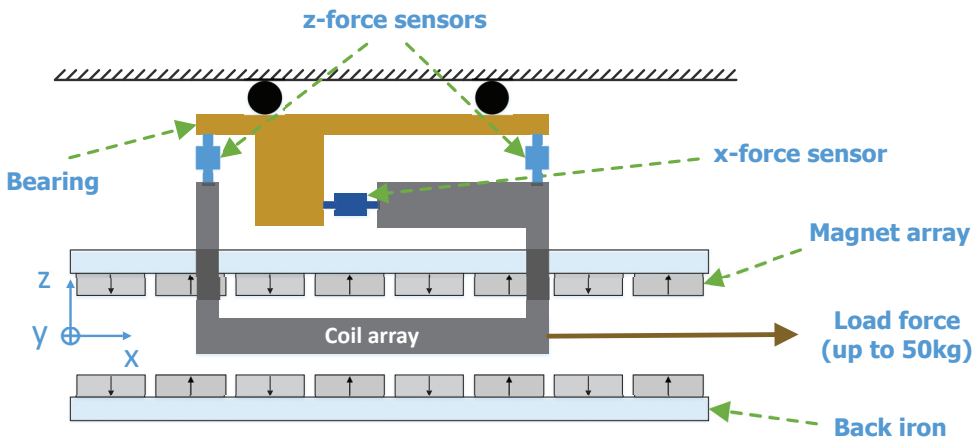
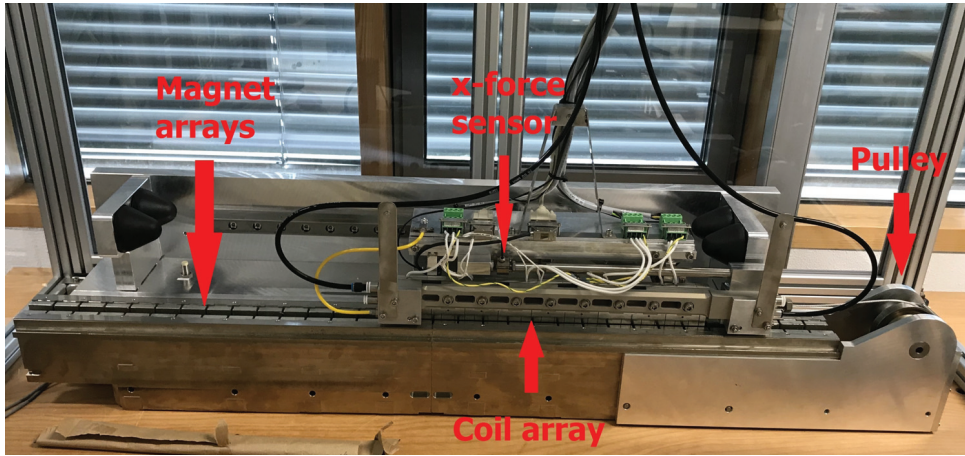


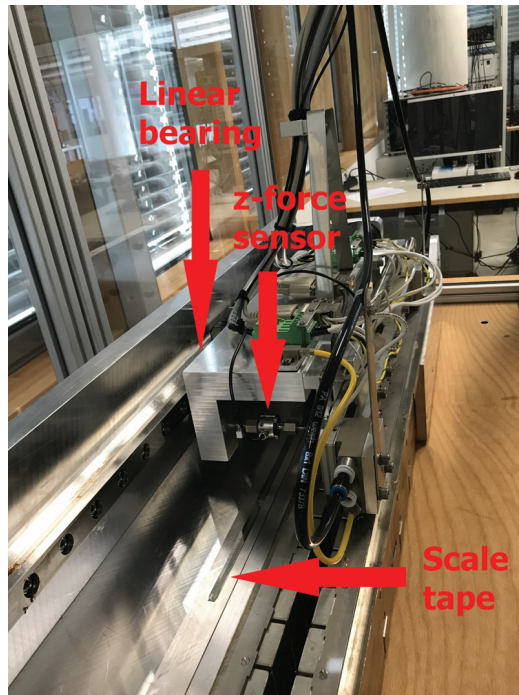
Figure 7.3: Mechanical design of the experimental setup.

In order to study the parasitic effects in the non-driving directions, the following functionalities are made available in our experimental setup:

- The  $z$ -position of the coil array can be adjusted. This functionality enables us to shift the coil array out of the center of the air gap, in order to study the effect of the misalignment of the coil in the  $z$ -direction. In this chapter, we perform experiments with the coil array shifted 0.7 mm out of the center of the air gap.
- Force sensors are added in the driving and non-driving directions. These force sensors are necessary to identify the model of the motor, especially in the non-driving directions, and to verify whether the developed commutation method helps to compensate for the parasitic forces and torques generated by the motor.
- A load force can be added by connecting masses to the coil array via a rope and pulley system. The added load force allows the linear motor to constantly generate high force without resulting in high acceleration and high speed, which is not safe for the limited stroke length of 0.4 m. As a result, we are



(a) Front view.



(b) Side view.

Figure 7.4: The linear motor setup construction.

able to perform experiments with high force and high current at low speed in which the parasitic effects are amplified. This allows us to easily observe the advantage of the new optimal commutation method compared to the classical commutation method.

It is noted that the mechanical bearing introduces parasitic effects which are larger than the parasitic effects caused by manufacturing tolerances discussed in Chapter 2. This causes difficulty for identification of the linear motor model in the driving direction. To deal with this problem, we have placed a force sensor in between the coil array and the bearing as illustrated in Figure 7.3. With this sensor we can measure any force created by the bearing like friction and inertial forces. The friction is therefore known and can be taken into account in the identification procedure. Furthermore, the 50 kg mass allows us to perform experiments with high force and high current at low speed. As the effects of the bearing on the dynamics is less significant at low speed, we are able to effectively apply the identification and commutation methods developed in the previous chapters. In the non-driving direction, the effect of the bearing is small. The developed identification and commutation methods can therefore be applied.

### 7.2.3 Hardware

In this section we present the specifications of the main components in the setup.

#### Linear motor

The setup consists of a real industrial coreless linear motor provided by ASML. The motor consists of three coil sets of three coils each. The length of the stroke is 0.4 m. The motor can reach a peak force of 5025 N. The design parameters of the motor, which are provided by the manufacturer, are summarized in Table 7.1.

Table 7.1: Parameters of the experimental linear motor.

<i>Parameter</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
Motor force constant	$k$	201	N/A
Magnet pole pitch	$\tau_p$	39	mm
Coil pitch	$\tau_c$	52	mm
Air gap	$\delta$	3.45	mm
Magnet width	$w_m$	36	mm
Magnet height	$h_m$	13.6	mm
Coil width	$w_c$	52	mm
Coil height	$h_c$	9	mm
Coil bundle width	$w_b$	22	mm
Remanent magnetization	$B_{\text{rem}}$	1.35	T
Magnet relative permeability	$\mu_r$	1.033	

### Power amplifiers

The setup consists of three identical three-phase current amplifiers also provided by ASML. The amplifier provides a three-phase output current according to the current setpoints it receives from the controller, and sends back the current measurements to the controller. The phase current range is  $\pm 25$  A peak. The phase-to-phase voltage range is  $\pm 550$  V.

### Encoder

For position measurement, we use an incremental linear encoder Numerik Jena LIA20 together with a SINGLEFLEX scale tape. The smallest measuring step is  $1 \mu\text{m}$ .

### Force sensors

For force measurement, we use three force sensors from HBM. One sensor HBM U9C/2kN is used to measure the force in the driving direction. Two sensors HBM U9C/100N are used to measure the force in the  $z$ -direction and the torque around the  $y$ -direction. The sensors have the accuracy of 0.2%. The measured signals are amplified by a signal amplifier HBM PX455 before being sent to the controller.

### dSPACE MicroLabBox

The control software is implemented on a dSPACE MicroLabBox development system. The MicroLabBox consists of a 2 GHz dual-core real-time processor and a user-programmable FPGA. It has 1 GB DRAM 128 MB flash memory. The control algorithm is programmed on the real-time processor. The MicroLabBox communicates with the power amplifiers through bidirectional differential digital I/O channels using a serial communication protocol. The serial communication protocol is programmed on the FPGA. The position measurements are received through an encoder sensor input. The force measurements are received through 14-bit analog input channels, with voltage range of  $\pm 10$  V.

## 7.2.4 Software

The control software is implemented in Matlab/Simulink. Simulink Coder generates C code from the Simulink block diagrams which can be downloaded to the MicroLabBox. The dSPACE Real-Time Interface (RTI) sets up connections between the Simulink model and MicroLabBox Input/Output (I/O). The dSPACE ControlDesk serves as an user interface which gives us access to the real-time applications during run-time.

## 7.3 Data-driven modeling experiments

### 7.3.1 Friction identification

The experimental setup has a force sensor placed in between the bearing and the translator of the motor as shown in Figure 7.3. The force measured by this force sensor is given by

$$F_{\text{measured}}^x = m_{\text{bearing}}a + F_{\text{friction}}. \quad (7.1)$$

where  $m_{\text{bearing}}$  is the mass of the bearing and  $a$  is the acceleration. Therefore, we can measure the friction by letting the motor move at a constant velocity, i.e.  $a = 0$ . We approximate the friction by the following model

$$F_{\text{friction}} = F_{\text{Coulomb}}(x) + F_{\text{viscous}}(v). \quad (7.2)$$

The position-dependent Coulomb friction is approximated by a Fourier series

$$F_{\text{Coulomb}}(x) = \sum_{i=1}^N \left( f_i \cos\left(\frac{2\pi}{L}x\right) + g_i \sin\left(\frac{2\pi}{L}x\right) \right) + h, \quad (7.3)$$

where  $L$  is the length of the stroke,  $f_i$ ,  $g_i$  and  $h$  are the Fourier coefficients. The viscous friction is linear with the velocity  $v$

$$F_{\text{viscous}}(v) = k_v v, \quad (7.4)$$

where  $k_v$  is the viscous friction coefficient.

We perform experiments in which the motor moves from one end to the other at a constant velocity. Several experiments with different values of the velocity are performed and the force sensor measurement data are collected. The parameters of the friction model are obtained by fitting the model to the measurement data. The identified Coulomb friction and viscous friction are plotted in Figure 7.5. The identified viscous friction coefficient  $k_v$  is 28.26 Ns/m.

### 7.3.2 Identification in the driving direction

#### Simultaneous identification of the static nonlinearity and the linear dynamical system

We first test the data-driven modeling method for the driving direction introduced in Section 3.2. The method aims to identify the model of the static nonlinearity and the linear dynamics simultaneously from the currents and position measurements.

The model structure of the linear motor in the driving direction is depicted in Figure 7.6. The linear dynamics part is modeled as a second-order discrete-time transfer function

$$G(q) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}. \quad (7.5)$$

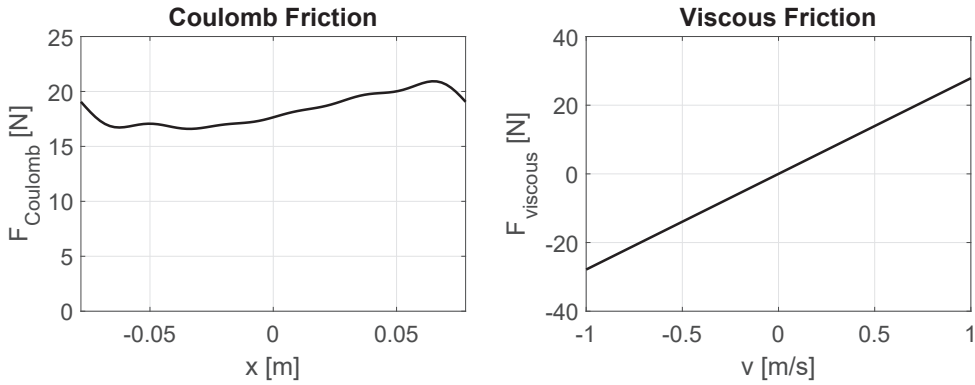


Figure 7.5: Identified Coulomb friction and viscous friction.

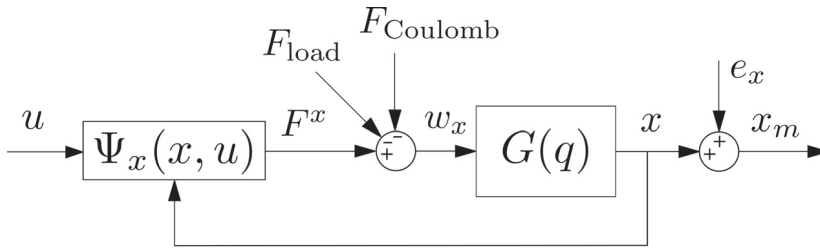


Figure 7.6: Model structure of the linear motor in the driving direction.

In order to keep the number of parameters low, only one coil set is used in the experiments. Therefore, the number of independent inputs is 2 due to the star connection. In the driving direction there is only Lorentz force which is linear with the currents. The driving force is given by

$$F^x = \begin{bmatrix} K_1^x(x) & K_2^x(x) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (7.6)$$

where  $K_l^x(x)$ ,  $l = 1, 2$ , are the position-dependent force functions. Only one harmonic component with frequency  $\frac{\pi}{\tau_p}$  is used in the Fourier model of the static nonlinearity. The force functions are modeled as

$$K_l^x(x) = c_l \cos\left(\frac{\pi}{\tau_p} x\right) + d_l \sin\left(\frac{\pi}{\tau_p} x\right), \quad l = 1, 2. \quad (7.7)$$

The total input force acting on the linear dynamics is given by

$$w_x = F^x - F_{\text{load}} - F_{\text{Coulomb}}, \quad (7.8)$$

where the load force comes from the mass attached to the motor. It should be noted that the viscous friction is modeled as a damper in the linear dynamics part and

therefore is not taken into account here. In addition, since the position-dependent part of the Coulomb friction is very small compared to the load force, we neglect the position dependency and assume that the Coulomb friction is constant along the stroke. The predictor model is given by

$$\begin{aligned}
 \hat{x}_{\text{NARX}}(t, \theta) &= - \sum_{j=1}^2 a_j x_m(t-j) + \sum_{k=1}^2 b_k w_x(t-k) \\
 &= - \sum_{j=1}^2 a_j x_m(t-j) + f \\
 &\quad + \sum_{k=1}^2 b_k \sum_{l=1}^2 c_l \cos(\omega_n x_m(t-k)) u_l(t-k) \\
 &\quad + \sum_{k=1}^2 b_k \sum_{l=1}^2 d_l \sin(\omega_n x_m(t-k)) u_l(t-k) \\
 &= \varphi_{\text{NARX}}^\top(t) \theta,
 \end{aligned} \tag{7.9}$$

where

$$f = - \sum_{k=1}^2 b_k (F_{\text{load}} + F_{\text{Coulomb}}), \tag{7.10}$$

and  $\theta$  is the parameter vector

$$\theta = [a_1 \ a_2 \ f \ b_1 c_1 \ b_1 d_1 \ b_1 c_2 \ b_1 d_2 \ b_2 c_1 \ b_2 d_1 \ b_2 c_2 \ b_2 d_2]^\top.$$

The identification experiments are performed as follows. The motor operates in closed position loop as depicted in Figure 3.3 (see page 40). A load of 50 kg is attached to the motor. To avoid the nonlinear behavior of the friction around zero velocity, we let the motor move in only one direction and keep the velocity positive. The motor moves from one end to the other with changing velocity. The changing velocity requires the motor force to keep changing, thereby generating excitation to the linear dynamics part. The input currents are calculated using the classical commutation. Then multi-sine excitation input currents are added

$$u = u_{\text{classical}} + u_{\text{excitation}} \tag{7.11}$$

The excitation inputs are calculated as follows. Assume that we have  $n_u$  independent inputs. Then  $n_u - 1$  inputs are excited. The last input is calculated such that the force produced by the motor is equal to the reference force  $F_{\text{ref}}^x$  given by the position controller. The measurements of the input currents and the output position are collected. The sampling frequency is 1 kHz. The data-driven modeling method for the driving direction introduced in Section 3.2 is then applied to identify the parameters of the linear motor model. The identified position-dependent force functions  $K_j^x(x)$  are plotted in Figure 7.7. The nominal force functions given by the motor specifications are also plotted for comparison. It can be seen that estimation of the force functions is very poor.

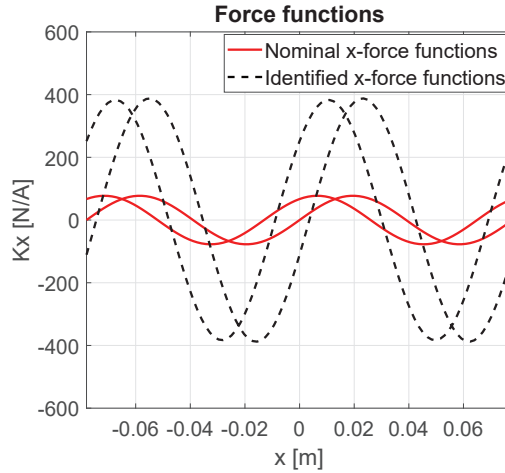


Figure 7.7: The nominal and identified force functions in the driving direction.

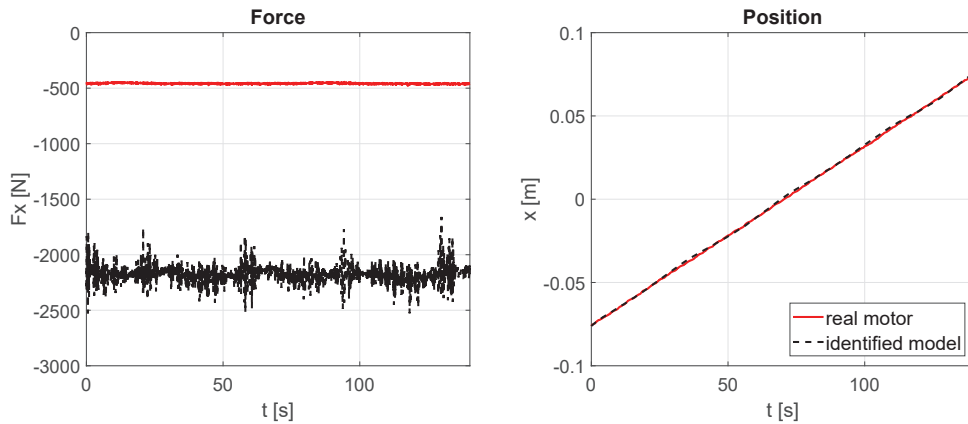


Figure 7.8: The forces and positions of the real motor and the identified model.

We simulate the identified model with the input currents which were supplied to the real motor. The resulting force and position obtained from the simulation are shown in Figure 7.8. It is observed that the estimate of the force is very poor. However, the estimate of the output position is still decent. This suggests that although the parameters vary considerably, they vary together such that the effect on the output is small. In other words, the correlation between the parameters is high. Let us check the correlation matrices between some parameters

$$R_{[a_1, b_1, c_1]} = \begin{bmatrix} 1 & 0.52 \\ 0.52 & 1 \end{bmatrix}, \quad R_{[a_1, b_1, c_2]} = \begin{bmatrix} 1 & 0.78 \\ 0.78 & 1 \end{bmatrix},$$



$$R_{[a_1, b_1 c_3]} = \begin{bmatrix} 1 & 0.78 \\ 0.78 & 1 \end{bmatrix}, \quad R_{[a_1, b_1 c_4]} = \begin{bmatrix} 1 & 0.78 \\ 0.78 & 1 \end{bmatrix}.$$

This verifies that indeed there is a high correlation between parameters and the parameters vary considerably together. As discussed in Section 3.2.5, due to the huge difference in the range of  $a$  compared to the range of  $bc$  and  $bd$ , a small relative variation in the estimate of  $a$  can cause a huge relative variation in the estimates of  $bc$  and  $bd$ . This causes difficulty in separately identifying the static nonlinearity part and the dynamics part. An alternative solution to this problem is to identify the static nonlinearity separately as will be described in the following subsection.

### Identification of the static nonlinearity

As an alternative, we employ the data-driven modeling method for the static nonlinearity only which is developed in Section 3.3 to identify the model of the static nonlinearity in the driving direction. This method requires measurement data of the input currents and the force produced by the motor. This is possible thanks to the force sensor in the driving direction.

The identification experiments are performed as follows. The motor operates in closed position loop. The motor moves from one end to the other by following a third-order reference motion profile. The input excitation is implemented as described in the previous subsection. During the constant velocity phase, the current measurements and force measurements are collected. We note that the force produced by the motor is equal to

$$F^x = m_{\text{total}} a + F_{\text{friction}} + F_{\text{load}}, \quad (7.12)$$

where  $m_{\text{total}}$  is the sum of the masses of the bearing, the translator and the load. The force measured by the force sensor in the driving direction is given by (7.1). Consequently, when  $a = 0$  we have that

$$F^x = F_{\text{measured}}^x + F_{\text{load}}, \quad (7.13)$$

Therefore, in the constant velocity phase, the force produced by the motor is calculated as the sum of the force measured by the force sensor and the constant load force.

All 3 coil sets are used in the experiments. Since each coil set has only 2 independent inputs due to the star connection, we have in total 6 independent inputs. The driving force is given by

$$F^x = \begin{bmatrix} K_1^x(x) & K_2^x(x) & \dots & K_6^x(x) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_6 \end{bmatrix}, \quad (7.14)$$

where  $K_l^x(x)$ ,  $l = 1, 2, \dots, 6$ , are the position-dependent force functions. As described in Section 3.3, the following predictor is used for identification

$$\hat{F}_{\text{NARX}}^x(t, \theta) = \sum_{l=1}^6 \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t)) u_l(t) + \sum_{l=1}^6 \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t)) u_l(t). \quad (7.15)$$

Here, the experiment range is  $[-2\tau_p, 2\tau_p]$ . We choose this experiment range as the base Fourier period. The 1st, 2nd, 4th, 6th and 10th harmonic components are included in the predictor model.

We perform experiments with different load forces and collect the measurement data. As analyzed in Chapter 3, since our  $x$ -position measurement noise is much smaller compared to the magnet pole pitch  $\tau_p$ , we can safely use the IV method developed in Section 3.3 without bias correction. The identified position-dependent force functions  $K_l^x(x)$  are plotted in Figure 7.9. The nominal force functions given by the motor specifications are also plotted for comparison. At first glance, it seems that the identified force functions are almost the same as the nominal force functions. But if we look closely, we can observe that, in contrast to the nominal force functions, the identified force functions of the coils in different coil sets are not exactly identical. The difference between the identified force functions and the nominal force functions are plotted in Figure 7.10.

To validate the identified model, we give the same input currents to the nominal model, the identified model and the real motor, and then compare their output forces. The input currents is calculated using classical commutation as shown in Figure 7.11. The results are plotted in Figure 7.11. It can be seen that the identified model provides a closer match with the real motor than the nominal model does. The root mean square (rms) value of the force error of the nominal model is 5.92 N. The rms value of the force error of the identified model is 2.89 N, which is much smaller than that of the nominal model.

### 7.3.3 Identification in the non-driving directions

To identify the model in the non-driving directions, we use the data-driven modeling method introduced in Section 3.3. Let us recall that the coil array is shifted 0.7 mm out of the center of the air gap. Consequently, there are both Lorentz force and reluctance force in the non-driving directions. The force in the  $z$ -direction is given by

$$F^z = K^z(x)u + u^\top G^z u. \quad (7.16)$$

As described in Section 3.3, the following predictor is used for identification

$$\begin{aligned} \hat{F}_{\text{NARX}}^z(t, \theta) = & \sum_{l=1}^6 \sum_{n=1}^{n_F} c_{l,n} \cos(\omega_n x_m(t)) \Lambda_l^{\text{Lor}}(u(t)) \\ & + \sum_{l=1}^6 \sum_{n=1}^{n_F} d_{l,n} \sin(\omega_n x_m(t)) \Lambda_l^{\text{Lor}}(u(t)) + \sum_{l=1}^{21} f_l \Lambda_l^{\text{rel}}(u(t)), \end{aligned} \quad (7.17)$$

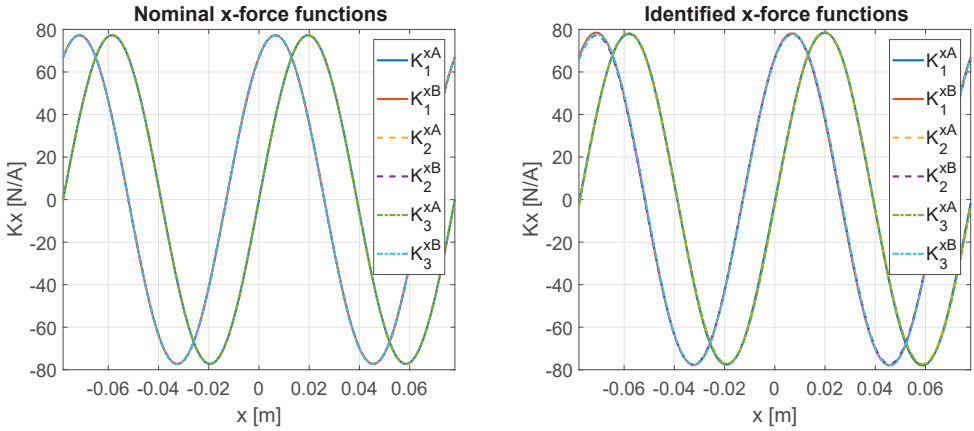


Figure 7.9: The nominal and identified force functions in the driving direction.

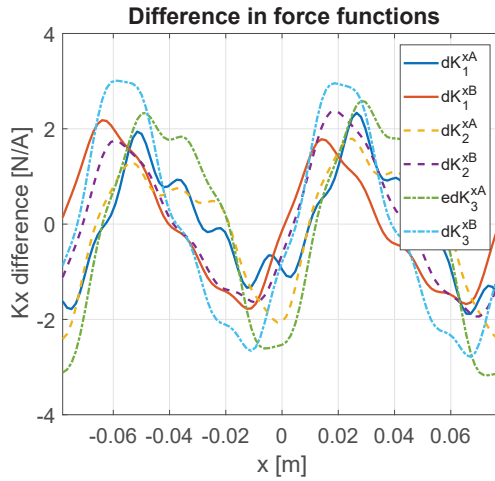


Figure 7.10: The difference between the identified and the nominal force functions in the driving direction.

where

$$\Lambda_l^{\text{Lor}}(u) = [u_1 \quad u_2 \quad \dots \quad u_6]^\top,$$

and

$$\Lambda_l^{\text{rel}}(u) = [u_1^2 \quad u_1 u_2 \quad \dots \quad u_1 u_6 \quad u_2^2 \quad u_2 u_3 \quad \dots \quad u_6^2]^\top.$$

Similar to the previous section, the experiment range  $[-2\tau_p, 2\tau_p]$  is chosen as the base Fourier period. The 1st, 2nd, 4th, 6th and 10th harmonic components are included in the Fourier series in the predictor model.

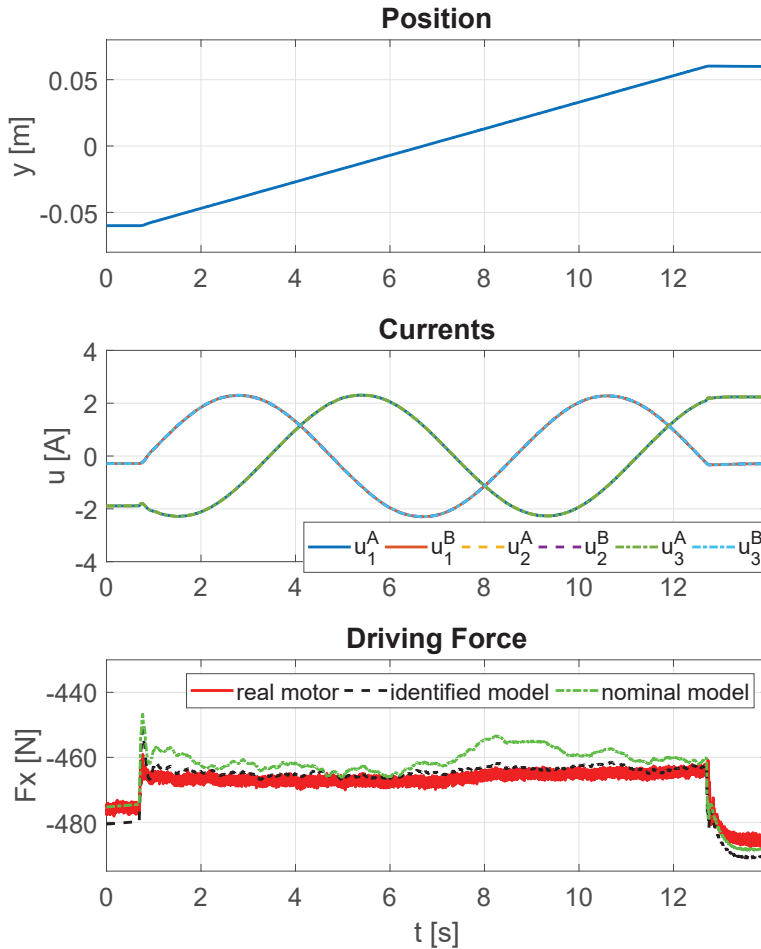


Figure 7.11: Model validation - Output forces of the nominal model, the identified model and the real motor in the driving direction.

The identification experiments are performed in the same ways as described in the previous section. The current measurements and force measurements are collected. As analyzed in Chapter 3, since our  $x$ -position measurement noise is much smaller compared to the magnet pole pitch  $\tau_p$ , we can safely use the IV method developed in Section 3.3 without bias correction. For illustration, the identified Lorentz position-dependent force functions of the force in the  $z$ -direction and the torque around the  $y$ -direction are plotted in Figure 7.12.

To validate the identified model, we supply the same input currents to the identified model and the real motor, then compare their output forces. We use the same input currents calculated using classical commutation as shown in Figure 7.11, which have also been used for model validation in the driving direction in the previous

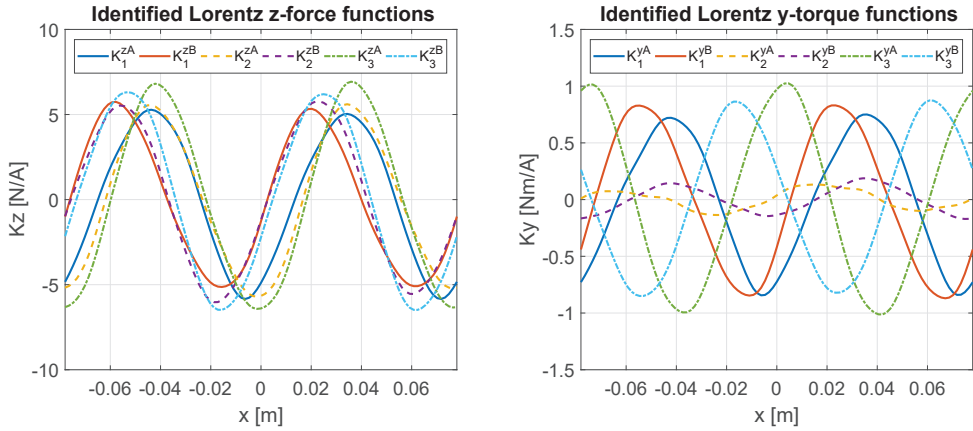


Figure 7.12: The identified Lorentz force functions in the non-driving directions.

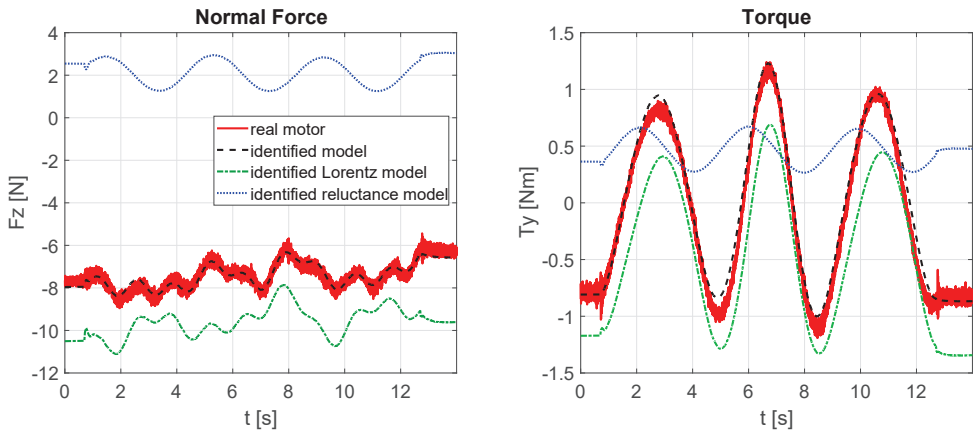


Figure 7.13: Model validation - Output forces of the identified model and the real motor in the non-driving directions.

section. The results are plotted in Figure 7.13. The Lorentz component and reluctance component are also plotted. It can be seen that the outputs of the identified model are close to the outputs of the real motor. The rms value of the  $z$ -force error is 0.21 N. The rms value of the  $y$ -torque error is 0.10 Nm. The validation results show that the identified models are accurate.

## 7.4 Optimal commutation experiments

In this section we present the results of the optimal commutation experiments. The experiments are performed as follows. The motor operates in closed position loop

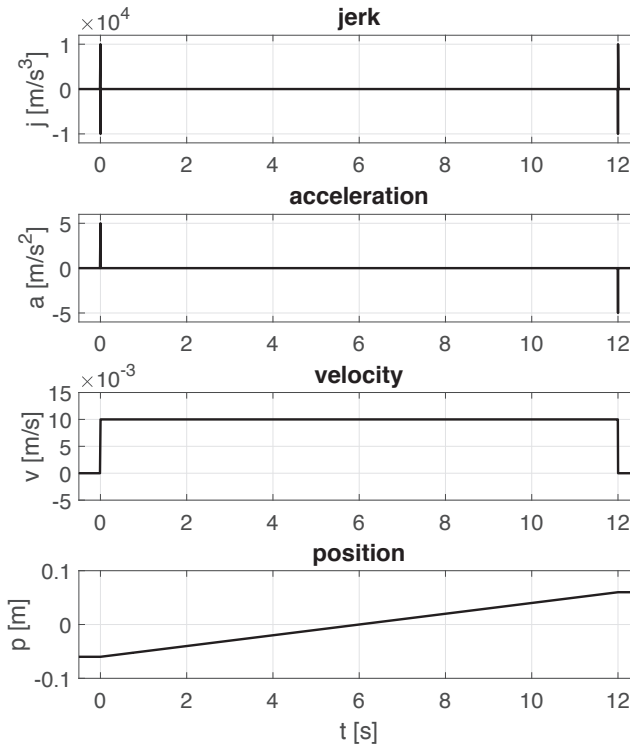


Figure 7.14: The reference motion profile.

as depicted in Figure 3.3. The bandwidth of the control loop is 10 Hz. A load of 50 kg is attached to the motor. The sampling frequency is 10 kHz.

The motor moves from one end to the other by following a third-order reference motion profile. The reference motion profile parameters are as follows: jerk= 10000  $\text{m/s}^3$ , acceleration= 5  $\text{m/s}^2$  and velocity= 0.01  $\text{m/s}$ . The reference motion profile is shown in Figure 7.14. It is noted that the maximum achievable jerk is limited by the maximum amplifier voltage and the motor inductance. For the motor and amplifier used, a jerk of 10000  $\text{m/s}^3$  is achievable. However, in this experiment we are mainly interested in the performance of the commutation methods in the constant velocity phase. It is not crucial to achieve the reference jerk and therefore no feedforward controller is included.

We first perform the experiment with classical commutation as described in Section 4.2. Then the optimal commutation method is implemented. The fast optimization algorithm introduced in Section 4.3.4 and analyzed in Chapter 6 is successfully implemented on the real-time hardware for solving the optimal commutation prob-

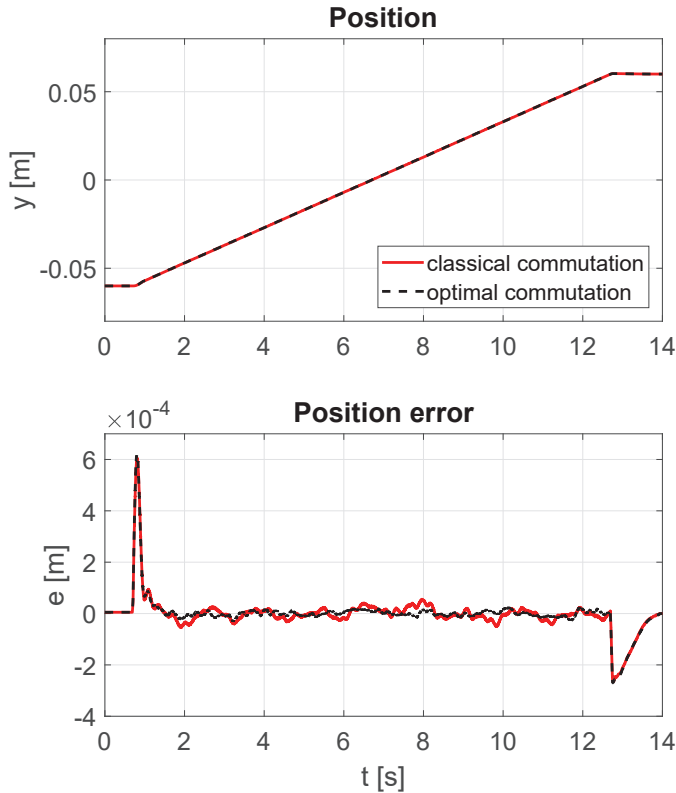


Figure 7.15: Position errors of the classical and optimal commutation methods.

lem. The results are discussed below.

Figure 7.15 shows the position errors of the two commutation methods. It is observed that the optimal commutation method using the models identified in Section 7.3 results in lower position error compared to the classical commutation method. During the constant velocity phase, the rms value of the position error of the classical commutation method is  $20.18 \mu\text{m}$ , while the rms value of the position error of the optimal commutation method is  $10.34 \mu\text{m}$ .

Figure 7.16 shows the comparison of the reference forces  $F_{\text{ref}}^x$  produced by the position controller and the output forces  $F^x$  produced by the motor in the two commutation methods. Since commutation is the inverse of the static nonlinearity, the quality of a commutation method can be evaluated by looking at the error between the reference force and the output force. The smaller the error is, the better the commutation method is. A perfect commutation will result in zero force error. Here, it is observed that in the force error of the classical commutation is much larger than that of the optimal commutation. During the constant velocity phase, the 3-sigma value of the force error of the classical commutation method is  $7.64 \text{ N}$ , while the

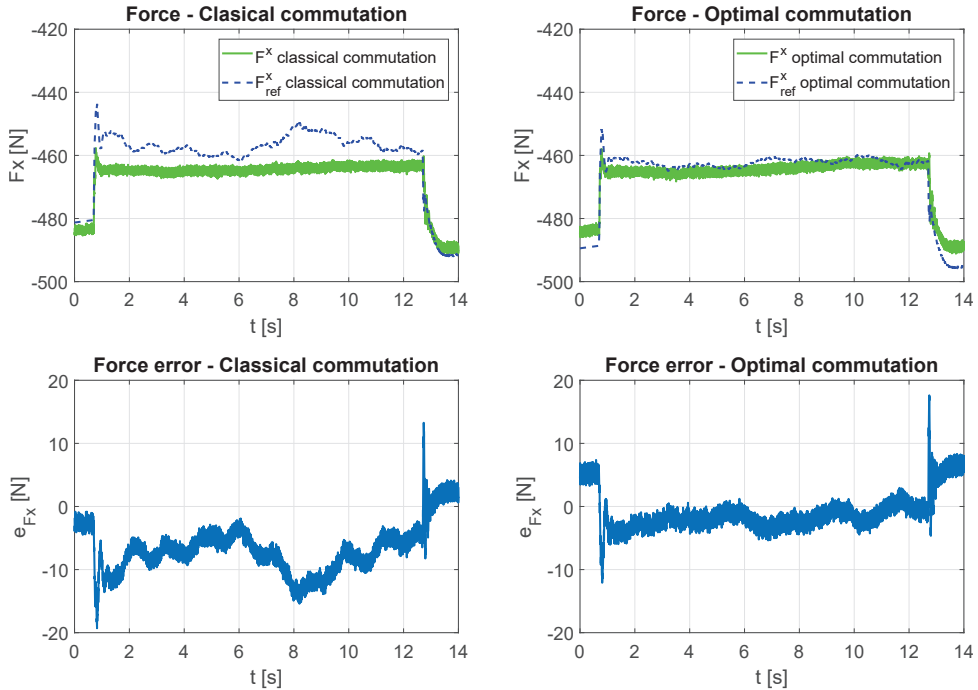


Figure 7.16: Forces and force errors of the classical and optimal commutation methods.

3-sigma value of the force error of the optimal commutation method is 3.63 N.

Figure 7.17 shows the force in the  $z$ -direction and the torque around the  $y$ -direction of the two commutation methods. It is observed that, using the optimal commutation method, the  $z$ -force and the  $y$ -torque are reduced in both offset and variation. The 3-sigma values of the  $z$ -force and  $y$ -torque during the constant velocity phase are summarized in Table 7.2.

Figure 7.18 shows the input currents of the two commutation methods. It can be seen that current waveforms of the optimal commutation method are not pure sine waves as those of the classical commutation.

A comparison of the two commutation methods is summarized in Table 7.2. The experimental results prove that the optimal commutation method can improve tracking performance in the driving direction and significantly reduce the parasitic forces and torques in the non-driving directions.

## 7.5 Conclusions

This chapter has been devoted to the experimental validation of the identification and commutation methods developed in the previous chapters. An experimental coreless linear motor setup has been constructed in collaboration with ASML for



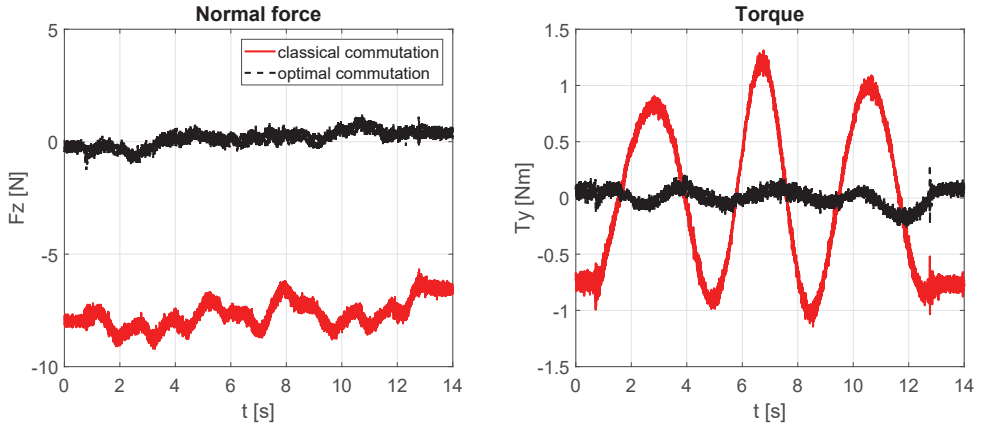


Figure 7.17: Force and torque in the non-driving directions of the classical and optimal commutation methods.

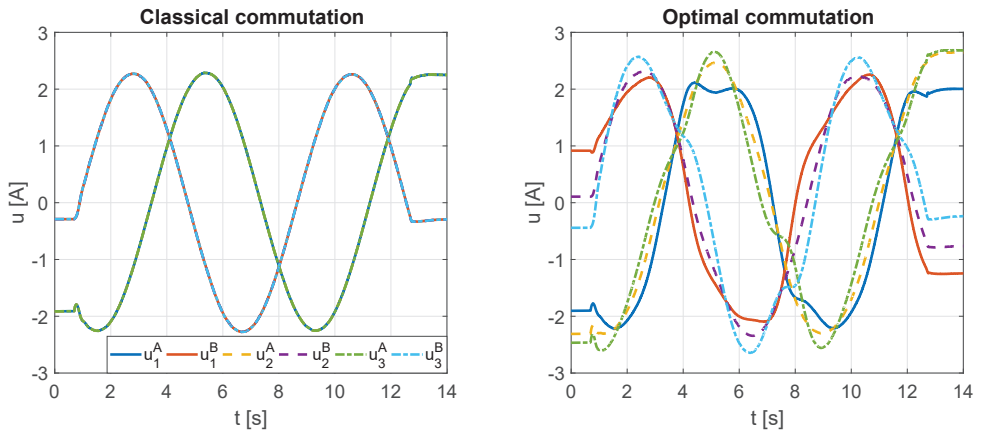


Figure 7.18: Input currents of the classical and optimal commutation methods.

this purpose. In this chapter we have presented the design and specifications of the experimental setup. The setup consists of a real industrial linear motor, power amplifiers and an encoder. Force sensors are added to measure the forces and torques in the driving and non-driving directions.

Identification experiments have been performed to identify the model of the linear motor in driving and non-driving directions, using the method introduced in Chapter 3. The IV identification method developed in Section 3.2, which aims to identify the static nonlinearity and the linear dynamical system simultaneously, does not provide an accurate estimate. The causes this problem are the high correlation between the parameters and the huge difference in the values of the parameters

Table 7.2: Comparison of the two commutation methods.

Method	Position error rms [ $\mu\text{m}$ ]	x-force error 3-sigma [N]	z-force 3-sigma [N]	y-torque 3-sigma [Nm]
Classical	20.18	7.64	1.62	2.06
Optimal	10.34	3.63	1.00	0.22

in the parameter vector. How to address this problem requires further research on experiment design.

As an alternative, we employ the IV identification method developed in Section 3.3, which identifies only the static nonlinearity, to estimate the model parameters in both driving and non-driving directions. The method successfully delivers accurate estimates.

The optimal commutation method using fast optimization algorithm introduced in Chapter 4 is then implemented successfully on the setup with a fast sampling frequency of 10 kHz. The experimental results show that, compared to the classical commutation method, the optimal commutation method reduces the position error in the driving direction by more accurately generating the control force. Furthermore, the force and torque in the non-driving directions are also reduced. The results prove that the optimal commutation method is able to reduce the parasitic forces and torques in driving and non-driving directions.

# Chapter 8

## Conclusions and recommendations

### 8.1 Conclusions

This thesis has been motivated by the future requirements of industrial positioning systems. Future generations of industrial positioning systems are facing increasing demands in precision and throughput. As a result, coreless linear motors are required to operate at higher acceleration and higher accuracy. One of the main challenges in meeting these requirements is the presence of parasitic effects, which lower the performance of a coreless linear motor and create disturbances to surrounding sensitive components. To address this problem, the main research objective of the thesis has been formulated as: "To identify and compensate for main parasitic effects in coreless linear motors in multiple degrees of freedom". In this thesis, the main research objective has been addressed step-by-step by investigating the six research questions formulated in Chapter 1. In the following, we summarize the results of our investigation.

#### **Q1. How to represent the main parasitic forces in a CLM mathematically in closed form expressions?**

Research question Q1 has been addressed in Chapter 2. As analyzed in Chapter 2, there are three main types of forces in a CLM: Lorentz force, reluctance force and drag force. The derivation of first-principle models of these main parasitic forces has been presented using the available Fourier modeling technique, under the assumption that the exact physical parameters of the CLM are known. The resulting models have been validated by comparing their outputs to the outputs of FEM models. The results are closely matched.

The resulting models are analytical which require low computational load and therefore are well-suited to controller design purposes. Furthermore, based on the

resulting first-principle models, model structures of the main parasitic forces have been derived, which are then utilized to develop a data-driven modeling method for CLMs in Chapter 3.

### **Q2. How to identify the parameters of the physical model structure of a CLM from measurement data?**

Research question Q2 has been investigated in Chapter 3. There are two main challenges in this identification problem. Firstly, the CLM must always operate in closed-loop for safety reason. Consequently, the input is correlated with the measurement noise, causing the simple LS estimate to be biased. The second challenge is the non-linear dependency of the CLM model on the unknown noise-free position output, which makes it difficult to find an appropriate predictor for identification.

In Chapter 3, we employ the IV identification framework for closed-loop identification. We have proved that using the IV identification framework together with the simple NARX predictor, the resulting estimate is biased. However, the bias has been shown to be negligible if the position measurement noise is small compared to the magnet pole pitch. When this does not hold, we have developed a bias-corrected IV method which has been proved to deliver a consistent estimate. This method requires knowledge of the probability distribution of the measurement noise.

### **Q3. How to derive an advanced commutation technique which can compensate for the main parasitic effects in a CLM based on its data-driven model?**

Research question Q3 has been investigated in Chapter 4. The commutation problem is formulated as finding the inverse of the static nonlinear model obtained by the data-driven modeling technique introduced in Chapter 3. Since the data-driven model has captured the main parasitic effects in a linear motor, the parasitic effects can be compensated if we can find the exact inverse of the model. When a CLM is over-actuated, i.e. there are more degrees of freedom than needed, there is an infinite number of solutions to the inverse problem. We can use the extra degrees of freedom to minimize the power losses in the coils. The commutation problem therefore becomes a quadratic optimization problem with nonlinear equality constraints. Solving this type of optimization problems generally requires numerical optimization algorithms, which are usually computationally expensive.

To address this problem, in Chapter 4 we have presented computationally efficient methods for solving the optimal commutation problem. A look-up table is a simple but efficient method, but on the other hand requires a large amount of memory and is not suitable for applications which requires control in multiple DOFs, since a multi-dimension table is required. As an alternative, sub-optimal analytical solutions have been introduced for cases when the number of coils in the translator is low. Furthermore, we have also developed a new optimization algorithm which is more computationally efficient than the classical Newton's method. The effectiveness of the proposed commutation methods has been validated in simulation with a

FEM model and in experiments with a real CLM setup.

**Q4. How to design a single nonlinear controller for a CLM instead of using a separate linear position controller and commutation?**

Research question Q4 has been addressed in Chapter 5. In this chapter, we have investigated the use of MPC in designing a nonlinear controller for linear motors. We have chosen MPC due to its ability to deliver high performance while guaranteeing constraint satisfaction. Two nonlinear controller schemes using MPC have been introduced: the Combined LMPC+OC scheme and the NMPC scheme. The Combined LMPC+OC scheme literally combines the LMPC problem and OC problem into a single optimization problem. It requires less computational effort than solving the LMPC problem and OC problem separately. However, it does not predict and minimize the future error caused by the static nonlinearity. The NMPC scheme includes the full nonlinear model for prediction and therefore may deliver better performance, at the price of higher computational effort, especially for high prediction horizon. Both of the introduced MPC schemes are able to compensate for parasitic forces and torques and also guarantee constraint satisfaction. The effectiveness of the new MPC schemes has been demonstrated in a simulation example.

**Q5. How to design a fast optimization solver such that the developed MPC and commutation algorithms can be implemented in real-time?**

Research question Q5 has been addressed in Chapter 6. In this chapter we have presented a new computationally efficient optimization method for solving nonlinear programming problems. The new method is a variant of SQP in which an arbitrary positive (semi-) definite Hessian approximation can be used. This is the main advantage of the method, since other SQP algorithms generally require good Hessian approximations which can be expensive to compute, especially for large-scale problems such as an MPC problem with high prediction horizon.

In our new method, the search direction is taken as the interpolation between the search direction provided by solving the QP subproblem and a feasible search direction. It is proved that the algorithm converges locally at linear rate to a local optimum point of the nonlinear programming problem. Furthermore, a strategy for global convergence is employed to guarantee convergence from any initial estimates. The effectiveness of the method has been illustrated in two numerical examples.

For NMPC of linear motors, it appears that the new algorithm is still not fast enough for real-time implementation. However, the algorithm is implemented in Matlab code, which is generally slower than C code. By converting from Matlab code to C code, we expect a significant reduction in computation time, which is promising for real-time implementation. This will be implemented in our future work.

### **Q6. Can we validate the developed identification, commutation and control methods in experiments with a real CLM setup?**

Research question Q6 has been addressed in Chapter 7. In order to validate the developed identification, commutation and control methods, we have designed and constructed an experimental CLM setup in collaboration with ASML. The setup consists of a real industrial linear motor, power amplifiers and an encoder provided by ASML. Additional force sensors are added in order to measure the forces and torques produced by the motor in driving and non-driving directions. The setup is controlled by a dSPACE MicroLabBox development system.

The data-driven modeling method and the optimal commutation method are validated on the experimental setup. First, identification experiments have been performed to identify the model of the linear motor in driving and non-driving directions, using the methods developed in Chapter 3. The IV identification method developed in Section 3.2, which aims to identify both the static nonlinearity and the linear dynamics, does not provide an accurate estimate. This problem is caused by the high correlation between the parameters and the huge difference in the values of the parameters in the parameter vector. How to address this problem requires further research on experiment design. As an alternative, the IV identification method developed in Section 3.3 which identify only the static nonlinearity is employed. The method successfully delivers an accurate estimate of the model parameters in both the driving and non-driving directions.

The optimal commutation method using fast optimization algorithm introduced in Chapter 4 is then implemented successfully on the setup with a fast sampling frequency of 10 kHz. The experimental results show that, compared to the classical commutation method, the optimal commutation method reduces the tracking error in the driving direction. Furthermore, the force and torque in the non-driving directions are also reduced significantly, in both offset and variation. This result is the first experimental evidence to prove that the optimal commutation method is able to compensate for the parasitic forces and torques in the non-driving directions in linear motors.

## **8.2 Future work and recommendations**

### **Investigation of the problem of the IV identification method for the driving direction**

In our identification experiments, the IV identification method for the driving direction developed in Section 3.2 does not provide accurate estimates of the model parameters. As analyzed in Chapter 7, this problem is caused by the high correlation between the parameters and the huge difference in the values of the parameters in the parameter vector. To deal with this problem, it is of interest to further investigate the design of the experiments.

### **Fast nonlinear optimization solver for NMPC of linear motors**

For future work, we will develop a fast nonlinear optimization solver for the NMPC of linear motors problem. The new optimization solver developed in Chapter 6 will be implemented in C code, which is expected to run faster than the Matlab code which is used for the simulation example in Section 6.8. The C code can then be deployed to a real-time processor for real-time implementation on the experimental linear motor setup.

Another interesting topic for future work is how to adapt the interpolation coefficient in the algorithm such that the fastest convergence rate is achieved. Furthermore, it is observed that the NMPC of linear motors problem has a special sparse structure. For future work, it is of interest to exploit the sparsity of the problem to write customized fast solver dedicated to solving the NMPC of linear motors problem.

### **Robustness and recursive feasibility of the commutation and NMPC problems**

In practice, linear motors may need to operate in the presence of high noise and disturbance. For future work, it is of interest to analyze the robustness properties of the commutation and NMPC problems. Furthermore, recursive feasibility of the NMPC for linear motors problem is an important topic which needs to be addressed in future research.

### **Extension to other types of electrical machines**

Although the identification and compensation methods in this thesis are developed for coreless linear motors, the methods can also be applied to other type of electrical machines. The main types of electromagnetic forces in electrical machines are similar: Lorentz force, reluctance force, cogging force and drag force. As a result, the identification method introduced in Chapter 3 is applicable to other types of machines. The optimal commutation problem in other types of machines can also be solved using the fast optimization solver introduced in Chapter 4.





# Bibliography

- [1] J. Abonyi, R. Babuška, F. Szeifert, and L. Nagy. Identification and control of nonlinear systems using fuzzy Hammerstein models. *Industrial & Engineering Chemistry Research*, 39(11):4302–4314, 2000.
- [2] R. Ahmed and D. G. Taylor. Optimal excitation of linear variable reluctance motors with coupled and uncoupled flux paths. In *Industrial Electronics, 2006 IEEE International Symposium on*, volume 3, pages 2498–2503, July 2006.
- [3] R. Ahmed and D. G. Taylor. Targeted-ripple optimal commutation of coupled and uncoupled linear variable reluctance motors. In *SoutheastCon 2015*, pages 1–8, April 2015.
- [4] S. R. Aleksandrov, D. C. J. Krop, D. T. E. H. van Casteren, T. T. Overboom, and E. A. Lomonova. Analytical modelling techniques for thrust force calculation of a permanent magnet linear motor. In *2017 11th International Symposium on Linear Drives for Industry Applications (LDIA)*, pages 1–6, Sept 2017.
- [5] E.-W. Bai. An optimal two-stage identification algorithm for Hammerstein–Wiener nonlinear systems. *Automatica*, 34(3):333 – 338, 1998.
- [6] N. Barahanov and R. Ortega. Necessary and sufficient conditions for passivity of the LuGre friction model. *IEEE Transactions on Automatic Control*, 45(4):830–832, Apr 2000.
- [7] L. Bascetta, P. Rocco, and G. A. Magnani. Force ripple compensation in linear motors based on closed-loop position-dependent identification. *Mechatronics, IEEE/ASME Transactions on*, 15(3):349–359, June 2010.
- [8] F. Baudart, B. Dehez, F. Labrique, and E. Matagne. Optimal sinusoidal currents for avoiding torque pulsations after the loss of one phase in polyphase SMPM synchronous motor. In *Power Electronics Electrical Drives Automation and Motion (SPEEDAM), 2010 International Symposium on*, pages 13–18, June 2010.

- [9] F. Baudart, E. Matagne, B. Dehez, and F. Labrique. Optimal current waveforms for torque control of permanent magnet synchronous machines with any number of phases in open circuit. *Mathematics and Computers in Simulation*, 90:1 – 14, 2013.
- [10] A. Ben-Israel. A Newton-Raphson method for the solution of systems of equations. *Journal of Mathematical Analysis and Applications*, 15(2):243 – 252, 1966.
- [11] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
- [12] D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton reference. Princeton University Press, 2009.
- [13] H. G. Bock. *Recent Advances in Parameter identification Techniques for O.D.E.*, pages 95–121. Birkhäuser Boston, Boston, MA, 1983.
- [14] H. G. Bock, E. Kostina, and J. P. Schlöder. *Direct Multiple Shooting and Generalized Gauss-Newton Method for Parameter Estimation Problems in ODE Models*, pages 1–34. Springer International Publishing, Cham, 2015.
- [15] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.
- [16] R. H. Byrd, J. Nocedal, and R. A. Waltz. *Knitro: An Integrated Package for Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006.
- [17] G. Cain and G. H. Meyer. *Separation of Variables for Partial Differential Equations: An Eigenfunction Approach*. CRC Press, 2005.
- [18] R. Cao and K. S. Low. Repetitive model predictive control of a precision linear motor drive. In *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, pages 1132–1137, Nov 2007.
- [19] M. Curti, J. J. H. Paulides, and E. A. Lomonova. An overview of analytical methods for magnetic field computation. In *2015 Tenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, pages 1–7, March 2015.
- [20] B. J. H. de Bruyn, J. W. Jansen, and E. A. Lomonova. Minimization of force ripples in coreless linear actuators. In *2014 International Conference on Electrical Machines (ICEM)*, pages 558–563, Sept 2014.
- [21] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, Germany, 2001.

- [22] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577 – 585, 2002.
- [23] Q. T. Dinh, C. Savorgnan, and M. Diehl. Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization. *SIAM Journal on Optimization*, 22(4):1258–1284, 2012.
- [24] S. Dwari and L. Parsa. Optimal current waveforms for five-phase permanent magnet motor drives under open-circuit fault. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pages 1–5, July 2008.
- [25] E. Eskinat, S. H. Johnson, and W. L. Luyben. Use of Hammerstein models in identification of nonlinear systems. *AIChE Journal*, 37(2):255–268, 1991.
- [26] M. Fiacchini, I. Alvarado, D. Limon, T. Alamo, and E. F. Camacho. Predictive control of a linear motor for tracking of constant references. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4526–4531, Dec 2006.
- [27] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.
- [28] U. Forssell and L. Ljung. Closed-loop identification revisited. *Automatica*, 35(7):1215 – 1241, 1999.
- [29] E. P. Furlani. *Permanent Magnet and Electromechanical Devices: Materials, Analysis, and Applications*. Academic Press, 2001.
- [30] M. Gajdusek. *Controlling a contactless planar actuator with manipulator*. PhD thesis, TUE : Department of Electrical Engineering, 2010.
- [31] J. Gieras and M. Godkin. Status of permanent magnet linear motors in the US. 1, Oct 2001.
- [32] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Some theoretical properties of an augmented lagrangian merit function. In *Advances in Optimization and Parallel Computing*, pages 101–128, North-Holland, Amsterdam, 1992.
- [33] P. E. Gill, M. A. Saunders, and E. Wong. *On the Performance of SQP Methods for Nonlinear Optimization*, pages 95–123. Springer International Publishing, Cham, 2015.
- [34] M. Gilson. What has instrumental variable method to offer for system identification? *IFAC-PapersOnLine*, 48(1):354 – 359, 2015. 8th Vienna International Conference on Mathematical Modelling.

- [35] M. Gilson and P. M. J. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2):241 – 249, 2005.
- [36] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [37] N. Gould and D. Robinson. A second derivative SQP method: Global convergence. *SIAM Journal on Optimization*, 20(4):2023–2048, 2010.
- [38] B. L. J. Gysen, K. J. Meessen, J. J. H. Paulides, and E. A. Lomonova. General formulation of the electromagnetic field distribution in machines and devices using Fourier analysis. *Magnetics, IEEE Transactions on*, 46(1):39–52, Jan 2010.
- [39] B. L. J. Gysen, K. J. Meessen, J. J. H. Paulides, and E. A. Lomonova. 3-d analytical and numerical modeling of tubular actuators with skewed permanent magnets. *IEEE Transactions on Magnetics*, 47(9):2200–2212, Sept 2011.
- [40] B. Gysen. *Generalized harmonic modeling technique for 2D electromagnetic problems : applied to the design of a direct-drive active suspension system*. PhD thesis, TUE : Department of Electrical Engineering, 2011.
- [41] B. Hague. *Electromagnetic Problems in Electrical Engineering*. Milford, 1929.
- [42] R. Hellinger and P. Mnich. Linear motor-powered transportation: History, present status, and future outlook. *Proceedings of the IEEE*, 97(11):1892–1900, Nov 2009.
- [43] K. Hirano, Y. Um, and Y. Kano. A simulation of MPC application for linear motor position control. *International Journal of Applied Electromagnetics and Mechanics*, 13(1-4):181–185, 2002.
- [44] J. J. E. Dennis and J. J. Moré. Quasi-newton methods, motivation and theory. *SIAM Review*, 19(1):46–89, 1977.
- [45] Z. P. J. Gieras and B. Tomczuk. *Linear Synchronous Motors: Transportation and Automation Systems*. CRC Press, Boca Raton, 2nd edition, 2011.
- [46] J. W. Jansen, E. A. Lomonova, and J. C. Compter. Analysis of the parasitic forces and torques in coreless linear motors. In *Electrical Machines and Systems (ICEMS), 2011 International Conference on*, pages 1–6, Aug 2011.
- [47] J. W. Jansen, E. A. Lomonova, A. J. A. Vandenput, and C. M. M. van Lierop. Analytical model of a magnetically levitated linear actuator. In *Fortieth IAS Annual Meeting. Conference Record of the 2005 Industry Applications Conference, 2005.*, volume 3, pages 2107–2113 Vol. 3, Oct 2005.
- [48] J. W. Jansen, J. P. C. Smeets, T. T. Overboom, J. M. M. Rovers, and E. A. Lomonova. Overview of analytical models for the design of linear and planar motors. *IEEE Transactions on Magnetics*, 50(11):1–7, Nov 2014.

- [49] J. Jansen. *Magnetically levitated planar actuator with moving magnets: electromechanical analysis and design*. PhD thesis, TUE : Department of Electrical Engineering, 2007.
- [50] J. Jin. *The Finite Element Method in Electromagnetics*. Wiley - IEEE. Wiley, 2014.
- [51] T. A. Johansen and T. I. Fossen. Control allocation—a survey. *Automatica*, 49(5):1087 – 1103, 2013.
- [52] L. V. Kantorovich and G. P. Akilov. *Functional Analysis in Normed Spaces*. Pergamon Press, New York, 1964.
- [53] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press.
- [54] V. Laurain, M. Gilson, and H. Garnier. Refined instrumental variable methods for identifying Hammerstein models operating in closed loop. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 3614–3619, Dec 2009.
- [55] Y. Levin and A. Ben-Israel. A Newton method for systems of  $m$  equations in  $n$  variables. *Nonlinear Analysis: Theory, Methods and Applications*, 47(3):1961 – 1971, 2001. Proceedings of the Third World Congress of Nonlinear Analysts.
- [56] F. J. Lin, P. H. Chou, C. S. Chen, and Y. S. Lin. DSP-based cross-coupled synchronous control for dual linear motors via intelligent complementary sliding mode control. *IEEE Transactions on Industrial Electronics*, 59(2):1061–1073, Feb 2012.
- [57] Z. Z. Liu, F. L. Luo, and M. A. Rahman. Robust and precision motion control system of linear-motor direct drive for high-speed x-y table positioning mechanism. *IEEE Transactions on Industrial Electronics*, 52(5):1357–1363, Oct 2005.
- [58] L. Ljung. *System Identification (2Nd Ed.): Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [59] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [60] E. A. Lomonova and J. W. Jansen. Lecture notes: Linear and planar motors for high-precision systems. Eindhoven University of Technology, 2017.
- [61] H. C. Lovatt and J. M. Stephenson. Computer-optimised smooth-torque current waveforms for switched-reluctance motors. *Electric Power Applications, IEE Proceedings -*, 144(5):310–316, Sep 1997.

- [62] S. Lu, X. Tang, B. Song, S. Zheng, and F. Zhou. Identification and compensation of force ripple in PMSLM using a JITL technique. *Asian Journal of Control*, 17(5):1559–1568, 2015.
- [63] M. Ma, L. Li, J. Zhang, J. Yu, H. Zhang, and Y. Jin. Analytical methods for minimizing detent force in long-stator pm linear motor including longitudinal end effects. *IEEE Transactions on Magnetics*, 51(11):1–4, Nov 2015.
- [64] D. C. Meeker. *Optimal Solutions to the Inverse Problem in Quadratic Magnetic Actuators*. PhD thesis, University of Virginia, 1996.
- [65] N. Moehle and S. Boyd. Optimal current waveforms for brushless permanent magnet motors. *International Journal of Control*, 88(7):1389–1399, 2015.
- [66] O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, 2001.
- [67] T. T. Nguyen, H. Butler, and M. Lazar. An analytical commutation law for parasitic forces and torques compensation in coreless linear motors. In *2016 European Control Conference (ECC)*, pages 2386–2391, June 2016.
- [68] T. T. Nguyen, M. Lazar, and H. Butler. Cancellation of normal parasitic forces in coreless linear motors. In *System Theory, Control and Computing (ICSTCC), 19th International Conference on*, pages 192–199, Oct 2015.
- [69] T. T. Nguyen, M. Lazar, and H. Butler. A Hessian-free algorithm for solving quadratic optimization problems with nonlinear equality constraints. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2808–2813, Dec 2016.
- [70] T. T. Nguyen, M. Lazar, and H. Butler. A method to guarantee local convergence for sequential quadratic programming with poor hessian approximation. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3781–3786, Dec 2017.
- [71] T. T. Nguyen, M. Lazar, and H. Butler. Nonlinear model predictive control of ironless linear motors. In *2nd IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2018.
- [72] T. T. Nguyen, M. Lazar, H. Butler, and P. M. J. Van den Hof. An instrumental variable method for closed-loop identification of coreless linear motors. In *American Control Conference (ACC)*, Jun 2018.
- [73] T. T. Nguyen, M. Lazar, and H. Butler. A computationally efficient commutation algorithm for parasitic forces and torques compensation in ironless linear motors. *IFAC-PapersOnLine*, 49(21):267 – 273, 2016. 7th IFAC Symposium on Mechatronic Systems MECHATRONICS, Loughborough University, Leicestershire, UK.

- [74] B. Ni, M. Gilson, and H. Garnier. Refined instrumental variable method for Hammerstein-Wiener continuous-time model identification. *IET Control Theory Applications*, 7(9):1276–1286, June 2013.
- [75] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, New York, NY, USA, 2nd edition, 2006.
- [76] M. W. Oppenheimer, D. B. Doman, and M. A. Bolender. Control allocation for over-actuated systems. In *2006 14th Mediterranean Conference on Control and Automation*, pages 1–6, June 2006.
- [77] G. Otten, T. J. A. de Vries, J. van Amerongen, A. M. Rankers, and E. W. Gaal. Linear motor motion control using a learning feedforward controller. *IEEE/ASME Transactions on Mechatronics*, 2(3):179–187, Sep 1997.
- [78] H. Ouarti and T. F. Edgar. The use of approximate models and exact linearization for control of nonlinear processes. In *1993 American Control Conference*, pages 2268–2273, June 1993.
- [79] T. T. Overboom. *Electromagnetic levitation and propulsion: force and torque decoupling in a planar motor with magnetic suspension and fail-safety*. PhD thesis, Eindhoven University of Technology, The Netherlands, 2015.
- [80] T. T. Overboom, J. P. C. Smeets, J. W. Jansen, and E. A. Lomonova. Semianalytical calculation of the torque in a linear permanent-magnet motor with finite yoke length. *IEEE Transactions on Magnetics*, 48(11):3575–3578, Nov 2012.
- [81] T. T. Overboom, J. P. C. Smeets, J. W. Jansen, E. A. Lomonova, and D. Mavrudieva. Power-optimal force decoupling in a hybrid linear reluctance motor. In *Proceedings of the 10th International Symposium on Linear Drives for Industry Applications (LDIA)*, July 2015.
- [82] M. J. D. Powell. *A fast algorithm for nonlinearly constrained optimization calculations*, pages 144–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.
- [83] L. B. Rall. *Computational Solution of Nonlinear Operator Equations*. Wiley, New York, 1969.
- [84] C. R. Rao and S. K. Mitra. Generalized inverse of a matrix and its applications. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics*, pages 601–620, Berkeley, Calif., 1972. University of California Press.
- [85] S. Rehman and D. G. Taylor. Piecewise modeling and optimal commutation of switched reluctance motors. In *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, volume 1, pages 266–271 vol.1, Jul 1995.



- [86] A. Ridge, P. Clifton, R. McMahon, and H.-P. Kelly. Force ripple compensation in a tubular linear generator for marine renewable generation. In *Electric Machines Drives Conference (IEMDC), 2011 IEEE International*, pages 480–485, May 2011.
- [87] C. Röhrig. Current waveform optimization for force ripple compensation of linear synchronous motors. In *42nd IEEE International Conference on Decision and Control*, volume 6, pages 5891–5896 Vol.6, Dec 2003.
- [88] C. Röhrig. Force ripple compensation of linear synchronous motors. *Asian Journal of Control*, 7(1):1–11, 2005.
- [89] C. Röhrig. Optimal commutation law for three-phase surface-mounted permanent magnet linear synchronous motors. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3996–4001, Dec 2006.
- [90] J. M. M. Rovers, J. W. Jansen, and E. A. Lomonova. Analytical calculation of the force between a rectangular coil and a cuboidal permanent magnet. *IEEE Transactions on Magnetics*, 46(6):1656–1659, June 2010.
- [91] S. D. Ruben and T.-C. Tsao. Optimal commutation laws by real-time optimization for multiple motor driven systems. In *American Control Conference (ACC), 2010*, pages 1942–1947, June 2010.
- [92] S. D. Ruben and T.-C. Tsao. Real-time optimal commutation for minimizing thermally induced inaccuracy in multi-motor driven stages. *Automatica*, 48(8):1566 – 1574, 2012.
- [93] P. Samuelsson, H. Norlander, and B. Carlsson. An integrating linearization method for Hammerstein models. *Automatica*, 41(10):1825 – 1828, 2005.
- [94] S. Shinnaka and T. Sagawa. New optimal current control methods for energy-efficient and wide speed-range operation of hybrid-field synchronous motor. *Industrial Electronics, IEEE Transactions on*, 54(5):2443–2450, Oct 2007.
- [95] G. Shorack. *Probability for Statisticians*. Springer, 2014.
- [96] T. Söderström and P. Stoica. Instrumental variable methods for system identification. *Circuits, Systems and Signal Processing*, 21(1):1–9, Jan 2002.
- [97] G. Sun and Z. Ma. Practical tracking control of linear motor with adaptive fractional order terminal sliding mode control. *IEEE/ASME Transactions on Mechatronics*, 22(6):2643–2653, Dec 2017.
- [98] K. K. Tan, S. N. Huang, and T. H. Lee. Robust adaptive numerical compensation for friction and force ripple in permanent-magnet linear motors. *IEEE Transactions on Magnetics*, 38(1):221–228, Jan 2002.



- [99] T. Therdbanker, P. Sanposh, N. Chayopitak, and H. Fujita. Parameter identification of a linear permanent magnet motor using particle swarm optimization. In *The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 173–177, May 2010.
- [100] C. S. Ting, Y. N. Chang, B. W. Shi, and J. F. Lieu. Adaptive backstepping control for permanent magnet linear synchronous motor servo drive. *IET Electric Power Applications*, 9(3):265–279, 2015.
- [101] J. Tjønnås and T. A. Johansen. Adaptive control allocation. *Automatica*, 44(11):2754 – 2765, 2008.
- [102] G. Torrisi, S. Grammatico, R. S. Smith, and M. Morari. A projected gradient and constraint linearization method for nonlinear model predictive control. *ArXiv e-prints*, October 2016.
- [103] G. Torrisi, S. Grammatico, R. S. Smith, and M. Morari. A variant to sequential quadratic programming for nonlinear model predictive control. In *IEEE 55th Conference on Decision and Control (CDC)*, pages 2814–2819, Dec 2016.
- [104] D. T. E. H. van Casteren, J. J. H. Paulides, and E. A. Lomonova. 3-d semianalytical surface charge model including relative permeability using polynomial approximation. *IEEE Transactions on Magnetics*, 50(11):1–4, Nov 2014.
- [105] C. M. M. van Lierop. *Magnetically levitated planar actuator with moving magnets*. PhD thesis, TUE : Department of Electrical Engineering, 2008.
- [106] C. M. M. van Lierop, J. W. Jansen, A. A. H. Damen, E. A. Lomonova, P. P. J. van den Bosch, and A. J. A. Vandenput. Model-based commutation of a long-stroke magnetically levitated linear actuator. *IEEE Transactions on Industry Applications*, 45(6):1982–1990, Nov 2009.
- [107] M. Verhaegen and D. Westwick. Identifying MIMO Hammerstein systems in the context of subspace model identification methods. *International Journal of Control*, 63(2):331–349, 1996.
- [108] R. Verschueren, N. van Duijkeren, R. Quirynen, and M. Diehl. Exploiting convexity in direct optimal control: a sequential convex quadratic programming method. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1099–1104, Dec 2016.
- [109] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.
- [110] S. Wendel, A. Dietz, and R. Kennel. Model predictive position control for permanent magnet synchronous linear motors. In *Innovative Small Drives and Micro-Motor Systems; 11th GMM/ETG-Symposium*, pages 1–6, Sept 2017.

- [111] A. P. Wu and P. L. Chapman. Simple expressions for optimal current waveforms for permanent-magnet synchronous machine drives. *Energy Conversion, IEEE Transactions on*, 20(1):151–157, March 2005.
- [112] D. Zhang, C. Kong, and Y. Chen. Modeling and precision control of permanent magnet linear motors. *IFAC Proceedings Volumes*, 41(2):2258 – 2263, 2008. 17th IFAC World Congress.
- [113] S. Zhao and K. Tan. Adaptive feedforward compensation of force ripples in linear motors. *Control Engineering Practice*, 13(9):1081 – 1092, 2005.
- [114] Y. W. Zhu, K. S. Jung, and Y. H. Cho. The reduction of force ripples of PMLSM using field oriented control method. In *2006 CES/IEEE 5th International Power Electronics and Motion Control Conference*, volume 2, pages 1–5, Aug 2006.
- [115] Z. Q. Zhu, D. Howe, E. Bolte, and B. Ackermann. Instantaneous magnetic field distribution in brushless permanent magnet DC motors. I. Open-circuit field. *Magnetics, IEEE Transactions on*, 29(1):124–135, Jan 1993.

# List of Abbreviations

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CLM	Coreless linear motor
DOF	Degree of freedom
EQP	Equality-constrained quadratic programming
FEM	Finite element method
GGN	Generalized Gauss-Newton
IQP	Inequality-constrained quadratic programming
IV	Instrumental variable
KKT	Karush–Kuhn–Tucker
LMPC	Linear model predictive control
LS	Least squares
MISO	Multiple-input single-output
MPC	Model predictive control
NARX	Nonlinear autoregressive exogenous
NLP	Nonlinear programming
NMPC	Nonlinear model predictive control
NOE	Nonlinear output error
OC	Optimal commutation
PID	Proportional–integral–derivative
PM	Permanent magnet
QP	Quadratic programming
SCP	Sequential convex programming
SCQP	Sequential convex quadratic programming
SISO	Single-input single-output
SQP	Sequential quadratic programming



# Acknowledgments

I would like to express my sincere gratitude to all the people who have contributed in one way or another to the completion of this thesis.

First of all, I would like to thank my supervisor Mircea Lazar for his constant support and guidance. You have always been a devoted supervisor ever since I started my internship with you during my Master's studies. I very much appreciate the time and effort you have invested in my project. Your scientific guidance has been crucial to the success of this project. Thank you for your continuous support and encouragement during the most difficult time of my Ph.D. journey. Your optimism and enthusiasm in research have been a great source of inspiration for me.

Second, I would like to express my gratitude to my second promotor Hans Butler. This thesis could not have been completed without your guidance. You have always been providing me with scientific suggestions that are not only theoretically interesting but also practically useful. Your intensive involvement in the design process has led to the successful construction of the experimental setup which is of great importance for my research. Thank you very much for all your support and guidance along the years.

Next, I would like to thank my first promotor Paul Van den Hof for his guidance and advice. I appreciate the freedom you have given me in my research while still providing support when needed. Thank you for your support in building the experimental setup for my research. Also, I very much appreciate your help with my research on identification. Furthermore, I want to thank you for your valuable feedback on my thesis draft. It has helped me to structure my research questions and strengthen the line of reasoning.

I am also very grateful to the members of my Ph.D. committee, Prof. Moritz Diehl, Prof. Marion Gilson-Bagrel, Prof. Henk Nijmeijer and Prof. Elena Lomonova. Thank you for accepting to be part of the committee and for your suggestions and remarks which have greatly improved my thesis.

In the beginning of my Ph.D. research, my knowledge on electrical machines was limited. I would like to thank Bart de Bruyn and Dave van Casteren from the EPE group for helping me with my questions on modeling of electrical machines. I am grateful to Peter Heijmans and Patricia Vreugdewater from ASML for their help with the FEM software Opera.

As part of my Ph.D. research, an experimental linear motor setup has been built

with the help of many people. I would like to thank Wim Symens, Wouter Aangenent and Joost Bolder for providing me with the linear motor and amplifiers. I want to thank Erwin Dekkers and Patrick de Laat for the mechanical design and construction of the setup. I am grateful to Jeroen van Duivenbode for his help in making the power amplifiers working. A special thank goes to Will Hendrix for the huge amount of time and effort he has spent on the hardware design and software implementation of the setup. The setup would not have been operational without your help. Furthermore, I would like to thank Prof. Johan Schoukens for the helpful discussions on identification of the setup which helped me gain new insight into the problem. I also want to thank Veaceslav for his advice in the early development of the setup. Thank you Ioannis for your tips and tricks with the experiments.

My time as a Ph.D. student has been really enjoyable thanks to my colleagues. Special thanks go to my officemates Constantijn, Henrik, Amritam, Daming, Bahadir, Paul, Giuseppe, Alina, Thinh, Koen, Handian for a nice working environment. I want to especially thank Yanin for the friendship over the years. Veaceslav, thank you for many helpful technical discussions and for the fun time drinking together. Daming, I really enjoyed our conversations and gym hours. Thank you Ruxandra for driving me safely through Death Valley under unfavorable weather conditions. To the members of Atlético Bitterballen, formerly known as the Team of Rob, I enjoyed playing futsal with you and thank you for the great team spirit. I would like to thank Pepijn, Ioannis, Mohsin, Sofie, Edwin, Dhruv, David, Alejandro, Ruben, Marcella, Harm, Esmail, Giulio, Jurre, Shenling, Maarten, Carlos, etc. for creating a pleasant atmosphere in the group. I am also very grateful to Barbara, Diana, Lucia and Hiltje for taking care of the organizational aspects.

I would like to thank my Vietnamese housemates Khanh, Cuong, Binh, Hoang for the memorable time together. Thank you Viet Anh, Trung and Khanh for being my friends and sharing your ideas and thoughts.

Finally, I would like to express my deepest gratitude to my parents and my sister. Your unconditional support has always been my greatest source of strength. Thank you for always being by my side.

Tuan Trong Nguyen  
Eindhoven, August 2018

# Curriculum Vitae

Tuan Trong Nguyen was born on February 4, 1988 in Hanoi, Vietnam. He received his B.Sc. degree in Industrial Automation from Hanoi University of Science and Technology, Vietnam in 2010. In 2013, he received his M.Sc. degree (cum laude) in Electrical Engineering, with specialization in Control Systems, from Eindhoven University of Technology, The Netherlands.

Thereafter, he started working towards his Ph.D. degree at the Control Systems group of the Electrical Engineering Department at the same institution. His Ph.D. project deals with identification and compensation of parasitic effects in coreless linear motors. The research is carried out in collaboration with ASML under the guidance of prof.dr.ir. P.M.J. Van den Hof, prof.dr.ir. H. Butler and dr. M. Lazar. The results of the research are presented in this thesis.

