Markus Hirvonen

ON THE ANALYSIS AND CONTROL OF A LINEAR SYNCHRONOUS SERVOMOTOR WITH A FLEXIBLE LOAD

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in Auditorium 1383 at Lappeenranta University of Technology, Lappeenranta, Finland, on 8th December, 2006, at noon.

Acta Universitatis
Lappeenrantaensis
257

Supervisor	Professor Heikki Handroos Department of Mechanical Engineering Lappeenranta University of Technology Finland
Reviewers	Associate Professor Alberto Bellini Department of Electric Machine and Drives University of Modena and Reggio Emilia Modena, Italy
	Professor Finn Conrad Department of Mechanical Engineering Technical University of Denmark Lyngby, Denmark
Opponents	Professor Torben Andersen Institute of Energy Technology Aalborg University Denmark
	Professor Finn Conrad Department of Mechanical Engineering Technical University of Denmark Lyngby, Denmark

ISBN 952-214-320-0
ISBN 952-214-321-9 (PDF)
ISSN 1456-4491

ABSTRACT

Markus Hirvonen

On the Analysis and Control of a Linear Synchronous Servomotor with a Flexible Load

Lappeenranta 2006

120 p.

Acta Universitatis Lappeenrantaensis 257

Diss. Lappeenranta University of Technology

ISBN 952-214-320-0, ISBN 952-214-321-9 (PDF), ISSN 1456-4491

Industry's growing need for higher productivity is placing new demands on mechanisms connected with electrical motors, because these can easily lead to vibration problems due to fast dynamics. Furthermore, the nonlinear effects caused by a motor frequently reduce servo stability, which diminishes the controller's ability to predict and maintain speed. Hence, the flexibility of a mechanism and its control has become an important area of research. The basic approach in control system engineering is to assume that the mechanism connected to a motor is rigid, so that vibrations in the tool mechanism, reel, gripper or any apparatus connected to the motor are not taken into account. This might reduce the ability of the machine system to carry out its assignment and shorten the lifetime of the equipment. Nonetheless, it is usually more important to know how the mechanism, or in other words the load on the motor, behaves.

A nonlinear load control method for a permanent magnet linear synchronous motor is developed and implemented in the thesis. The purpose of the controller is to track a flexible load to the desired velocity reference as fast as possible and without awkward oscillations. The control method is based on an adaptive backstepping algorithm with its stability ensured by the Lyapunov stability theorem. As a reference controller for the backstepping method, a hybrid neural controller is introduced in which the linear motor itself is controlled by a conventional PI velocity controller and the vibration of the associated flexible mechanism is suppressed from an outer control loop using a compensation signal from a multilayer perceptron network. To avoid the local minimum problem entailed in neural networks, the initial weights are searched for offline by means of a differential evolution algorithm. The states of a mechanical system for controllers are estimated using the Kalman filter.

The theoretical results obtained from the control design are validated with the lumped mass model for a mechanism. Generalization of the mechanism allows the methods derived here to be widely implemented in machine automation. The control algorithms are first designed in a specially introduced nonlinear simulation model and then implemented in the physical linear

motor using a DSP (Digital Signal Processor) application. The measurements prove that both controllers are capable of suppressing vibration, but that the backstepping method is superior to others due to its accuracy of response and stability properties.

Keywords: Backstepping, dynamic model, linear motor, vibration control.

UDC 621.313.32 : 681.587.7 : 681.532.1

ACKNOWLEDGEMENTS

The research work for this thesis was carried out during the years 2001-2006 in the Institute of Mechatronics and Virtual Engineering at the Department of Mechanical Engineering of Lappeenranta University of Technology as part of research projects financed by the Academy of Finland and the National Technology Agency of Finland (TEKES).

I would like to express my deepest thanks to all the people who have influenced my work. First I would express my appreciation for my supervisor, Professor Heikki Handroos, who made this interesting research possible. I would also thank Professor Olli Pyrhönen of the Department of Electrical Engineering for his valuable comments and advice in the field of control engineering. I would like to express my gratitude to the personnel of the Institute of Mechatronics and Virtual Engineering for their support in my work, especially Hassan Yousefi, M.Sc., for his help in the field of neural networks and evolution algorithms. I am very grateful to secretary Kaija Tammelin for her kind help in practical details. I would also like to thank Mr. Malcolm Hicks for his valuable work in proof-reading the final manuscript.

I would also like to thank the reviewers of the thesis, Associate Professor Alberto Bellini of the University of Modena and Reggio Emilia, Italy, and Associate Professor Finn Conrad of the Technical University of Denmark, Lyngby, Denmark.

My great appreciation for financial encouragement goes to the Finnish Cultural Foundation and its South Carelian Regional Fund, the Jenny and Antti Wihuri Foundation, the Walter Ahlström Foundation and the Technology Foundation.

To my Mother and Father

Hämeenlinna, 12th November 2006 Markus Hirvonen

CONTENTS

NOMENCLATURE

Roman letters

A_{ri}	amplitude of harmonic i in the cogging force model
$\mathbf{A}$	state matrix
b	damping coefficient
b_{e}	equivalent damping
b_{k}	bias term
b_{s}	structural damping
$\mathbf{B}$	input matrix
c_{i}	ith controller gain
C	capacitance
CR	crossover control constant
$\mathbf{C}$	output matrix
$\mathcal{C}$	controllability matrix
D	number of network weights
$\mathbf{D}$	feedthrough matrix
e	error
$\dot{e}$	time derivative of error e
E	error function
f	function, frequency in hertz
f_{mod}	modulation frequency
f_{n}	nominal frequency
F	force, scale factor in differential evolution
F_{cogging}	cogging force
F_{detent}	detent force
F_{dist}	disturbance force
F_{dx}	electromagnetic thrust in the direction x
F_{max}	maximum force
F_{n}	nominal force
F_{s}	spring force
F_{set}	force command
F_{tot}	resultant force
F_{μ}	friction force
F^{*}	force command
$\hat{F}_{s}$	estimated spring force
g	function
$\mathbf{H}$	observer gain, discrete output matrix
i	current
i_{A}, i_{B}, i_{C}	phase currents
i_{f}	magnetizing current
i_{ad}	d-axis armature current
i_{aq}	q-axis armature current
i_{D}	d-axis damper winding current
i_{max}	maximum current
i_{n}	nominal current

i_{qset}	q-axis current command
i_{Q}	q-axis damper winding current
i_{set}	current command
i_0	zero current
$\overline{i}$	current vector in stationary frame
k	spring constant
K_{d}	derivative gain of a controller
K_{i}	integral gain of a controller
K_{m}	motor constant
K_{p}	proportional gain of a controller
K_{s}	scaling factor
K_1, K_2, K_3	Kalman gains
$\mathbf{K}$	feedback gain
$\mathbf{K}_{\mathrm{e}}$	observer gain
$\mathbf{K}_{\mathrm{k}}$	Kalman gain
L	inductance or stroke
L_{ad}	d-axis armature self-inductance
L_{aq}	q-axis armature self-inductance
$L_{a\sigma}$	leakage inductance
L_{md}	d-axis magnetizing inductance
L_{mq}	q-axis magnetizing inductance
L_{sd}	d-axis resultant armature inductance
L_{sq}	q-axis resultant armature inductance
L_{D}	d-axis damper winding inductance
$L_{\mathrm{D}\sigma}$	d-axis leakage inductance
L_{Q}	q-axis damper winding inductance
$L_{\mathrm{Q}\sigma}$	q-axis leakage inductance
NP	population size
$\mathcal{O}$	observability matrix
p	pole pair number
p_{elm}	electromagnetic power
p_{in}	power input to three-phase armature
p_{n}	nominal power
$\mathbf{P}_{\mathrm{G}}$	population
$\mathbf{P}_{\mathrm{k}}$	estimation error covariance
r	measurement residual
R	armature winding resistance
R_{D}	d-axis damper winding resistance
R_{Q}	q-axis damper winding resistance
$\mathbf{R}$	measurement covariance
$\mathbf{Q}$	noise covariance
t	time
t_{k}	discrete time
T_{s}	sampling time
u	control input
u_{d}	d-axis terminal voltage
u_{n}	motor voltage

u_q	q-axis terminal voltage
u_s	inverter bus voltage
$\overline{u}$	voltage vector in a stationary frame
v_A, v_B, v_C	phase voltages
v	velocity
v_k	output of a neuron's summing junction
v_n	max. speed at nominal thrust
v_s	synchronous velocity
v_l	load velocity
v_m	mover velocity
V_i	*i*th Lyapunov function
V	Lyapunov function
$\mathbf{v}$	mutated vector or measurement noise
w	synaptic weight
$\mathbf{w}$	vector of synaptic weights or random process
W	continuous function
$\mathbf{W}_k$	innovation covariance
x_i	state *i*
x_L	load position
x_M	mover position
x_0	initial condition
$\dot{x}_d$	desired velocity
$\dot{x}_L$	load velocity
$\dot{x}_M$	mover velocity
$\dot{x}_{ref}$	reference velocity
$\ddot{x}_L$	load acceleration
$\ddot{x}_M$	mover acceleration
$\hat{\ddot{x}}_L$	estimated load acceleration
$\mathbf{x}$	state vector, or vector of inputs
$\hat{\mathbf{x}}$	estimated state vector
y_k	output of a neuron
$\mathbf{y}$	output vector

Greek letters

α_{init}	initial angle between stator and rotor
α	virtual control
$\dot{\alpha}$	time derivative of virtual control
γ	adaptation gain
γ_i	coefficient of the Stribeck effect in the friction model
$\mathbf{\Gamma}$	discrete input matrix
δ	logarithmic decrement, or adaptation gain
Δx	position difference
$\Delta \dot{x}$	velocity difference
$\Delta \mathbf{w}$	adjustment term for synaptic weights

ε	expectation
θ	unknown parameter in backstepping
θ	angle between coordinates
$\hat{\theta}$	estimate for an unknown parameter
θ_1	$=-k/m_M$; unknown parameter
θ_2	$=k/m_L$; unknown parameter
ξ	estimation error, or damping ratio
σ_v	standard deviation of the measurement noise
σ_w	standard deviation of the process noise
τ	pole pitch
φ_i	wavenumber of the *i*th harmonic
$\varphi(.)$	activation function
$\mathbf{\Phi}$	discrete state matrix
ψ	flux linkage
ψ_d	d-axis flux linkage
ψ_D	d-axis flux linkage of damping windings
ψ_{PM}	flux linkage of permanent magnet
ψ_q	q-axis flux linkage
ψ_Q	q-axis flux linkage of damping windings
$\bar{\psi}_s$	flux linkage vector in a stationary frame
ω	electrical angular velocity
ω_n	natural frequency in radians
$\bar{\omega}$	vector of angular velocity

Subscripts

a	armature
a0	zero armature current
A,B,C	phases
cogging	cogging force
d	direct axis
detent	detent force
dist	disturbance
dx	x-directional
D	direct axis component of damping winding
e	equivalent
elm	electromagnetic
f	field magnetizing
G	generation to which a population belongs
i	integral
k	discrete sampling
L	load
m	magnet
mod	modulation
M	mover
n	nominal or neural

p	proportional
pm	permanent magnet
q	quadrature axis
Q	quadrature axis component of damping winding
ref	reference
s	synchronous, stator, structural or spring
v	measurement noise
w	process noise
σ	leakage
μ	friction

Superscripts

*	command signal or desired value
º	angle
T	transpose
-	a priori
^	estimate

Acronyms

AC	Alternating Current
ACC	American Control Conference
ANN	Artificial Neural Network
CLF	Control Lyapunov Function
DC	Direct Current
DE	Differential Evolution
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
FFT	Fast Fourier Transformation
GA	Genetic Algorithm
GAS	Global Asymptotic Stability
ISS	Input to State Stability
ITAE	Integral of Time multiplied by the Absolute value of Error (criterion)
LKF	Linear Kalman Filter
LQ	Linear Quadratic (control theory)
LQG	Linear Quadratic Gaussian (control theory)
MLP	Multilayer Perceptron
NeFeB	Neodymium-Iron-Boron (magnet material)
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PM	Permanent Magnet
PMLSM	Permanent Magnet Linear Synchronous Motor
SPM	Surface Permanent Magnet
PWM	Pulse Width Modulation
SPWM	Sinusoidal Pulse Width Modulation
UAS	Uniformly Asymptotic Stability
ZOH	Zero-Order Hold

Chapter 1

1 Introduction

1.1 The Motor System to be Studied

Industry's growing need for faster development times and more accurate manufacturing is placing new demands on the technology used in automation. Systems are becoming more and more complex, consisting of mechanisms, actuators, controllers and sensors. Even simple components will contain sensors which are used as a feedback for their controllers. Conventional hydraulic and pneumatic systems and electric motors with gearings have been superseding by new forms of motor and drive technology. In particular, permanent magnet direct drive motors are becoming more and more popular in machine automation nowadays. The advantages of permanent magnet motor drives are their gearless structure, better control characteristics and better efficiency. Permanent magnet motors are used in lifts, paper machines, propulsion units of ships, windmills *etc*.

One of the special applications of permanent magnet motors is a permanent magnet linear synchronous motor (PMLSM), the moving part (mover) of which consists of a slotted armature and three-phase windings, while the surface permanent magnets (SPMs) are mounted along the whole length of the path (stator). The principle of a PMLSM is shown in Figure 1.1. The linear motor is an old invention, but it is only recently that, as a result of the development of permanent magnets and their reduced costs, permanent magnet linear motors have become a viable alternative to rotating motors fitted with linear transmission. In machine automation, linear movement has traditionally been produced by pneumatics or hydraulics, or transmitted from a rotary actuator by means of a ball screw, rack and pinion or belt. A linear motor provides linear motion without the potential complications associated with pneumatics

and hydraulics, and simplifies the mechanical structure of transmission, eliminating the contact-type nonlinearities caused by backlash and compliance. Additionally, the main benefits of a linear motor include high force without sacrificing speed or precision. Linear motors are therefore the most natural choice in applications where accurate positioning is needed, high demands are placed on the dynamics, and the movement required is over one metre. In longer strikes the traditional rotating motor with linear transmission, *e.g.* a ball-screw, has detrimental resonance frequencies, upon which the overall velocity becomes too slow. Linear motors are mainly used in CNC, assembly, and manufacturing machines.

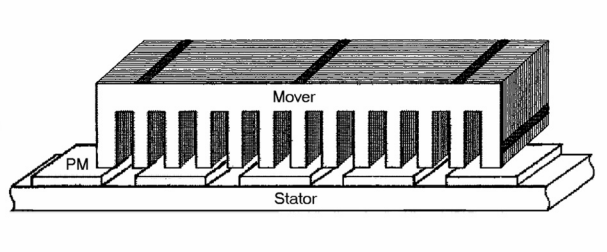


Figure 1.1. The principle of a PMLSM.

The need for linear motors in Finland is greatest in the mechanical wood processing and electronics industries. A greatest barrier to permanent magnet linear motors becoming the standard actuators in industry is their price, which is due to the expensive materials used in permanent magnets. NeFeB magnets have expensive components such as the rare earth metal neodymium. Prices are expected to drop in response to rapidly growing production and to developments in manufacturing technology. The price of a linear motor is dependent on the length of the motor, and is usually 1.5 to 3 times higher than that of a traditional rotating motor fitted with linear transmission. The supply of linear motors is much better nowadays than it was three years ago, for example, and this will have an advantageous effect on the forces produced by them and on their prices, which have dropped by 10 to 15 percent annually over the last couple of years [1].

The mechanism connected to a linear motor should be very stiff and light, with as high a natural frequency as possible, which usually means that it must be optimized for these things. If it does not deliver equally high performance, the benefit of the linear motor will be lost, and if the system does not have adequate stiffness the linear motor may be no more effective than

less expensive alternatives [2]. On the other hand, making the system more rigid through constructional modifications will result in an inefficient, slow, heavy mechanism. One possibility is to make the mechanism lighter and counteract the deformation with a controller. An attempt has been made to solve this problem in the field of flexible lightweight robots [3] [4], yielding systems that are lighter, faster and less expensive than their rigid counterparts and have a better load-carrying capacity and lower energy consumption, use smaller actuators, are safer to operate due to the reduction in inertia, are easier to transport and have a more compliant structure for assembly [5]. In some cases, structural flexibility may even be desirable in order to avoid damage in the process or in the case of accident. Studying the present case the mechanism was chosen to be very flexible in order to test the ability of the controller to suppress mechanical vibrations in such systems. Based on a generalization of the concept of a flexible mechanism, the mechanism is modelled as a lumped mass spring, a model that is suitable for estimating the performance of the actual system.

1.2 Vibration Suppression Methods for Flexible Mechanisms

A great deal of attention is usually paid to the control of the linear motor itself, as in [6], where a neural network controller is implemented for linear motor motion control. In such cases vibrations in the tool mechanism, reel, gripper or any apparatus connected to the motor are not taken into account. This might reduce the ability of the machine system to carry out its assignment and impair the lifetime of the equipment. Nonetheless, it is usually more important to know how the load on the motor behaves. Load control, in other words the control of flexible mechanisms, has come increasingly into focus recently, although mechanical flexibilities with motor nonidealities make the design of a control system a very challenging task.

This thesis compares possible control strategies for suppressing mechanical vibrations in a PMLSM. The derived controllers are of the velocity type, which is very common in many applications, especially in manufacturing. Velocity control loops can also be used as inner loops in more complex systems. For example, it is easy to implement a position loop for a derived velocity controller. The aim of derived load controllers is to drive a flexible load to a velocity reference value in such a way that the load follows the reference value as accurately

as possible but without awkward vibration. The reference can be a certain velocity profile, which is a set of velocities vs. time. Velocity profiles may include several speed levels maintained over given time intervals, or they can be programmed to change on the basis of test results or a given arbitrary trajectory.

One of the most traditional methods of suppressing resonance in an electromechanical system is to allow only slow changes in the reference command. Various kinds of filter are used in a reference signal to suppress mechanical vibrations, for example, which means that the bandwidth of the motor system is limited so that the natural mechanical frequencies are not excited. Dumetz *et al.* [7] have studied bi-quad and low pass filters in a control loop, and also as reference filters, and another filter that is widely used for vibration suppression is the Notch filter [8]. A closed loop filter makes it possible to compensate for poles and zeros in the transfer function on the motor side, and a reference filter compensates for the poles of the transfer function on the load side. The drawbacks of filtering are its low sensitivity to parameter variations and a reduction in the dynamic properties of a servo system.

It is possible to control and eliminate the oscillation of a flexible mechanism by directly sensing or estimating the tip motion. One method that is widely used in vibration control is state feedback control. This needs not only values for the states to be measured, such as motor force and motor velocity, but also information on an unknown input and internal states. One possibility for obtaining all of these values is to use an estimator. One example of this kind of control system is given in Reference [9], where an LQ-based speed controller with the Kalman filter in a 2-mass motor drive system is studied, but a more promising method is to use acceleration compensation to suppress load vibration. In this method, the motor is controlled by a simple PI controller and load acceleration can be measured or estimated and used as compensation feedback. Kang *et al.* [10] and Lee *et al.* [11] have used this kind of method successfully for vibration control in elevators. Montanaro *et al.* [12] combined the LQG method with acceleration compensation.

The main problem with the methods mentioned above is that they do not ensure the global stability of the system if it contains large nonlinearities. If the task of a control system involves large-range and/or high-speed motions, nonlinear effects will be significant in the

dynamics and nonlinear control may be necessary to achieve the desired performance [13]. Nonlinear controllers such as feedback linearization [14] [15], sliding mode [16] [17], backstepping and neural network controllers [18] are becoming more popular nowadays due to developments in design methods and computer software. The biggest obstacle to the use of nonlinear controllers in industry is their difficult implementation in the case of practical problems.

This thesis introduces a systematic method for nonlinear vibration control in a flexible mechanism, known as adaptive backstepping control. This is a newly developed technique for the control of uncertain nonlinear systems. The numbers of journal and conference papers concerning the backstepping method mentioned in IEEE Xplore [19] in the years 1992 to 2005 are given in Figure 1.2. The total number is:

- 272 conference papers,
- 51 journal papers,
- 3 journal papers and 2 conference papers with backstepping + linear motor in the title.

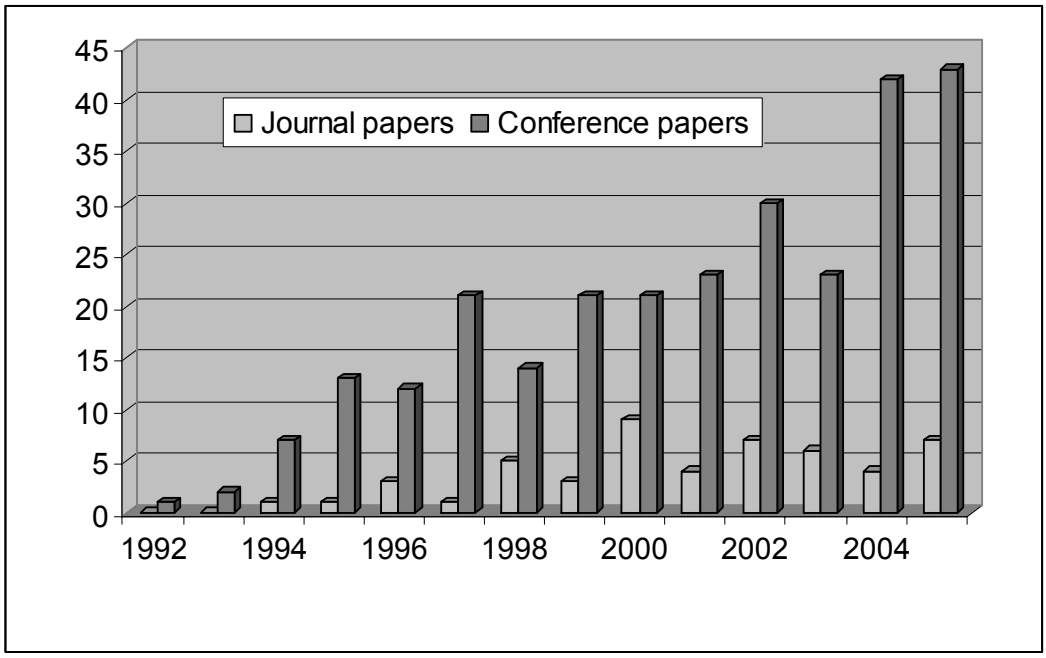


Figure 1.2. Backstepping in the title, 1992-2005 (IEEE Xplore).

The backstepping procedure was first introduced by Kokotovic et al. [20]. The most appealing point of it is that the virtual control variable can be used to make the original high-order system simple, so that the final control outputs can be derived step by step through suitable Lyapunov functions. An adaptive robust nonlinear controller can be straightforwardly derived using this method. Sometimes we may not have a precise knowledge of the system parameters, or the parameter may change with time. In these cases an adaptive control scheme is appropriate.

Adaptive backstepping is a systematic, recursive design methodology for nonlinear feedback control. The idea of backstepping design is to select recursively some appropriate functions of state variables as pseudo-control inputs for lower-dimension subsystems of the overall system. Each backstepping stage results in a new virtual control design, expressed in terms of the virtual control design from the preceding design stage. The procedure terminates a feedback design for the true control input, which is achieved from a final Lyapunov function formed by summing the Lyapunov functions associated with each individual design stage [21]. As backstepping breaks the design problem for a full system into a sequence of design problems for lower-order subsystems, it can often solve control problems under conditions that are less restrictive than those encountered in other methods.

A hybrid neural controller is introduced as the reference controller. In this method the idea of acceleration compensation for a flexible mechanism is extended so that the compensation signal is produced by a neural network. This is termed a hybrid neural controller. The advantage of neural networks is their ability to learn system behaviour even from noisy signals, while the drawback is their need for loads of data and large calculation capacity. Therefore neural networks cannot be used in systems where insufficient measurement data is available. One possibility for reducing the calculation time is to teach the network in a simulation model. We found the initial weights for the multilayer perceptron (MLP) network by means of the accurate nonlinear simulation model introduced in Chapter 2, using the differential evolution (DE) algorithm. After the weights had been initialized, they were transferred to the neural network compensator in the experimental setup. The derived neural network controller is not of the adaptive type, *i.e.* it uses rigid weights.

It is often not possible to achieve acceptable performance using only state variables that can be measured, and if the system is observable, it is possible to estimate the state variables that are not directly accessible. By using these estimates rather than measured values one can usually achieve an acceptable level of performance. State-variable estimates may in some circumstances even be preferable to direct measurements, because the errors in the instruments that provide the direct measurements may be larger than the errors in estimating these variables. The use of estimated values in control feedback is a consequence of the separation principle, which allows the problem to be decomposed into two sub-problems: the design of a state observer and the design of a controller. Unavailable states of the flexible mechanism are estimated here using the Kalman filter, the theory of which is introduced in Section 3.3.1.

1.3 Scope of the Work and Outline of the Thesis

The scope of this thesis is two-fold and will focus on (i) the development of an analytical nonlinear model for a permanent magnet linear synchronous motor (PMLSM), and (ii) a synthesis of nonlinear adaptive control for a mechanical system with flexibility. The work is distributed as follows.

The nonlinear simulation model for a PMLSM is introduced in Chapter 2. This is used to study different control strategies for mechanical flexibilities. The modelling of a PMLSM is based on the space vector theory introduced in Section 2.1. The nonlinearities in the physical linear motor are analysed and the derived mathematical models introduced in Section 2.1.4, and a simplified model of a flexible mechanism for control system development is introduced in Section 2.2.

Chapter 3 discusses control strategies for flexible mechanisms. Basic linear control methods and their weaknesses for vibration suppression in nonlinear systems are introduced in Section 3.1. The main focus in the thesis is on studying the use of a adaptive backstepping controller for vibration control, as introduced in Section 3.2.1, which this is compared with other nonlinear controller based on a neural network compensator. The basic theory of the method

is introduced in Section 3.2.2. Both controllers use estimated system states due to lack of measured states, and for this purpose the Kalman filter is introduced in Section 3.3.1.

The experimental work and results are introduced in Chapter 4. The nonlinear simulation model is verified with a physical linear motor and the results are compared in Section 4.1. The controllers introduced in Chapter 3 are first derived and analysed in a nonlinear simulation model, after which they are implemented for a physical laboratory setup. The velocity responses measured from the motor are showed in Section 4.5. Chapter 5 summarizes the conclusions reached in the work.

1.4 Contribution of the Work

The work presented in this thesis is mainly based on the following articles in journals and conference proceedings:

Hirvonen, M. J., Pyrhönen, O., and Handroos, H., 2006, "Adaptive Nonlinear Velocity Controller for a Flexible Mechanism of a Linear Motor", *International Journal of Mechatronics*, **16**(5), pp. 279-290.

Hirvonen, M. J., Pyrhönen O., and Handroos, H., 2006, "Force Ripple Compensator for a Vector Controlled PM Linear Synchronous Motor", *Informatics in Control, Automation and Robotics,* **I**, pp. 135-142.

Hirvonen, M. J., and Handroos, H., 2006, "Suppressing Mechanical Vibrations in a PMLSM Using Feed Forward Compensation and State Estimates", in: Arai, E. and Arai, T. (Eds.), *Mechatronics for Safety, Security and Dependability in a New Era*, Elsevier Science, Amsterdam.

Yousefi, H., Hirvonen, M. J., and Handroos, H., "Feedforward Neural Network Compensation and State Estimation in Suppressing Mechanical Vibrations in a PMLSM", *Journal of Control Engineering Practice*, accepted for publication.

The above articles were produced under the supervision of Prof. Olli Pyrhönen (El. Eng.) and Prof. Heikki Handroos (Mech. Eng.). In the fourth article Hassan Yousefi, M.Sc., acted as a co-author and specialist in differential evolution algorithms and neural networks.

The following contributions made by the thesis can be highlighted:

- An accurate dynamic model for a PMLSM system based on the space vector theory.
- An analysis of nonlinearities in a PMLSM.
- Analysis of the control of a flexible mechanism.
- Design and implementation of adaptive backstepping control for a flexible mechanism attached to a PMLSM.
- Implementation of a neural network hybrid controller for a flexible mechanism attached to a PMLSM.
- Use of a differential evolution algorithm to find global minima for the initial weights in a neural network-based controller.
- Output feedback control of a nonlinear system.

Chapter 2

2 Modelling of a PM Linear Synchronous Motor

This chapter introduces a nonlinear model for a permanent magnet linear synchronous motor (PMLSM). Developments in computers and software have made it feasible to simulate the more detailed dynamic behaviour of machine systems. This thesis discusses the use of a detailed nonlinear dynamic model for the analysis of a linear transmission system. All the equations necessary for modelling a complete linear motor system with current controller and external mechanism are introduced. The mathematical modelling of a linear synchronous motor in Section 2.1 is mainly based on the theory of rotating synchronous motors. The main difference between rotating and linear motor models is the finite length of the rotor part in a linear motor, while in a rotating motor the structure is enclosed. The mechanical nonlinearities derived from the physical application of the linear motor are included in the model, their mathematical representations being introduced in Section 2.1.4. Last of all, a model for a flexible mechanism for control system design is introduced in Section 2.2.

2.1 Model for an Electrical Linear Motor

Since the understanding of three-phase voltages and currents in the modelling of AC motors is usually awkward, dynamics modelling of a linear synchronous motor should be based on space-vector theory, in which the model is expressed in the form of a reference frame that rotates at a synchronous speed of ω. All variables are expressed on orthogonal or mutually decoupled direct and quadrature axes which move at a linear synchronous velocity v_s. The synchronous speed in a linear motor is

$$\omega = \frac{\pi v_s}{\tau}, \tag{2.1}$$

where τ is the pole pitch. In order to derive a simplified model for the motor, the stator voltage equation can be given in vector form as [22]

$$\bar{u} = R\bar{i} + \frac{d\bar{\psi}}{dt}, \tag{2.2}$$

where $\bar{u}, \bar{i}$ and $\bar{\psi}$ are instantaneous voltage, current, and flux linkage vectors, respectively, in the stationary frame, and R is the winding resistance. Equation (2.2) can now be written in the form

$$\bar{u} = R\bar{i} + \frac{d\bar{\psi}}{dt} + \bar{\omega} \times \bar{\psi}, \tag{2.3}$$

where the vectors are the same as in Equation (2.2), but the term $\bar{\omega} \times \bar{\psi}$ has been added. The cross product $\bar{\omega} \times \bar{\psi}$ is defined as a speed voltage due to rotating of the reference frame. After determining the cross product, Equation (2.3) can be written in terms of component d-axis and q-axis voltages in a synchronous rotating frame, as

$$u_d = Ri_{ad} + \frac{d\psi_d}{dt} - \omega\psi_q, \tag{2.4}$$

$$u_q = Ri_{aq} + \frac{d\psi_q}{dt} + \omega\psi_d, \tag{2.5}$$

where u_d and u_q are the d-axis and q-axis components of the terminal voltage and i_{ad} and i_{aq} the d-axis and q-axis components of the armature current. The equivalent circuit of the PMLSM is introduced in Figure 2.1.

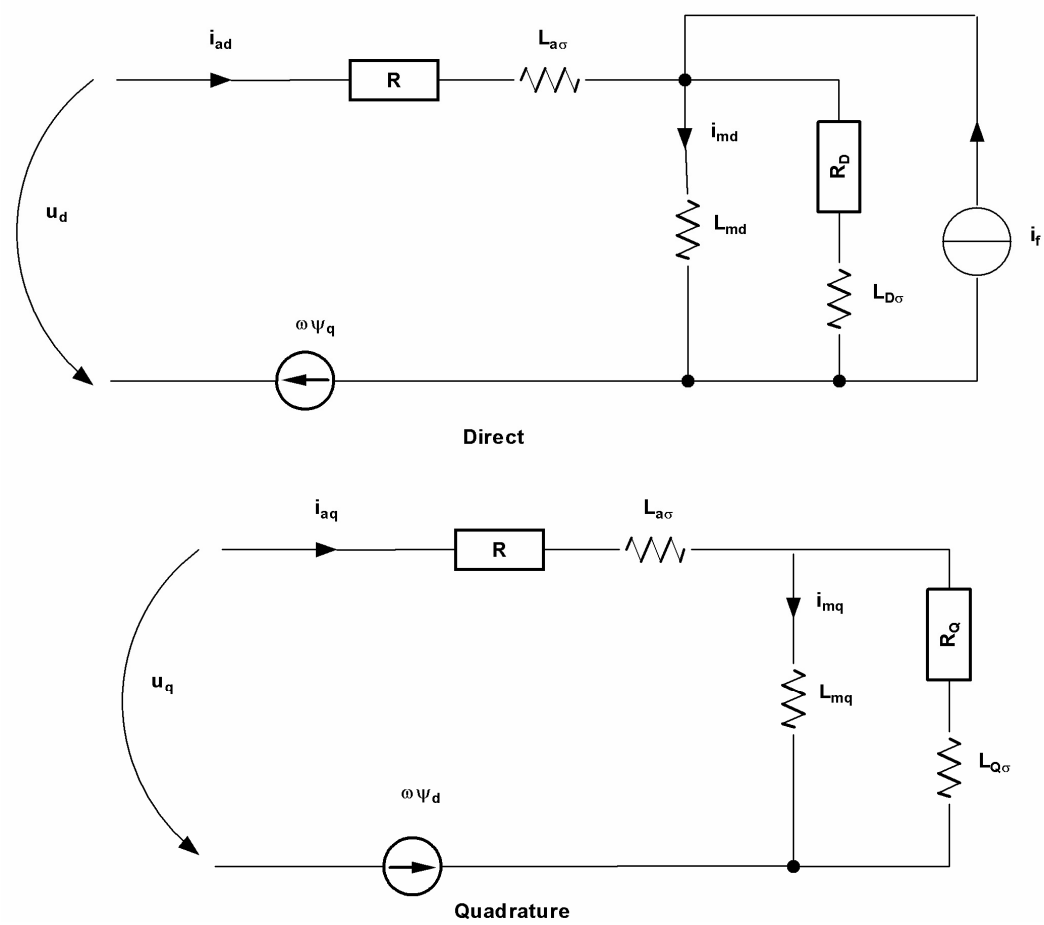


Figure 2.1. Equivalent circuit of a PMLSM.

The armature winding flux linkages in the previous equations and the damper winding flux linkages are

$$\psi_d = L_{ad} i_{ad} + L_{md} i_D + \psi_{pm}, \tag{2.6}$$

$$\psi_q = L_{aq} i_{aq} + L_{mq} i_Q, \tag{2.7}$$

$$\psi_D = L_{md} i_{ad} + L_D i_D + \psi_{pm}, \tag{2.8}$$

$$\psi_Q = L_{mq} i_{aq} + L_Q i_Q, \tag{2.9}$$

where L_{ad} and L_{aq} are the d-axis and q-axis components of the armature self-inductance, L_D and L_Q the d-axis and q-axis components of the damper winding inductance, L_{md} and L_{mq} the d-axis and q-axis components of the magnetizing inductance and ψ_{pm} is the flux linkage of the permanent magnet per phase. Although the physical system does not contain a damper, which in a PMLSM usually takes the form of an aluminium cover on the PMs, virtual damping must

be included in the model due to eddy currents. The voltage equations of the short-circuited damper winding are

$$0 = R_D i_D + \frac{d\psi_D}{dt}, \tag{2.10}$$

$$0 = R_Q i_Q + \frac{d\psi_Q}{dt}, \tag{2.11}$$

where R_D and R_Q are the d-axis and q-axis components of the damper winding resistance and i_D and i_Q the d-axis and q-axis components of the damper winding current. The armature and damper inductances are

$$L_{ad} = L_{md} + L_{a\sigma}, \tag{2.12}$$

$$L_{aq} = L_{mq} + L_{a\sigma}, \tag{2.13}$$

$$L_D = L_{md} + L_{D\sigma}, \tag{2.14}$$

$$L_Q = L_{mq} + L_{Q\sigma}, \tag{2.15}$$

where $L_{i\sigma}$ is the leakage inductance of the respective component. The currents from equation (2.6)-(2.9) are shown below

$$i_{ad} = \frac{1}{A}\left[B\left(\psi_d - \psi_{pm}\right) + C\left(\psi_D - \psi_{pm}\right)\right], \tag{2.16}$$

$$i_{aq} = \frac{1}{A}\left(D\psi_q + E\psi_Q\right), \tag{2.17}$$

$$i_D = \frac{1}{A}\left[C\left(\psi_d - \psi_{pm}\right) + F\left(\psi_D - \psi_{pm}\right)\right], \tag{2.18}$$

$$i_Q = \frac{1}{A}\left(E\psi_d + G\psi_Q\right), \tag{2.19}$$

where A…G are

$$A = L_{ad}L_D(L_{aq}L_Q - L_{mq}{}^2) - L_{md}{}^2(L_{aq}L_Q - L_{mq}{}^2), \tag{2.20}$$

$$B = L_D(L_{aq}L_Q - L_{mq}{}^2), \tag{2.21}$$

$$C = L_{md}(L_{mq}{}^2 - L_{aq}L_Q), \tag{2.22}$$

$$D = L_Q(L_{ad}L_D - L_{md}{}^2), \tag{2.23}$$

$$E = L_{mq}(L_{md}{}^2 - L_{ad}L_D), \tag{2.24}$$

$$F = L_{ad}(L_{aq}L_Q - L_{mq}{}^2), \tag{2.25}$$

$$G = L_{aq}(L_{ad}L_D - L_{md}{}^2). \tag{2.26}$$

Solving the flux linkage differentials from (2.4), (2.5), (2.10) and (2.11), and substituting (2.16) to (2.19) into these equations, we obtain equations which can be used to create the dynamic model for a linear motor, *i.e.*

$$\frac{d\psi_d}{dt} = u_d - \frac{R}{A}\left[B\left(\psi_d - \psi_{pm}\right) + C\left(\psi_D - \psi_{pm}\right)\right] - \omega\psi_q, \tag{2.27}$$

$$\frac{d\psi_q}{dt} = u_q - \frac{R}{A}\left(D\psi_q + E\psi_Q\right) + \omega\psi_d, \tag{2.28}$$

$$\frac{d\psi_D}{dt} = -\frac{R_D}{A}\left[C\left(\psi_d - \psi_{pm}\right) + F\left(\psi_D - \psi_{pm}\right)\right], \tag{2.29}$$

$$\frac{d\psi_Q}{dt} = -\frac{R_Q}{A}\left(E\psi_q + G\psi_Q\right). \tag{2.30}$$

The equation for the electromagnetic thrust based on the previously introduced flux and current equations is derived in the next section.

2.1.1 Electromagnetic Thrust

The instantaneous power input to a three-phase armature is [23]

$$p_{in} = v_A i_{aA} + v_B i_{aB} + v_C i_{aC} = \frac{3}{2}\left(v_d i_{ad} + v_q i_{aq}\right). \tag{2.31}$$

where v_A, v_B and v_C are phase voltages, i_{aA}, i_{aB}, and i_{aC} are phase currents, and v_d and v_q d- and q-axis components. The power balance equation is obtained from Equations (2.4) and (2.5), *i.e.*

$$v_d i_{ad} + v_q i_{aq} = Ri_{ad}^2 + \frac{d\psi_d}{dt} i_{ad} + Ri_{aq}^2 + \frac{d\psi_q}{dt} i_{aq} + \omega\left(\psi_d i_{aq} - \psi_q i_{ad}\right). \tag{2.32}$$

The last term accounts for the electromagnetic power of a single phase, two pole synchronous machine. For a three-phase machine,

$$p_{elm} = \frac{3}{2}\omega\left(\psi_d i_{aq} - \psi_q i_{ad}\right) = \frac{3}{2}\omega\left[\left(L_{sd} i_{ad} + \psi_{pm}\right) i_{aq} - L_{sq} i_{ad} i_{aq}\right] \\ = \frac{3}{2}\omega\left[\psi_{pm} + \left(L_{sd} - L_{sq}\right) i_{ad}\right] i_{aq} \quad . \tag{2.33}$$

where L_{sd} and L_{sq} are the resultant armature inductances. The electromagnetic thrust of a PMLSM with p pole pairs is the electromagnetic power p_{elm} in Equation (2.33) divided by the linear velocity v_s in Equation (2.1), *i.e.*

$$F_{dx} = \frac{p_{elm}}{v_s} = \frac{3}{2} p \frac{\pi}{\tau}(\psi_d i_{aq} - \psi_q i_{ad}) \, . \tag{2.34}$$

A model for the current vector controller will be introduced in the next section.

2.1.2 Current Controller for a Linear Motor

A motion control system can contain three control loops: position, velocity, and force loops. The force loop is usually closed, with an existing drive, and appears as a gain in the motion control system. Force control in the present system is implemented in the form of a current vector control based on the space vector theory of electrical machines, and can therefore be easily implemented in the motor model, which is also based on the space vector theory. It is suitable for force (torque) control in both induction and synchronous motors. The current and flux components are generally analysed separately in vector control theory, using a mathematical model for the machine and separate algorithms to control these components. If the inductance in a motor is low, it is not worth taking the control of the flux component into consideration. In the present vector control the direct axis current i_{ad} is set to zero ($i_d^*=0$), assuming that it has any influence on the force production, *i.e.* Equation (2.34) is transformed to

$$F_{dx} = \frac{\pi}{\tau}(\psi_d i_{aq}) \, . \tag{2.35}$$

This means that the angle θ between the armature current and q-axis always remains at 0° and the thrust is proportional to the armature current $i_a=i_{aq}$. One drawback with vector control is

the low robustness of the machine parameters. In physical systems the resistance values change considerably due to variations in temperature and the inductances are rapidly driven to saturation. A vector controller is nevertheless appropriate for applications which require good dynamics and/or accurate velocity control.

Vector control is presented in many ways in the literature. We use here a simulation principle in which the incoming thrust command F_{set} is converted to a i_{qset} current component by dividing the force value by the motor constant K_m. Current control algorithms are executed in flux coordinates and the outputs of the controllers are transformed back to the rotating reference frame, so that these values u_A, u_B, and u_C control the inverter (Figure 2.2).

The modulation technique used in the simulation model is sinusoidal pulse width modulation (SPWM) with ideal switches. The PWM inverter utilizes the voltage-stiff DC link but combines both voltage control and frequency control within the inverter itself [24].

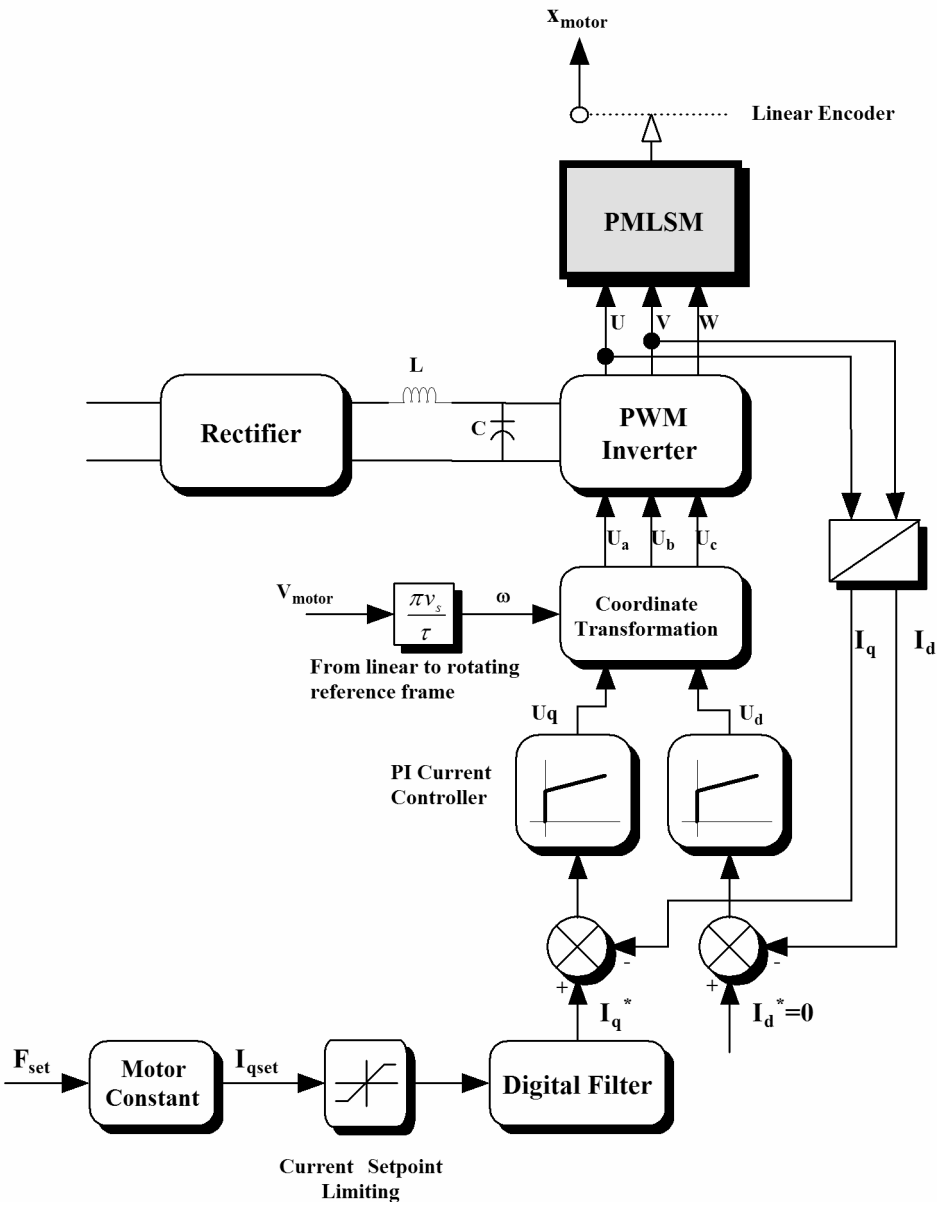


Figure 2.2. Current controller of a PMLSM.

2.1.3 Coordinate Transformations

Coordinate transformations occupy a vital position in vector control mathematics, where complex differential equations are transformed from one coordinate system to another. The following coordinate transformations are performed using armature currents, but the equations are also valid for armature voltages and armature fluxes. The transformations from three-phase to two-phase quantities are

$$\begin{aligned} i_x &= \frac{2}{3}\left[i_A - \frac{1}{2}\left(i_B + i_C\right) \right], \\ i_y &= \frac{1}{\sqrt{3}}\left[i_B - i_C \right], \end{aligned} \tag{2.36}$$

and for zero current

$$i_0 = \frac{1}{3}\left[i_A + i_B + i_C \right]. \tag{2.37}$$

The reverse relations are

$$\begin{aligned} i_A &= i_x + i_0, \\ i_B &= -\frac{1}{2} i_x + \frac{\sqrt{3}}{2} i_y + i_0, \\ i_C &= -\frac{1}{2} i_x - \frac{\sqrt{3}}{2} i_y + i_0. \end{aligned} \tag{2.38}$$

When transferring from a stationary coordinate to a rotating coordinate we also have to know the angle θ between the coordinates. As no zero component is included in the current vector, this possibility must be taken into consideration separately.

$$\begin{aligned} i_{ad} &= i_x \cos\theta + i_y \sin\theta, \\ i_{aq} &= -i_x \sin\theta + i_y \cos\theta, \\ i_{a0} &= i_0, \end{aligned} \tag{2.39}$$

while the inverse relation is

$$\begin{aligned} i_x &= i_{aq} \cos\theta - i_{aq} \sin\theta, \\ i_y &= i_{ad} \sin\theta + i_{aq} \cos\theta, \\ i_0 &= i_{a0}. \end{aligned} \tag{2.40}$$

The transformations straight from phase quantities to rotor coordinates, *i.e.* the relationships between i_{ad}, i_{aq} and the phase currents i_A, i_B and i_C, are

$$
\begin{aligned}
i_{ad} &= \frac{2}{3}\left[i_A \cos\theta t + i_B \cos\left(\theta t - \frac{2\pi}{3} \right) + i_C \cos\left(\theta t + \frac{2\pi}{3} \right) \right], \\
i_{aq} &= -\frac{2}{3}\left[i_A \sin\theta t + i_B \sin\left(\theta t - \frac{2\pi}{3} \right) + i_C \sin\left(\theta t + \frac{2\pi}{3} \right) \right], \\
i_{a0} &= \frac{1}{3}\left[i_A + i_B + i_C \right].
\end{aligned}
\tag{2.41}
$$

The reverse relations, obtained by simultaneous solution of Equation (2.41), are

$$
\begin{aligned}
i_A &= i_{ad} \cos\theta t - i_{aq} \sin\theta t + i_{a0}, \\
i_B &= i_{ad} \cos\left(\theta t - \frac{2\pi}{3} \right) - i_{aq} \sin\left(\theta t - \frac{2\pi}{3} \right) + i_{a0}, \\
i_C &= i_{ad} \cos\left(\theta t + \frac{2\pi}{3} \right) - i_{aq} \sin\left(\theta t + \frac{2\pi}{3} \right) + i_{a0}.
\end{aligned}
\tag{2.42}
$$

The equations required for modelling a PMLSM have now been introduced. To complete the simulation model, the nonlinearities must also be included in the present motor model. These nonlinearities are application-specific, and must always be analysed individually in different applications. This is due to differences in guideway structure (friction) and the shape and configuration of the permanent magnet (cogging). These nonlinearities are introduced in the following section.

2.1.4 Nonidealities of a PMLSM

In high-performance motion systems such as pick-and-place machines, nonlinearities such as detent force and friction can introduce severe negative side-effects such as tracking error or a large settling time. The appropriate response of the system can be achieved if the nonlinearities are taken into consideration in the design of the control system. To this end it is useful that any nonlinearities present in the system should be modelled with the appropriate mathematical models and that the corresponding parameters should be identified. The main nonidealities in PMLSMs are the detent force and friction. The derived detent force model is based on measurements made on a physical application, and the friction model is a numerically effective continuous model which takes viscous, Stribeck and Coulomb effects into consideration.

The total disturbance force equation can be described using the equations for the cogging force, F_{cogging}, and the friction force, F_{μ}; *i.e.* the disturbance force, F_{dist}, is

$$F_{dist} = F_{cogging} - F_{\mu}. \tag{2.43}$$

The cogging force and friction were analysed experimentally using the simple gradient method, in which a linear motor system was placed in a tilted position to produce a constant force F and the currentless moving part allowed to slide down freely, while the motion of the moving part was measured using an acceleration sensor. The equation of motion of this experimental arrangement is

$$m_M \ddot{x}_M = F + F_{dist}. \tag{2.44}$$

where m_{M} and $\ddot{x}_M$ are the mass and acceleration of the mover, respectively. It was known that the friction force F_{μ} is a constant at constant velocity and the cogging force pulsates around zero, so that it was possible to separate these two components. Friction was measured at several tilt angles and the results were used to plot friction as a function of velocity.

Cogging Force

The force ripple of a PMLSM is larger than that of a rotary motor because of the finite length of the mover and the wide slot opening. The cogging force in a PMLSM is caused mainly by the detent force generated between the PMs and the armature. This type of force can be divided into two components; tooth and core-type detent force. The tooth detent force component is generated between the PMs and the primary teeth, while the core-type detent force component is generated between the PMs and the primary core. The wavelength of the core component is usually the length of the primary core, while that of the tooth component is one pole pitch τ. The core-type detent force can be efficiently reduced by optimising the length of the moving part or smooth-forming the edges of the mover, and the tooth-type detent force can be reduced by skewing the magnets and chamfering the edges of the teeth [25] [26] [27] [28] [29].

The detent force effect tends to shift the mover to a position in which the energy of the magnetic circuit is at its minimum. This phenomenon attempts to stall the mover at the stator pole positions and is always present, even when no current is flowing through the coils of the motor [6]. The ripple of the detent force produces both vibrations and noise and reduces controllability [30]. It is dominant at low velocities and accelerations, while at higher velocities it is relatively small and the influence of dynamic effects (acceleration and deceleration) is more prominent [6]. The detent force, F_{detent}, can be described by sinusoidal harmonic functions of the mover position, x_{M}, with a wavenumber of φ and an amplitude of A_{r}, *i.e.*

$$F_{detent} = f_1\left(x_M\right) = K_s \sin(\varphi_1 x_M 2\pi)\left[A_{r1} + A_{r2}\sin(\varphi_2 x_M 2\pi)\right]. \tag{2.45}$$

where the wavenumber φ_1 is $1/\tau$, and φ_2 is related to the primary core length. The parameters of the detent force model are given in Table 1.

Table 1. Parameters of the detent force model

Symbol	Parameter	Value
φ_1	1st wavenumber	67.2 1/m
φ_2	2nd wavenumber	8.5 1/m
A_{r1}	Amplitude of 1st harmonic	35 N
A_{r2}	Amplitude of 2nd harmonic	15 N
K_s	Scaling factor	-0.7

The result of the simulation model is compared with measurements in the reference system in Figure 2.3.

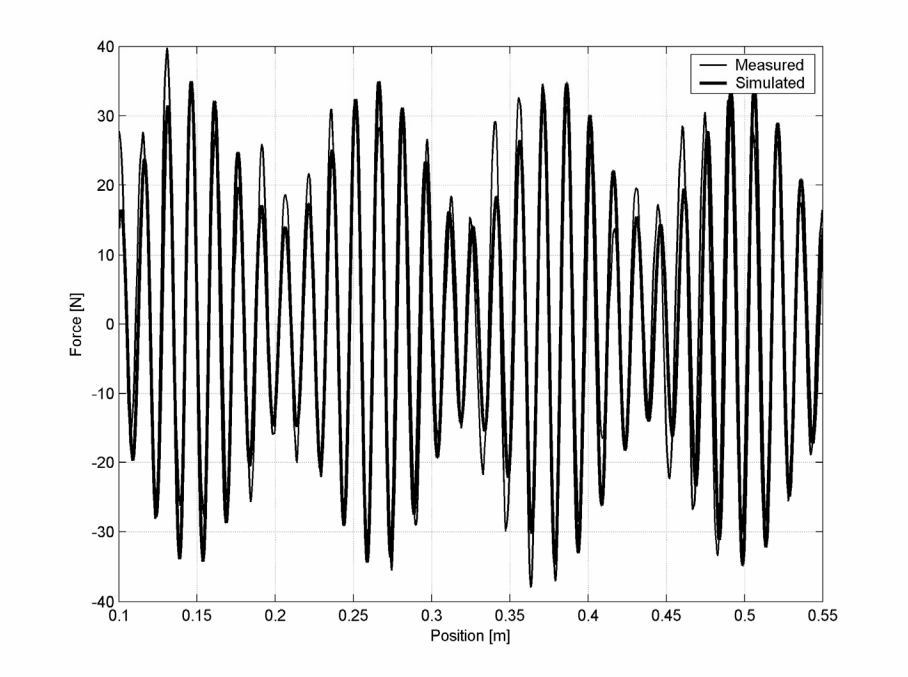


Figure 2.3. Comparison of the measured and simulated detent forces.

The reluctance force is another phenomenon that occurs in linear synchronous motors. A lot of research has been carried out into reluctance in linear induction machines, in which it depends on velocity, but it has been studied to a lesser extent in PMLSMs. The reluctance force is due to variations in the self-inductance of the windings with respect to their relative positions between the mover and the magnets [31]. It was observed to be relatively small in the present reference system, and was therefore not included in the model.

Friction

Friction is very important in control engineering where high-performance systems are concerned and is relatively high particularly in motors of the types discussed here due to the attraction force between permanent magnets and the steel armature. Several friction models have been developed in the past decades [32] [33], but only few are valid for effective simulations, due to their discontinuous behaviour. The main disadvantage when using such models for simulations or control purposes is high discontinuity at near-zero speed, which gives rise to problems such as numerical chatter. Friction is modelled as continuous, in order

to avoid numerical problems, and takes into account the Coulomb (static), viscous (dynamic) and Stribeck effects. The friction force as a function of mover velocity is [34]

$$F_\mu = f_2\left(\dot{x}_M\right) = \gamma_1\left(\tanh\left(\gamma_2\dot{x}_M\right) - \tanh\left(\gamma_3\dot{x}_M\right)\right) + \gamma_4\tanh\left(\gamma_5\dot{x}_M\right) + \gamma_6\dot{x}_M, \quad (2.46)$$

where $\dot{x}_M$ is the velocity of the mover and $\gamma_i \in \mathbb{R}\forall i = 1,2,...,6$ denote unknown positive constants. The parameters of the friction model are shown in Table 2.

Table 2. Parameters of the friction model

Symbol	Parameter	Value
γ_1	Stribeck effect coefficient	50 N
γ_2	Stribeck effect coefficient	100 s/m
γ_3	Stribeck effect coefficient	50 s/m
γ_4	Coulomb coefficient	43.94 N
γ_5	Coulomb coefficient	400 s/m
γ_6	Viscous coefficient	122.043 Ns/m

The continuous friction force model (2.46) is plotted as a function of velocity in Figure 2.4. The force is composed of Coulomb (static), viscous (dynamic) and Stribeck effects.

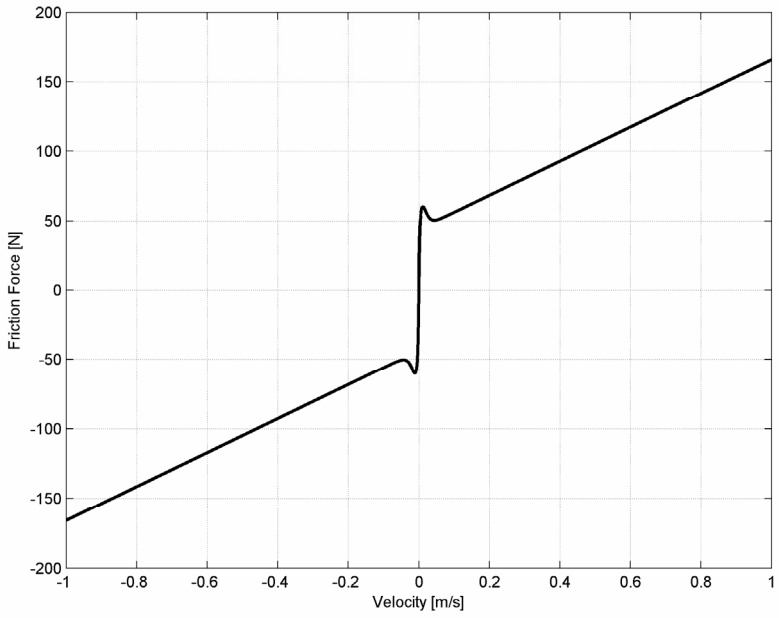


Figure 2.4. Modelled continuous friction force as a function of velocity.

The resultant disturbance force component combines with the electromotive force to influence the dynamic behaviour of the linear motor system. A block diagram model of the PMLSM with mechanical nonlinearities and PI velocity controller is presented in Figure 2.5.

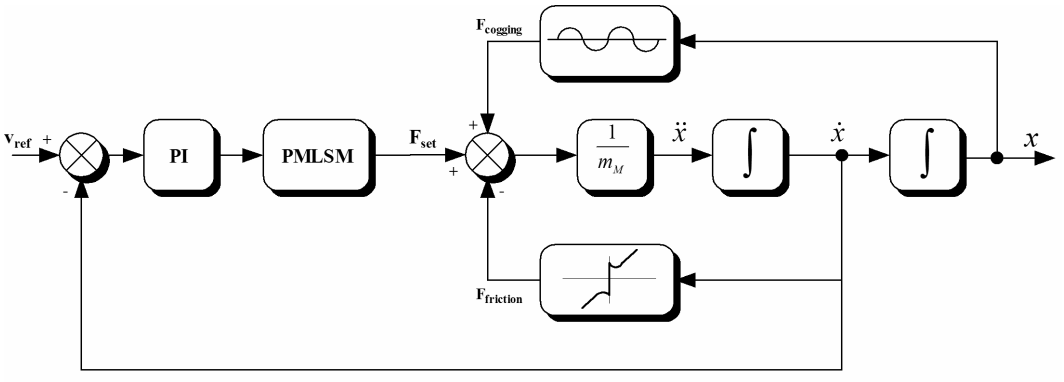


Figure 2.5. Block diagram of the nonlinear PMLSM model.

2.2 The Flexible Mechanism

The traditional approach to the dynamic analysis of mechanisms and machines is based on the assumption that systems are composed of rigid bodies. When a mechanism is operating under high-speed conditions, however, the rigid body assumption is no longer valid and the load should be considered flexible. The flexibility of a mechanism causes a disturbing difference between the reference and load velocity, especially in the fast transient state. The mechanism employed here was constructed according to the ACC benchmark [35], *i.e.* it is a two-mass model, as presented in Figure 2.6.

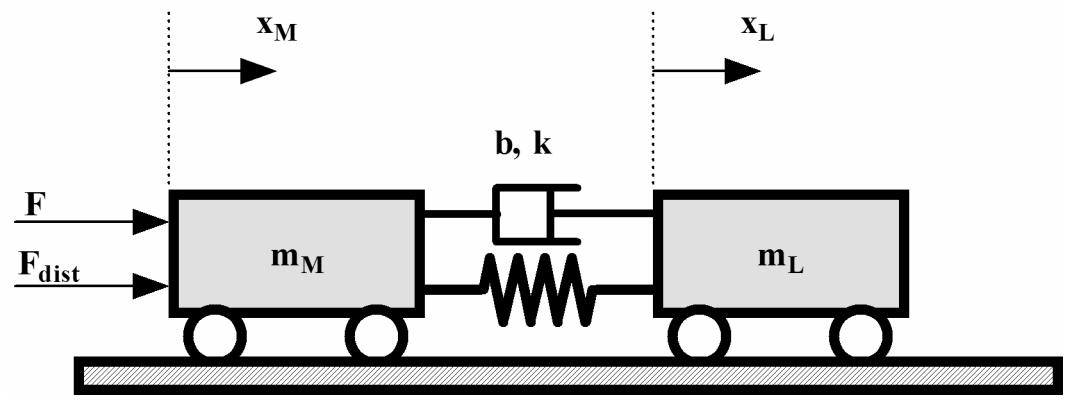


Figure 2.6. Model of the Flexible Mechanism.

The ACC benchmark problem, which is usually used for testing robust controllers, is applied here to the adaptive control problem, due to the fact that most of the flexible mechanism can be simplified as a two-mass model. This conclusion may be reached from the dynamic response when the some term or terms may contribute so small a component to the response that they may be omitted and the order of the model reduced. In the case of flexible structures there are many, dynamic modes present, perhaps even an infinite number, and removing those that have negligible effects is important in simulations. This is clarified in Figure 2.7 in the frequency domain and in a pole/zero plot, where only the dominant natural mode is taken into account.

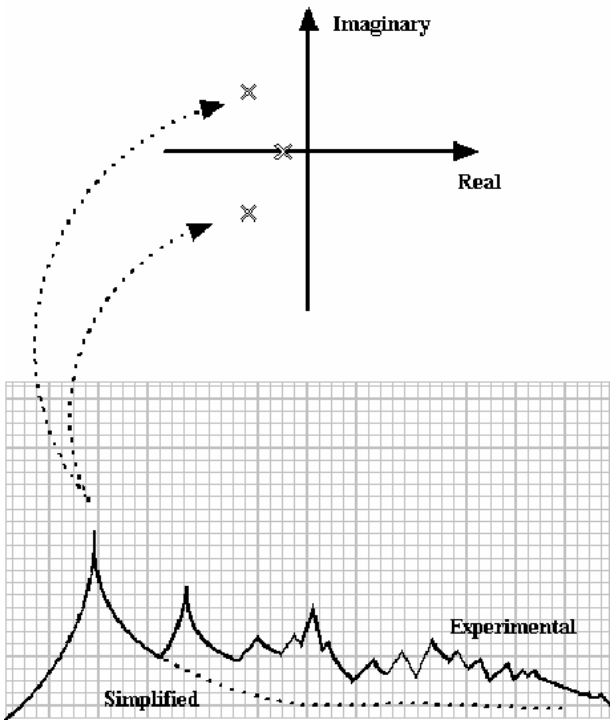


Figure 2.7. Simplification of the mechanism for control system design.

The simplified model has equivalent inertia, damping and stiffness characteristics to the physical counterpart. Several assumptions have to be made in order to obtain a simplified model which is more amenable to analytical studies. A reasonably simplified model can make the analysis much more straightforward by reducing the number of variables and the complexity of the resulting dynamic equations [36]. The motion equations of the 2-mass mechanism in Figure 2.6 are

$$\begin{aligned} m_M \ddot{x}_M &= k\left(x_L - x_M\right) + b\left(\dot{x}_L - \dot{x}_M\right) + F, \\ m_L \ddot{x}_L &= -k\left(x_L - x_M\right) - b\left(\dot{x}_L - \dot{x}_M\right), \end{aligned} \tag{2.47}$$

where k is the spring constant, b the damping coefficient, m_M and m_L are mover and load masses, respectively, and x_i, $\dot{x}_i$ and $\ddot{x}_i$ are the displacement, velocity and acceleration of the masses m_M and m_L. For relative small displacement, as in the case of a vibrational structure, the restoring force of the spring can be modelled as a linear function [37]. Usually the masses m_M and m_L are known, but the spring constant k and damping coefficient b must be identified somehow. In the present work a transient response is used. The measured velocity of the load when the motor is controlled by a PI velocity controller with a step function and amplitude of 0.15 m/s and the load can vibrate freely is depicted in Figure 2.8. The measured mass of the load was 4.2 kg.

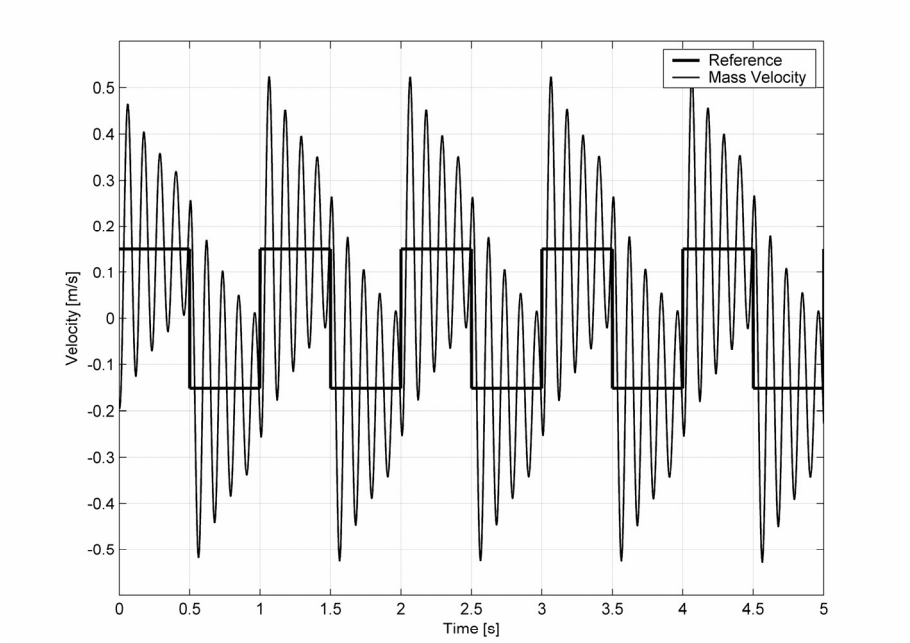


Figure 2.8. Measured velocity of the load in the case of PI motor control.

From Figure 2.8 we can deduce parameters for a controller and system model in the Kalman filter. The natural frequency f_n of the mechanism in hertz can be calculated directly from the figure. Substituting frequency f_n into Equation (2.48) and assuming that the mass of the load m_L is known, we can solve the spring constant k as follows:

$$\omega_n = \sqrt{\frac{k}{m_L}} \rightarrow k = \omega_n{}^2 m_L = \left(2\pi f_n\right)^2 m_L . \tag{2.48}$$

The damping coefficient b of an underdamped system can be solved from the step or impulse response by comparing the amplitudes of two consecutive waves. The logarithmic decrement δ is given by [36]

$$\delta = \ln \frac{A_i}{A_{i+1}}, \tag{2.49}$$

where A_i is the amplitude of ith period. The structural damping coefficient b_s is calculated from

$$b_s = \frac{2\delta}{\sqrt{\left(2\pi\right)^2 + \delta^2}} . \tag{2.50}$$

The equivalent viscous damping coefficient can then be determined as

$$b_e = \frac{kb_s}{\omega_n} = b_s \sqrt{km_L} . \tag{2.51}$$

This damping coefficient b_e can now be used as b in the mathematical model for the 2-mass system in Equation (2.47). Damping is usually very low in mechanical systems, and it can be neglected without affecting the accuracy of the analysis of the control system [38]. The damping coefficient is taken into consideration in the thesis, however, when deriving the state space model for the Kalman filter. The spring constant can also be solved from the frequency response by determining the nominal frequency of the mechanism and calculating k using Equation (2.48). In the physical system it was possible to change the load mass from 2.6 kg to 4.2 kg. The nominal frequencies of the mechanism analysed on the basis of the physical linear motor system are 11.7 Hz and 9.1 Hz, respectively. The mechanical parameters of the physical system are given in Table 3.

Table 3. Mechanical Parameters

Symbol	Parameter	Value
m_M	Mover Mass	18 kg
m_L	Load Mass	2.6 kg / 4.2 kg
k	Spring Constant	13700 N/m
b	Damping Coefficient	6 Ns/m

These parameters are used in the simulation model for the mechanism as initial parameters for the backstepping controller introduced in Section 4.2 and in the state space model for the Kalman filter introduced in Section 4.4.

Chapter 3

3 Control of a PMLSM with a Flexible Mechanism

This chapter introduces control methods for use with a flexible mechanism. High-gain controllers can be used to control linear motors, to compensate for the main nonlinearities in the motor. When an external flexible mechanism is mounted on the mover and the aim is to control this mechanism, the control task is totally different and more sophisticated methods must be used. Two nonlinear controllers, based on backstepping and a neural network, are introduced in this thesis. Analysis of the system begins with the linearized motor model, which is used to analyse the problematic nature of conventional linear controllers.

3.1 Conventional Linear Controllers

3.1.1 PID Controller

We will first analyse a PMLSM system in which a conventional PI controller is used for the flexible load. PI/PID controllers are the most common type used in industry, due to their simplicity and reliability. A block diagram model of a linearized PMLSM system with a PI velocity controller is shown in Figure 3.1. The inputs for the controller are the outputs of actuators, in this case electrical actuators with their own complex dynamics. The actuator dynamics are usually neglected in the first step of control design, on the assumption that they are stable and considerably faster than the inertial dynamics of the masses.

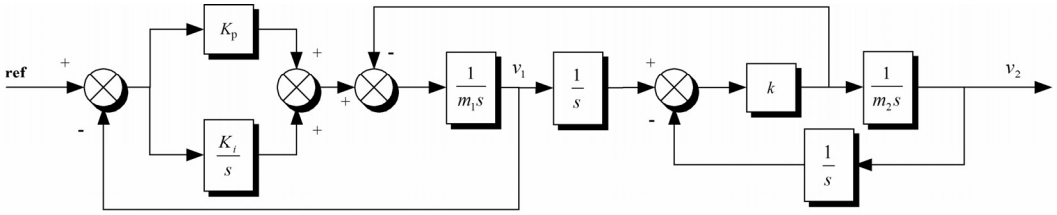


Figure 3.1. Block diagram of a linearized PMLSM system.

The open loop bode plots from the driving force F to the mover velocity $\dot{x}_M$ (v_1/F) and load velocity $\dot{x}_L$ (v_2/F) are shown in Figure 3.2.

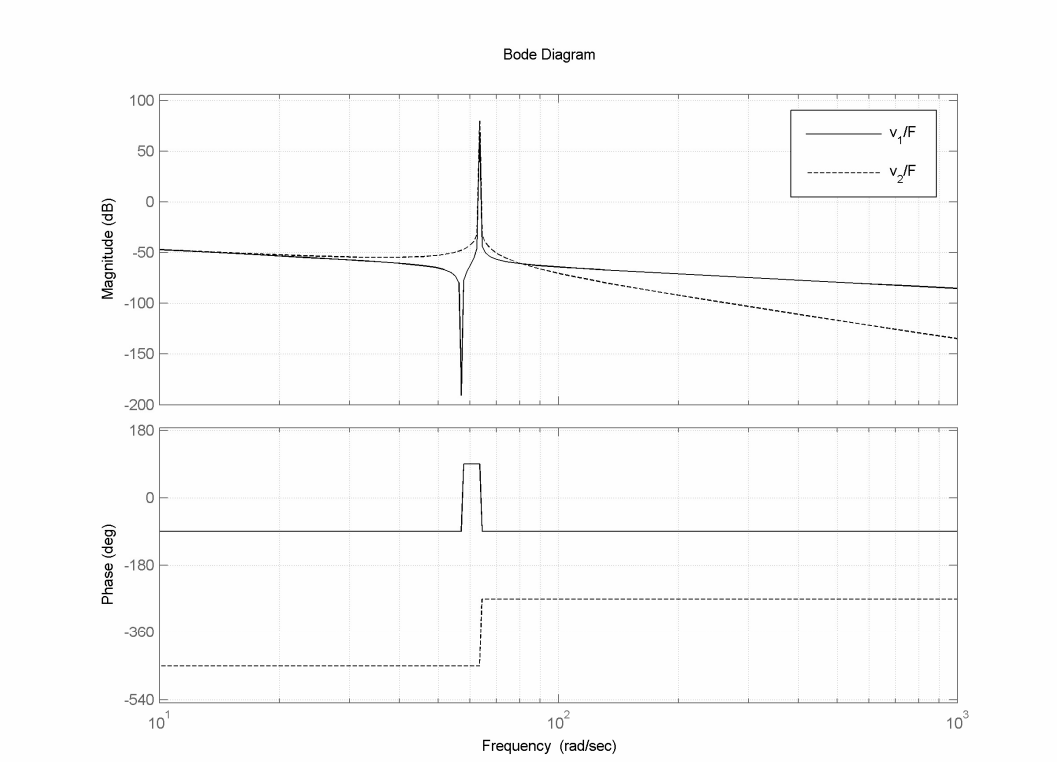


Figure 3.2. Open loop Bode plot of the 2-mass system.

As can be seen in Figure 3.2, the velocity difference is too high in the case of anti-resonance to permit feasible control over the load velocity. When we add a PI controller to the system, the closed loop transfer functions from the reference velocity $\dot{x}_{ref}$ to the mover velocity $\dot{x}_M$, (3.1) and load velocity $\dot{x}_L$ (3.2) are

$$\frac{\dot{x}_M}{\dot{x}_{ref}} = \frac{m_L K_p s^3 + m_L K_i s^2 + K_p k s + K_i k}{m_M m_L s^4 + m_L K_p s^3 + \left(m_L K_i + m_L k + m_M k\right) s^2 + K_p k s + K_i k} \,, \qquad (3.1)$$

$$\frac{\dot{x}_L}{\dot{x}_{ref}} = \frac{k\left(K_p s + K_i\right)}{m_M m_L s^4 + m_L K_p s^3 + (m_M k + m_L K_i + m_L k) s^2 + K_p k s + K_i k} \,, \qquad (3.2)$$

where K_p is the proportional gain and K_i the integrator gain. It is not possible to affect the behaviour of the systems (3.1) and (3.2) arbitrarily, because these are fourth-order models and we have only two control parameters, K_p and K_i. It would not be possible to control the system even if the derivative term K_d were added to the controller. The root locus of system (3.2) when the proportional gain K_p varies from 1 to 10000 is shown in Figure 3.3.

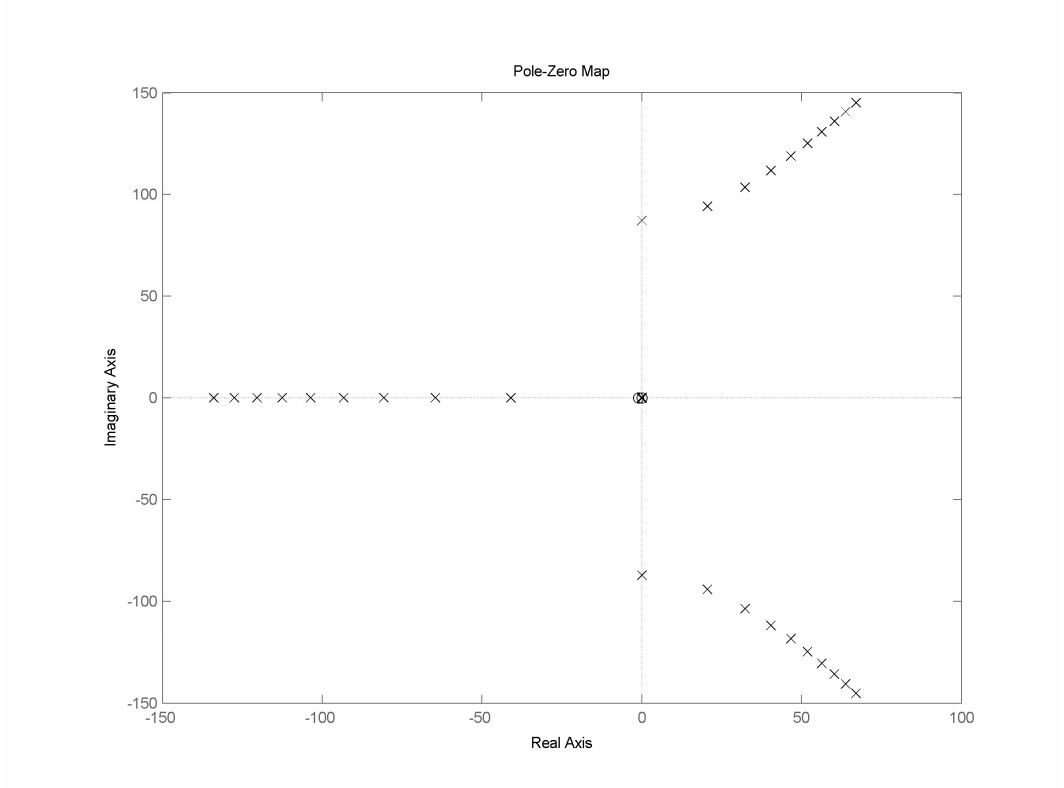


Figure 3.3. Pole-Zero map of system (3.2) when the value of K_p is varied from 1 to 10000 with a step size of 1000.

In theory a conventional linear controller (PI/PID) can suppress the vibration of the load in the linear system, as illustrated in Figure 3.3. There are small gain margins in the root locus where the system is stable. However, the velocity becomes unstable very quickly as the gains

are increased when controlling the load by means of a simple PI controller, and the physical linear motor application is highly nonlinear, so that this conventional controller fails to suppress load vibrations in motors of this kind when the load is as flexible as in the present instance. The disadvantage with the system in Figure 3.1 is that the variable which needs controlling is not measured. One possibility is to take a feedback signal for the controller straight from the load that is to be controlled, but although this seems the most logical solution, it is often impossible to implement and causes instability in the process. A block diagram for the system is shown in Figure 3.4.

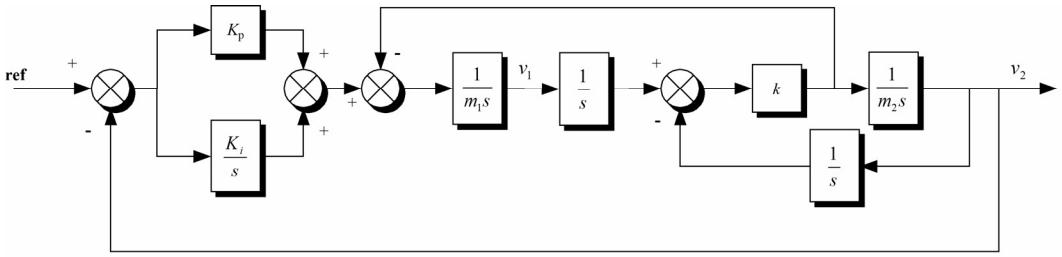


Figure 3.4. Block diagram of the load feedback system.

Due to instability problems in such systems, it is therefore necessary to have other control strategies than those based on conventional PI controllers.

3.1.2 State Feedback Controller

One possibility for stabilization is to use a state feedback controller in which all the state variables can be measured, or one version would be an output feedback controller in which only an output vector with a dimension typically less than that of the state can be measured. The feedback stabilization problem is easy when the system is linear and time-invariant:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t). \end{aligned} \tag{3.3}$$

where **A** is called the state matrix, **B** the input matrix, **C** the output matrix and **D** the direct transition matrix. In this case, the state feedback control u=-**Kx***(t)* preserves the linearity of the open-loop system, and the origin of the closed-loop system

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) \tag{3.4}$$

is asymptotically stable if and only if the matrix **A-BK** is of the *Hurwitzh* type, *i.e.* the eigenvalues lie strictly in the left half plane [37]. Thus the state feedback stabilization problem reduces to a problem of designing a vector **K** to assign the eigenvalues of **A-BK** in the open left-hand complex plane. Linear control theory confirms that the eigenvalues of **A-BK** can be arbitrarily assigned provided the pair (**A**,**B**) is controllable (Appendix A.2). If we can only measure the output *y*, we can use dynamic compensation, such as an observer-based controller

$$\begin{aligned} u(t) &= -\mathbf{K}\hat{\mathbf{x}}(t) \\ \dot{\hat{\mathbf{x}}}(t) &= \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{H}\left(y(t) - \mathbf{C}\hat{\mathbf{x}}(t) - \mathbf{D}u(t)\right) \end{aligned} \tag{3.5}$$

to stabilize the system. If we are to be able to observe states of the system, the pair (**A**,**C**) must provide observability (Appendix A.2). A block diagram of the state feedback system is provided in Figure 3.5. The feedback gain **K** is designed as in the state feedback case, such that **A-BK** is of the *Hurwitzh* type, while the observer gain **H** is designed such that **A-HC** is of the *Hurwitzh* type. The closed-loop eigenvalues will consist of the eigenvalues of **A-BK** and of **A-HC**. This theorem is a separation principle, since the closed-loop eigenvalues can be assigned in separate tasks for the state feedback and observer problems. For a general nonlinear system the problem is more difficult and well less understood. The most practical way to approach such a stabilization problem is to appeal to the neat results available in the linear case, that is, via linearization.

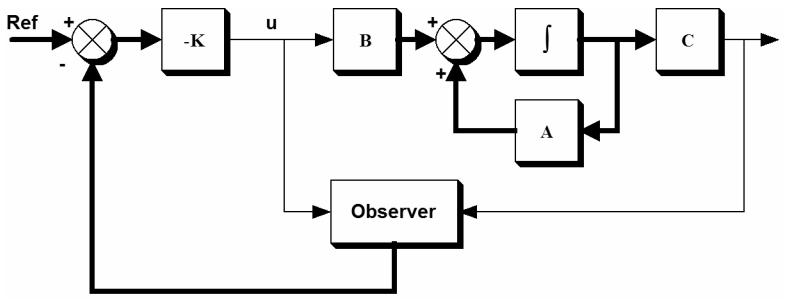


Figure 3.5. State feedback controller with observer.

3.2 Nonlinear Controllers

Although numerous linear methods have been developed to control linear processes, most physical systems are in practise nonlinear and the use of conventional linear control design methods for nonlinear systems usually fails. The first step in analysing a nonlinear system is usually to linearize it, if possible, about some nominal operating point and analyse the resulting linear model. A linearized model of this kind can help us to learn about the behaviour of the nonlinear system. Linearization is not enough, however, because it is an approximation in the neighbourhood of an operating point, *i.e.* it can only predict the "local" behaviour of the nonlinear system. When the required range of operation is large, a linear controller is likely to perform very poorly or be unstable, because the nonlinearities in the system cannot be properly compensated for [13]. It cannot predict the "non-local" behaviour of the system far away from the operating point, and certainly not its "global" behaviour. Control systems have many common nonlinearities whose discontinuous nature does not allow linear approximation, e.g. Coulomb friction, saturation, dead zones, backlash, and hysteresis. Another problem lies in the uncertainties entailed in the model. It is assumed that the parameters of the system model for a linear controller are well known, but this is not usually the case, and many control problems involve uncertainties in the model parameters. For systems of this kind there are two classes of nonlinear controllers: robust controllers and adaptive controllers. We prefer to use an adaptive control scheme.

While there are several design techniques for linear controllers in the time domain or in the frequency domain, there are no universal design techniques for nonlinear controllers. Many methods have been proposed for analysing nonlinear control systems, but those that are mainly used are phase plane analysis and the Lyapunov method. Two nonlinear methods for the analytical designing of a controller for a linear motor system are introduced in the next sections. Both controllers are observer-based, *i.e.* unavailable states are estimated by the technique introduced in Section 3.3.

3.2.1 The Backstepping Controller

A promising method for nonlinear control is the *control Lyapunov function* (CLF), the derivative of which depends on the control and can be made negative by feedback. This is a

difficult problem because no general design methods are available. CLFs for a large class of systems can be constructed by backstepping, a recursive means of designing a system with nonlinearities not constrained by linear bounds [20]. The key idea of backstepping is to derive an error equation and construct a control law and a parameter adjustment law so that the state of the error equation reverts to zero [39]. To be precise, the idea is to start with a system which is stabilizable with a known feedback law for a known Lyapunov function and then to add an integrator to its input. For the augmented system a new stabilizing feedback law is explicitly designed and shown to be capable of stabilizing a new Lyapunov function, and so on [20]. The name backstepping comes from the design procedure, where we step back towards the control input, starting with the scalar equation, which is separated from it by the largest number of integrations. The backstepping method can be successfully implemented for many applications [40] [41].

Although the Lyapunov theory has for a long time been an important tool for both linear and nonlinear control, its use for nonlinear control has been complicated by difficulties in finding a Lyapunov function for a given system. If one can be found, it is known that the system will be stable, but the task of finding such a function has often been left to the imagination and experience of the designer. It is for this reason that constructive tools for nonlinear control design based on Lyapunov theory, such as backstepping, are becoming more and more popular among control engineers.

The method used in a backstepping controller is referred to as Lyapunov's direct method. There are basically two ways of using this for control design, and both have a trial and error flavour. The first technique involves hypothesizing one form of control law and then finding a Lyapunov function to justify the choice, while the second technique, used in the present control design, works in such a way that first a Lyapunov function *V(x)* is selected and then efforts are made to find a feedback control *u(x)* that renders $\dot{V}\left(x,u\left(x\right)\right)$ negative definite. This attempt may fail with an arbitrary choice of *V(x)*, but if *V(x)* is a CLF, a stabilizing control law *u(x)* can be found. The theory of Lyapunov's stability criterion and *control Lyapunov function* (CLF) will be introduced in the next chapters.

Lyapunov Stability

The stability of the equilibrium point is usually characterized in the sense of Lyapunov, the basic idea of this criterion being that an equilibrium point is stable if all solutions starting at nearby points remain nearby; otherwise, it is unstable. It is asymptotically stable if all solutions starting at nearby points not only remain nearby but also tend towards the equilibrium point as time approaches infinity. These notions can be stated in the form of the following definitions, as in [13], [37] and [42], on which this review of Lyapunov stability is based. Consider the time-varying system

$$\dot{x} = f(x,t), \tag{3.6}$$

where $x \in \mathbb{R}^n$. Let us first define the equilibrium point.

Definition 3.1 (Equilibrium point)

The origin x=0 is the equilibrium point for (3.6) *if*

$$f(0,t) = 0, \qquad \forall t \geq 0.$$

□

The concepts related to Lyapunov stability describe continuity properties of $x(x_0,t_0)$ with respect to x_0. If the initial condition x_0 is perturbed to x_0^*, then, for stability, the resulting perturbed solution $x(x_0^*,t_0)$ is required to remain close to $x(x_0,t_0)$ for all $t \geq t_0$. For asymptotic stability, the error $x(x_0^*,t_0)$-$x(x_0,t_0)$ is required to vanish as $t \to \infty$. This can be defined as follows.

Definition 3.2 (Lyapunov Stability)

The solution $x(x_0,t_0)$ is

- *Stable if for each $\varepsilon > 0$ there exists $\delta(\varepsilon,t_0) > 0$ such that*

$$\left\| x_0^* - x_0 \right\| < \delta \Rightarrow \left\| x\left(x_0^*, t_0\right) - x\left(x_0, t_0\right) \right\| < \varepsilon , \qquad \forall\ t \geq t_0 ,$$

- *Unstable if it is not stable*
- *Asymptotically stable if it is stable and in addition there exists δ such that*

$$\left\| x_0^* - x_0 \right\| < \delta \Rightarrow \left\| x\left(x_0^*, t_0\right) - x\left(x_0, t_0\right) \right\| \to 0 \qquad as \quad t \to \infty .$$

□

The stability properties of $x(x_0,t_0)$ in general depend on the initial time t_0. When the constants are independent of t_0, the corresponding properties are *uniform*, *i.e.,* the system is time-invariant. For adaptive systems, *uniform stability* is more desirable than mere stability. Even more desirable is *uniform asymptotic stability* (UAS) [42], since the system can deal better with perturbations and disturbances. In general adaptive design achieves less than uniform asymptotic stability but more than uniform stability, because it forces the tracking error to converge to zero. Asymptotic stability is a highly desirable property in a control system, but it would be even more favourable if the state tended towards equilibrium from an arbitrary initial condition, which leads us to the following definition.

Definition 3.3

The equilibrium point of a system (3.6) *is globally asymptotically stable (GAS) if it is asymptotically stable for all initial states.*

□

Global asymptotic stability (GAS) is what we will strive towards in our control design. As seen from the definitions above, in order to show a certain type of stability we now have to determine the explicit solution to (3.6). This solution cannot generally be found analytically, but fortunately there are other ways of proving stability. Lyapunov came up with the idea of using the state vector $\mathbf{x}(t)$ to construct a scalar function $V(x)$ that would measure how far the system is from the equilibrium. $V(x)$ is an energy-like, radially unbounded and positive definite function. If $V(x)$ can be shown to decrease continuously, the system itself must be

moving towards an equilibrium [43]. This approach to showing stability is called Lyapunov's direct method, and can be found in [13]. Before we go any further, let us clarify the concepts that we will use throughout this thesis.

Definition 3.4

A scalar function V(x) is said to be

- *Positive definite if V(0)=0 and*

  $V(x) > 0, x \neq 0$,

- *Positive semidefinite if V(0)=0 and*

  $V(x) \geq 0, x \neq 0$,

- *Negative (semi-)definite if –V(x) is positive (semi-)definite*

- *Radially unbounded if*

  $V(x) \to \infty \; as \; \|x\| \to \infty$.

□

Now we can state the main theorem for proving stability.

Theorem 3.1 (LaSalle-Yoshizawa)

Let x=0 be an equilibrium point of (3.6)*. Let V be a scalar, continuously differentiable function of the state x such that*

- *V(x) is positive definite*
- *V(x) is radially unbounded*

- $\dot{V}(x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leq -W(x) \leq 0$, *where W(x) is a positive semidefinite continuous function.*

Then all solutions to (3.6) *satisfy*

$$\lim_{t \to \infty} W(x(t)) = 0.$$

In addition, if W(x) is positive definite, then the equilibrium x=0 is GUAS.

Proof

A proof of this theorem is presented in Kristic *et al.* [42] and Slotine *et al.* [13].

□

A designed system is usually presented as a time-invariant system

$$\dot{x} = f(x). \tag{3.7}$$

The asymptotic stability of this system is summarized in the next corollary by LaSalle.

Corollary 3.1

Let x=0 be the equilibrium point of (3.7). *Let V(x) be a scalar, continuously differentiable function of the state x such that*

- *V(x) is positive definite*
- *V(x) is radially unbounded*
- $\dot{V}(x)$ *is negative semidefinite*

Let $E = \{x : \dot{V}(x) = 0\}$ *and suppose that no other solution than* $x(t) \equiv 0$ *can remain in E for ever. Then x=0 is GAS.*

Proof

See Kristic *et al.* [42]. □

These results are not constructive, because they do not give any clue as to how to find a proper V to conclude GAS. In the following section the Lyapunov criterion will be extended to include a control input, so that the desirable stability properties can be ensured.

Control Lyapunov functions (CLF)

Let us add a control input and consider the time-invariant system

$$\dot{x} = f(x, u). \tag{3.8}$$

Since the objective of this thesis is to design a closed-loop system with desirable stability properties rather than to analyse the properties of the system itself, we are interested in an extension of the Lyapunov function that may be called a *control Lyapunov function* (labelled CLF for convenience). Given the stability results in the previous section, we want to find a control law

$$u = \alpha(x) \tag{3.9}$$

such that the desired state of the closed-loop system

$$\dot{x} = f(x, \alpha(x)) \tag{3.10}$$

is a globally asymptotically stable equilibrium point. Once again, for simplicity, we consider the origin to be the goal state. We can choose a function $V(x)$ as a Lyapunov candidate and require that its derivate should satisfy $\dot{V}(x) \leq -W(x)$, where $W(x)$ is a positive definite

function. Its closed-loop stability then follows from Theorem 3.1. We therefore need to find $\alpha(x)$ to guarantee that for all $x \in \mathbb{R}^n$

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x) f(x, \alpha(x)) \leq -W(x). \tag{3.11}$$

The pair *V(x)* and *W(x)* must be chosen carefully, otherwise (3.11) will not be solvable. This motivates the following definition, which can be found in Kristic *et al.* [42]:

Definition 3.5 (Control Lyapunov Function)

A smooth positive definite and radially unbounded function $V: \mathbb{R}^n \to \mathbb{R}_+$ *is called a control Lyapunov function (CLF) for* (3.8) *if for every* $x \neq 0$

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x) f(x, u) < 0 \qquad \text{for some } u.$$

□

The significance of this definition lies in establishing the fact that the existence of a globally stabilizing control law is equivalent to the existence of a CLF. If we have a CLF for the system then we can certainly find a globally stabilizing control law. The following example clarifies the CLF. The time-invariant system is

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}, \quad f(0,0) = 0. \tag{3.12}$$

The goal is to design a feedback control law $\alpha(x)$ for the control variable u such that the equilibrium $x=0$ of the closed loop system

$$\dot{x} = f(x, \alpha(x)) \tag{3.13}$$

is *globally asymptotically stable* (GAS). The derivative of a Lyapunov candidate *V(x)* is required to satisfy $\dot{V}(x) \leq -W(x)$, where *W(x)* is a positive definite function. We need to find $\alpha(x)$ to guarantee that for all $x \in \mathbb{R}^n$

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x) f\left(x, \alpha(x)\right) \leq -W(x). \tag{3.14}$$

A function is called a *control Lyapunov function* (CLF) for Equation (3.12) if

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x) f(x,u) < 0, \quad \forall x \neq 0. \tag{3.15}$$

For the nonlinear system

$$\dot{x} = f(x) + g(x)u \tag{3.16}$$

the CLF inequality in Equation (3.14) becomes

$$\frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(x)\alpha(x) \leq -W(x). \tag{3.17}$$

It should be noted that Equation (3.17) can be satisfied only if, for all $x \neq 0$,

$$\frac{\partial V}{\partial x} g(x) = 0 \Rightarrow \frac{\partial V}{\partial x} f(x) < 0. \tag{3.18}$$

When *V(x)* is a CLF, there are many control laws that render $\dot{V}\left(x, u(x)\right)$ a negative definite, so that a backstepping method should be used.

Integrator Backstepping

Backstepping is a procedure which finds both a CLF and a control law simultaneously. The following example shows how this is done. Consider a one-dimensional system with input *u*

$$\dot{x}_1 = f(x_1) + u, \tag{3.19}$$

where f is a known function and $f(0)=0$. In this case it is easy to find a control Lyapunov function, and any positive quadratic function $V_1(x_1)=x_1^2/2$. We want to choose $u_1=u_1(x_1)$ such that V_1 has a negative derivative. The time derivative of V_1 is

$$\dot{V}_1 = x_1\dot{x}_1 = x_1\left[f\left(x_1\right)+u\right]. \tag{3.20}$$

We can choose the control input u to render the derivative V_1 negative definite such a way that it is equal to $-c_1x_1^2$, *i.e.*

$$\dot{V}_1 = x_1\dot{x}_1 = x_1\left[f\left(x_1\right)+u\right] = -c_1x_1^2 \le 0\,, \tag{3.21}$$

where the controller gain $c_1>0$. Now the controller u can be solved from (3.21) as follows:

$$u = -c_1x_1 - f\left(x_1\right). \tag{3.22}$$

To obtain a more realistic model, the system is extended with an integrator at the input, giving

$$\begin{aligned}\dot{x}_1 &= f\left(x_1\right)+g\left(x_1\right)x_2\\ \dot{x}_2 &= u.\end{aligned} \tag{3.23}$$

This system can be viewed as a cascade connection of two components, as in Figure 3.6 (a). The same approach as above can be used if we first regard x_2 as a virtual control. Let the desired value be $x_2^* = \alpha(x_1)$, so that the first state will be stabilized. For this purpose we introduce the error state

$$e = x_2 - x_2^* = x_2 - \alpha\left(x_1\right). \tag{3.24}$$

Adding x_2 from (3.24) to (3.23), we obtain

$$\begin{aligned}\dot{x}_1 &= f\left(x_1\right)+g\left(x_1\right)\left(e+\alpha\left(x_1\right)\right),\\ \dot{x}_2 &= u,\end{aligned} \tag{3.25}$$

which is shown in Figure 3.6 (b). The time derivative of e is

$$\dot{e} = \dot{x}_2 - \dot{\alpha} . \tag{3.26}$$

The following equations can now be derived:

$$\begin{aligned} \dot{x}_1 &= \left[f(x_1) + g(x_1)\alpha(x_1) \right] + g(x_1)e, \\ \dot{e} &= u - \dot{\alpha}. \end{aligned} \tag{3.27}$$

This equation is shown in Figure 3.6 (c). The time derivative of α is

$$\dot{\alpha} = \frac{\partial \alpha}{\partial x_1}(x_1)\left[f(x_1) + g(x_1)\left(\alpha(x_1) + e\right) \right]. \tag{3.28}$$

Consider the Lyapunov function candidate for the whole system

$$V_2 = V_1 + \frac{1}{2}e^2 \tag{3.29}$$

whose derivative is

$$\dot{V}_2 = \dot{V}_1 + e\dot{e} = \frac{\partial V_1}{\partial x}\left(f + g\left(\alpha + e\right)\right) + e\left[u - \frac{\partial \alpha}{\partial x_1}\left(f + g\left(\alpha + e\right)\right) \right]. \tag{3.30}$$

After a few manipulations Equation (3.30) can be written as

$$\dot{V}_2 = \frac{\partial V_1}{\partial x_1}\left(f + g\alpha \right) + e\left[u - \frac{\partial \alpha}{\partial x_1}\left(f + g\left(\alpha + e\right)\right) + \frac{\partial V_1}{\partial x_1} g \right]. \tag{3.31}$$

According to the LaSalle-Yoshizawa theorem (Theorem 3.1), Equation (3.31) can be written as

$$\dot{V}_2 \le -W(x_1) + e\left[u - \frac{\partial \alpha}{\partial x_1}\left(f + g\left(\alpha + e\right)\right) + \frac{\partial V_1}{\partial x_1} g \right]. \tag{3.32}$$

The simplest way to make the derivative of V_2 negative definite is to choose a control input to render the last term in Equation (3.32) equal to $-ce^2$, *i.e.*

$$\dot{V}_2 \leq -W(x_1) - ce^2 . \tag{3.33}$$

The control law u can now be solved from Equations (3.32) and (3.33):

$$\begin{aligned} & u - \frac{\partial \alpha}{\partial x_1}\left(f + g(\alpha + e)\right) + \frac{\partial V_1}{\partial x_1} g = -ce \\ & \rightarrow u = -c\left(x_2 - \alpha(x)\right) + \frac{\partial \alpha}{\partial x_1}(x_1)\left[f(x_1) + g(x_1)x_2\right] - \frac{\partial V}{\partial x_1}(x_1)\, g(x_1) \end{aligned}, \tag{3.34}$$

where $c \geq 0$. This guarantees that the system is GAS.

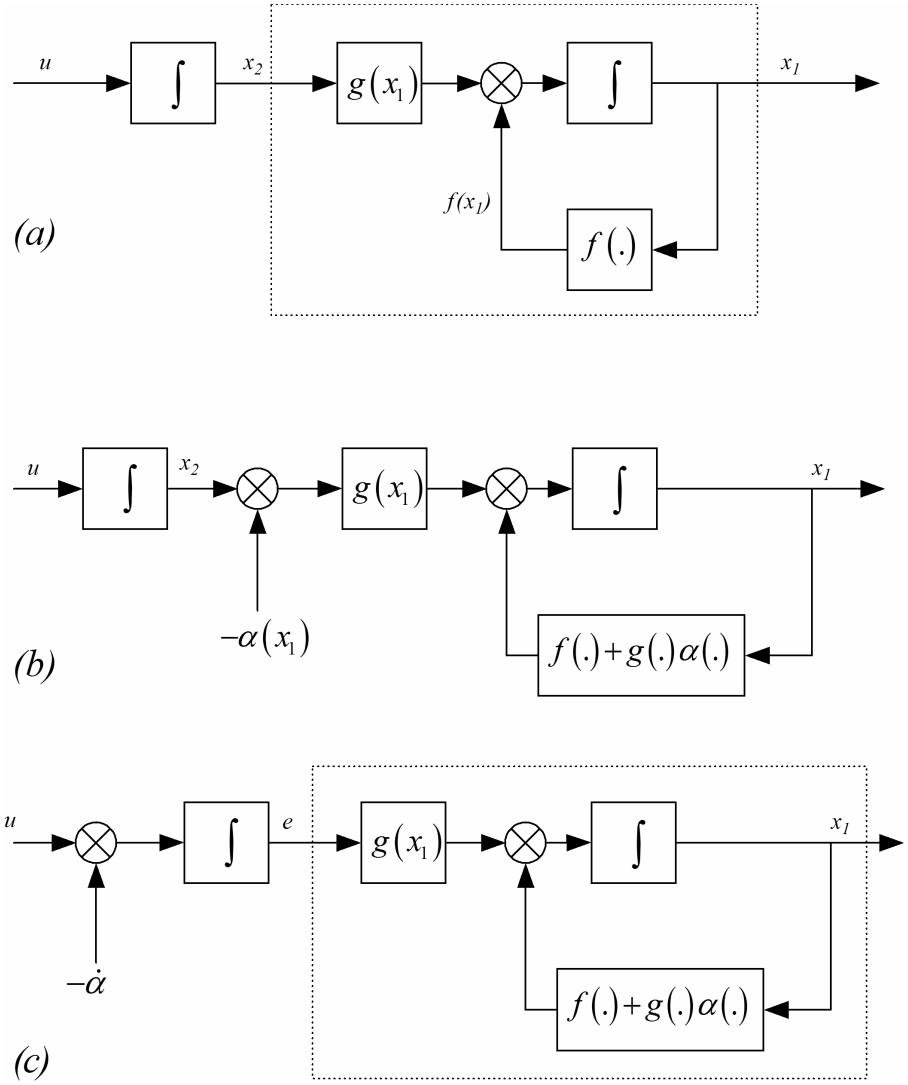


Figure 3.6. (a) Block diagram of the system (3.23), (b) the same with virtual control α(x1), and (c) "backstepping" of α(x1) through the integrator.

To be able to apply backstepping to a system, it must have a "lower triangular form". A pure feedback system (3.35) is one example of this form [42].

$$\begin{aligned}
&\dot{x} = f(x) + g(x)x_1 \\
&\dot{x}_1 = f_1(x, x_1, x_2) \\
&\vdots \\
&\dot{x}_{k-1} = f_{k-1}(x, x_1, \cdots, x_k) \\
&\dot{x}_k = f_k(x, x_1, \cdots, x_k, u)
\end{aligned} \tag{3.35}$$

where $x_i \in \mathbb{R}$. Systems which can be written in strict feedback form (3.36) can also be handled:

$$\begin{aligned}
&\dot{x} = f(x) + g(x)x_1 \\
&\dot{x}_1 = f_1(x, x_1) + g_1(x, x_1)x_2 \\
&\dot{x}_2 = f_2(x, x_1, x_2) + g_2(x, x_1, x_2)x_3 \\
&\vdots \\
&\dot{x}_{k-1} = f_{k-1}(x, x_1, \cdots, x_{k-1}) + g_{k-1}(x, x_1, \cdots, x_{k-1})x_k \\
&\dot{x}_k = f_k(x, x_1, \cdots, x_k) + g_k(x, x_1, \cdots, x_k)u
\end{aligned} \tag{3.36}$$

Many physical systems cannot be written in a lower triangular form, although this can be done by neglecting some physical properties when modelling the system, so that it is then possible to apply the backstepping technique. Some form of analysis or simulation has to be carried out, of course, to verify that the neglected physical property does not affect the stability of the closed-loop system.

Adaptive Backstepping Controller

There are several well-known adaptive schemes that solve the control problem for linear systems with unknown parameters, but there are also restricted classes of nonlinear systems for which the design problem is solvable. Adaptive controllers are dynamic and therefore more complex than static controllers. An adaptive controller guarantees not only that the plant state *x* remains bounded, but also that it tends towards a desired constant value or asymptotically tracks a reference signal. This is achieved by extending the Lyapunov function *V(x)* with a term penalizing the estimation error $\theta = \hat{\theta} + \xi$. The idea is to employ backstepping to design a control law for the system as if all the parameters were known and then replace the unknown parameters with their estimates, a *certainty equivalence* way of thinking. We can consider the plant [42]

$$\dot{x} = u + \theta x\,, \tag{3.37}$$

where u is the control parameter and θ is the unknown constant parameter. The aim is to achieve regulation of the state $x(t)$: $x(t)\to 0$, $t\to\infty$. Here we are seeking for a parameter update law for the estimate $\hat{\theta}(t)$,

$$\dot{\hat{\theta}} = \tau\left(x, \hat{\theta}\right), \tag{3.38}$$

which, along with the control law $u = \alpha\left(x, \hat{\theta}\right)$, will make the derivative of the CLF $V\left(x, \hat{\theta}\right)$ negative. As mentioned in the preceding part of this section, one of the terms in the CLF is intended to penalize the estimation error ξ. A simple choice is the quadratic term $\frac{1}{2}\xi^2$. This result in the CLF

$$V(x, \hat{\theta}) = \frac{1}{2}x^2 + \frac{1}{2}\left(\hat{\theta} - \theta\right)^2 , \tag{3.39}$$

which is a radially unbounded function of time. We express the derivative of V as a function of u and $\dot{\hat{\theta}}$ and seek $\alpha\left(x, \hat{\theta}\right)$ and $\tau\left(x, \hat{\theta}\right)$ to guarantee that $\dot{V} \le -cx^2$ with c>0.

$$\dot{V} = x\left(u + \theta x\right) + \left(\hat{\theta} - \theta\right)\dot{\hat{\theta}} \le -cx^2 . \tag{3.40}$$

By rearranging the terms we obtain

$$xu + \hat{\theta}\dot{\hat{\theta}} + \theta\left(x^2 - \dot{\hat{\theta}}\right) \le -cx^2 . \tag{3.41}$$

Since neither α nor τ is allowed to depend on the unknown θ, we must take $\tau = x^2$,

$$\dot{\hat{\theta}} = x^2 . \tag{3.42}$$

The remaining condition

$$xu + \hat{\theta}x^2 \leq -cx^2 \tag{3.43}$$

allows us to select u in various ways. For instance, we can choose

$$u = -\left(c + \hat{\theta}\right)x\,. \tag{3.44}$$

Along with the update law, this controller renders the derivative of CLF negative, and the closed-loop system is guaranteed to be stable. Although this example applies to linear systems, it shows the principle of Lyapunov-based design, which is also valid for nonlinear systems. This exhibits some features of the more general scheme. A block diagram of the adaptive control principle is presented in Figure 3.7.

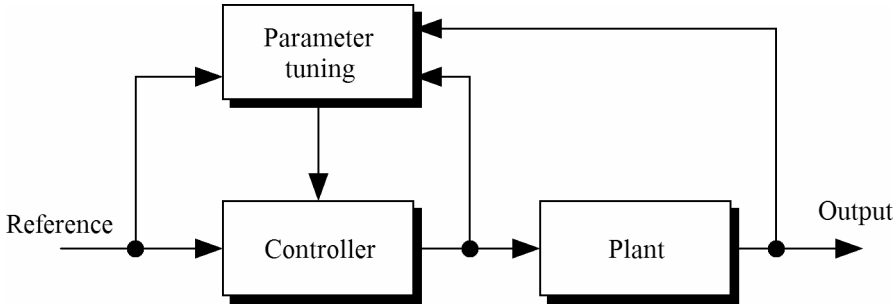


Figure 3.7. Adaptive control scheme.

3.2.2 Neural Network-Based Controller

Neural networks have been studied in many areas, from finance to technology, for many years. The first innovations in this field were made in the early 1940s, since when a number of neural network structures and teaching algorithms have been developed. Developments in computers and algorithms allow neural networks to be applied to numerous practical problems nowadays, as they can handle complex input-output mapping without any detailed analytical model of the system.

The simplified functionality of artificial neural networks (ANN) has been inspired by biological neurons. Basic neural networks consist of neurons, weights and activation

functions. The weights are adapted to achieve mappings between the input and output sets in order to track the reference model. In biological systems a typical neuron collects signals from others through a host of fine structures called dendrites and then sends out electrical signals through a long, thin strand known as an axon, which splits signals into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects. When a neuron receives an excitatory input that is sufficiently large relative to its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes [44]. A simplified biological neural system is compared with the block diagram of an artificial neuron in Figure 3.8.

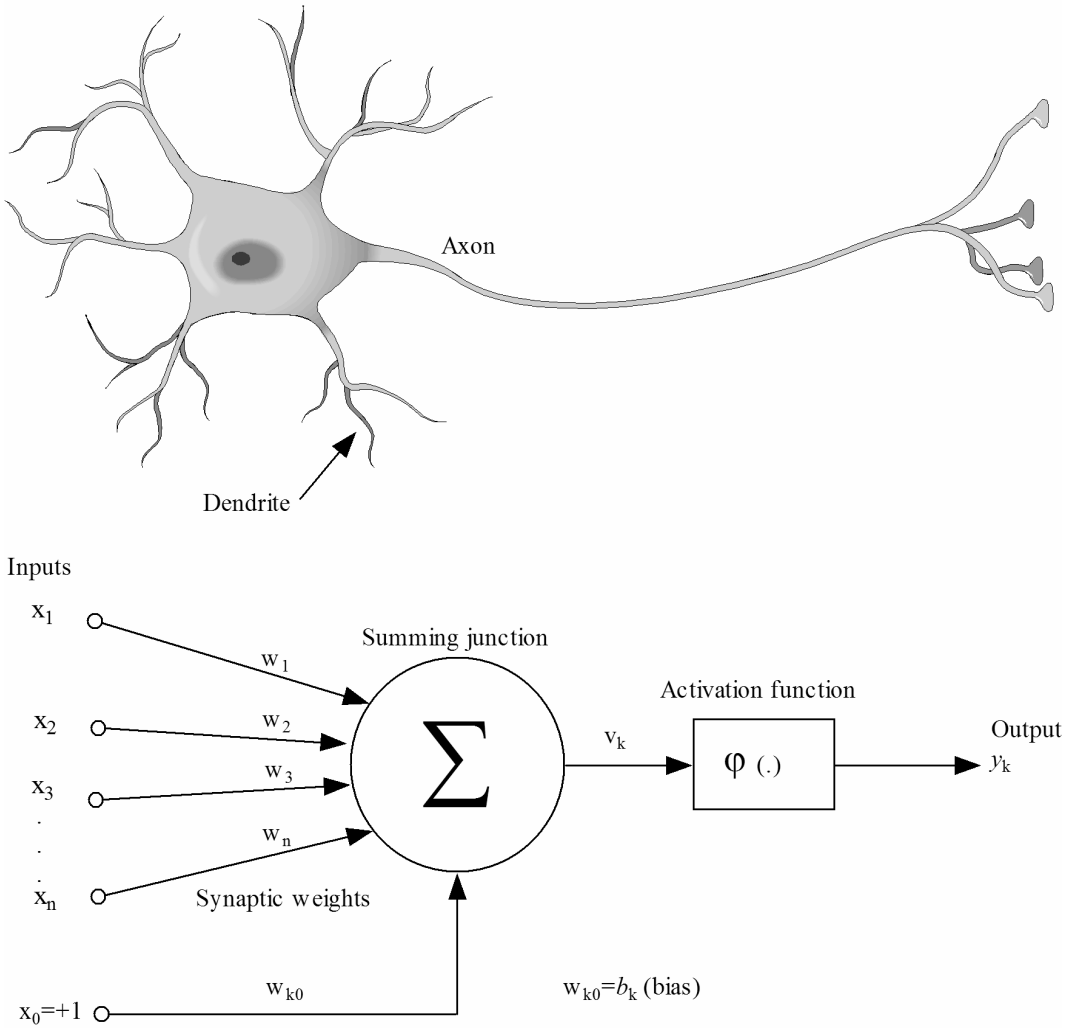


Figure 3.8. Biological (above) and artificial (below) neurons.

The operation of an artificial neuron is the following: A neuron updates its output from weighted inputs received from all the neurons connected to it. The decision to update the actual state of the neuron is taken by the activation function. Let us consider the neuron introduced in Figure 3.8. The synaptic weights are denoted by $w_1,w_2,\ldots,w_n$. Correspondingly, the inputs applied to the neuron are denoted by $x_1,x_2,\ldots,x_n$. The externally applied bias is denoted by b_k. From the model of Figure 3.8 we find that the output of the summing junction is

$$v_k = \sum_{i=1}^{n} w_i x_i + b_k = \mathbf{w}^T \mathbf{x} + b\,, \tag{3.45}$$

where **w** denotes the vector of weights and **x** the vector of inputs. The new state of a neuron is calculated by putting the output v_k through an activation function $\varphi(.)$. This can be written mathematically as

$$y_k = \varphi\left(\sum_{i=1}^{n} w_i x_i + b_k \right) = \varphi\left(\mathbf{w}^T \mathbf{x} + b \right)\,. \tag{3.46}$$

The simplest form of neural network is a perceptron, which consists of a single neuron with adjustable synaptic weights and threshold, as shown in Figure 3.8. Rosenblatt's original perceptron [45], [46] used a threshold function as its activation function $\varphi(.)$, but nowadays, and especially in multilayer networks, a sigmoid or hyperbolic tangent function is often chosen as the activation function. Three basic types of activation function are shown in Figure 3.9: Threshold Step, Piecewise-Linear and Sigmoid functions [47]. These are all of the bipolar type. The difference between *unipolar* and *bipolar* activation functions is that unipolar ones have values in the range [0,1], while bipolar values are in the range [-1,1]. The two first-mentioned functions are discontinuous and are usually able to adjust weights only in single-layer networks, while *sigmoid* or bipolar *hyperbolic tangent functions* are continuous functions, which allow for the gradient-based training of multilayer networks.

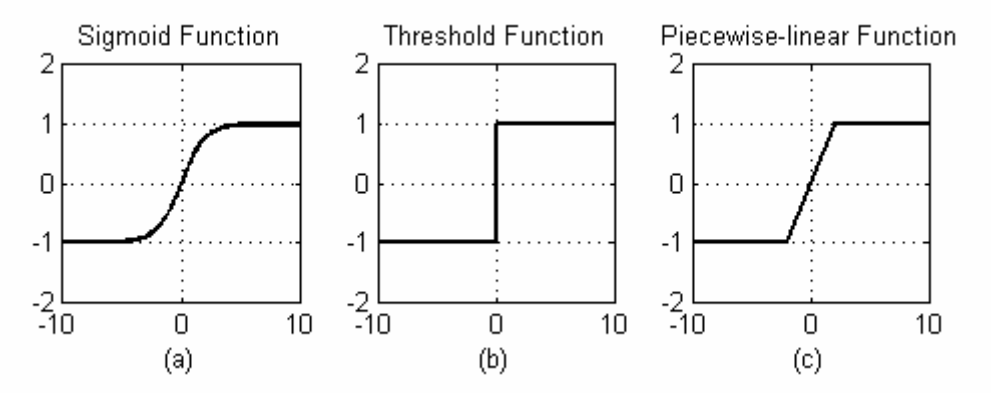


Figure 3.9. The most common activation functions.

A single perceptron is not very useful because of its limited mapping ability, but perceptrons can be used in clusters. The basic structure of all neural networks is similar: some of the neurons interface with the real world to receive inputs and other neurons provide the real world with the network's outputs. All the rest of the neurons are hidden from view. A typical multilayer perceptron (MLP) network consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes and an output layer of nodes. An example of a 3-3-2 multilayer perceptron is given in Figure 3.10.

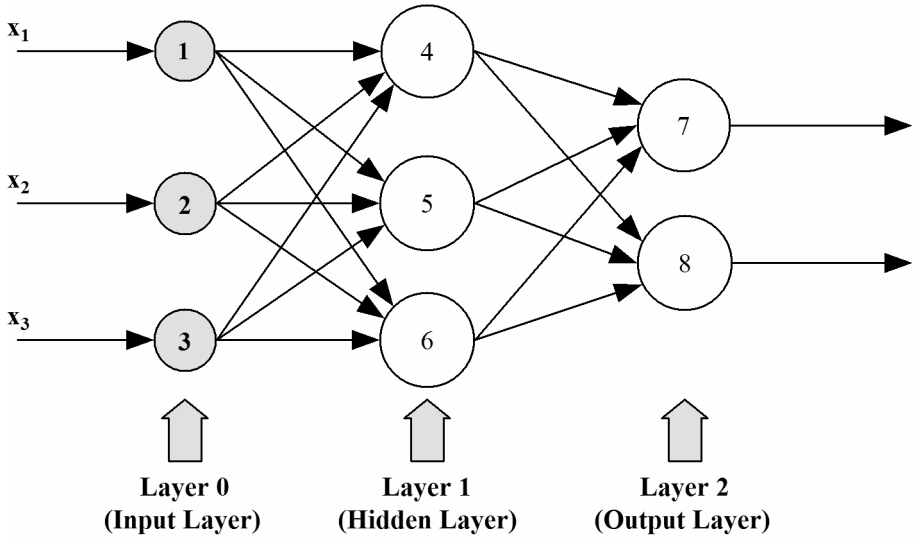


Figure 3.10. A multilayer perceptron with one hidden layer.

The Multilayer Perceptron (MLP) is the most common network structure for describing static and dynamic processes. MLP networks generate a nonlinear representation of the input-output relationship forming a nonlinear function to describe how the output is dependent on the input. A multilayer perceptron has an input layer of source nodes and an output layer of neurons (computation nodes). These two layers connect the network to the outside world. In addition, it usually has one or more layers of hidden neurons, which are so called because they are not directly accessible. The hidden neurons extract important features contained in the input data. The output vector **y** of the network in Figure 3.10 is given by

$$\mathbf{y} = \varphi_2\left(\mathbf{w}_2\varphi_1\left(\mathbf{w}_1\mathbf{x} + b_1\right) + b_2\right), \tag{3.47}$$

where **x** is the input vector, $\mathbf{w}_i$ is the weight matrix, b_i is a bias, and φ_i is the activation function. The weights in the network can be fixed, as in Hopfield models, or adaptive. In fixed-weight networks the weights are computed in advance using an evolution algorithm, for example. Fixed-weight models have limited applications, since they cannot adapt to different environments, and adaptive networks are more useful, as their weights are adjusted through a training procedure. Learning is a process by which the free parameters (*i.e.* weights and bias levels) of a neural network are adapted by the environment in which the network is embedded through a continuous process of stimulation. The weights can be updated as follows: Let $\mathbf{w}_j(n)$ denote the vector of the synaptic weights $\mathbf{w}_j$ at time n, when an adjustment $\Delta\boldsymbol{w}_j(n)$ is applied to the synaptic weights $\mathbf{w}_j(n)$, yielding the update value $\mathbf{w}_j(n+1)$. We may thus write the adapted weights of layer j as [47]

$$\mathbf{w}_j\left(n+1\right) = \mathbf{w}_j\left(n\right) + \Delta\mathbf{w}_j\left(n\right), \tag{3.48}$$

where $\mathbf{w}_j(n)$ and $\mathbf{w}_j(n+1)$ may be viewed as the old and new values of the synaptic weights $\mathbf{w}_j$. Various learning algorithms have been developed for adjusting $\Delta\mathbf{w}_j$, but only a few of them are suitable for multilayer neural networks. Some use only local signals in the neurons, while others require information from outputs; some require a supervisor who knows what the outputs for a given pattern should be, and other, unsupervised algorithms need no such information [48]. In a general sense, the learning process may be classified as follows:

- Learning with a teacher, also referred to as supervised learning.
- Learning without a teacher, also referred to as unsupervised learning.

Supervised learning is usually performed with feed-forward networks.

Differential Evolution

Global optimization methods are under continuous development, and *genetic algorithms* (GAs) and *evolution strategies* have been studied lately and found to be promising stochastic optimization methods [49]. There are numerous combinations of neural networks and *evolutionary algorithms* (EAs) that are used to solve classification problems. EAs have been used to train networks, design their architecture and select feature subsets [50]. A differential evolution (DE) algorithm is used here to find the initial weights for a feed-forward neural network. DE is a simple, powerful population-based, direct-search algorithm for globally optimizing functions defined especially on functions with real-valued parameters. Real parameter optimization makes up a large and important class of practical problems in science and engineering [51].

The back-propagation method and its variations may achieve fast convergence rates for reasonably sized networks. When the number of network parameters grows drastically, the number of suitable training algorithms decreases correspondingly. The other drawback with using gradient-based methods lies in the local minimum problem. The search space is high-dimensional and may contain numerous local minima, depending on the error measure and the input data. Traditional gradient-based methods may become trapped in these local minima. EAs do not use any gradient information and are likely to avoid becoming trapped in a local minimum by sampling multiple regions of the shape simultaneously.

Differential evolution can be applied to search for initial weights for a typical feed-forward neural network. The output of such a network is a function of the weights **w** and the inputs **x**, *i.e.* **y**=*f*(**x**,**w**), as in Equation (3.47). The adaptation can be carried out by minimizing the network error function *E,* which is of the form [49]

$$E\left(\mathbf{y}, f\left(\mathbf{x},\mathbf{w}\right)\right):\left(\mathbf{y}^{D_1},\mathbf{x}^{D_2},\mathbf{w}^{D_3},f\right)\rightarrow R\,. \tag{3.49}$$

The objective function in Equation (3.49) is minimized by optimizing the values of the network weights $w_1,w_2,\ldots,w_D$. In many ways DE is a typical evolution algorithm (EA). It operates on a population, $\mathbf{P}_G$, of candidate solutions, not just a single solution, and it generates a randomly distributed initial population $\mathbf{P}_{G=0}$ of NP D-dimensional object variable vectors $\mathbf{w}_{i,G}$, where i is the index to the population and G is the generation to which the population belongs. The current population, $\mathbf{P}_G$, is composed of those vectors, $\mathbf{w}_{i,G}$, that have already been found to be acceptable either as initial points, or by comparison with other vectors [52]. The population $\mathbf{P}_G$ is written as:

$$\mathbf{P}_G = \left(\mathbf{w}_{1,G},\ldots,\mathbf{w}_{NP,G}\right), G = 0,\ldots,G_{\max}. \tag{3.50}$$

Additionally, each vector contains D network weights for network training purposes:

$$\mathbf{w}_{i,G} = \left(w_{1,i,G},\ldots,w_{D,i,G}\right), i = 1,\ldots,NP, G = 0,\ldots,G_{\max}. \tag{3.51}$$

The DE population reproduction scheme is different from other evolutionary algorithms. After initialization, DE mutates and recombines the population $\mathbf{P}_{G+1}$ (sometimes referred to as crossover) to produce a population of NP trial vectors, $\mathbf{u}_{j,i,G+1}$, for each parent vector $\mathbf{w}_{j,i,G}$, as follows:

$$\begin{aligned}\mathbf{v}_{j,i,G+1} &= \mathbf{w}_{j,r3,G} + F\left(\mathbf{w}_{j,r1,G} - \mathbf{w}_{j,r2,G}\right)\\ \mathbf{u}_{j,i,G+1} &= \begin{cases}\mathbf{v}_{j,i,G+1}, \text{ if } rand_j[0,1) \le CR\\ \mathbf{w}_{j,i,G}, \text{ otherwise}\end{cases},\end{aligned} \tag{3.52}$$

where $\mathbf{v}_{j,i,G+1}$ is a mutant vector, and

$i = 1,\ldots,NP, j = 1,\ldots,D$
$r_1,r_2,r_3 \in \{1,\ldots,NP\}$, randomly selected, except $r_1 \ne r_2 \ne r_3 \ne i$.
$CR \in [0,1], F \in (0,1+]$

Both CR and F are user-specified control variables. CR is a real-valued crossover factor that expresses the probability that a trial vector parameter will come from the randomly chosen,

mutated vector $\mathbf{v}_{j,i,G+1}$ instead of the current vector $\mathbf{w}_{j,i,G}$. In general, *F* and *CR* affect the convergence speed and robustness of the search process. Their optimal values depend both on objective function characteristics and the population size *NP*, and therefore the selection of such optimal values is an application-dependent task [49]. The indices r_1, r_2, and r_3 are randomly chosen population indices that are mutually different and also different from *i*, which indexes the current object vector. Consequently, the population size, *NP*, must be greater than 3. *CR* represents probability, and therefore it ranges from 0 to 1. *F*, however, is a scaling factor that typically belongs to the interval (0,1+) [51].

If the trial vector, $\mathbf{u}_{i,G+1}$, has an objective function value equal to or lower than that of its target vector, $\mathbf{w}_{i,G}$, it replaces the target vector in the next generation; otherwise, the target retains its place in the population for at least one more generation [52]. The population for the next generation, $\mathbf{P}_{G+1}$, is selected from the current population $\mathbf{P}_G$ or the child population according to the rule

$$\mathbf{w}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1}, \text{ if } E\left(\mathbf{y}, f\left(\mathbf{x}, \mathbf{w}_{i,G+1}\right)\right) \le E\left(\mathbf{y}, f\left(\mathbf{x}, \mathbf{w}_{i,G}\right)\right) \\ \mathbf{w}_{i,G}, \text{ otherwise} \end{cases} . \qquad (3.53)$$

Once the new population is installed, the process of mutation, recombination and selection is repeated until the optimum is located or a pre-specified termination criterion is satisfied, *e.g.*, the number of generations reaches a preset maximum, G_{max} [52]. A flow chart of DE is shown in Figure 3.11.

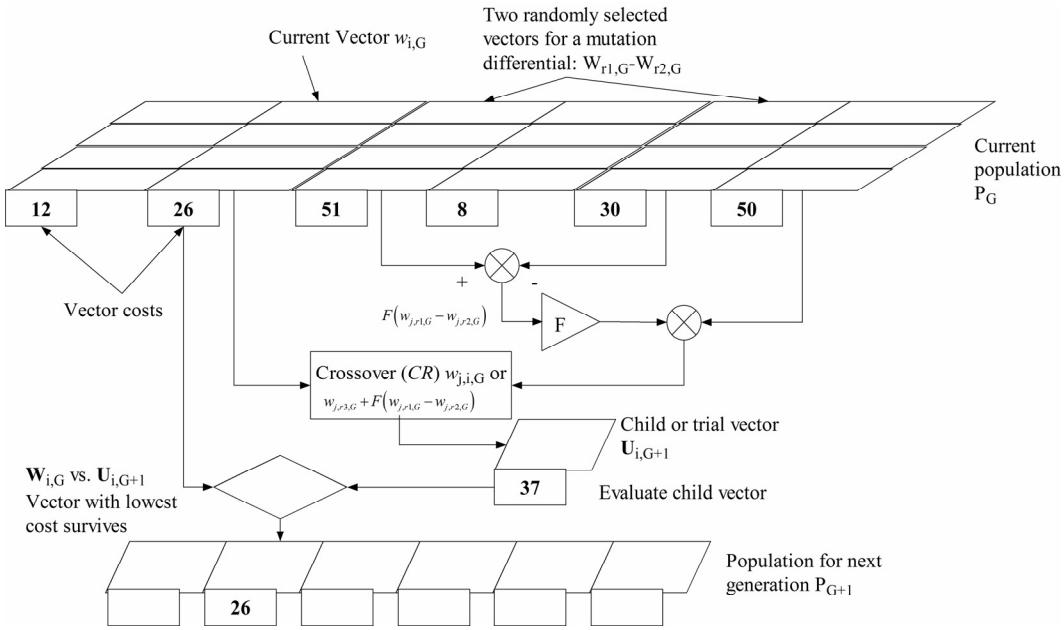


Figure 3.11. The procedure of the Differential Evolution algorithm [51].

The Reference Model

The desired linear time-invariant second-order reference model was selected to form a reference signal for the DE algorithm in the offline teaching process. The transfer function of the reference model is

$$G_{ref}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{3.54}$$

where the positive constants ζ and ω_n are chosen to shape the reference signal for a given input signal. Various methods exist for finding parameters for the reference model, such as the ITAE or Bessel transfer functions. It is the Bessel function that is used here. This does not have an overshoot when ω_n equals one, but when ω_n is greater than one the overshoot appears [53]. Parameters of the system were chosen such a way that the system was as fast as possible. The natural frequency ω_n was set to 100 rad/sec, with a damping ratio ζ of 1.732.

3.3 Observer-Based Nonlinear Controllers

Although it is assumed in state feedback control that all the state variables are available for feedback, not all state variables are measured for feedback in practice, and therefore we need to estimate the unavailable state variables. The *separation principle* is used for this purpose, as it allows the output-feedback problem to be decomposed into two subproblems, which can be solved separately: the design of a state observer and the design of a state-feedback controller. This theorem is valid on the assumption that the observer error converges to zero. Considering a linear time-invariant system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{aligned} \tag{3.55}$$

where **A** is called the state matrix, **B** the input matrix, **C** the output matrix and **D** the direct transition matrix, the presence of matrix **D** means that there is a direct connection between the input **u**(t) and the output **y**(t), without the intervention of the state **x**(t). Since the matrix **D** is absent in an overwhelming majority of practical applications, the assumption that **D**=0 is valid in most cases. The system is assumed to be controllable and observable (Appendix A.2). For such a system it is straightforward to show that the separation principle applies for both a full-order and reduced-order observer. For nonlinear systems the separation principle does not in general apply, and state observer design is generally coupled with controller design [20]. For some particular classes of systems, however, such as bilinear systems, or systems with certain structural interconnection properties, the separation principle can be used [54]. When the Input-to-State-Stability (ISS) property can be assured for disturbances entering the states additively in a stabilizing state feedback law, estimates from a converging state observer can be used in a certainty equivalence approach [54]. The idea of the separation principle is presented in Figure 3.12. The overall control law uses a certainty equivalence combination of a linear time-varying state-feedback controller and a separately designed state observer.

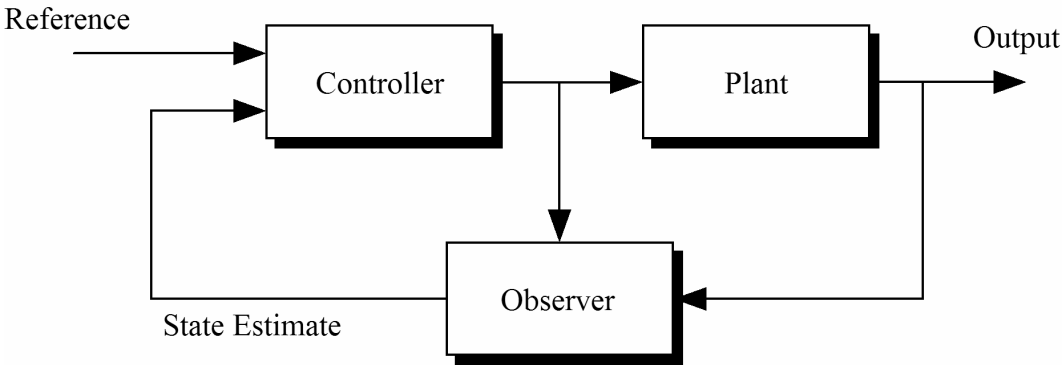


Figure 3.12. An observer-based control scheme.

In this thesis the mechanism is simplified in the form of a spring-mass system, which is easy to model with linearized differential equations without loss of generality. The system states are estimated by the function of the mover position. This approach allows use of the separation principle with a nonlinear controller. The Kalman filter theory, as used in the thesis, will be introduced in the following chapter.

3.3.1 The Kalman Filter

Although classical control system theory assumes that all state variables are available for feedback, this is not so in practice. Consequently, we need to estimate the unavailable state variables, which means that the system must satisfy the observability criterion (Appendix A.2). There are several methods for estimating unmeasurable state variables without a differentiation process, a procedure commonly called observation [55]. If the state observer observes all the state variables in a system regardless of whether or not they are available for direct measurement, it is called a full-order observer. An observer that estimates fewer than n state variables, where n is the dimension of the state vector, is called a reduced-order state observer, or simply, a reduced-order observer. If the order of a reduced-order observer is the minimum possible, the observer is called a minimum-order observer.

The Kalman filter is an optimum observer, implying that the observer gain, called here the Kalman gain, is optimally chosen, whereas with a linear observer the gains are positioned arbitrarily. The following Kalman filter equations are appropriate for linear systems, and in

particular for full order observers. If the measured state is non-linear, then a modified form of the Kalman filter known as an extended Kalman filter (EKF) can be used.

The dynamic model of a system in the Kalman filter may be presented in state space form as in Equation (3.55). We can assume that the state **x** is to be approximated by the state $\hat{\mathbf{x}}$ of the dynamic model:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_e(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}), \tag{3.56}$$

which represents the state observer. The last term on the right-hand side of this model equation is a correction term that involves the difference between the measured output **y** and the estimated output $\mathbf{C}\hat{\mathbf{x}}$. This difference is usually called the measurement residual *r,* and the matrix $\mathbf{K}_e$ serves as a weighting matrix. The correction term monitors the state $\hat{\mathbf{x}}$. If we now choose the gain matrix $\mathbf{K}_e$ so that the error dynamics are asymptotically stable at an adequate speed of response, the equation takes the form of the linear observer which was introduced by D. Luenberger in [56]. The design technique for the Luenberger observer is equivalent to pole placement in feedback system design, where the residual *r* has the role of the error typical of a feedback system.

If we now choose the gain matrix $\mathbf{K}_e$ optimally, it will take the form of the Kalman filter introduced in [57], the essential properties of which are derived from the requirements that the state estimate should be a linear combination of the previous state estimate and current measurement information unbiased with respect to the true state and optimal in terms of having minimum variance with respect to the true state. State equations for a continuous Kalman filter are usually introduced as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{v}(t), \end{aligned} \tag{3.57}$$

where **w** is a random process that accounts for the noise arising from mismodelling and **v** is a random noise process accounting for measurement noise. These general models allow a linear Kalman filter (LKF) to be derived in a discrete-time formulation. Since we can assume that

the control is applied with a computer through a ZOH, the states are represented by the sampled signal $\{\mathbf{x}_k : k = ..., -1, 0, 1, ...\}$, where $\mathbf{x}_k$ is the state at some future time t_k and the state vector at the next sampling time t_{k+1} is the linear function of $\mathbf{x}_k$ and $\mathbf{u}_k$. The linear difference equations for the discrete Kalman Filter are therefore

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}\mathbf{u}_k + \mathbf{w}_k , \tag{3.58}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k , \tag{3.59}$$

where the **H** is the same matrix as **C** in a continuous system, and

$$\mathbf{\Phi} = e^{\mathbf{A}T_s} , \tag{3.60}$$

$$\mathbf{\Gamma} = \int_0^{T_s} e^{\mathbf{A}s} ds \mathbf{B} , \tag{3.61}$$

where T_s is the time step for periodic sampling. The random variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise, respectively. They are assumed to be independent, white and with a normal probability distribution. The white noise covariances for **w** and **v** are **Q** and **R**, respectively. In practice, the process noise covariance **Q** and measurement covariance **R** matrices may change with each step or measurement, although we assume here that they are constant. The definitions for **w** and **v** are such that they are random sequences with a zero mean, that is

$$\varepsilon\{\mathbf{w}_k\} = \varepsilon\{\mathbf{v}_k\} = 0 , \tag{3.62}$$

have no time correlation or are "white" noise, that is

$$\varepsilon\{\mathbf{w}_i \mathbf{w}_j^T\} = \varepsilon\{\mathbf{v}_i \mathbf{v}_j^T\} = 0 \text{ if i} \neq \text{j}, \tag{3.63}$$

and have covariances or mean square "noise levels" defined by

$$\varepsilon\{\mathbf{w}_k \mathbf{w}_k^T\} = \mathbf{Q} \text{ and } \varepsilon\{\mathbf{v}_k \mathbf{v}_k^T\} = \mathbf{R} . \tag{3.64}$$

The correlation among the measured data points can be computed and put into the measurement covariance matrix **R**. The measurement noise covariance **R** is usually measured prior to operation of the filter. It can be obtained from off-line sample measurements in order to determine the variance of the measurement noise.

The process noise covariance **Q** is generally more difficult to determine, because we are typically unable to observe the processes we are estimating directly. The process noise is usually used as a tuning parameter to ensure that the filter is operating correctly [48]. If the target is known to be highly manoeuvrable, then a large value for the standard deviation σ_w of the covariance ($\mathbf{Q}=[\sigma_w^2]$) is appropriate, but for targets of low manoeuvrability a small value of σ_w is appropriate. Sometimes a relatively simple process model can produce acceptable results if one injects enough uncertainty into the process. Overall, the errors are smoother for a smaller value of σ_w than they are for a larger value. In a sense, the choice of σ_w can be thought of as being analogous to setting the cut-off frequency of a low pass filter acting on the estimation error. Furthermore, the selection of **Q** and **R** is one of the central features in tuning the Kalman filter to provide the optimum sensor performance.

We now allow the Kalman gain $\mathbf{K}_k$, which is equal to $\mathbf{K}_e$ in equation (3.56), to vary with the time step, and attempt to select $\mathbf{K}_k$ so that the estimate for $\mathbf{x}_k$ is optimal. Let us pretend that without using the current measurement $\mathbf{y}_k$ we already have an a priori estimate for the state at the time of a measurement, which we will call $\hat{\mathbf{x}}_k^-$. The problem at this point is to update this old estimate based on the current new measurement [58]. The object in deriving Kalman filter equations is to find an equation that gives an a posteriori state estimate $\hat{\mathbf{x}}_k$ as a linear combination of an a priori estimate $\hat{\mathbf{x}}_k^-$ and a weighted difference between measurement $\mathbf{y}_k$ and a predicted measurement $\mathbf{H}\hat{\mathbf{x}}_k^-$ [56]. The solution equation is

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^-), \tag{3.65}$$

where the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \mathbf{W}_k^{-1}. \tag{3.66}$$

The term $\mathbf{W}_k$ is referred to as the innovation covariance. The whole equation for the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R} \right)^{-1}. \tag{3.67}$$

The term $\mathbf{P}_k$ is the covariance of the new estimate, which is

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \mathbf{P}_k^- , \tag{3.68}$$

where $\mathbf{I}$ is a N×N identity matrix. The estimated error covariance $\mathbf{P}_k$ is independent of the measurement, as seen in Figure 3.13. It could even be pre-calculated before any measurements are made. This is a direct consequence of the underlying assumption that the noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ are known a priori [59]. In practice, these covariances can be used to control the rate of convergence of the Kalman filter estimator, with the optimum values depending on the manoeuvrability of the target being tracked.

We can now derive a real-time algorithm for the Kalman filter. The first step is to initialize the process by making suitable guesses at an estimate for the state vector $\hat{\mathbf{x}}_k$ and the covariance of estimation error $\mathbf{P}_k$. This procedure is called the time update:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi} \hat{\mathbf{x}}_{k-1} + \mathbf{\Gamma} \mathbf{u}_{k-1}, \tag{3.69}$$

$$\mathbf{P}_k^- = \mathbf{\Phi} \mathbf{P}_{k-1} \mathbf{\Phi}^T + \mathbf{Q} . \tag{3.70}$$

The next step is to calculate the Kalman gain, update the state estimate by means of measurements and update the error uncertainty. This procedure is called a measurement update:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R} \right)^{-1}, \tag{3.71}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right), \tag{3.72}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H} \right) \mathbf{P}_k^- . \tag{3.73}$$

After each pair of time and measurement updates, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. A block diagram depicting the form of the discrete Kalman filter is presented in Figure 3.13.

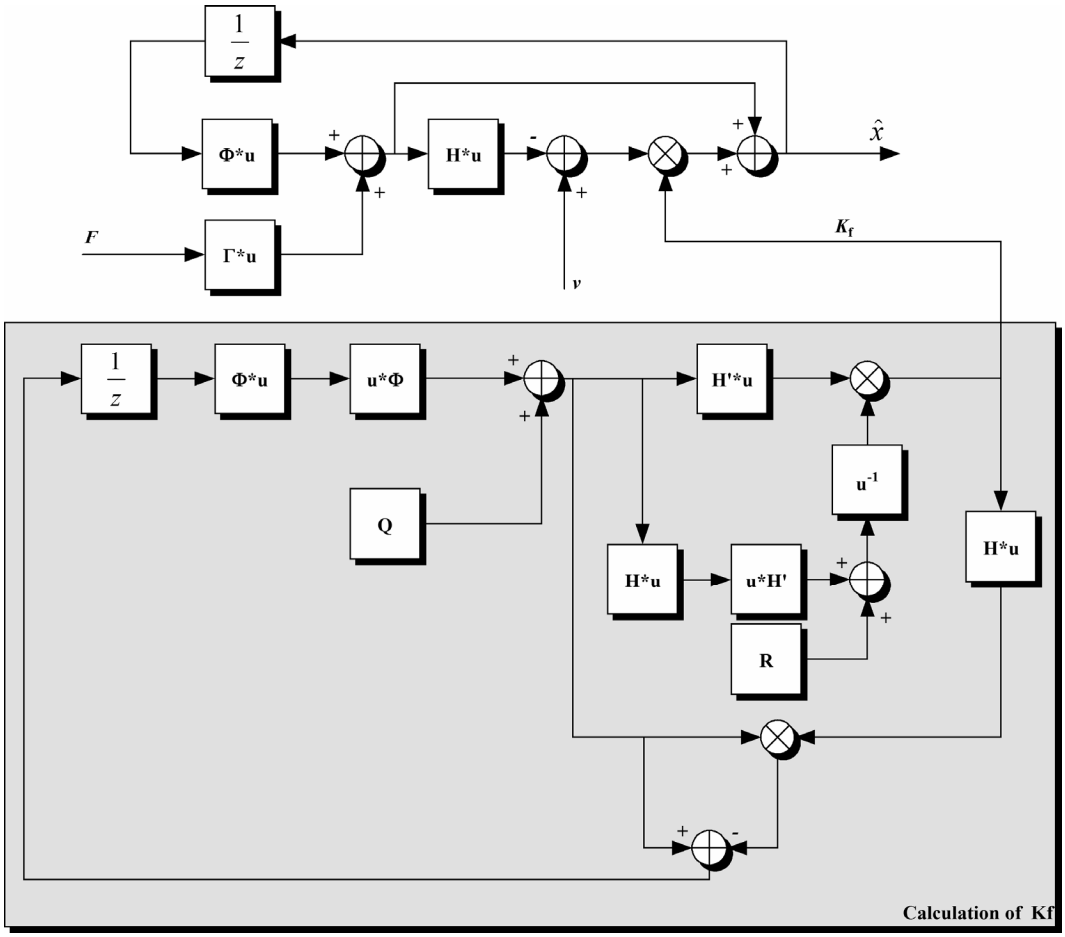


Figure 3.13. Block diagram of the discrete Kalman filter.

As Bishop [48] notes, it should be stressed that the Kalman filter is not the state observer algorithm that is best suited for all applications. Its strength lies in its light computational requirements and the real-time availability of a state estimate in the presence of accurate measurement information. Batch techniques such as least-squares estimation may be more appropriate, however, in applications where the dynamic process is to be modelled to a high degree of fidelity, the measurements are not of uniform accuracy and real-time operation is not an issue.

Chapter 4

4 Experimental Work and Results

The PMLSM model introduced in Chapter 2 is verified with measurements from a physical linear motor application in Section 4.1, employing parameters analysed or measured from the laboratory setup. Section 4.2 then describes the implementation of an adaptive backstepping controller for velocity control in a flexible mechanism and provides a detailed analysis based on the simulation software. The implementation of the hybrid neural controller is reported in Section 4.3, and the state space model of the mechanism for the Kalman filter is introduced in Section 4.4. The remaining responses obtained from the backstepping and hybrid neural controller are compared in Section 4.5.

4.1 Verification of the Simulation Model

The simulation results were compared with the measurements performed using the laboratory setup. The application studied here is a commercial three-phase permanent magnet linear synchronous motor (PMLSM) with a rated force of 675N. The parameters given by the motor and inverter manufacturer are in listed in Appendix A.3. The moving part (mover) consists of a slotted armature, while the surface permanent magnets (SPMs) are mounted along the whole length of the path (stator). The permanent magnets are slightly skewed (1.7°) in relation to the normal, which reduces the detent force, the attractive force between the PMs and the slotted core of the armature, and some higher space harmonics [23]. The moving part is set up on an aluminium base with four recirculating roller bearing blocks on steel rails. The position of the linear motor was measured using an optical linear encoder with a resolution of one micrometre. The spring-mass mechanism was built on a tool base in order to act as a flexible

tool (e.g. a picker that increases the level of excitation). The mechanism consists of a moving mass which can be altered in order to change the natural frequency of the mechanism and a break spring which is connected to the moving mass on the guide.

The commercial inverter was analysed by means of a Bode diagram for the current control loop (x_M/i_{set}). The bandwidth of the system (-3dB closed-loop) was 650 Hz. The manufacturer's pre-set current controller parameters were captured in the simulation model. The Bode diagram for the current control loop of the laboratory setup is shown in Figure 4.1.

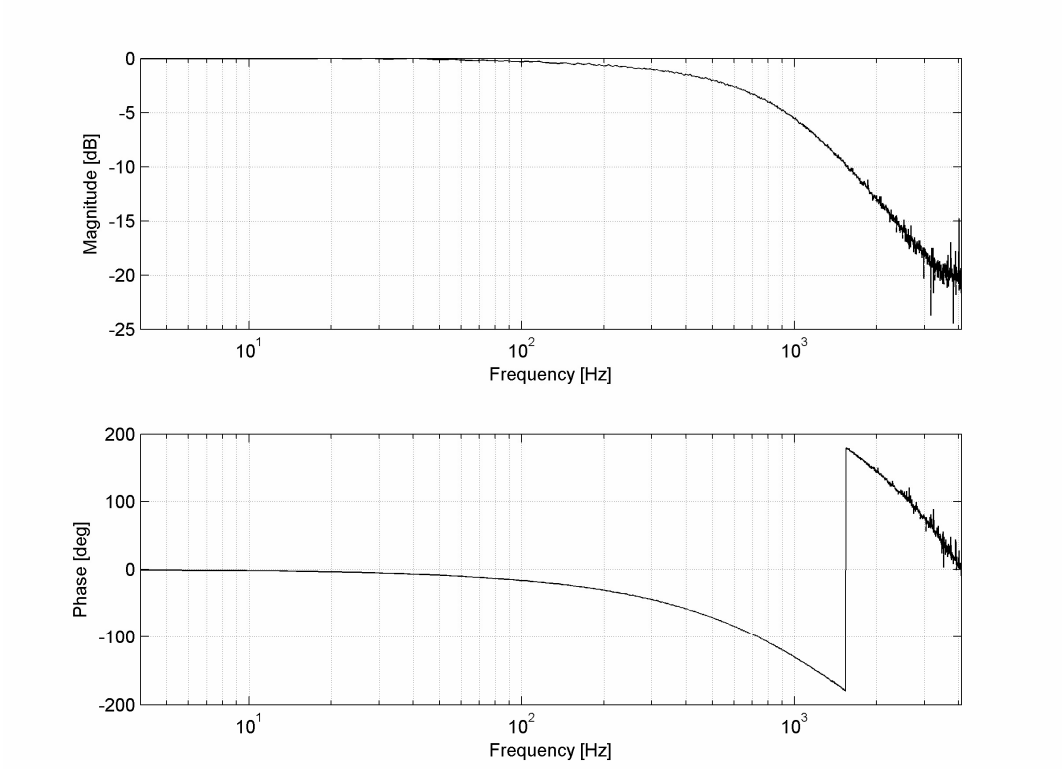


Figure 4.1. Bode diagram of the current controller loop.

The simulation model was implemented and analysed in the MatLab/Simulink® software. The PWM inverter was modelled as an ideal voltage source, and common Simulink blocks were used for the model. The time step of the ode4 Runge-Kutta integrator in the analysis was 10 µs, except for the velocity controller, which had a time step of 1 ms (sampling rate 1 kHz). The physical linear motor application was driven in such a way that the proposed velocity controller was implemented in Simulink to gain the desired force reference. The derived algorithm was transferred to C code for real-time use by the dSPACE© digital signal

processor (DSP). The force command F^* was fed into the drive of the linear motor (Figure 2.2) using the DS1103 controller board. The A/D and D/A converters in this system both have 16bit resolution. The laboratory setup is presented in Figure 4.2.

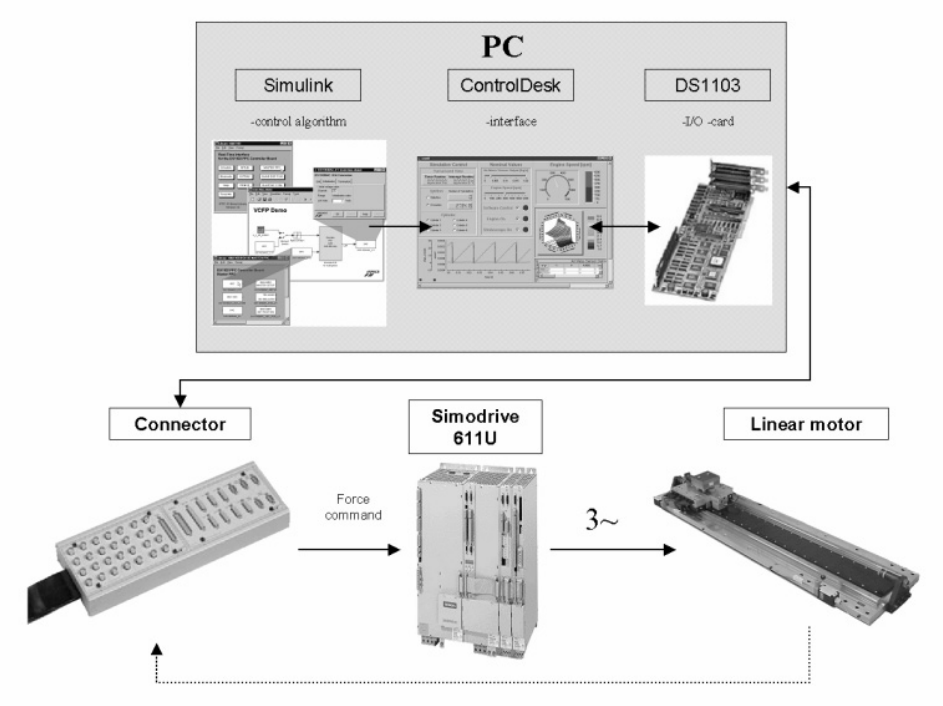
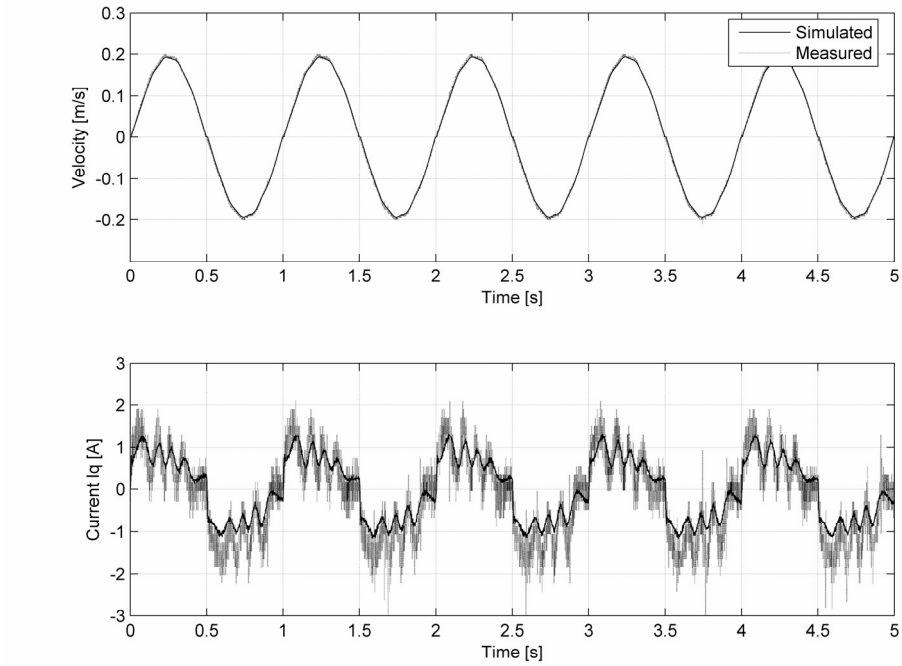
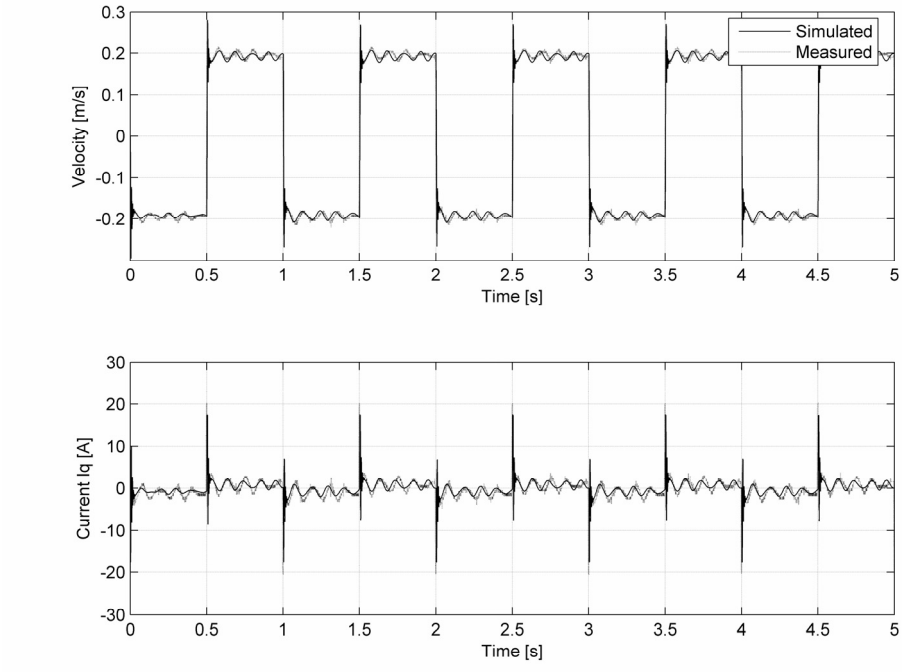


Figure 4.2. The laboratory setup.

The simulation results were compared with those measured in the reference system, as shown for the measured and simulated velocity responses and force generating quadrature currents in Figure 4.3. The sine and step functions were used as reference velocities, the amplitude being 0.2 m/s in both cases and the frequency 1 Hz.



(a)



(b)

Figure 4.3. Comparison of the measured and simulated quadrature currents and velocity responses in the case of (a) a sine velocity reference and (b) a step velocity reference.

As can be seen in the figure, the responses are very close to each other. The simulation model was used to design load controllers for the PMLSM system. We will look next at the design of the backstepping controller and neural network-based hybrid controller for the flexible mechanism.

4.2 Backstepping Control for a Flexible Mechanism

We will first derive equations for a mechanical system. It is assumed that the motor dynamics are negligible, *i.e.* the dynamics of the mechanism predominate. This is due to the fast dynamics of a linear motor relative to a flexible mechanism. In control system design, the mechanism can usually be simplified to a 2-dof system, as mentioned in Section 2.2. The equations of motion for this system are

$$\begin{aligned} m_M \ddot{x}_M &= k\Delta x + b\Delta\dot{x} + u \\ m_L \ddot{x}_L &= -k\Delta x - b\Delta\dot{x} \end{aligned}, \tag{4.1}$$

where k is the spring constant, b is the damping factor, m_{M} and m_{L} are the masses of the mover and load, x_{M} and x_{L} are the positions of the mover and load, and Δx and $\Delta\dot{x}$ are the position and velocity differences between the masses, respectively. The state variables for the controller are set as

$$\begin{aligned} x_1 &= \dot{x}_L \\ x_2 &= \Delta x = x_L - x_M \\ x_3 &= \Delta\dot{x} = \dot{x}_L - \dot{x}_M \end{aligned}. \tag{4.2}$$

In an adaptive control problem the uncertainty is reflected by an unknown parameter θ, the estimate for which, $\hat{\theta}(t)$, is used to define an adaptive control law. As the spring constant k and load mass m_{L} are unknown, let us define the two unknown parameters as

$$\theta_1 = -\frac{k}{m_M}, \tag{4.3}$$

$$\theta_2 = \frac{k}{m_L}. \tag{4.4}$$

Using the state variables and assuming that the damping factor is negligible, we can rewrite the equations of motion as

$$\begin{aligned} \dot{x}_1 &= \theta_2 x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -(\theta_1 + \theta_2)x_2 + \frac{1}{m_M}u \end{aligned} \tag{4.5}$$

$$y = x_1, \tag{4.6}$$

where u is the input force. The control input y is the velocity of the load. The objective for the system is to design a control to force the load velocity to track a desired velocity, denoted by $\dot{x}_d$. We will assume that $\dot{x}_d$ and its first time derivative are continuous and bounded for all time [13]. Since we are solving the tracking problem, we will denote the associated tracking error e_1 as

$$e_1 = x_1 - \dot{x}_d. \tag{4.7}$$

The controller is designed to drive the tracking error, denoted by Equation (4.7), towards zero. The time derivative of the error is

$$\dot{e}_1 = \dot{x}_1 - \ddot{x}_d. \tag{4.8}$$

We can now assume that $\ddot{x}_d = 0$, as a result of which

$$\dot{e}_1 = \dot{x}_1 = \theta_2 x_2. \tag{4.9}$$

We now set a positive semidefinite Lyapunov function as a candidate for e_1. There is no universal method for constructing Lyapunov functions for a stable system, but in the case of an electrical or mechanical system there are natural Lyapunov functions such as total energy functions that serve as candidates. In other cases, it is basically a matter of trial and error. A

good first attempt is to test quadratic functions [39], whereupon the function depends on the unknown parameter θ_2, *i.e.* the Lyapunov function is

$$V_1 = \frac{1}{2\theta_2} e_1^2 . \tag{4.10}$$

This gives the time derivative of Equation (4.10):

$$\dot{V}_1 = \frac{1}{\theta_2} e_1 \dot{e}_1 = e_1 x_2 . \tag{4.11}$$

Let us also define

$$e_2 = x_2 - x_2^* = x_2 + c_1 e_1 , \tag{4.12}$$

where x_2^* is the first virtual control and c_1 is a positive design constant. Solving x_2 in Equation (4.12) and substituting it into Equation (4.11), we now obtain

$$\dot{V}_1 = e_1 (e_2 - c_1 e_1) = -c_1 e_1^2 + e_1 e_2 . \tag{4.13}$$

Thus the time derivative of x_2* is

$$\dot{x}_2^* = -c_1 \dot{e}_1 = -c_1 \theta_2 x_2 . \tag{4.14}$$

Solving the time derivative of Equation (4.12) and substituting Equation (4.14) into it, we obtain

$$\dot{e}_2 = \dot{x}_2 - \dot{x}_2^* = x_3 + c_1 \theta_2 x_2 . \tag{4.15}$$

The second Lyapunov function is

$$V_2 = V_1 + \frac{1}{2} e_2^2 , \tag{4.16}$$

and differentiation leads to

$$\dot{V}_2 = \dot{V}_1 + e_2\dot{e}_2 = -c_1e_1^2 + e_1e_2 + \left(x_3 + c_1\theta_2x_2\right)e_2 . \tag{4.17}$$

The unknown parameter θ_2 is the sum of the estimate $\hat{\theta}_2$ and the estimation error ξ_2, *i.e.*

$$\theta_2 = \hat{\theta}_2 + \xi_2 . \tag{4.18}$$

Let

$$e_3 = x_3 - x_3^* , \tag{4.19}$$

where the virtual control x_3^* is

$$x_3^* = -c_2e_2 - e_1 - c_1\hat{\theta}_2x_2 . \tag{4.20}$$

Substituting (4.18) and x_3 from Equation (4.19) into Equation (4.17), we obtain

$$\begin{aligned}\dot{V}_2 &= -c_1e_1^2 + e_1e_2 + e_2\left(x_3 + c_1\theta_2x_2\right) = -c_1e_1^2 + e_1e_2 + e_2\left(e_3 + x_3^* + c_1\theta_2x_2\right)\\ &= -c_1e_1^2 + e_1e_2 + e_2\left(e_3 - c_2e_2 - e_1 - c_1\hat{\theta}_2x_2 + c_1\theta_2x_2\right)\\ &= -c_1e_1^2 + e_2e_3 - c_2e_2^2 + e_2c_1x_2\xi_2 .\end{aligned} \tag{4.21}$$

The Lyapunov function for the whole system is now

$$V_3 = V_2 + \frac{1}{2}e_3^2 + \frac{1}{2\gamma}\xi_1^2 + \frac{1}{2\delta}\xi_2^2 . \tag{4.22}$$

The time derivative of Equation (4.19) is

$$\dot{e}_3 = \dot{x}_3 - \dot{x}_3^* = -\left(\theta_1 + \theta_2\right)x_2 + \frac{u}{m_M} - \dot{x}_3^* \tag{4.23}$$

and the parameter $\dot{x}_3^*$, from Equation (4.20), is

$$\begin{aligned}\dot{x}_3^* &= -c_2\dot{e}_2 - \dot{e}_1 - c_1\dot{\hat{\theta}}_2 x_2 - c_1\hat{\theta}_2\dot{x}_2 = -c_2\left(x_3 + c_1\theta_2 x_2\right) - \theta_2 x_2 - c_1\dot{\hat{\theta}}_2 x_2 - c_1\hat{\theta}_2\dot{x}_2 \\ &= -c_2 x_3 - c_1 c_2\theta_2 x_2 - \theta_2 x_2 - c_1\dot{\hat{\theta}}_2 x_2 - c_1\hat{\theta}_2 x_3.\end{aligned} \tag{4.24}$$

Inserting this equation into Equation (4.23), we obtain

$$\begin{aligned}\dot{e}_3 &= -\left(\theta_1 + \theta_2\right)x_2 + \frac{u}{m_M} + c_2 x_3 + c_1 c_2\theta_2 x_2 + \theta_2 x_2 + c_1\dot{\hat{\theta}}_2 x_2 + c_1\hat{\theta}_2 x_3 \\ &= -\theta_1 x_2 + \frac{u}{m_M} + c_2 x_3 + c_1 c_2\theta_2 x_2 + c_1\dot{\hat{\theta}}_2 x_2 + c_1\hat{\theta}_2 x_3.\end{aligned} \tag{4.25}$$

The unknown parameter θ_1 is defined in the same way as θ_2 in Equation (4.18), as

$$\theta_1 = \hat{\theta}_1 + \xi_1. \tag{4.26}$$

The time derivative of the Lyapunov function in Equation (4.22) is

$$\dot{V}_3 = \dot{V}_2 + e_3\dot{e}_3 + \frac{1}{\gamma}\xi_1\dot{\xi}_1 + \frac{1}{\delta}\xi_2\dot{\xi}_2. \tag{4.27}$$

Inserting equations (4.21) and (4.25) into Equation (4.27) and defining $\dot{\xi} = -\dot{\hat{\theta}}$, we obtain

$$\begin{aligned}&\dot{V}_3 = \dot{V}_2 + e_3\dot{e}_3 + \frac{1}{\gamma}\xi_1\dot{\xi}_1 + \frac{1}{\delta}\xi_2\dot{\xi}_2 = -c_1 e_1^2 - c_2 e_2^2 \\ &+e_3\left(e_2 - x_2\hat{\theta}_1 + \frac{u}{m_M} + c_2 x_3 + c_1 c_2 x_2\hat{\theta}_2 + c_1\dot{\hat{\theta}}_2 x_2 + c_1\hat{\theta}_2 x_3\right) \\ &+\xi_1\left(-e_3 x_2 - \frac{1}{\gamma}\dot{\hat{\theta}}_1\right) + \xi_2\left(c_1 e_2 x_2 + e_3 c_1 c_2 x_2 - \frac{1}{\delta}\dot{\hat{\theta}}_2\right).\end{aligned} \tag{4.28}$$

If the gains in the estimation errors ξ_1 and ξ_2 in this equation are set at zero, we can solve the differentials of the estimates $\hat{\theta}_1$ and $\hat{\theta}_2$, *i.e.*

$$-e_3 x_2 - \frac{1}{\gamma}\dot{\hat{\theta}}_1 = 0 \rightarrow \dot{\hat{\theta}}_1 = -\gamma e_3 x_2, \tag{4.29}$$

$$c_1 e_2 x_2 + e_3 c_1 c_2 x_2 - \frac{1}{\delta}\dot{\hat{\theta}}_2 = 0 \rightarrow \dot{\hat{\theta}}_2 = \delta c_1 x_2 \left(e_2 + c_2 e_3\right). \tag{4.30}$$

To achieve global stability, the Lyapunov function must be a negative semi-definite, *i.e.* Equation (4.28) must satisfy

$$\dot{V}_3 = -c_1 e_1^2 - c_2 e_2^2 - c_3 e_3^2. \tag{4.31}$$

If the estimation errors in Equation (4.28) are set at zero and the last terms of Equations (4.28) and (4.31) are assumed to be equal, we get

$$-c_3 e_3 = e_2 - \hat{\theta}_1 x_2 + \frac{u}{m_M} + c_2 x_3 + c_1 c_2 \hat{\theta}_2 x_2 + c_1 \dot{\hat{\theta}}_2 x_2 + c_1 \hat{\theta}_2 x_3. \tag{4.32}$$

Substituting Equation (4.30) into Equation (4.32), we can solve the final control input for the system, which is

$$u = m_M \left[-c_3 e_3 + \hat{\theta}_1 x_2 - e_2 - c_2 x_3 - c_1^2 x_2^2 \delta \left(e_2 + c_2 e_3\right) - c_1 \hat{\theta}_2 \left(c_2 x_2 + x_3\right)\right]. \tag{4.33}$$

Since we can assume that γ, δ, c_1, c_2, $c_3 > 0$, the adaptation laws (4.29) and (4.30) and the control (4.33) achieve asymptotic tracking of the system with unknown parameters m_L and k. The parameters for the controller are the adaptive gains δ and γ and the controller gains c_1, c_2 and c_3 (Table 4). These and other parameters are first tuned by a trial and error method in the control of the nonlinear simulation mode, but analytical methods can also be used to choose the controller gains, one of these being introduced in Reference [60]. A nonlinear control system is usually reformulated as a linear, time-invariant system in these methods, but unfortunately this might be a very tricky task and less work is required when selecting the parameters by trial and error in an accurate simulation model. It was observed that the control gain c_1 has very large impact on the system dynamics, and tuning of this gain is essential when choosing the parameters, as clarified in Figure 4.4, which compares the responses of the load at different values of c_1, given that the other control parameters remain constant.

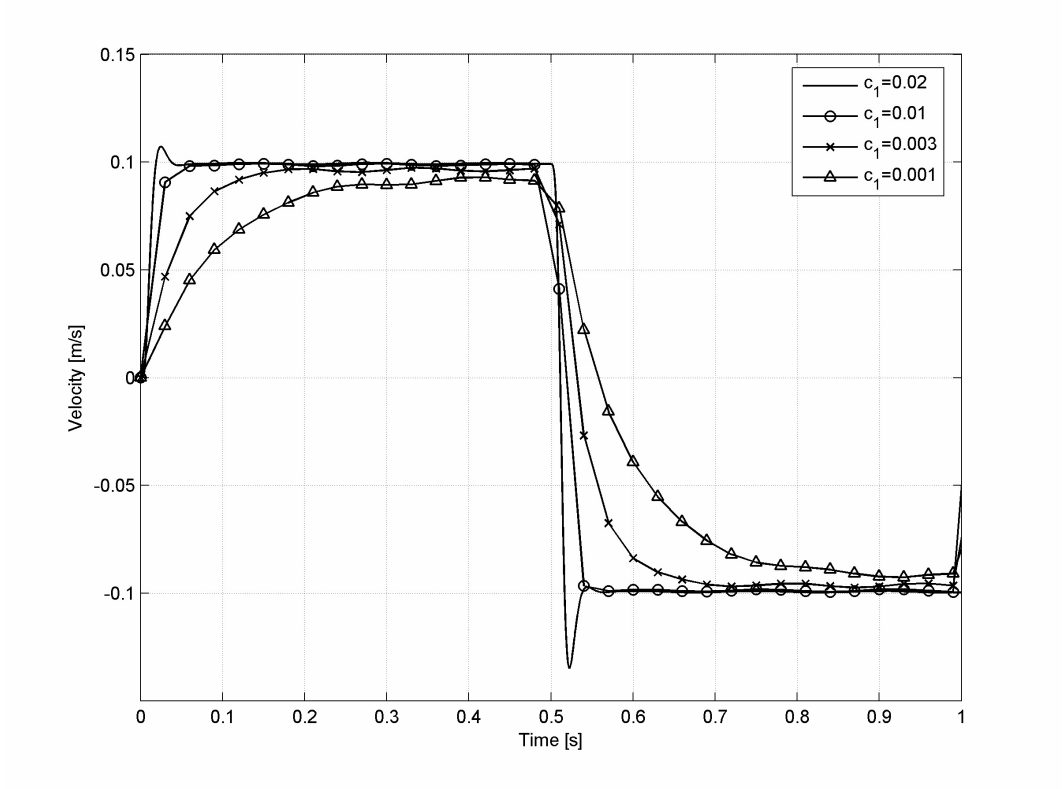


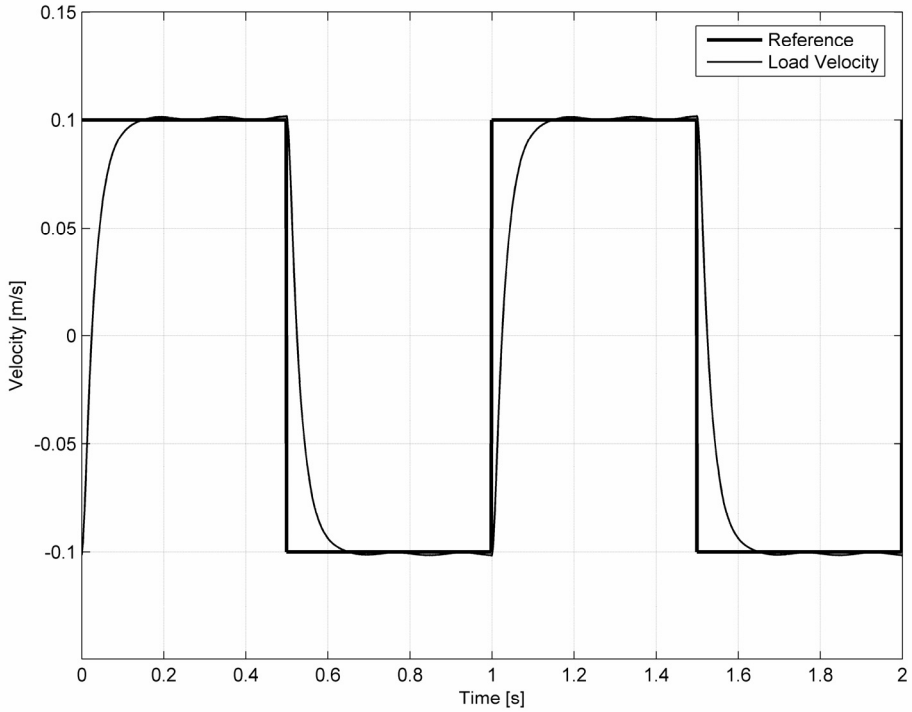
Figure 4.4. Velocities of the load at different values of the controller gain c_1.

The value of c_1 was thus chosen to be 0.01. The other parameters do not affect the system response so much and they have wider range within which to choose correct value. The controller parameters are listed in Table 4.

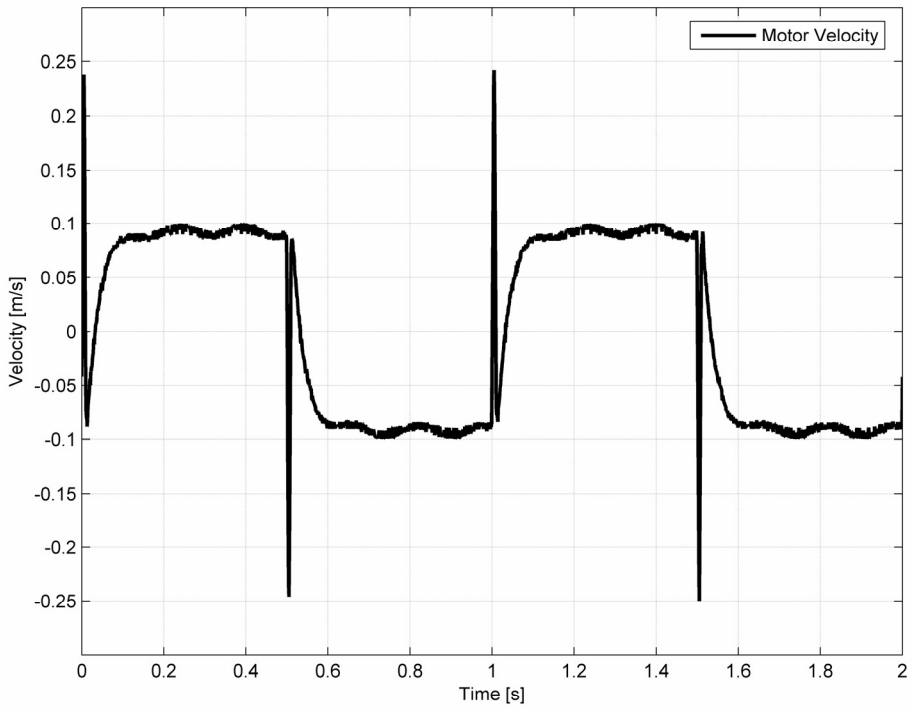
Table 4. Control Parameters

Symbol	Parameter	Value
c_1	Control Parameter	0.01
c_2	Control Parameter	1300
c_3	Control Parameter	250
γ	Adaptation Gain	50000
δ	Adaptation Gain	50000

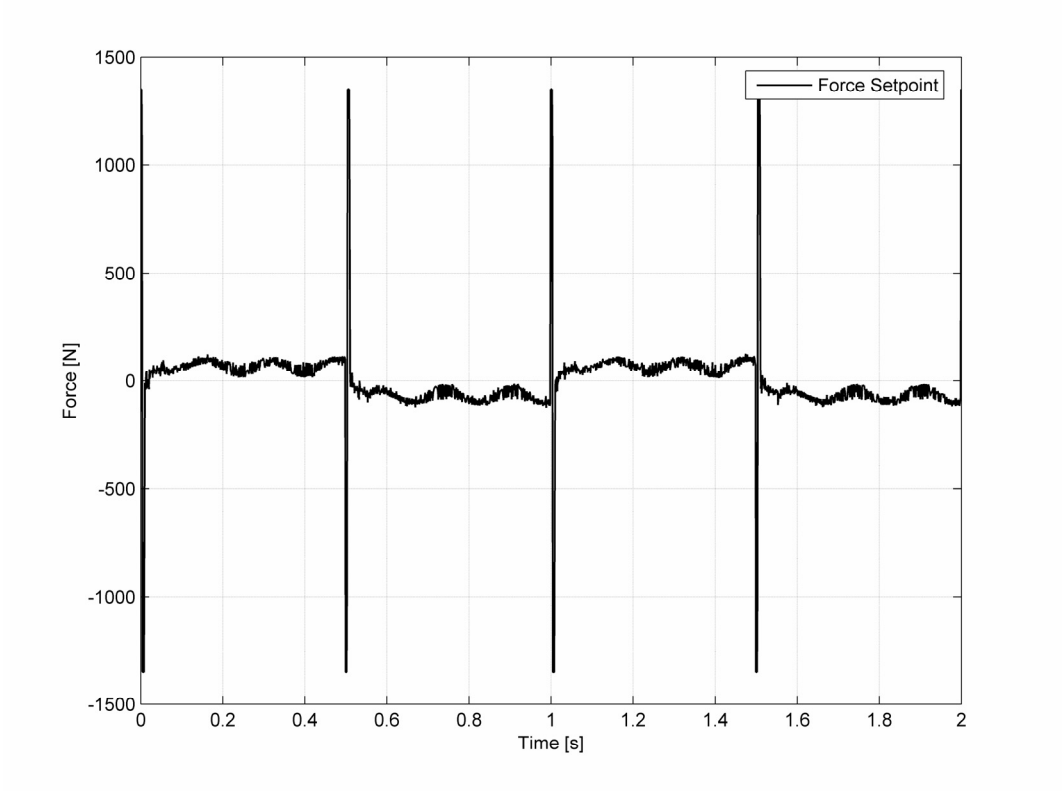
After testing the control in the simulation model, it was implemented in the physical linear motor application and final tuning was carried out. The velocity response of the load in the physical linear motor when using an adaptive backstepping controller is shown in Figure 4.5 (a) and the mover velocity and force set point of the controller in Figure 4.5 (b) and (c), respectively. The reference was a step function of amplitude 0.1 m/s and frequency 1 Hz.


Reference
Load Velocity
Velocity [m/s]
Time [s]
0.15
0.1
0.05
0
-0.05
-0.1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

(a)


Motor Velocity
Velocity [m/s]
Time [s]
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

(b)



(c)

Figure 4.5. Measured velocity of the load (a), velocity of the mover (b) and force setpoint (c) in the case of a backstepping load controller.

The velocity of the load follows the reference command accurately, and even the system stiffness is relatively loose. The mover velocity, shown in Figure 4.5 (b) reflects the suppression movement of the mover due to active vibration control. The controller was next tested more accurately with the simulation model.

Lyapunov stability is not enough in many applications. For example, when a load is disturbed from its nominal position, we not only want it to maintain its position in a range determined by the magnitude of the disturbance, *i.e.*, Lyapunov stability, but also require that the position should gradually return to its original value, *i.e.* asymptotic stability. This was tested in the simulation model by setting an initial displacement for the load. Systems with and without the backstepping controller are compared in Figure 4.6. In the first case the load was damped naturally (grey curve) and in the second case vibration control was used (black curve). The load (2.6 kg) had an initial displacement of 0.05 m.

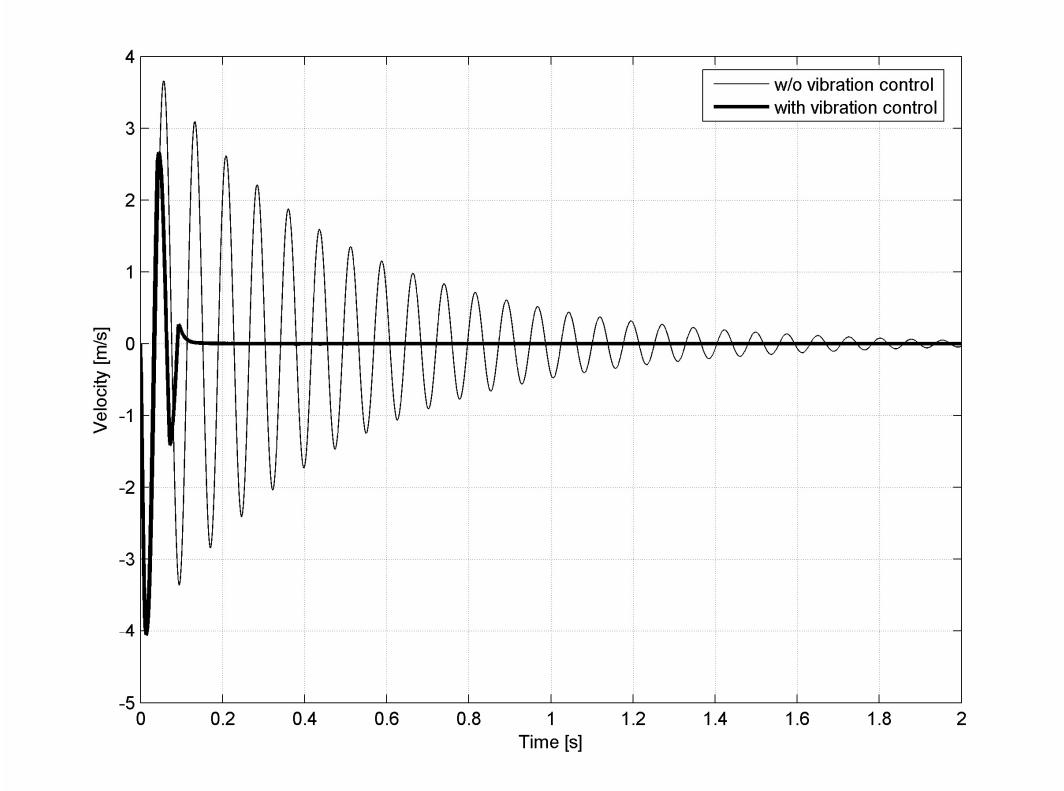


Figure 4.6. Suppression of vibration of the load by natural damping and by active vibration control.

In this system, with the parameters listed in Table 4, the suppression time in active vibration control is about 5% of that in the system without active vibration control, as seen in Figure 4.6. In the following figure the convergence is shown in phase-plane, in which the trajectories of the simulation model are compared using different initial load positions (0.06 m, 0.04 m and 0.02 m). The state estimator is not used in these simulations, in order to avoid estimation error. States x_1 and x_2 are the position and velocity of the load, respectively. The figure consists of two curves: the phase plane as a function of time and curves projected in two dimensions.

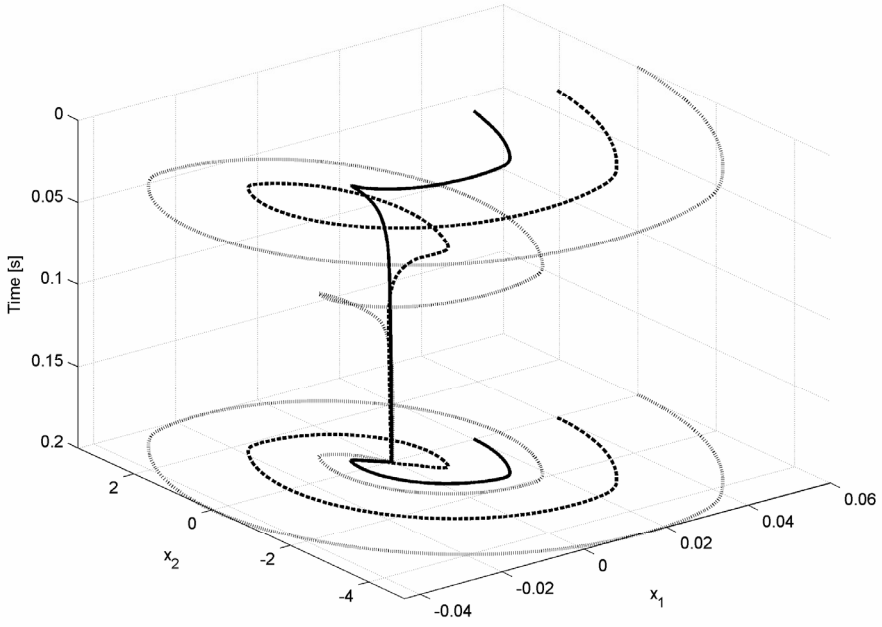


Figure 4.7. Phase plane plot of the load.

As mentioned earlier, asymptotic stability is a highly desirable property in a control system. Even more favourable, however, is GAS, *i.e.* that the state tends towards an equilibrium from an arbitrary initial condition over an infinite time. This requirement is verified in terms of the Lyapunov stability theorem (Definition 3.2) in Figure 4.7, where the trajectories from all feasible initial positions converge to an equilibrium, which in this case is zero. In small initial positions the controller is even exponentially stable. It should be keep in mind that these control parameters used in the simulations are tuned to a step function reference. Exponential stability can also be achieved for higher initial values using other parameters.

The dynamics of the adaptation law were then tested with the simulation model, which was excited with the step command. The mass of the load was altered from 2 kg to 5 kg at 10 seconds and the spring constant changed from 13700 N/m to 23700 N/m at 20 seconds. The unknown controller parameters in this simulation are presented in Figure 4.8.

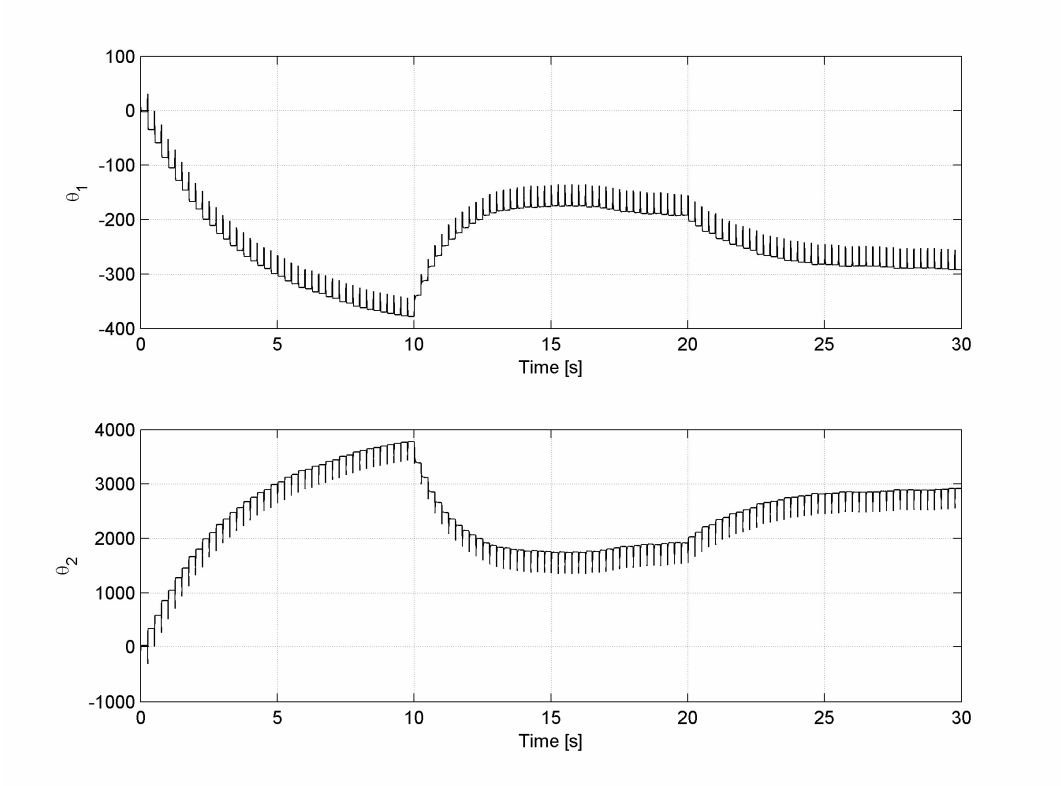


Figure 4.8. Dynamics of the adaptation law in the case of variations in load mass and spring constant.

The interesting phenomenon in the proposed backstepping controller is that the estimates do not converge to the right values. This is due to the estimation equations, which are differential functions of controller states and parameters, so that the final values are highly dependent on the control parameters. The higher the adaptation gains are, the faster the adaptation dynamics. On the other hand, adaptation gains that are too high might make the controller unstable, and therefore final values are a compromise between the dynamics of adaptation and the response of the system.

Finally, the effect of the estimator (the Kalman filter) on the system dynamics was studied with the simulation model by first controlling the system with the backstepping controller without an estimator and then including the estimator in the model. The responses of the load velocities are compared in Figure 4.9, in which the difference in load velocity between the backstepping controller with and without estimator is examined.

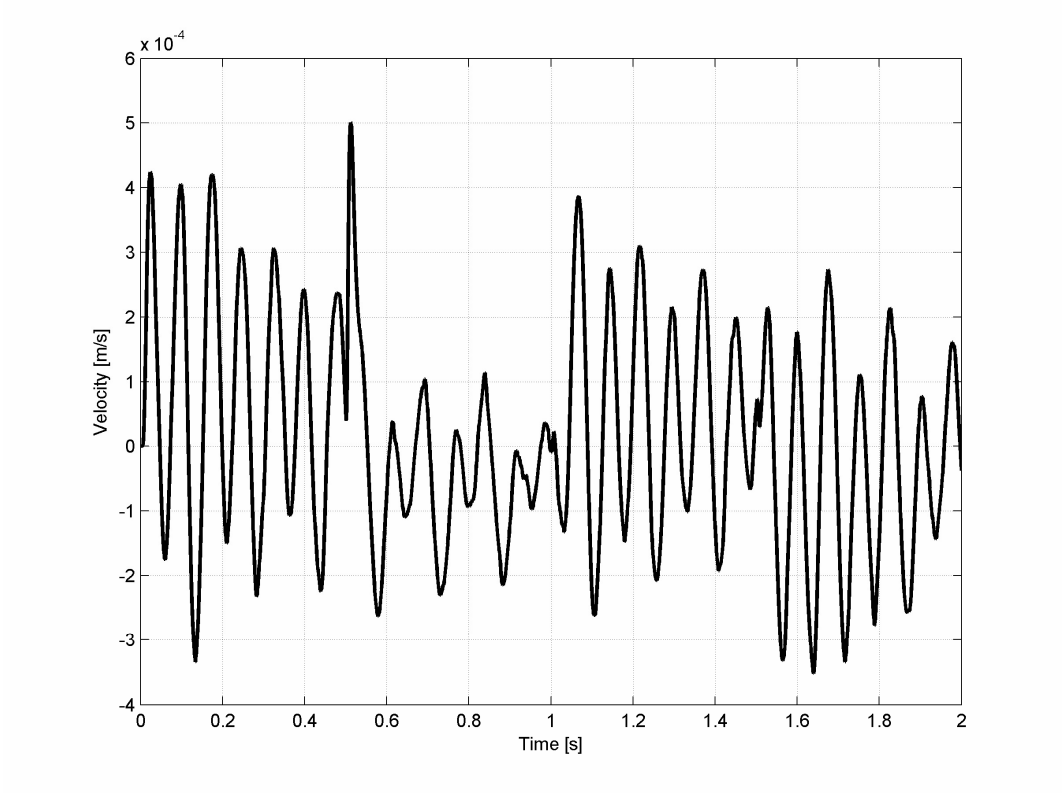


Figure 4.9. Velocity difference between backstepping controlled loads with and without the estimator.

In practice the velocity difference is so small that the estimated states are very close to the measured ones. The problem arises in the situation when the estimated states are used and the system parameters are changed drastically. As the estimator is not of the adaptive type, it fails in the estimation in which the system parameter is changed. One possibility would be to use a nonlinear estimator, but this topic is not included in the present thesis. This issue is discussed in Reference [61], where the Extended Kalman Filter (EKF) is used to estimate not only system states but also system parameters. The problem can naturally be avoided if it is possible to measure the state of a flexible mechanism and use it or estimated states derived from it as a feedback for the controller.

4.3 Hybrid Neural Controller

Several neural network-based adaptive architectures for motor control have been studied, as in [6], [62] and [63]. Taking into account the industrial specifications, the most effective scheme is nevertheless a hybrid neural controller [44], [64], [65], in which a neural network-based controller is usually used in conjunction with another controller to improve the performance of the system. In this work the aim of using a neural network was to increase the damping of the system, so that a load acceleration compensator was added to a conventional velocity PI motor controller in order to reduce mechanical vibration, which can be assumed to be a disturbance force that is added to a flexible load. The advantage of the proposed method is that it suppresses vibrations without degrading the overall velocity control performance. The structure of the hybrid controller is presented in Figure 4.10.

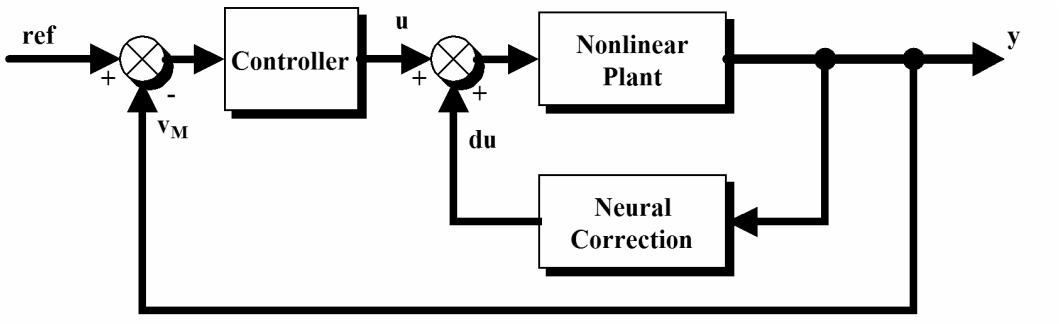


Figure 4.10. Structure of the proposed hybrid neural controller.

Several kinds of neural network structure with one or two hidden layers were tested, and the compensation signal was calculated by means of a multi-layer perceptron (MLP) network, which has three hidden layers; one input and one output layer. The inputs of the neural network are acceleration of the load, $\hat{\ddot{x}}_L(k)$, its delayed signal, $\hat{\ddot{x}}_L(k-1)$, and the output of the previous step $u_n(k-1)$.

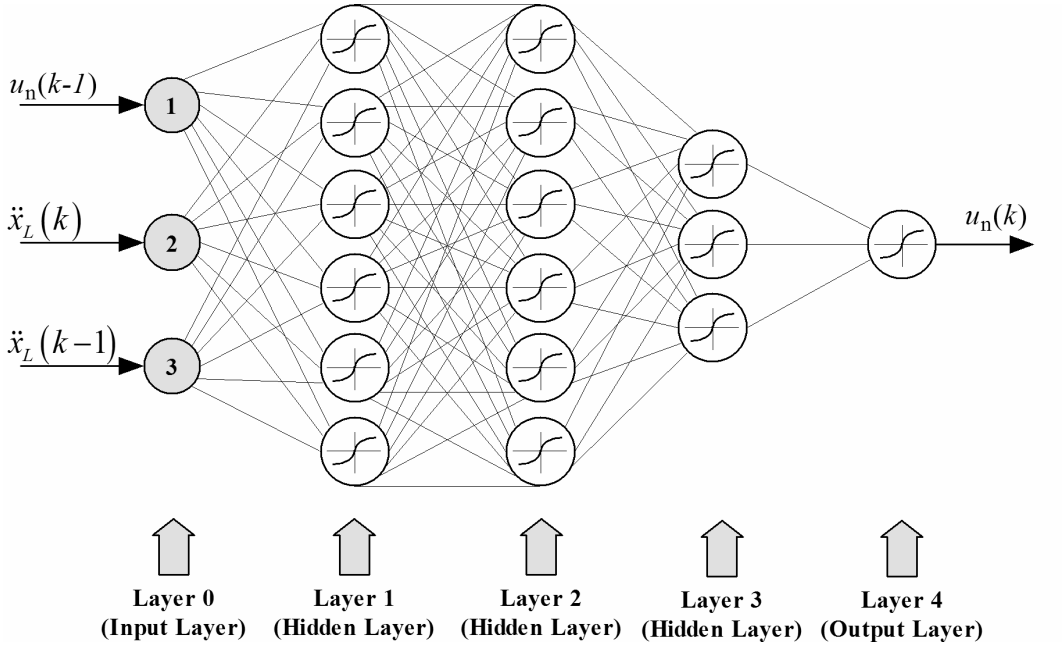


Figure 4.11. Structure of proposed neural network.

The output u_n of the proposed neural corrector is given by

$$u_n = \varphi_4\left(\mathbf{w}_4\varphi_3\left(\mathbf{w}_3\varphi_2\left(\mathbf{w}_2\varphi_1\left(\mathbf{w}_1\mathbf{x}+b_1\right)+b_2\right)+b_3\right)+b_4\right), \tag{4.34}$$

where φ_j is an activation function, $\mathbf{w}_j$ is a weighting matrix and b_j is a bias. The activation function used in the neural network is the hyperbolic tangent function, which has a monotonic increasing in the range –1 to 1. The mathematical model of the hyperbolic tangent can be written in the form

$$f(z)=\frac{e^z-e^{-z}}{e^z+e^{-z}}=\tanh(z). \tag{4.35}$$

In the physical system a Kalman filter was used to produce the estimated acceleration (Equation (4.43)) for the neural network. The estimated filter is noiseless and delayless relative to a filtered signal from an acceleration sensor. A block diagram of the experimental control setup is presented in Figure 4.12.

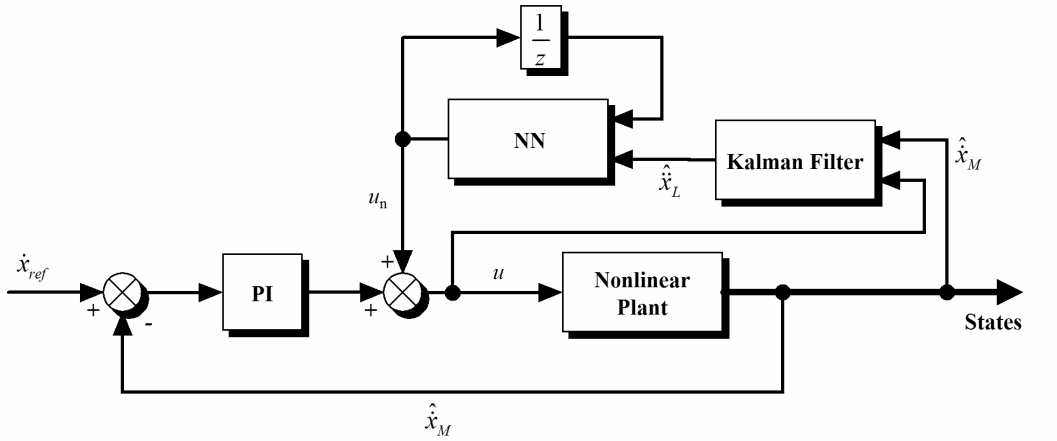


Figure 4.12. Block diagram of the neural network compensator.

A differential evolution algorithm was used to find the optimal weights for the nonlinear simulation model introduced in Chapter 2, after which the weights were transferred to the neural network in the physical system. The parameters selected for this strategy are listed in Table 5.

Table 5. Parameters of the differential evolution algorithm

Symbol	PARAMETER	Value
D	Number of parameters	91
NP	Size of Population	120
F	Scale Factor	0.8
CR	Crossover Control Constant	0.7

The cost of the system is considered in terms of the sum of absolute errors for one period (one second with a one millisecond sampling time), so that the overall cost is the sum of the absolute errors for one thousand points. The initial upper and lower bounds are [1,-1], respectively. Note that generally the size of the population (*NP*) is five times the number of parameters (*D*). Because of a memory saturation problem, the maximum *NP* used here was 120. The results show that *DE* can find the global minimum cost for the system, and depending on the expected value of the cost, it will then find the proper weights. The cost of the system is defined here as follows:

$$F(x_{i,G}) = \sum_{k=1}^{K=1000} |e(k)|, \tag{4.36}$$

where

$$\left|e(k)\right| = \left|\dot{x}_{ref}(k) - \dot{x}_L(k)\right|. \tag{4.37}$$

The minimum cost for the system was 5.6 and the related weights were used in the feed-forward neural network algorithm. A plot of the cost when the differential evolution algorithm is used to search for the global minimum weight is shown in Figure 4.13.

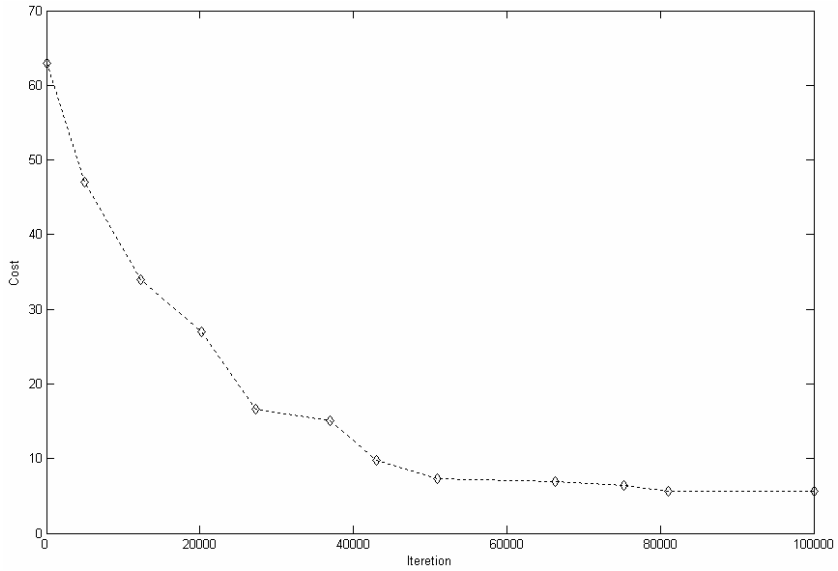


Figure 4.13. Cost of the system plotted against the number of iterations.

The weight matrices used in the neural network compensator are transposes of the matrices in Appendix A.4. The number of weights for the neural network is 91 (a bias is considered for all layers except the output layer). The DE algorithm finds the global minimum more reliably than the other methods. Note that algorithms of this kind are useful in offline training, as the approaching speed is not high, and can be used for preference to validate the optima attained. The velocity response of the load when a neural network compensator is used is shown in Figure 4.14. In this case, the reference was a step function of amplitude 0.1 m/s and frequency 1 Hz.

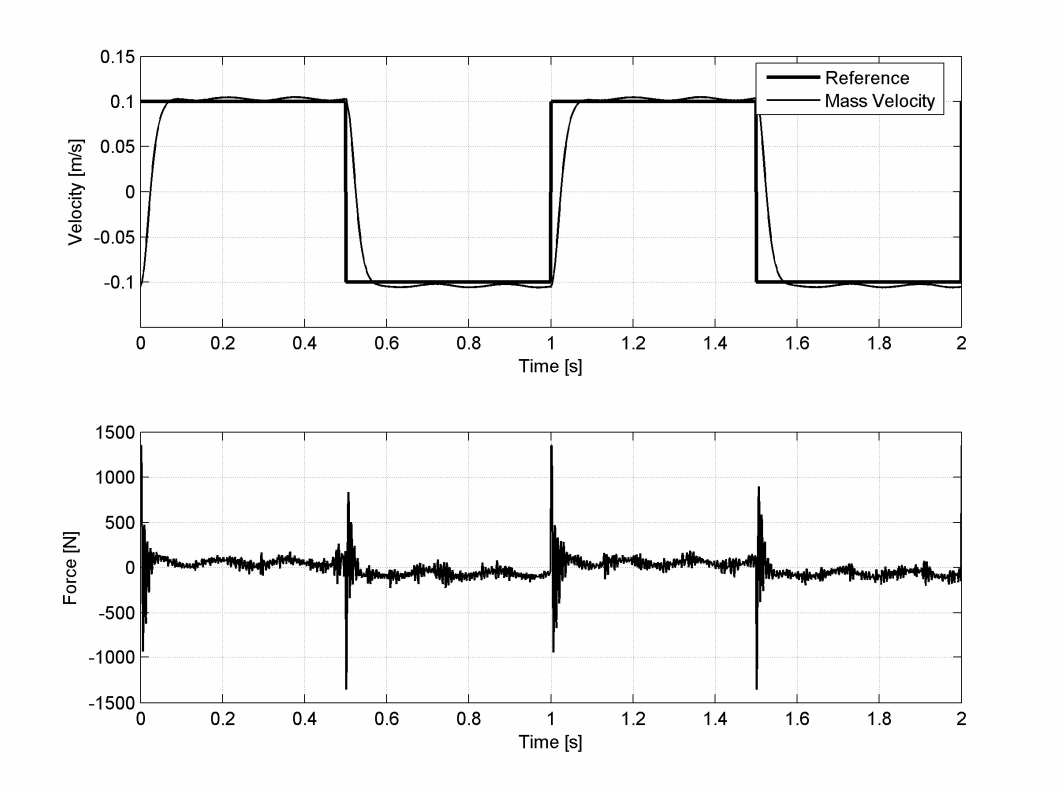


Figure 4.14. Measured velocity of the load and the driving force of the motor in the case of a neural network compensator under a step excitation.

4.4 Estimation of the System States

A linear state-space model of the mechanical system was derived for the Kalman filter introduced in the previous chapter. The equations of motion for the system in Figure 2.6 are

$$\begin{aligned} m_M \ddot{x}_M &= k\left(x_M - x_L\right) + b\left(\dot{x}_M - \dot{x}_L\right) + u \\ m_L \ddot{x}_L &= -k\left(x_M - x_L\right) - b\left(\dot{x}_M - \dot{x}_L\right) \end{aligned} . \tag{4.38}$$

The estimated states of the system are the velocity of the mover $\dot{x}_M$, velocity of the load $\dot{x}_L$ and spring force F_s, *i.e.* the state vector is

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \dot{x}_M \\ \dot{x}_L \\ F_s \end{bmatrix} . \tag{4.39}$$

The state matrix **A**, input matrix **B** and output matrix **C** in Equation (3.3) are

$$\mathbf{A}=\begin{bmatrix} -\frac{b}{m_M} & \frac{b}{m_M} & -\frac{1}{m_M} \\ \frac{b}{m_L} & -\frac{b}{m_L} & \frac{1}{m_L} \\ k & -k & 0 \end{bmatrix}, \mathbf{B}=\begin{bmatrix} \frac{1}{m_M} \\ 0 \\ 0 \end{bmatrix}, \tag{4.40}$$

$$\mathbf{C}=\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

where b is the damping constant and k the spring constant. The input to the Kalman filter is the velocity of the mover $\dot{x}_M$ and the control input u in this application is the motor thrust F_e. These matrices are discretized for the real-time Kalman filter using Equations (3.60) and (3.61). The process noise covariance **Q** in this application is

$$\mathbf{Q}=\begin{bmatrix} 100 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.41}$$

and the measurement covariance R=0.01 is scalar due to the existence of only one input to the Kalman filter, and was measured from the sensor signal. The Kalman gains K_1 and K_2, from Equation (3.65), are shown in Figure 4.15. Gain K_1 is related to the measured mover velocity and is one. The Kalman gains converge to a final value and their estimated states can be said to be stable.

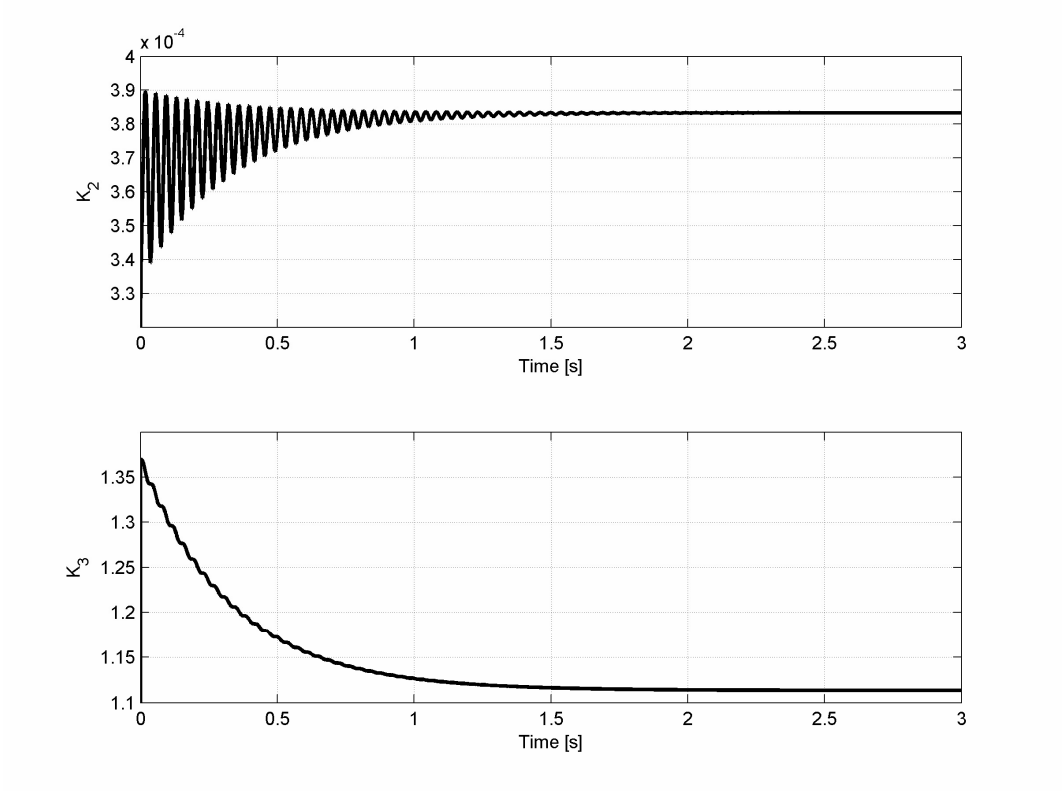


Figure 4.15. Convergence of the Kalman gains K_2 and K_3.

The position difference $\Delta x = x_M - x_L$ used in the backstepping controller is calculated from the estimated spring force $\hat{F}_s$, by dividing it by the spring constant k :

$$\Delta \hat{x} = \frac{\hat{F}_s}{k}. \tag{4.42}$$

Similarly the acceleration estimate $\hat{\ddot{x}}_L$ used in the neural network-based hybrid controller is measured from the estimated spring force $\hat{F}_s$, by dividing it by the load mass m_L, *i.e.* the acceleration estimate is

$$\hat{\ddot{x}}_L = \frac{\hat{F}_s}{m_L}. \tag{4.43}$$

Although the linear motor is highly nonlinear, the use of the Kalman filter for estimating the states of the system for nonlinear control is justified because we are estimating the states of a

mass-spring system which is assumed to behave linearly. The position of the nonlinear motor part was measured using an optical encoder with a resolution of 1 micrometre and its time derivative, *i.e.* velocity, was used to estimate the states of the mechanical system. The small nonlinear components in the mechanism (friction) are assumed to be system noise, which the Kalman filter handles as a random process. The estimated states are very close to the real ones, as seen in Figure 4.9.

4.5 Comparison of Results

The measured responses of the load from a physical linear motor when backstepping and hybrid neural controllers are used are presented in Figure 4.16. A step function of amplitude 0.1 m/s and frequency 1 Hz was used as a reference.

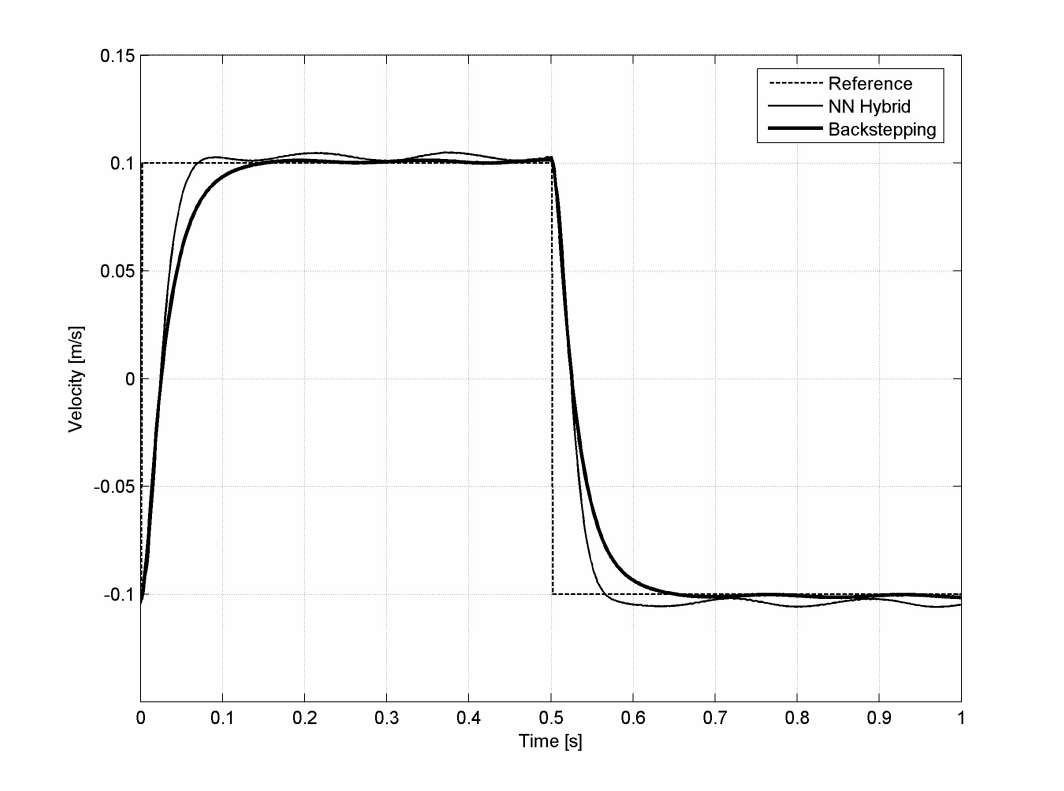


Figure 4.16. Comparison of load velocities in the physical linear motor application.

The hybrid neural controller has a slightly faster step response, but the ripple is larger than in the case of the backstepping controller. This means that the latter handles nonlinearities in the system better. On the other hand, the system with a backstepping controller converged to

equilibrium faster in an impulse test where the load had an initial displacement of 0.05 m. The phase plane plots of the impulse test are compared in Figure 4.17.

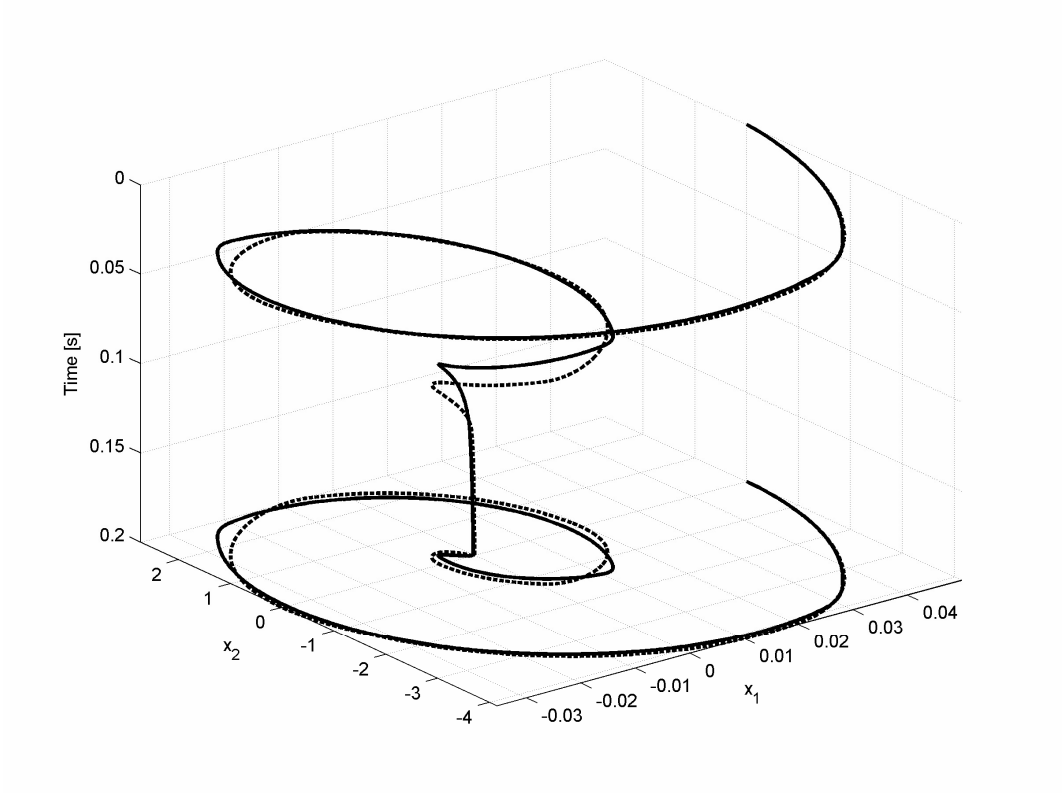


Figure 4.17. Comparison of phase plane plots of the backstepping controller (solid) and hybrid neural controller (dotted).

5 Conclusions

The objective of this thesis was two-fold and focused on (i) the development of an analytical nonlinear model for a permanent magnet linear synchronous motor (PMLSM), and (ii) a control synthesis for a mechanical system with flexibility. All the equations required to model and control the PMLSM with flexible mechanism were introduced, and the problematic nature of conventional linear controllers for mechanical flexibilities was considered and two nonlinear controllers proposed and compared. It was shown in the thesis that the adaptive backstepping controller is suitable for suppress mechanical vibrations in a linear motor application.

Mathematical modelling and simulations play an important role in control system engineering, where linear motors are usually represented as first-order models with nonlinearities. A more detailed model for a permanent magnet linear synchronous motor (PMLSM) was derived here, in which its dynamics were modelled using a space vector theory and a current controller was included in the model. Nonlinearities play an important role in system dynamics, the main nonlinearities being friction between a moving part and a stator and the detent force caused by interaction between the permanent magnets and the teeth in the armature. These nonlinearities were analysed on the basis of a laboratory setup and mathematical models were derived. It was shown that the derived simulation model enabled the system behaviour to be analysed accurately and the model could be used successfully for control system development purposes.

The control objective was to develop a velocity controller for use with a flexible mechanism in a linear motor application. The mechanism was modelled using a lumped mass model, *i.e*, the two-mass spring model, in which the dominant nominal frequency was taken into consideration. A reduction model of this kind allows a sufficiently accurate low-order model to be extracted from a physical mechanism, a valid generalization in control system engineering. The parameters for the mechanism model can be analysed on the basis of a physical application using the transient and/or frequency response. The transient method is presented in the thesis due to low natural frequency of the mechanism, which means clear and readable transient response. A complementary method would be to analyse the system in the

frequency domain and to use FFT or a Bode diagram to analyse the frequencies. These methods are useful for high-frequency applications.

The adaptive backstepping load control method for a PMLSM introduced here was successfully implemented in the physical linear motor application. The stability of the controller was ensured in terms of the Lyapunov stability theorem. The backstepping controller handled the nonlinearities involved in the motor part and controlled the load accurately. In the proposed adaptive control method the uncertain plant parameters, *i.e.* the mass of the load and the spring constant of the system, which can vary in the middle of the working cycle, were estimated online using the adaptive backstepping approach. This adaptive backstepping achieved global stabilization in the presence of unknown parameters. The vibration of the load was considerably reduced and the proposed controller was perceived to be asymptotically stable under all conditions. The backstepping controller proved to have faster convergence and better disturbance rejection than the reference controller.

The derived backstepping controller was compared with a hybrid neural controller in which the motor velocity was controlled by a conventional PI controller and vibration in the load was suppressed using the neural network compensator. The proposed structure was able to emulate the acceleration feedback and improve the dynamic behaviour of the system. The Differential Evolution algorithm proved to be an efficient tool for finding initial weights for the feed-forward neural network. More traditional teaching methods might have encountered a local minimum problem due to their gradient-based mode of calculation, *i.e.* they frequently lead to local optima. DE is not based on the gradient method, and therefore the local minimum problem was avoided. The problem with DE is its speed in approaching the global minimum, which means that it is useful only in offline training. The combination of a neural network hybrid controller and a state estimator reduced the vibration in the load considerably. The problem with neural controllers lies in their stability properties. If the input signal for the neural network includes combinations which are not used in the teaching process, the network output may become unstable. Furthermore, with most neural control schemes the choices of neural network structure, initial weights and training speed are often non-systematic.

The states of the mechanical system were estimated using the Kalman Filter, due to difficulties in measuring the physical states of the flexible mechanism. In the case of nonlinear systems it is not possible to combine a linear observer with nonlinear control, but in this case the mechanism acted as a linear system and the Kalman filter could be used to estimate the exponentially converging states. The proposed observer is valid for contactless applications, because it is not of an adaptive type. This means that the estimator is unable to adapt system parameters based on the velocity of the motor if the spring constant or mass of the load changes during a run. This problem can naturally be avoided if it is possible to measure the state of the flexible mechanism and use it directly as feedback or estimate other states of the mechanism for feedback purposes using this measured signal.

Adaptive nonlinear controllers are still very difficult to implement for solving industrial problems because no general methods have been introduced. Of course it is impossible to build a theory for nonlinear control that works for everything, but the method introduced here is valid for many mechanical systems. The adaptive backstepping method seems to be a potential and effective candidate for industrial use, because it solves many of the problems which have arisen in previously developed linear and nonlinear controllers, e.g. the lack of a constructive design method for nonlinear systems and the need for global robust stabilization.

There are few issues related to the backstepping method that needs further investigation. A critical one is tuning of control parameters. In the thesis control parameters and adaptation gains were experimentally tuned, but it is a challenging task to tune control parameters analytically. This is usually done by reformulating a nonlinear control system to linear time-invariant system. Another future work is to make tests with a more complex tool mechanism and verify backstepping controller's functionality in that kind of a electromechanical system. The issue which needs also future investigation but is not directy related to backstepping controller is development of an adaptive state estimator, which is suitable for wider range of applications, not only for contactless applications.

References

[1] Maloney, L. D., 2005, "Linear Motors Hit Their Stride", *Design News (Reed Business)*, **10**.

[2] Paula, G., 1998, "Linear Motors Take Center Stage", *Mechanical Engineering*, **March**.

[3] Cetinkunt, S., and Wayne, J. B., 1990, "Performance Limitations of Joint Variable-Feedback Controllers Due to Manipulator Structural Flexibility", *IEEE Trans. on Robotics and Automation*, **6**(2), pp. 219-231.

[4] Wai, R.-J., and Lee, M.-C., 2004, "Intelligent Optimal Control of Single-Link Flexible Robot Arm", *IEEE Trans. on Ind. Electronics*, **51**(1), pp. 201-220.

[5] Yue, S. G., Herich, D., Xu, W. L., and Tso, S. K., 2001, "Trajectory Planning in Joint Space for Flexible Robots with Kinematic Redundancy", *Proceedings of the IASTED International Conference of Modeling, Identification, and Control*.

[6] Otten, G., de Vries, T. J. A., van Amorengen, J., Rankers, A. M., and Gaal, E. W., 1997, "Linear Motor Motion Control Using a Learning Feedforward Controller", *IEEE/ASME Trans. Mechatronics*, **2**(3), pp. 179-187.

[7] Dumetz, E., Vanden Hende, F., and Barre, P. J., 2001, "Resonant Load Control Method Application to High-Speed Machine Tool with Linear Motor", *Proceedings of Emerging Technologies and Factory Automation*, **2**, pp. 23-31.

[8] Ellis, G., and Lorenz, R. D., 2000, "Resonant Load Control Methods for Industrial Servo Drives", *IEEE Industry Application Society Annual Meeting*, **3**, pp. 1438-1445.

[9] Jun-Keun, J., and Seung-Ki, S., 1995, "Kalman Filter and LQ Based Speed Controller for Torsional Vibration Suppression in a 2-mass Motor Drive System", *IEEE Trans. on Industrial Electronics*, **42**(6), pp. 564-571.

[10] Kang, J-K., and Sul, S.-K., 2000, "Vertical-Vibration Control of Elevator Using Estimated Car Acceleration Feedback Compensation", *IEEE Trans. on Industrial Electronics*, **47**(1), pp. 91-99.

[11] Lee, Y.-M., Kang, J.-K., and Sul, S.-K., 1999, "Acceleration Feedback Control Strategy for Improving Riding Quality of Elevator System", *Proceedings of the IAS*, **2**, pp. 1375-1379.

[12] Montanaro, J., and Beale, G., 1998, "Feedback Control for Canceling Mechanical Vibrations", *Proceedings of the IECON*, **3**, pp. 1433-1438.

[13] Slotine, J.-J., and Li, W., 1991, *Applied Nonlinear Control*, Prentice-Hall, New Jersey.

[14] Olivier, P. D., 1991, "Feedback Linearization of DC motors", *IEEE Trans. on Industrial Electronics*, **38**(6), pp. 498-501.

[15] Wai, R.-J., Duan, R.-Y., and Chang, L.-J., 2001, "Grey Feedback Linearization Speed Control for Induction Servo Motor Drive", *Proceedings of the IECON*, **1**, pp. 580-585.

[16] Furuhashi, T., Sangwongwanich, S., and Okuma, S. A., 1992, "Position-and-Velocity Sensorless Control for Brushless DC Motor Using an Adaptive Sliding Mode Observer", *IEEE Trans. on Industrial Electronics*, **39**(2), pp. 89-95.

[17] Lin, F.-J., Chiu, S.-L., and Shyu, K.-K., 1998, "Novel Sliding Mode Controller for Synchronous Motor Drive", *IEEE Trans. on Aerospace and Electronic Systems*, **34**(2), pp. 532-542.

[18] Hongjie, H., Jingquan, C., and Lianjie, E., 2001, "Neural Adaptive Tacking Control of a Low Speed DC Servo System", *Proceedings of the Fifth International Conference on Fluid Power*.

[19] *IEEE Xplore release 2.1*. Retrieved April 4, 2006, from http://ieeexplore.ieee.org/.

[20] Kokotovic, P., and Arcak, M., 2001, "Constructive Nonlinear Control: A Historical Perspective", *Automatica*, **37**, pp. 637-662.

[21] Wai, R.-J., Lin, F.-J., and Hsu, S.-P., 2001, "Intelligent Backstepping Control for Linear Induction Motor Drive", *IEEE Trans. on Proc.-Control Theory Appl.*, **148**(3), pp. 193-202.

[22] Bose, B. K., 1986, *Power Electronics and AC Drives*, Prentice-Hall, New Jersey.

[23] Gieras, J. F., and Piech, Z. J., 2001, Linear synchronous motors: Transportation and automation systems, CRC, Boca Raton.

[24] Novotny, D. W., and Lipo, T. A., 1996, *Vector Control and Dynamics of AC Drives,* Oxford Press, New York.

[25] Hyun, D.-S., Jung, I.-S., Shim, J.-H., and Yoon, S.-B., 1999, "Analysis of Forces in a Short Primary Type Permanent Magnet Linear Synchronous Motor", *IEEE Trans. Energy Conversion*, **14**(4), pp. 1265-1270.

[26] Inoue, M., and Sato, K., 2000, "An Approach to a Suitable Stator Length for Minimizing the Detent Force of Permanent Magnet Linear Synchronous Motors", *IEEE Trans. Magn.*, **36**(4), pp. 1890-1893.

[27] Zhu, Z. Q., Xia, Z. P., Howe, D., and Mellor, P. H., 1997, "Reduction of Cogging Force in Slotless Linear Permanent Magnet Motors", *Proceedings of IEEE Electr. Power Appl*., **144**(4), pp. 277-282.

[28] Hor, P. J., Zhu, Z. Q., Howe, D., and Rees-Jones, J., 1998, "Minimization of Cogging Force in a Linear Permanent Magnet Motor", *IEEE Trans. Magn.*, **34**(5), pp. 3544-3547.

[29] Jung, S.-Y., and Jung, H.-K., 2002, "Reduction of Force Ripples in Permanent Magnet Linear Synchronous Motor", *Proceedings of International Conference on Electrical Machines*, pp.452-458.

[30] Chun, J.-S., Jung, H.-K., and Jung, J.-S., 2000, "Analysis of the End-Effect of Permanent Magnet Linear Synchronous Motor Energized by Partially Exited Primary", *Proceedings of International Conference on Electrical Machines,* **1**, pp. 333-337.

[31] Tan, K. K., Huang S. N., and Lee, T. H., 2002, "Robust Adaptive Numerical Compensation for Friction and Force Ripple in Permanent-Magnet Linear Motors", *IEEE Trans. Magn.*, **38**(1), pp. 221-228.

[32] Olsson, H., Åström, K. J., de Wit, C. C., Gäfvert M., and Lichinsky, P., 1998, "Friction Models and Friction Compensation," *European Journal of Control,* **4**(3), pp. 176-195.

[33] Hensen, R. H. A., 2002, *Controlled Mechanical Systems with Friction*, Doctoral Dissertation, Eindhoven University of Technology, Holland.

[34] Makkar, C., Dixon, W. E., Sawyer, W. G., and Hu, G., 2005, "A New Continuously Differentiable Friction Model for Control System Design," *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 600-605.

[35] Wie, B., and Bernstein, D. S., 1990, "A Benchmark Problem for Robust Control Design", *Proceedings of the American Control Conference*, pp. 961-962.

[36] Shabana, A., A., 1996, *Theory of Vibration*, Second Edition, Springer, New York.

[37] Khalil, H. K., 2000, *Nonlinear Systems*, Prentice-Hall, New Jersey.

[38] Zhang, G., and Furusho, J., 2000, "Speed Control of Two-Inertia System by PI/PID Control", *IEEE Trans. on Ind. Electronics*, **47**(3), pp. 603-609.

[39] Åström, K., and Wittenmark, B., 1995, *Adaptive Control*, Addison-Wesley, USA.

[40] Dawson, D., Carroll, J., and Schneider, M., 1994, "Integrator Backstepping Control of a Brush DC Motor Turning a Robot Load", *IEEE Trans. on Control System Technology*, **2**(3), pp. 233-244.

[41] Lin, J.-S., and Kanellakopoulos, I., 1997, "Nonlinear Design of Active Suspensions", *IEEE Control Systems Magazine*, **17**(3), pp. 45-59.

[42] Krstic, M., Kanellakopoulos, I., and Kokotovic, P., 1995, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, New York.

[43] Härkegård, O., 2001, *Flight Control Design Using Backstepping*, Licentiate Thesis, Linköping University, Sweden.

[44] Madani, K., 2006, "Real-World Industrial Applications of Artificial Neural Networks – Illusion or reality?", *Informatics in Control, Automation, and Robotics,* **1**, pp. 11-26.

[45] Rosenblatt, F., 1958, "The Percepton: A Probabilistic Model for Information Storage and Organization in the Brain", *Psychological Review*, **65**, pp. 386-408.

[46] Rosenblatt, F., 1962, *Principles of Neurodynamics*, Spartan, New York.

[47] Haykin, S., 1994, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, USA. p. 696.

[48] Bishop, R. H., 2002, *The Mechatronics Handbook*, CRC Press, USA.

[49] Ilonen, J., Kamarainen, J.-K., and Lampinen, J., 2003, "Differential Evolution Training Algorithm for Feed-Forward Neural Networks", *Neural Processing Letters,* **7**(1), pp. 93-105.

[50] Cant-Paz, E., and Kamath, C., 2005, "An Empirical Comparison of Combinations of Evolutionary Algorithms and Neural Networks for Classification Problems", IEEE Transactions on Systems, Man, and Cybernetics, **B**, pp. 915-927.

[51] Corne, D., *et al.*, 1999, *New Ideas in Optimization*, McGraw-Hill, UK.

[52] Price, K. V., Storn, R. M., and Lampinen, J. A., 2005, *Differential Evolution – A Practical Approach to Global Optimization*, Springer, Heidelberg, Germany.

[53] Franklin, G. F., *et al.*, 1986, *Feedback Control of Dynamic Systems*, Addison-Wesley, USA.

[54] Robertsson, A, 1999, *On Observer-Based Control of Nonlinear Systems*, Doctoral Thesis, Lund Institute of Technology, Sweden.

[55] Ogata, K., 1997, *Modern Control Engineering*, Prentice-Hall, London.

[56] Luenberger, D. G., 1971, "An Introduction to Observers", *IEEE Trans. on Automatic Control*, **16**(6), pp. 596-602.

[57] Kalman, R. E. A., 1960, "New Approach to Linear Filtering and Prediction Problems", *Transaction of the ASME – Journal of Basic Engineering*, pp. 35-45.

[58] Franklin, G. F., 1998, *Digital Control of Dynamic System*, Addison-Wesley, Menlo Park.

[59] Leleux, D. P., *et al.*, 2001, "Application of Kalman Filtering to Real-Time Trace Gas Concentration Measurements", *Lasers and Optics*, **June**, pp. 85-93.

[60] Yu, J.-T., Hu, J., and Chang, J., 2002, "The Analysis of Gain Selection in Adaptive Backstepping Motion Control", *Proceedings of the Power Conversion Conference*, **2**, pp. 902-906.

[61] Schytte, F., Beineke, S., Rolfsmeier, A., and Grotstollen, H., 1997, "Online Identification of Mechanical Parameters Using Extended Kalman Filter", *Proceedings of the IEEE Industry Applications Society Annual Meeting*, pp. 501-508.

[62] Low, T.-S., and Lee, T.-H., 1993, "A Methodology for Neural Network Training for Control of Drives with Nonlinearities", *IEEE Trans. on Ind. El.*, **39**(2), pp. 243-249.

[63] Lin, C.-M., and Hsu, C.-F., 2002, "Neural-Network-Based Adaptive Control for Induction Servomotor Drive System", *IEEE Trans. on Ind. El.*, **49**(1), pp. 115-123.

[64] Low, T.-S., Lee T.-H. and Lim, H.-K., 2001, "Hybrid Control Using Recurrent Fuzzy Neural Network for Linear-Induction Motor Servo Drive", *IEEE Trans. on Fuzzy Systems*, **9**(1), pp. 102-115.

[65] Lin, C.-H., Chou, W.-D., and Lin, F.-J., 2001, "Adaptive Hybrid Control Using a Recurrent Neural Network for a Linear Synchronous Motor Servo-Drive System", *IEEE Proc.-Control Theory Appl.*, **148**(2), pp. 156-168.

Appendices

Appendix A.1

Simulation Parameters

Symbol	Parameter	Value
α_{init}	Electrical angle between stator and rotor	0
ω_n	Electrical angular velocity	984.37 °/s
ψ_{PM}	Magnetic flux	0.3 Wb
τ_M	Pole pitch (180° electrical)	15 mm
f_N	Nominal motor frequency	50 Hz
F_{max}	Maximum thrust	1650 N
F_n	Nominal thrust	675 N
i_{max}	Maximum current	19.1 A
i_n	Nominal current	7.2 A
K_t	Motor constant	94
L_{ad}	d-axis armature inductance	30 mH
L_{aq}	q-axis armature inductance	40 mH
L_{md}	d-axis magnetizing inductance	20 mH
L_{mq}	q-axis magnetizing inductance	20 mH
L_D	d-axis damper winding inductance	30 mH
L_Q	q-axis damper winding inductance	40 mH
p	Pole pair number	5
R	Winding resistance	4.8 Ω
R_D	d-axis damper winding resistance	2.4 Ω
R_Q	q-axis damper winding resistance	2.4 Ω
u_s	Inverter bus voltage	720 V
u_n	Motor voltage	400 V

Appendix A.2

Controllability and Observability of the System

Consider the system described by [55]

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{A.1}$$

Where $\mathbf{x}$ = state vector

$\mathbf{u}$ = control vector

$\mathbf{y}$ = output vector

$\mathbf{A}$ = n x n matrix

$\mathbf{B}$ = n x r matrix

$\mathbf{C}$ = m x n matrix.

Controllability

The system (A.1) is said to be controllable if the rank of the n x nr controllability matrix

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \cdots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} \tag{A.2}$$

is n, *i.e.* $rank(\mathcal{C}) = n$.

Observability

The system (A.1) is said to be observable if the rank of the n x nm observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{A.3}$$

is n, *i.e.* $rank(\mathcal{O}) = n$.

Appendix A.3

Parameters given by a motor and inverter manufacturer

Symbol	Parameter	Value
F_N	Nominal force	675 N
A_N	Nominal current	7.2 A
K_m	Motor constant	94 N/A
R_P	Electric resistance	4.8 Ω
L_P	Winding inductance	20 mH
V_n	Max. speed at nominal thrust	2.1 m/s
τ_M	Pole pitch (180° electrical)	15 mm
P_N	Nominal power	3910 W
L	Stroke	1000 mm
Kp	P gain of velocity controller	10000 Ns/m
Ki	I gain of velocity controller	0.01 s
Kp	P gain of current controller	50.3 V/A
Ki	I gain of current controller	0.002 s
f_{mod}	Modulation frequency (PWM)	4000 Hz
V	Inverter bus voltage	720 V

Appendix A.4

Values of the weight matrices

The matrices used in the controller are transposes of the following.

Weight matrix W_1 for layer 1			
-0.429806	-0.490136	0.261634	-1.336779
0.363570	0.959710	0.763704	0.264503
-1.707644	-0.395742	-0.450741	1.186434
-0.786033	0.987219	0.566839	0.654641
0.176158	-0.326978	1.555741	0.709490
-2.631994	-0.551799	0.350849	1.1258239

Weight matrix W_2 for layer 2						
-0.078877	0.401151	-0.372676	0.288755	-0.470446	-0.924121	-0.886906
-0.338182	-0.839938	1.481111	1.661285	-0.147407	0.836702	-0.084553
1.029182	-0.450955	0.772902	-0.133162	0.400379	0.791054	1.303503
-1.619169	-0.198272	0.936287	0.358100	-0.305366	-1.857181	1.063913
-0.229639	-0.382553	0.210725	-0.587176	0.400449	0.175861	1.311136
0.429760	0.414340	0.168668	0.850356	-0.118835	-0.509728	-0.074963

Weight matrix W_3 for layer 3						
0.634231	0.967565	0.035364	-1.248977	-0.106021	-0.943593	-0.240842
0.933318	0.323732	0.233129	0.303268	0.620870	1.626560	-0.790759
1.403531	-0.069303	-0.574637	0.989509	0.069376	-0.151322	0.785669

Weight matrix W_4 for the output layer			
-0.898459	0.781132	-0.048206	-0.249697

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

209. FORSMAN, HELENA. Business development efforts and performance improvements in SMEs. Case study of business development projects implemented in SMEs. 2005. 209 s. Diss.

210. KOSONEN, LEENA. Vaarinpidosta virtuaaliaikaan. Sata vuotta suomalaista tilintarkastusta. 2005. 275 s. Diss.

211. 3rd Workshop on Applications of Wireless Communications. 2005. 62 s.

212. BERGMAN, JUKKA-PEKKA. Supporting knowledge creation and sharing in the early phases of the strategic innovation process. 2005. 180 s. Diss.

213. LAAKSONEN, PETTERI. Managing strategic change: new business models applying wireless technology as a source of competitive edge. 2005. 142 s. Diss.

214. OVASKA, PÄIVI. Studies on coordination of systems development process. 2005. U.s. Diss.

215. YANG, GUANGYU. Control and simulation of batch crystallization. 2005. U.s. Diss.

216. MUSTONEN–OLLILA, ERJA. Information system process innovation adoption, adaptation, learning, and unlearning: a longitudinal case study. 2005. U.s. Diss.

217. SAINIO, LIISA-MAIJA. The effects of potentially disruptive technology on business model – a case study of new technologies in ICT industry. 2005. 205 s. Diss.

218. SAINIO, TUOMO. Ion-exchange resins as stationary phase in reactive chromatography. 2005. 175 s. Diss.

219. CONN, STEFFEN. Software tools to support managers: development and application to new product development and strategic alliance management. 2005. 168 s. Diss.

220. TYNJÄLÄ, TERO. Theoretical and numerical study of thermomagnetic convection in magnetic fluids. 2005. U.s. Diss.

221. JANTUNEN, ARI. Dynamic capabilities and firm performance. 2005. 210 s. Diss.

222. KOLA-NYSTRÖM, SARI M. In search of corporate renewal: how to benefit from corporate venturing? 2005. 190 s. Diss.

223. SARÉN, HANNU. Analysis of the voltage source inverter with small DC-link capacitor. 2005. 143 s. Diss.

224. HUUHILO, TIINA. Fouling, prevention of fouling, and cleaning in filtration. 2005. U.s. Diss.

225. VILJAINEN, SATU. Regulation design in the electricity distribution sector – theory and practice. 2005. 132 s. Diss.

226. AVRAMENKO, YURY. Case-based design method for chemical product and process development. 2005. U.s. Diss.

227. JÄRVINEN, KIMMO. Development of filter media treatments for liquid filtration. 2005. U.s. Diss.

228. HURMELINNA-LAUKKANEN, PIA. Dynamics of appropriability – finding a balance between efficiency and strength in the appropriability regime. 2005. U.s. Diss.

229. LAARI, ARTO. Gas-liquid mass transfer in bubbly flow: Estimation of mass transfer, bubble size and reactor performance in various applications. 2005. U.s. Diss.

230. BORDBAR, MOHAMMAD HADI. Theoretical analysis and simulations of vertically vibrated granular materials. 2005. U.s. Diss.

231. LUUKKA, PASI. Similarity measure based classification. 2005. 129 s. Diss.

232. JUUTILAINEN, ANNELI. Pienen matkailuyrityksen yrittäjän taival. Oppiminen yrittäjyysprosessissa. 2005. 191 s. Diss.

233. BJÖRK, TIMO. Ductility and ultimate strength of cold-formed rectangular hollow section joints at subzero temperatures. 2005. 163 s. Diss.

234. BELYAEV, SERGEY. Knowledge discovery for product design. 2005. U.s. Diss.

235. LEINONEN, KARI. Fabrication and characterization of silicon position sensitive particle detectors. 2006. U.s. Diss.

236. DUFVA, KARI. Development of finite elements for large deformation analysis of multibody systems. 2006. U.s. Diss.

237. RITVANEN, JOUNI. Experimental insights into deformation dynamics and intermittency in rapid granular shear flows. 2006. U.s. Diss.

238. KERKKÄNEN, KIMMO. Dynamic analysis of belt-drives using the absolute nodal coordinate formulation. 2006. 121 s. Diss.

239. ELFVENGREN, KALLE. Group support system for managing the front end of innovation: case applications in business-to-business enterprises. 2006. 196 s. Diss.

240. IKONEN, LEENA. Distance transforms on gray-level surfaces. 2006. 132 s. Diss.

241. TENHUNEN, JARKKO. Johdon laskentatoimi kärkiyritysverkostoissa. Soveltamismahdollisuudet ja yritysten tarpeet. 2006. 270 s. Diss.

242. KEMPPINEN, JUKKA. Digitaaliongelma. Kirjoitus oikeudesta ja ympäristöstä. 2006. 492 s.

243. PÖLLÄNEN, KATI. Monitoring of crystallization processes by using infrared spectroscopy and multivariate methods. 2006. U.s. Diss.

244. AARNIO, TEIJA. Challenges in packaging waste management: A case study in the fast food industry. 2006. 260 s. Diss.

245. PANAPANAAN, VIRGILIO M. Exploration of the social dimension of corporate responsibility in a welfare state. 2006. 239 s. Diss.

246. HEINOLA, JANNE-MATTI. Relative permittivity and loss tangent measurements of PWB materials using ring resonator structures. 2006. U.s. Diss.

247. SALMELA, NINA. Washing and dewatering of different starches in pressure filters. 2006. U.s. Diss.

248. SISSONEN, HELI. Information sharing in R&D collaboration – context-dependency and means of governance. 2006. 246 s. Diss.

249. PURANEN, JUSSI. Induction motor versus permanent magnet synchronous motor in motion control applications: a comparative study. 2006. 147 s. Diss.

250. PERÄLÄ, KARI. Kassanhallintakäytännöt Suomen kunnissa. 2006. 213 s. Diss.

252. LIHAVAINEN, VELI-MATTI. A novel approach for assessing the fatigue strength of ultrasonic impact treated welded structures. 2006. 87 s., liitt. Diss.

253. TANG, JIN. Computational analysis and optimization of real gas flow in small centrifugal compressors. 2006. 93 s. Diss.