CHALMERS

Linear Model Predictive Control of a 3D Tower Crane for Educational Use

Double Master's Thesis

PATRICK BREUNING

Supervisors:
Prof. Bo Egardt, Chalmers University of Technology
Dipl.-Ing. Daniel Coupek, University of Stuttgart

Examiners:
Prof. Bo Egardt, Chalmers University of Technology
Prof. Dr.-Ing. Peter Klemm, University of Stuttgart

Department of Signals & Systems
Chalmers University of Technology

Institute for Control Engineering of Machine Tools and Manufacturing Units
University of Stuttgart

Gothenburg, Sweden 2015

Elaboration of a laboratory guidance for linear Model Predictive Control of a 3D Tower Crane

This thesis was conducted within the Double Master's Programme in Systems, Control and Mechatronics at Chalmers University of Technology, Gothenburg, Sweden and Mechatronics at University of Stuttgart, Germany. The responsible departments are the Department of Signals and Systems at Chalmers University of Technology and the Institute for Control Engineering of Machine Tools and Manufacturing Units at University of Stuttgart.

Master's Thesis 2015:01/02

Department of Signal and System
Chalmers University of Technology
SE-41296 Göteborg
Sweden
Tel. +46-(0)31 772 1000

Institute for Control Engineering of Machine Tools and Manufacturing Units
University of Stuttgart
DE-70569 Stuttgart
Germany
Tel. +49 (0)711 / 685-82410

Abstract

Since numerical computation has become very efficient, Model Predictive Control (MPC) has become more relevant for control tasks, apart from process control, over the last years. This is the reason why more universities are offering MPC courses to their students these days.

The purpose of this thesis was to implement an linear quadratic MPC to an educational control system in order to use it in an MPC student laboratory. Therefore a tower crane system was selected. A highly nonlinear mathematical model was derived by using the Lagrangian approach as well as physical properties of the actuators. In order to use linear methods, the model was linearized and decoupled into three subsystems. Special attention was paid to discuss advantages and disadvantages of different Receding Horizon Control (RHC) structures as well as comparing computation times of different quadratic program solvers. Using a steady state target selector, the RHC works with deviation variables. Zero Offset control was achieved by introducing an augmented disturbance model to the steady state target selector. Not measurable states as well as the disturbance are estimated by a time varying Kalman filter. Finally, real time control was successfully implemented by using the code generation system FORCES Pro.

As a result of this work a laboratory guidance was obtained. This guidance will be used for student assignments in the MPC course (SSY280) at the Chalmers University of Technology.

Acknowledgements

I would like to thank professor Bo Egardt, for giving me the possibility to do my master's thesis with him as supervisor and examiner. Academical discussions with him have always opened new points of view to me.

Furthermore I would like to thank my supervisor Daniel Coupek and examiner professor Peter Klemm for the support from Stuttgart.

Discussions with Faisal Altaf and Sandor lies, who both have worked with the tower crane in there studies, have helped me a lot. Thank you for that.

Last but not least I would like to thank the founding team of the embotech GmbH for giving me the opportunity to use the code generation system FORCES Pro for free.

Gothenburg, 27/5/15

Contents

List of Figures

List of Tables

Notation

Mechanical Model Parameters

X_w	Position of the trolley on the arm
L	Length of the playload
α	Swing angle of the load in the XZ-plane
β	Swing angle of the load in the YZ-plane
θ	Angle of the arm
$\vec{p_l}$	Vector of the position of the playload
$\vec{p_t}$	Vector of the position of the trolley
$\vec{\omega_T}$	Angular velocity of the tower
m	Mass of the playload
M	Mass of the trolley
J_o	Moment of inertia of the tower and the jib about the z-axis
F_x	Generalized force in X-direction
F_θ	Generalized force in θ-direction
B_x	Viscous friction coefficient in X-direction
B_α	Viscous friction coefficient in θ-direction
B_θ	Viscous friction coefficient in α-direction
B_β	Viscous friction coefficient in β-direction
τ_{dx}	System delay in x-direction
$\tau_{d\theta}$	System delay in θ-direction
τ_{dC}	System delay in Z-direction

Electrical Model Parameters

V	Armature voltage
R_a	Armature resistance
i_a	Armature current
L_a	Armature Inductance
ω	Angular velocity of the motor shaft
T_d	Motor torque
T_e	External part of motor torque
k_m	Torque constant
k_u	Counter-electromotive force constant
k_g	Gear ratio
B	Motor viscous damping coefficient
η_x	Trolley motor gearbox and motor efficiency
η_θ	Arm motor gearbox and motor efficiency
r_x	Radius of pulley
G_{ax}	Amplifier gain x-direction
$G_{a\theta}$	Amplifier gain θ-direction
G_{ac}	Amplifier gain Z-direction
U_x	PWM signal X-direction
U_θ	PWM signal θ-direction
U_c	PWM signal Z-direction

State Space Model Parameters

x	State vector
u	Input vector
y	Output vector
A	State matrix
B	Input matrix
C	Output matrix

MPC Parameters - Part 1

N	Prediction horizon
M	Control horizon
n	Number of system states
$\mathbb{R}^{m\times n}$	The class of m by n real matrices
m	Number of system inputs
p	Number of system outputs
Q	State deviation penalization matrix
Q_f	State deviation penalization matrix of final states
R	Input magnitude penalization matrix
V_N	Cost function over horizon N
$\mathcal{X}$	State constraint set
$\mathcal{X}_f$	Terminal state constraint set
$\mathcal{U}$	Control constraint set
H	Matrix in quadratic programming problem
f	Vector in quadratic programming problem
A_{eq}	Equality constraints matrix
b_{eq}	Equality constraints vector
A_{ineq}	Inequality constraints matrix
b_{ineq}	Inequality constraints vector
z	Vector of optimization variables
$\bar{Q}$	diag$\{Q....Q,Q_f\}$
$\bar{R}$	diag$\{R....R\}$
X	Vector of future state variables
U	Vector of future control variables
ψ	Vector in the batched representation of the system dynamics
Φ	Matrix in the batched representation of the system dynamics

x_s	Steady state
u_s	Steady state control signal
y_sp	Steady state set points
p_z	Number of controlled system outputs
C_z	Reduced output matrix
z_sp	Reduced steady state set points

MPC Parameters - Part 2

d	Disturbance in augmented disturbance model
B_d	Vector in the disturbance augmented model (states)
C_d	Vector in the disturbance augmented model (outputs)
n_d	Dimension of disturbance d
G_k	Input disturbance vector
$w(k)$	Input disturbance
$v(k)$	Output disturbance
L	Kalman filter innovation gain
P	State estimation error covariance
K	Process noise covariance
S	Sensor noise covariance
$\hat{x}$	State estimate
$\hat{d}$	Disturbance estimate

Acronyms

MPC	Model Predictive Control
PID	Proportional-Integral-Derivative
DMC	Dynamic Matrix Control
LQR	Linear-Quadratic Regulator
LQG	Linear-Quadratic-Gaussian Regulator
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
SIMO	Single Input Single Output
MIMO	Multiple Input Multiple Output
IP	Inverted Pendulum
PWM	Pulse-Width Modulation
RHC	Receding Horizon Control

1 Introduction

This chapter provides a general overview of the thesis i.e. what the thesis is about, which problems are solved and what the goal is. After some basic information of Model Predictive Control is given in section 1.1, problems to solve are stated in section 1.3. The purpose and objective of this thesis are discussed in 1.2 and 1.4, respectively. Some limitations of the work are given in section 1.5.

1.1 Background

Model Predictive Control (MPC), developed by engineers, is the only advanced control technique that is more advanced than standard PID-Control, which have had a significant impact to process control, [2]. The first official description of MPC applications was presented by Richalet in 1976 but engineers at Shell Oil developed their own independent MPC technology, named DMC, in the early 1970s, with an initial application in 1973 and an official presentation in 1979, [3]. There are several different theoretical foundations that underlay MPC like the Hamilton Jacobi Bellman theory (Dynamic Programming) and the further development of the LQR to the LQG by Kalman, [4]. Nevertheless, none of these foundations provides the possibility to solve control problems by considering system constraints, which is the main advantage of MPC. Furthermore it is possible to apply MPC to multivariable systems. Controlling process units near to their constraints increases throughput and efficiency and is thus highly profitable. This main advantage was ignored by the academic community and they only payed attention years after the first MPC publications. Then stability was proven under specific conditions and today MPC is still a big research area.

An MPC solves a finite horizon optimal control problem on-line at each sampling instant. Only the first optimal control input is applied to the process, the others are discarded. After that the time horizon is shifted one time step forward and the computation is done again. This is why MPC is also known as receding horizon control (RHC). In order to obtain a precise prediction of the system states a good mathematical model of the process is needed. The optimization is done by minimizing a cost function like in the finite LQR case but differs by the inclusion of constraints. Since this optimization problem must be solved at every sample step, the biggest challenge is to make the computation time or rather the time delay of the feedback control as small as possible. This is also the reason why MPC was only applied to slow processes in its early years when computers were much slower than today.

1.2 Purpose

In the last years numerical computation has become very efficient and thus MPC became more relevant for systems with a fast dynamic. Furthermore theoretical advances have led to improved understanding of MPC properties, in particular stability, and the techniques of MPC have been extended to nonlinear systems, [5]. Because of this fast progression and the growing relevance, an MPC course was offered by the signals and systems department of the Chalmers University of Technology three years ago. Part of this course are two assignments which the students have to study carefully. Unfortunately these two assignments are only simulations carried out with Matlab yet. After successful completion of this thesis at least the second assignment shall be applied to a real control system.

1.3 Problem

The problems to solve in this thesis can be split up into four different parts. In the first part a suitable system to apply MPC must be found. After that a mathematical model must be derived by using physical properties as well as experimentally identified parameters in part two. In part three the main problem is solved, namely the MPC design and implementation with an appropriate optimization algorithm. In the last part an assignment guidance for the MPC course (SSY280) must be drawn up.

1.4 Objective

The overall objective of this thesis is the creation of an assignment guidance for MPC course (SSY280). To obtain that, an MPC shall be designed and implemented successfully to an educational system, with must also be selected with different criteria. Advantages and disadvantages of different RHC structures shall be discussed.

1.5 Scope

A linear quadratic MPC will be designed and implemented to a nonlinear system. Quadratic means that the cost function which has to be optimized is quadratic and linear means that the model of the system is linear. Thus the nonlinear dynamics of the systems must be linearized. The thesis will not deal with nonlinear MPC.

2
Methods

In order to solve the problems, stated in section 1.3, and achieve the objectives of the thesis, stated in section 1.4, different methods will be used. These methods will be described in chronological order in this chapter. Figure 2.1 outlines the structure of the thesis and the methods.

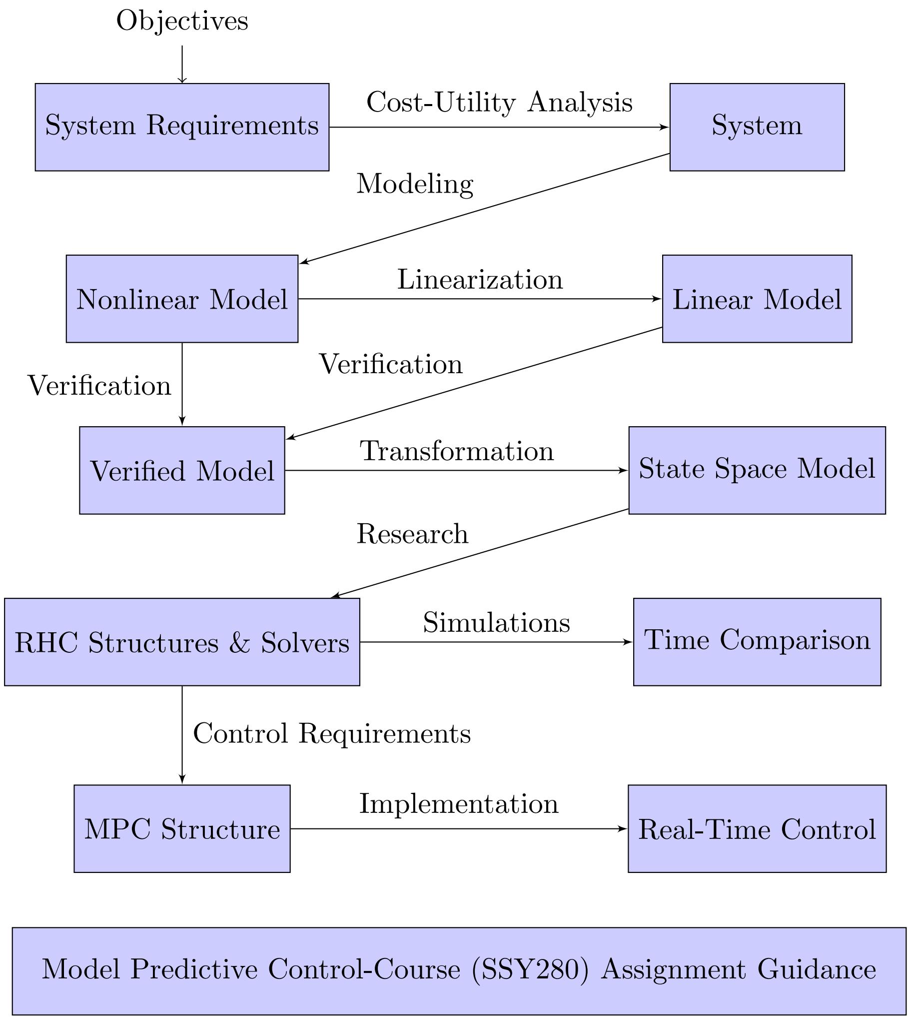

Figure 2.1: Structure of Thesis

The objective of the thesis is to implement an linear quadratic MPC to an educational control system in order to use it in an MPC student laboratory. Therefore an educational system must be chosen in the first step. General and specific system requirements shall be defined in order to score and weight them according to their importance for the project. Different systems shall be compared in a cost-utility analysis.

The system shall be modeled by using the kinetic and potential energy of the system. The driving force shall be calculated from physical properties of the motors. Friction of the system shall be included in additional terms. The obtained, nonlinear model, shall be linearized in order to apply linear MPC. Both models shall be verified with the real system.

Special attention shall be paid to discuss advantages and disadvantages of different Receding Horizon Control (RHC) structures. Different quadratic program solvers shall be compared regarding computation time of the mentioned RHC structures, in simulations.

An MPC structure, witch controls the system with zero offset in real-time, shall be implemented.

Finally an assignment guidance for the MPC-Course (SSY280) shall be drawn up.

3 System Selection

In this chapter a supply market analysis for educational control systems is done. For this purpose, both general and specific requirements are introduced in section 3.1. After that a scoring for the requirements is defined as well as a weighting according to their importance for the project. Finally the possible systems are compared in a cost–utility analysis and presented in table 3.1 in section 3.2. On basis of that analysis a decision is made which system to use in section 3.3.

3.1 Requirements

Before a supply market analysis can be done, conditions of the control system must be set. Since this thesis is done for the MPC course the system must meet different MPC specific requirements as well as general ones. These requirements are explained in this section. Furthermore they are scored and weighted for the cost–utility analysis.

3.1.1 General

In order to make the cost-benefit of the system better it should be **applicable for other courses** besides the Model Predictive Control course (SSY280). For example for the Linear Control System Design (SSY285) or Nonlinear and Adaptive Control (ESS076) course. Since the University of Chalmers owns the **Matlab/Simulink** license the system must be executable with it and **real time** control must be possible. Furthermore the time limitation of the thesis requires a short **delivery time** as well as it should be **Plug & Play**. Because of the high costs of purchased control systems, the **price** must be considered also. Due to insufficient laboratory space the **required space** of the system should be as small as possible. According to that it would be nice to have an **expandable system** for further projects, to save space. The need of a **PCI or PCIe slot** or **FIREwire** must also be noted since it is not available for any computer. Last but not least the system should be **interesting** to awaken the passion of the students.

3.1.2 MPC Course

Additional to the general requirements there are some MPC specific ones. First of all the system should have at least two outputs i.e. it shall be a **SIMO or MIMO** system. A MIMO system would be more appreciated. Since the big advantage of MPC is the possibility to deal with constrains the system shall have at least one **constrained input** as well as one **constrained state**. Furthermore the students shall practice the integration of a state observer, which means the system shall have more states then outputs. Dealing with **set point** and **reference trajectory changes** shall also be possible.

3.1.3 Scoring and Weighting

A scoring for each requirement is defined below. If the requirement is perfectly meet, six points are given. If it does not met the requirement at all, zero points are given. The requirements “Executable with Matlab/Simulink” and “Real time control” are not listed because these are hard ones. All compared systems meet these two requirements. Furthermore the requirements are weighted according to their importance for the project.

1. **Price**(€)
   Weighting: 10
   Scoring: 0-3: 6, 3-6: 5, 6-9: 4, 9-12: 3, 12-15: 2, 15-18: 1, >18: 0
2. **Delivery time**(Weeks)
   Weighting: 9
   Scoring: 3: 6, 4: 5, 5: 4, 6: 3, 7: 2, 8: 1, >8: 0
3. **SIMO/MIMO**
   Weighting: 8
   Scoring: MIMO: 6, SIMO: 3, SISO: 0
4. **Constraints**
   Weighting: 8
   Scoring: Input and Output: 6, Input or Output: 3, No Constraints: 0
5. **Set Point**
   Weighting: 6
   Scoring: Is possible: 6, Is not possible: 0
6. **Interesting**
   Weighting: 5
   Scoring: Very: 6, Normal: 4, A little: 2, Not: 0
7. **Required space**
   Weighting: 4
   Scoring: Part of table: 6, Whole table or floor: 3, Big area on floor: 0
8. **Applicable for other courses**
   Weighting: 3
   Scoring: Pendulum up swing: 6, Usable for other course: 4, Not usable: 0

9. **PCI, PCIe, FIREwire**
   Weighting: 2
   Scoring: Needs neither: 6, Needs one: 3, Needs both: 0

10. **Expandability**
   Weighting: 1
   Scoring: Is expandable: 6, Is not expandable: 0

3.2 Cost–Utility Analysis

In table 3.1, 24 systems are listed, sorted by companies. Educational systems of the companies Quanser, ECP Systems, Feedback, INTECO, Leypold and Googoltech were considered. The requirement "Interesting" of a system is evaluated by at least two different persons. Nevertheless it is clear that this requirement is more or less subjective. The weighted result is used to make a decision for a system in section 3.3.

Table 3.1: Cost-Utility Analysis of Suitable Control Systems

	General							MPC			
	Courses	Price	Space	Delivery	Expandability	PCI, FIREwire	Interesting	SIMO/MIMO	Cons. In,- Output	Set Point	Weighted Result
Weight:	3	10	4	8	1	2	5	8	8	6	
Quanser											
01. Linear Inverted Pendulum	6	3	3	6	6	6	2	3	6	6	256
02. Double Inverted Pendulum	6	3	3	6	6	6	4	3	6	6	266
03. Flexible Joint with IP	6	2	3	6	6	6	4	3	6	6	256
04. High Fidelity IP	6	0	3	6	6	6	4	3	6	6	226
05. Rotary Inverted Pendulum	6	4	6	6	6	6	4	3	3	6	254
06. 2 DOF Inverted Pendulum	4	1	6	6	6	6	6	6	6	0	250
07. 2 DOF Ball Balancer	4	1	6	6	6	3	6	6	6	6	280
08. 2 DOF Helicopter	4	2	3	6	0	6	6	6	6	6	278
09. 3 DOF Helicopter	4	0	0	6	0	6	6	6	6	6	246
10. 3 DOF Hover	4	0	3	6	0	6	6	6	6	6	258
11. 3 DOF Crane	4	0	0	6	0	6	6	6	6	6	246

Table 3.1 Continued

	General							MPC			
	Courses	Price	Space	Delivery	Expandability	PCI, FIREwire	Interesting	SIMO/MIMO	Cons. In,- Output	Set Point	Weighted Result
Weight:	3	10	4	8	1	2	5	8	8	6	
ECP System											
12. Inverted Pendulum	6	2	6	0	0	3	2	3	6	6	186
Feedback											
13. Digital Pendulum	6	4	0	0	0	3	2	3	6	6	182
14. Twin Rotor MIMO	4	4	6	0	0	3	6	6	6	6	244
INTECO											
15. 3D Crane	4	4	3	4	0	6	6	6	6	6	288
16. Pendulum & Cart Control	6	5	0	4	0	6	2	3	6	6	248
17. Tower Crane	4	4	3	4	0	6	6	6	6	6	288
18. Two Rotor Aerodynamical	4	5	3	4	0	6	6	6	6	6	274
Leypold											
19. Invertiertes Pendel	6	0	3	0	0	3	2	3	6	6	154
Googoltech											
20. Inverted Pendulum	6	5	3	6	6	3	2	3	6	6	270
21. Circular IP	6	5	3	0	0	3	2	3	6	6	180
22. Planar IP	4	4	3	0	6	3	4	3	6	6	204
23. 3 DOF Helicopter Simulator	4	4	3	6	0	3	6	6	6	6	292
24. Ball and Plate System	4	4	6	0	0	3	6	6	6	6	244

3.3 Decision

As can be seen in table 3.1 the 3 DOF Helicopter Simulator of the company Googoltech took place one with 292 points followed by the 3D Crane and the Tower Crane of the company INTECO with 288 points. Since the company INTECO is located not that far from Sweden in Poland, and because of its many references, a decision was made for the Tower Crane. This System can also be used for the Nonlinear Control course and maybe it is also suitable for the Linear Control course. The system also gives the possibility to expand the linear MPC course to a nonlinear MPC course in further years. Related work investigation has shown that it is possible to control the Tower Crane with a linear model and some modifications. This is important since linear MPC shall be applied to the system. The Tower Crane, the software and the real-time control module are described in more detail in chapter 4.

4 INTECO's Tower Crane

In this chapter first some general information about INTECO is given section 4.1. After that all different parts of the Tower Crane System, shown in figure 4.1, are described in detail. Technical information about the real-time control device RT-DAC/USB2, the RT-CON Professional development toolbox and the Tower Crane it self, is given in section 4.2.

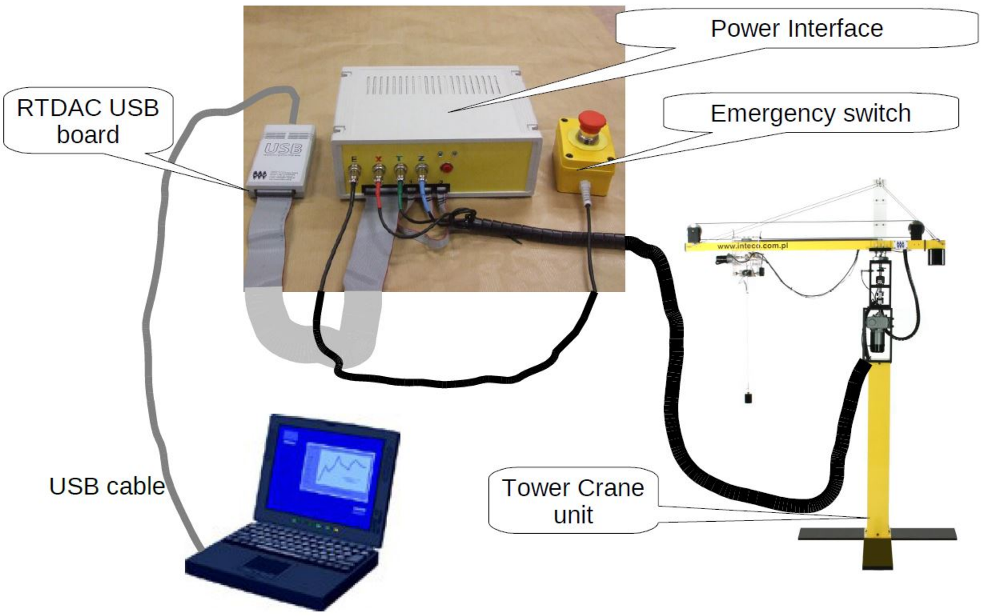

Figure 4.1: Tower Crane System[1]

4.1 INTECO

In formations about INTECO can be found on their website, [6]. The following quote about the company itself is taken from there.

"INTECO is an European company founded in 1997 by the researchers and engineers from University of Science and Technology (AGH) in Krakow – Poland. From the beginning till now our interest and major line of business have been focused on mechatronic devices for use in control engineering research and education. We have designed several innovative digitally controlled laboratory plants, all supplied with full mathematical/simulation models and control hardware and software. INTECO has been involved more and more in its own solutions of integrated control. We offer real-time kernels for control applications in MS Windows, rapid prototyping toolboxes for automatic code generation, and control/data acquisition systems. Most of our applications are controlled via RT-CON – our Windows based real-time software. We perform custom modifications and custom design to meet special needs like field bus drivers, PLC target applications or integration with SCADA systems. Our solutions are supported by our RT-DAC real-time I/O boards. Since January 1999 INTECO is an official MATLAB partner."

4.2 Technical Information

Basic information about the INTECO Tower Crane system can be found on their homepage, [6] and in manual [7]. The following technical descriptions of the different system parts are taken from their.

Real-Time Data Acquisition and Control

"The RT-DAC/USB2 is a multifunction analog and digital I/O board dedicated to real-time data acquisition and control in the Windows 95/98/NT/2000/XP environments. The board contains a Xilinx® FPGA chip. All boards are built as the OMNI version. It means the boards can be reconfigured to introduce a new functionality of all inputs and outputs without any hardware modification. The default configuration of the FPGA chip accepts signals from incremental encoders and generates PWM outputs, typical for mechatronic control applications and is equipped with the general purpose digital input/outputs (GPIO), A/D and D/A converters, timers, counters, frequency meters and chronometers."[7]

In this work, the analog and digital RT-DAC/USB2 is used. Specifications for the analog and digital inputs and outputs as well as for the PWM outputs can be found in manual [7].

Software

"The RT-CON software enables to develop real-time applications directly from Simulink models. Special extensions for Simulink and Real-Time Workshop help to build real-time applications for the MS Windows operating systems."[6]

Tower Crane

The following picture shows sensors, actuators and the cart, which is refereed later on as trolley.

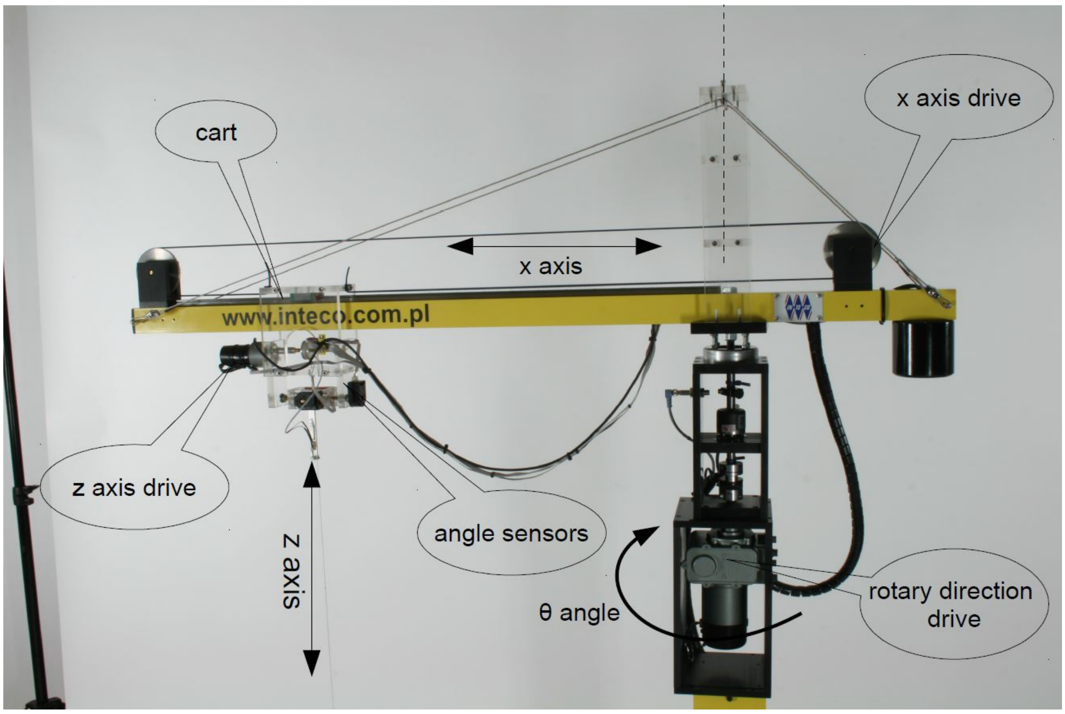

Figure 4.2: Tower Crane[1]

“ The Tower Crane is a nonlinear electromechanical system having a complex dynamic behavior and creating challenging control problems. It is controlled from a PC. Therefore it is delivered with hardware and software which can be easily mounted and installed in a laboratory. You obtain the mechanical unit together with the power supply and interface to the PC and the dedicated digital board configured in the Xilinx® technology. The software operates under MS Windows® using MATLAB® and RTW toolbox package.

The Tower Crane setup (Fig. 4.2) consists of a payload hanging on a pendulum-like lift line wound by a motor mounted on a trolley. The payload is lifted and lowered in the z direction. Both the arm and the trolley are capable of horizontal motion: the trolley in the radial x direction along the arm and the arm in the rotary direction. The angular position of the arm is expressed by the θ angle. The payload attached to the end of the lift line can move freely in three dimensions. The Tower Crane is driven by three DC motors. There are five measuring encoders measuring five state variables: the trolley co-ordinates on the polar coordinates plane, the lift line length, and two deviation angles of the payload. The encoders measure movements with a high resolution equal up to 4096 pulses per rotation (ppr). These encoders together with the specialized mechanical solution create a unique measurement unit. The deviation of the load is measured with a high accuracy equal to 0.0015 rad. The power interface amplifies the control signals which are transmitted from the PC to the DC motors. The PC equipped with the RT-DAC/USB multipurpose digital I/O board communicates with the power interface. The whole logic necessary to activate and read the encoder signals and to generate the appropriate sequence of pulses of PWM to control the DC motors is configured in the Xilinx® chip of the RT-DAC/USB board. All functions of the board are accessed from the Tower Crane toolbox which operates directly in the MATLAB®&Simulink® environment."[6]

The following list states the key features of the Tower Crane defined by INTECO in manual [7].

Key Features of Tower Crane:

- Three-dimensional laboratory model of industrial crane.
- A highly nonlinear MIMO system.
- It can be easily installed.
- There are high-resolution sensors – unique 2D angle measuring unit.
- The set-up is fully integrated with MATLAB®&Simulink® and operates in real-time in MS Windows®.
- Real-time control algorithms can be rapidly prototyped. No C code programming is required.
- The software includes complete dynamic models.
- User's Manual contains the library of basic controllers and a number of preprogrammed experiments which familiarise the user with the system in a fast way.
- It is ideal for illustrating complex nonlinear control algorithms.

5 Model

INTECO delivers a mathematical model of their system which is stated in manual [1] and derived in Master Thesis [8]. Unfortunately, the model only considers dynamics of the payload itself and neglects dynamics of the trolley and the tower. Furthermore, the acceleration of the trolley and the tower are used as inputs for the system. However, voltage supply of the motors is the only variable which can be manipulated. Therefore, a new mathematical model is derived by using the kinetic and potential energy in the Lagrange's equations in order to obtain the equations of motion in section 5.1. The driving force is calculated from the physical properties of the motors. Missing parameters of the tower system and the motors are estimated experimentally by comparing with the real system. The highly nonlinear system is simplified in section 5.2. Both models are compared and verified with the real system. A transformation of the simplified model into a linear state-space representation as well as a decoupling into three separate systems is done in section 5.3.

In the following figure the system outputs L, X_w, α, β and θ of the model are shown. These outputs will be controlled later on. The introduced coordinate system (X,Y,Z) has its origin in the middle of the tower and is moving with the arm. The X-axis goes through the point where the payload is mounted to the trolley. This mechanical model of the INTECO Tower Crane is used further.

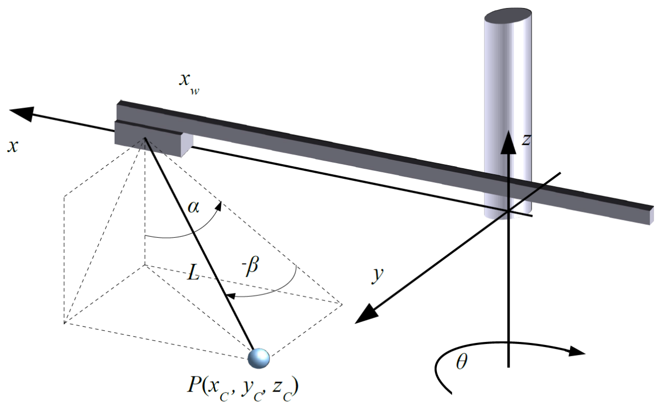

Figure 5.1: Tower Crane Model[1]

5.1 Nonlinear Model

A mathematical nonlinear model of the system without cable dynamics is derived by using the Lagrangian approach in order to obtain the equations of motion. Physical properties of the motors, electrical as well as mechanical, are used to model the actuators of the system. Viscous friction of the trolley and the arm is included. For simplicity of the model the cable length is assumed to be constant. Through this assumption it is possible to include the cable length as a parameter. Cable dynamics are stated later on.

5.1.1 Equations of Motion

The first step in deriving the equations of motion is to construct the trolley and load position, $\vec{p_t}$ and $\vec{p_l}$, in the coordinate system (X,Y,Z). These can be directly derived from figure 5.1.

$$\vec{p_l} = \{X_w - L\cos(\beta)\sin(\alpha),\ L\sin(\beta),\ -L\cos(\beta)\cos(\alpha)\} \tag{5.1}$$
$$\vec{p_t} = \{X_w,\ 0,\ 0\} \tag{5.2}$$

In order to derive the Lagrange's equations, the Lagrangian $\mathcal{L} = T - V$ must be constructed. Therefore the kinetic energy T and the potential energy V are needed. The whole kinetic energy of the system consists of the kinetic energy of the trolley and the payload as well as of the rotational energy of the arm. The trolley and the load are both modeled as point masses and therefore no rotational energy of these parts must be considered. The potential energy has its lowest value when the load is at its lowest point and increases for increasing angles α and β.

$$T = \frac{1}{2}m\dot{\vec{p_l}}\cdot\dot{\vec{p_l}} + \frac{1}{2}M_t\dot{\vec{p_t}}\cdot\dot{\vec{p_t}} + \frac{1}{2}J_o\dot{\theta}^2 \tag{5.3}$$
$$V = -mgL\cos(\beta)\cos(\alpha) \tag{5.4}$$

The velocities of trolley and payload can be calculated by using $\vec{p_l}$ and $\vec{p_t}$ in the following equation:

$$\dot{\vec{p}} = \frac{d\vec{p}}{dt} + \vec{\omega_T}\times\vec{p} \tag{5.5}$$

Using equation (5.5), where the angular velocity of trolley and load around the tower is described by $\vec{\omega_T} = \{0,0,\dot{\theta}\}$, in (5.3) and (5.4), finally allows to calculate the Lagrange's equations, where the generalized forces are described by $\vec{F} = \{F_x{,}0{,}F_\theta{,}0\}$ regarding the displacement vector $\vec{q} = \{X_w{,}\alpha{,}\theta\,\beta\}$:

$$\frac{d}{dt}(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}) - \frac{\partial \mathcal{L}}{\partial q_j} = F_j - B_{q_j}\dot{q}_j, \quad j = 1{,}2{,}3{,}4 \tag{5.6}$$

The four nonlinear equations of motion were obtained from (5.6) as follows:

$$\begin{aligned}
&M\ddot{X}_w + m\Big(-L\cos(\Theta)\dot{y}\dot{\Theta} + L\cos(\Theta)\sin(\Phi)\dot{\Theta}^2 + 2L\cos(\Phi)\sin(\Theta)\dot{\Phi}\dot{\Theta}\\
&+ L\cos(\Theta)\sin(\Phi)\dot{\Phi}^2 + \ddot{X}_w - L\sin(\Theta)\ddot{y} + L\sin(\Theta)\sin(\Phi)\ddot{\Theta} - L\cos(\Theta)\cos(\Phi)\ddot{\Phi}\Big)\\
&- \Big[\Big(m + M\Big)X_w - Lm\cos(\Theta)\sin(\Phi)\Big]\dot{y}^2 + Lm\cos(\Theta)\dot{\Theta}\dot{y} = F_x - B_x\dot{X}_w
\end{aligned} \tag{5.7}$$

$$\begin{aligned}
&-\frac{1}{2}Lm\Big[-2\cos(\Theta)\Big(-2L\dot{\Theta}\sin(\Theta)\dot{\Phi} + L\cos(\Theta)\ddot{\Phi} + L\sin(\Theta)\cos(\Phi)\ddot{y}\\
&- \cos(\Phi)\ddot{X}_w\Big) + L\dot{y}\left(\sin(2\Theta)\dot{\Phi}\sin(\Phi) - 2\dot{\Theta}\cos(2\Theta)\cos(\Phi)\right)\\
&- 2\dot{X}_w\left(\dot{\Theta}\sin(\Theta)\cos(\Phi) + \cos(\Theta)\dot{\Phi}\sin(\Phi)\right)\Big] - Lm\Big[-g\cos(\Theta)\sin(\Phi)\\
&+ \cos(\Theta)\cos(\Phi)\dot{y}^2\Big(L\cos(\Theta)\sin(\Phi) - X_w\Big) - L\dot{y}\left(\dot{\Theta}\cos(\Phi) + \sin(\Theta)\cos(\Theta)\dot{\Phi}\sin(\Phi)\right)\\
&+ \dot{X}_w\left(\dot{\Theta}\sin(\Theta)\cos(\Phi) + \cos(\Theta)\dot{\Phi}\sin(\Phi)\right)\Big] = -B_\alpha\dot{\alpha}
\end{aligned} \tag{5.8}$$

$$\begin{aligned}
&2MX_w\dot{X}_w\dot{y} + J\ddot{y} + MX_w^2\ddot{y} + m\Big[-L^2\ddot{\Theta}\sin^2(\Theta)\sin(\Phi) - L^2\ddot{\Theta}\cos^2(\Theta)\sin(\Phi)\\
&- 2L^2\dot{\Theta}\sin^2(\Theta)\dot{\Phi}\cos(\Phi) + L^2\sin(\Theta)\cos(\Theta)\ddot{\Phi}\cos(\Phi) - L^2\sin(\Theta)\cos(\Theta)\dot{\Phi}^2\sin(\Phi)\\
&+ L^2\cos^2(\Theta)\sin^2(\Phi)\ddot{y} + L^2\sin^2(\Theta)\ddot{y} - L\sin(\Theta)\ddot{X}_w + LX_w\ddot{\Theta}\cos(\Theta) - LX_w\dot{\Theta}^2\sin(\Theta)\\
&- 2LX_w\cos(\Theta)\sin(\Phi)\ddot{y} + L\dot{y}\Big(\dot{\Theta}[L\sin(2\Theta)\cos^2(\Phi) + 2X_w\sin(\Theta)\sin(\Phi)]\\
&+ \cos(\Theta)\dot{\Phi}[L\cos(\Theta)\sin(2\Phi) - 2X_w\cos(\Phi)]\Big) + 2\dot{X}_w\dot{y}\Big(X_w - L\cos(\Theta)\sin(\Phi)\Big) + X_w^2\ddot{y}\Big]\\
&= F_\theta - B_\theta\dot{\theta}
\end{aligned} \tag{5.9}$$

$$\begin{aligned}
&Lm\Big[L\ddot{\Theta} - L\sin(\Phi)\ddot{y} - L\cos(\Phi)\dot{\Phi}\dot{y} + \sin(\Theta)\sin(\Phi)\ddot{X}_w + \dot{X}_w\Big(\dot{\Theta}\cos(\Theta)\sin(\Phi)\\
&+ \sin(\Theta)\dot{\Phi}\cos(\Phi) + \cos(\Theta)\dot{y}\Big) + X_w\left(\cos(\Theta)\ddot{y} - \dot{\Theta}\sin(\Theta)\dot{y}\right)\Big]\\
&- Lm\Big[-\sin(\Theta)\Big(g\cos(\Phi) + L\cos(\Theta)\dot{\Phi}^2\Big) + \dot{y}\Big(L\cos(2\Theta)\dot{\Phi}\cos(\Phi) - X_w\dot{\Theta}\sin(\Theta)\Big)\\
&+ \sin(\Theta)\dot{y}^2\Big(L\cos(\Theta)\cos^2(\Phi) + X_w\sin(\Phi)\Big) + \dot{X}_w\Big(\dot{\Theta}\cos(\Theta)\sin(\Phi) + \sin(\Theta)\dot{\Phi}\cos(\Phi)\\
&- \cos(\Theta)\dot{y}\Big)\Big] = -B_\beta\dot{\beta}
\end{aligned} \tag{5.10}$$

5.1.2 Motor Model

The dynamic equation for a permanent magnet dc motor is given by the following equation. The last term describes the counter-electromotive force which is produced by the rotation of the armature within the permanent magnet.

$$V_a = R_a i_a + L_a \frac{i_a}{dt} + k_g k_u \omega \tag{5.11}$$

Since the electrical response is much faster than the mechanical response, L_a can be assumed zero. The produced torque of a motor is given by the two following equations.

$$T_d = \eta k_g k_m i_a \tag{5.12}$$
$$T_d = T_e + B k_g \omega \tag{5.13}$$

In SI units k_m and k_u must be equal for DC motors. Therefore only k_m is used in further equations. Since $T_e \gg B k_g \omega$ for low speed applications, the second term in (5.13) was neglected for simplicity. Using (5.12) and (5.13) in (5.11) allows to calculate the external torque of the motor.

$$T_e = \frac{\eta k_g k_m}{R_a} V_a - \frac{\eta k_g^2 k_m^2}{R_a} \omega \tag{5.14}$$

Equation (5.14) can now be used to calculate the specific forces in X- and θ-direction, F_x and F_θ. Since the motion of the arm is rotational, F_θ can be calculated directly.

$$F_\theta = \frac{\eta_\theta k_{g\theta} k_{m\theta}}{R_{a\theta}} G_{a\theta} U_\theta - \frac{\eta_\theta k_{g\theta}^2 k_{m\theta}^2}{R_{a\theta}} \dot{\theta} \tag{5.15}$$

For calculation of F_x, the translational motion of the trolley must be transferred into a rotatory motion. This is done by using the following relations.

$$\dot{X}_w = r_x \omega \tag{5.16}$$
$$T_e = F_x r_x \tag{5.17}$$

Finally F_x can be calculated as follows, where r_x is the radius of the deflection pulley.

$$F_x = \frac{\eta_x k_{gx} k_{mx}}{R_{ax} r_x} G_{ax} U_x - \frac{\eta_x k_{gx}^2 k_{mx}^2}{R_{ax} r_x^2} \dot{X}_w \tag{5.18}$$

5.1.3 Cable Model

Until now the cable length was assumed to be constant in the models for the trolley and arm subsystem. The model for the cable dynamics was obtained from the authors of article [9]. The system was modeled as a first order plus integrator system. The equation of motion for the cable dynamics with constant payload mass can be stated as follows, where τ_{dL} describes the input time delay.

$$0.02\ddot{L} + \dot{L} = 0.0108 G_{aC} U_c(t - \tau_{dC}) \tag{5.19}$$

5.1.4 Model Parameters

This section describes how the parameters for the mechanical tower system as well as for the electrical motor model were obtained. In the case of experimentally estimated parameters, the measurements can be found in section 5.1.6, where the whole nonlinear model is verified.

The parameters m and M were obtained from the tower crane manual [1]. Furthermore, B_x, B_α, B_θ, B_β, η_{gx}, $\eta_{g\theta}$, η_{mx}, $\eta_{m\theta}$, R_{ax}, $R_{a\theta}$, k_{mx}, $k_{m\theta}$, k_{gx}, $k_{g\theta}$ were used as first estimates from paper [9]. The parameters J, r_x, G_{ax}, $G_{a\theta}$ and the system input delays τ_{dx} and $\tau_{d\theta}$ were used as first estimates from master thesis [10]. The remaining parameters, G_{ac} and τ_{dc}, were obtained with experiments.

The parameters which have been used as first estimates were improved experimentally by applying basic test like step responses or just swinging the payload in each direction.

The following table shows all final parameters which were used for the nonlinear model.

Table 5.1: Identified Model Parameters

Expression	Value	Expression	Value	Expression	Value
M	0.7	η_x	0.36	R_{ax}	25
m	0.32	η_θ	0.24	$R_{a\theta}$	0.5
B_x	14	k_{mx}	0.032	$G_{a\theta}$	12.05
B_α	0.0055	$k_{m\theta}$	0.0195	G_{ax}	15
B_θ	11	τ_{dx}	0.03	G_{aC}	12
B_β	0.018	$\tau_{d\theta}$	0.06	J	2.4
k_{gx}	76.84	τ_{dC}	0.01		
$k_{g\theta}$	275	r_x	0.0375		

5.1.5 Real System vs. Nonlinear Model

Figure 5.2 shows the outputs of the model on the right side and the outputs of the real system on the left side. This difference is caused by the fact, that the model was derived before the manual of the Tower Crane was obtained from INTECO. Also the zero position of the trolley differs in the systems, which can not be seen in the figure. Furthermore the cable in the real system is mounted in such a way that it differs from the model. This offset was mostly measured by swinging the payload and comparing the frequency which depends directly on the cable length.

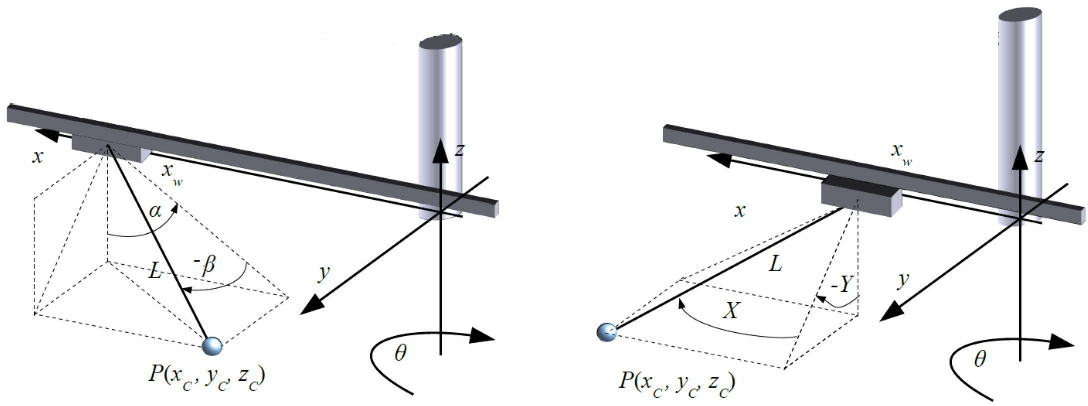

Figure 5.2: Measured Variables at the Laboratory System[1]

In order to control the system later on the measured states must be converted into the corresponding model values. This can be done by using the following relations.

Table 5.2: Measured Angles - Real System vs. Model

Measured States	States in Model
X_w	$X_w + 0.22$
θ	θ
L	$L - 0.15$
X	$\alpha = \arcsin(-\frac{\sin(X)}{\cos(\arcsin(\sin(Y)\cos(X)))})$
Y	$\beta = \arcsin(\frac{\sin(Y)}{\cos(X)})$

When the system was used the first time it was observed that the angles α and β are dependent on the angle θ. This could be attributed later to the fact that either the tower by itself is not 100% straight or the floor is not plane. The next figure shows this behavior.

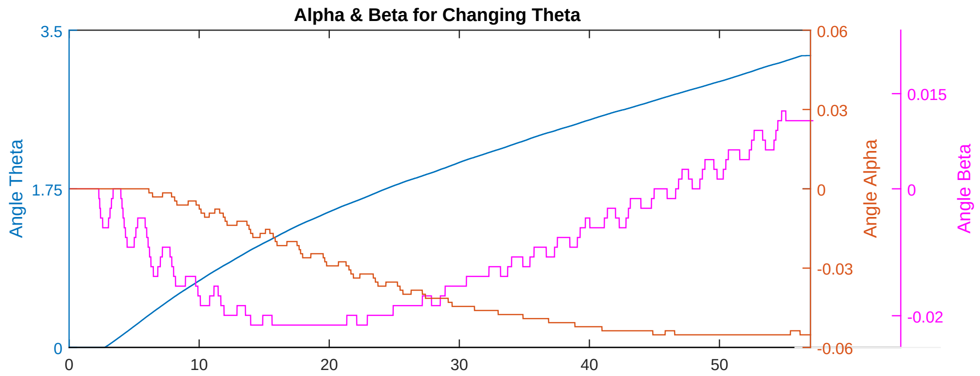

Figure 5.3: Angles for Changing Theta - Unfitted

In order to control the system in an efficient way the measured angles were fitted into a function depended on the angle θ by using the curve fitting tool from Matlab. This curve is used together with the other conversions, given above, to calculate the values which are finally used for controlling the system. The following Figure shows the angles α and β for changing angle θ after conversion.

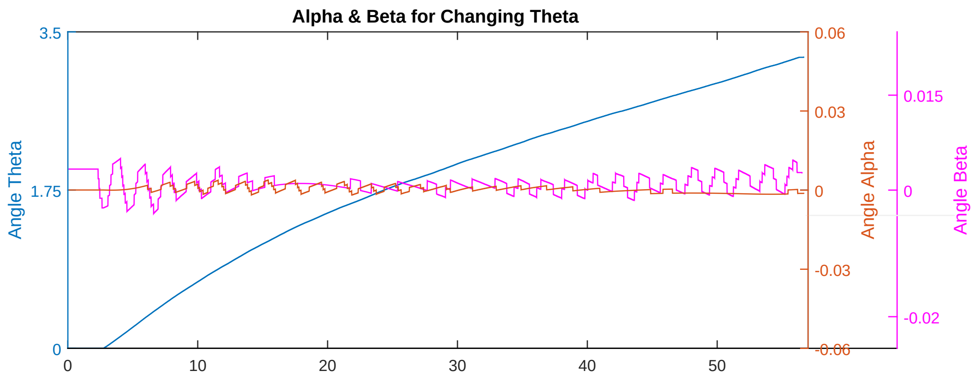

Figure 5.4: Angles for Changing Theta - Fitted

5.1.6 Model Verification

In this subsection the nonlinear model, derived in section 5.1, is verified by comparison with measurements of the real system. In figure 5.5 and 5.6 swinging the payload in α- and β-direction without an input signal is shown. With help of these measurements the viscous friction coefficients as well as the cable length offset were estimated. Step response measurements for the trolley and arm position as well as for the cable length are depicted in figures 5.7, 5.8 and 5.9, respectively. These were used to obtain system delays, amplifier gains, viscous friction coefficients as well as the arm inertia. The third signal in each measurement represents the control signal u.

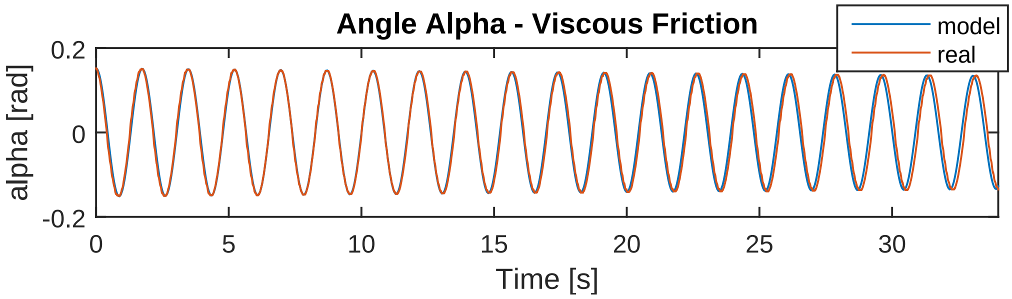

Figure 5.5: Nonlinear - Swinging - α

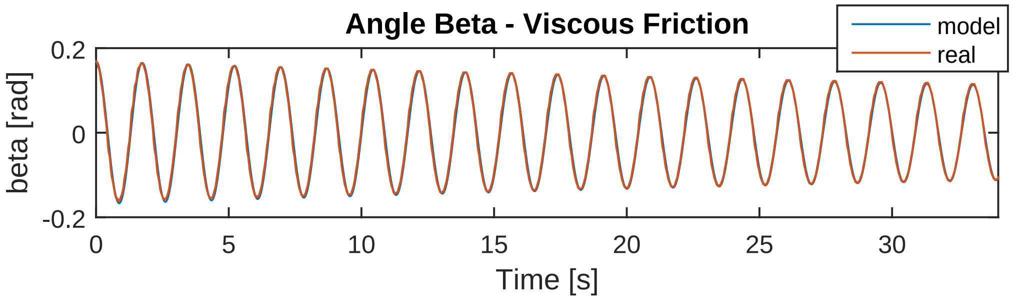

Figure 5.6: Nonlinear - Swinging - β

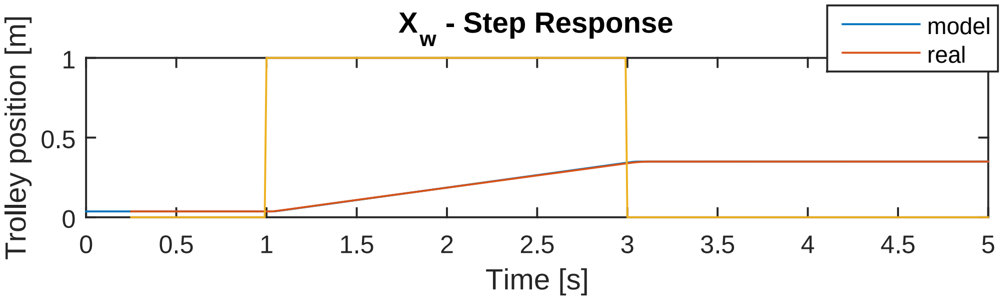

Figure 5.7: Nonlinear - Step Response - X_w

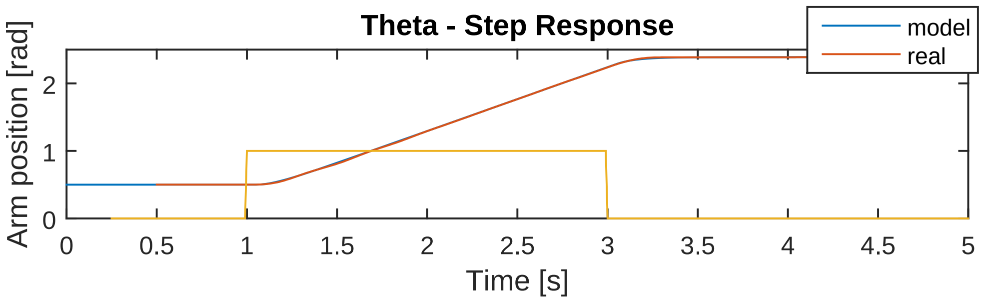

Figure 5.8: Nonlinear - Step Response - θ

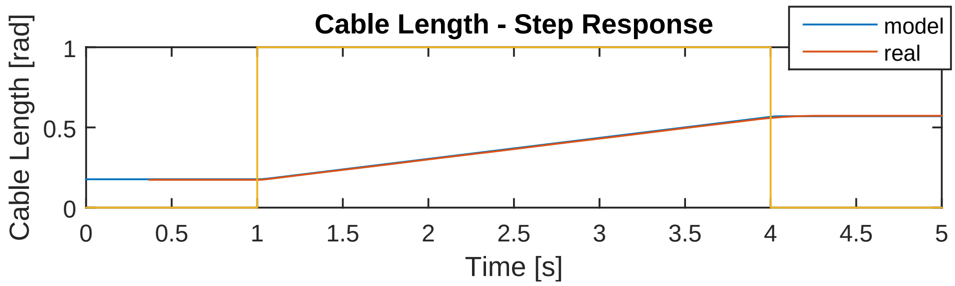

Figure 5.9: Nonlinear - Step Response - L

Furthermore a input test signal was applied to the real system and the model. The behavior of the trolley, the arm and the cable length is shown in the next three figures, respectively.

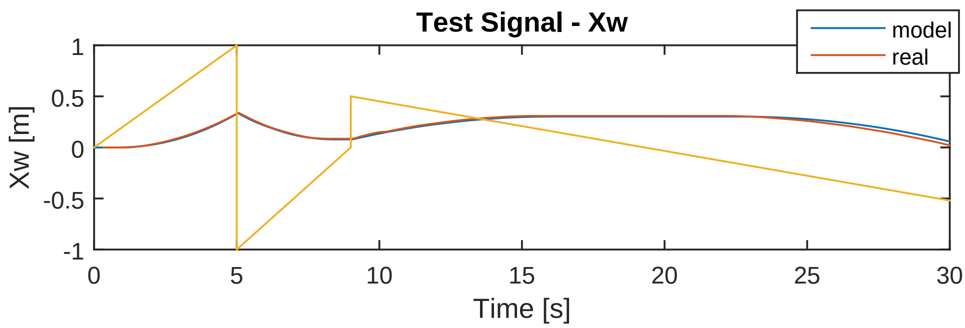

Figure 5.10: Nonlinear - Test Signal - X_w

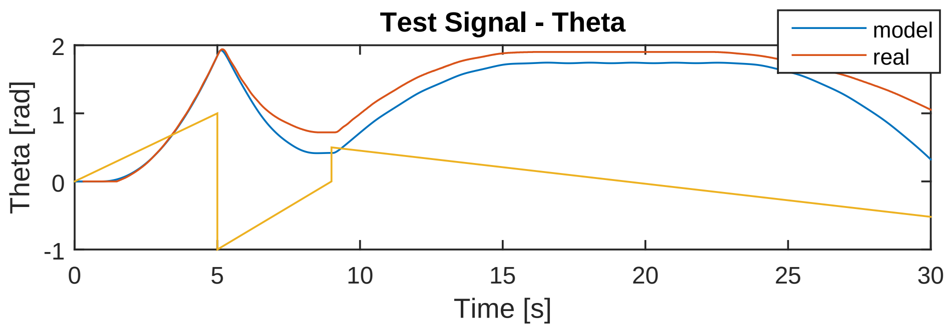

Figure 5.11: Nonlinear - Test Signal - θ

Mainly two important facts could be observed through this measurements. First of all the friction for the arm is not equal in both directions. Nevertheless the MPC controller for the arm works fine as will be stated in later chapters. The more important fact is that there is a huge stiction in the trolley as well as in the arm movement. This can be

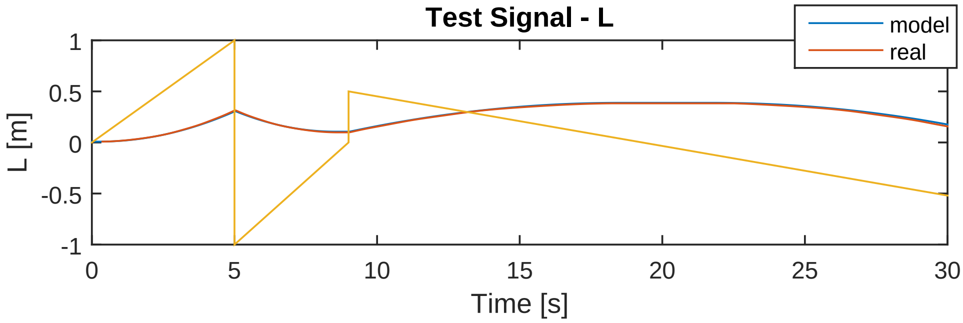

Figure 5.12: Nonlinear - Test Signal - L

seen especially between 15s and 25s in all three figures. Stiction is here already included in the Simulink model with $V_{dx} = 0.16$, $V_{d\theta} = 0.165$ and $V_{dc} = 0.16$. In order to get a sufficient control some integrator behavior must be integrated into the control structure.

The following figures are showing the behavior for the angle α and β, receptively.

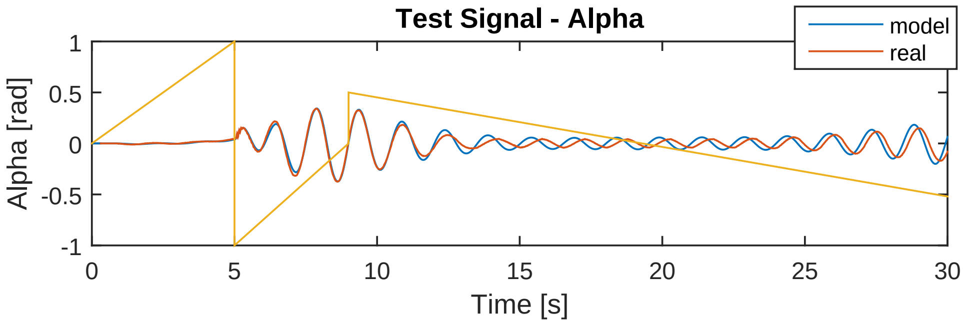

Figure 5.13: Nonlinear - Test Signal - α

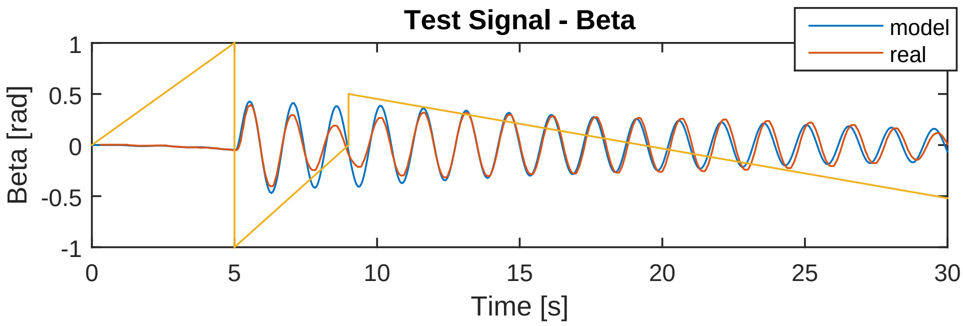

Figure 5.14: Nonlinear - Test Signal - β

Except for very small deviations the model matches the real system. What is needed to be said is that an additional factor was included to reduce the influence of the arm movement to the angle α. In first measurements the amplitude of α was higher in the model than in the real system. That effect was attributed to different mounting of the cable in the real system than in the model. This is not stated in more detail here because this factorized term is not used at all in the linear model.

5.2 Simplified Model

Since the derived equations of motion are highly nonlinear and complex some simplifications were done for the purpose of control. Small swing angles were assumed and the assumption that the rate of change of X_w and θ are the same order of magnitude as the swing angles α and β and their rates was made. For decoupling and controller design two terms, marked red in the following equations, were neglected for decoupling the system [11]. These neglected terms are for example effects from the arm acceleration to α caused by the Coriolis force. Tests have shown that these terms have a, maybe not negligible, influence on the dynamics as can be seen later in figures. If in the end the controlling is insufficient it would be necessary to consider how to include these terms into the control structure. The simplified model can finally be written as follows without the red terms.

Simplified model:

$$
\begin{aligned}
0.02\ddot{L} + \dot{L} &= 0.0108 G_{aL} U_c(t - \tau_{dL}) \\
\ddot{X}_w + \left(\frac{\eta_x k_{gx}^2 k_{mx}^2}{R_{ax} r_x^2} + B_x\right)\frac{1}{M}\dot{X}_w + \frac{m}{M} g\alpha &= \frac{\eta_x k_{gx} k_{mx}}{R_{ax} r_x M} G_{ax} U_x(t - \tau_{d_x}) \\
L\ddot{\alpha} + g\alpha - \ddot{X}_w - B_\alpha \dot{\alpha} + L\ddot{\theta}\beta &= 0 \\
\left(1 + \frac{M}{J} X_w^2\right)\ddot{\theta} + \left(\frac{\eta_\theta k_{g\theta}^2 k_{m\theta}^2}{R_{a\theta}} + B_\theta\right)\frac{1}{J}\dot{\theta} - \frac{m}{J} g X_w \beta &= \frac{\eta_\theta k_{g\theta} k_{m\theta}}{R_{a\theta} r_\theta J} G_{a\theta} U_\theta(t - \tau_{d_\theta}) \\
L\ddot{\beta} + g\beta + X_w\ddot{\theta} - B_\beta \dot{\beta} - L\ddot{\theta}\alpha &= 0
\end{aligned}
\tag{5.20}
$$

The simplified model was verified with the same input test signal as the nonlinear model. Figures 5.15 to 5.18 are showing the behavior of the simplified model compared to the real system. The deviation in figure 5.17 is caused by the neglected terms mentioned in the paragraph before. But also the angle is not “small” any more, which was assumed for simplification. The Cable length measurements are not stated again since nothing changed through the simplifications. The third signal in each measurement represents the control signal u.

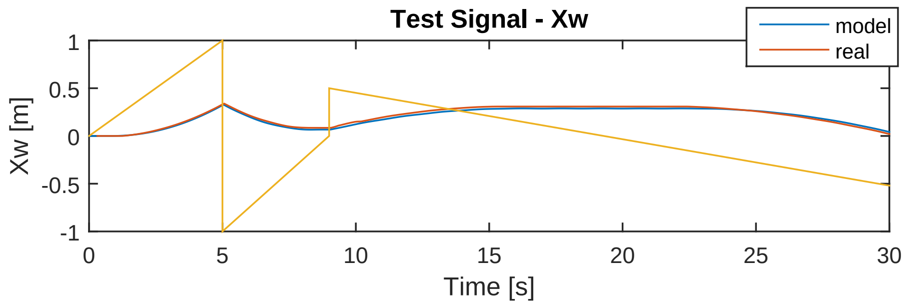

Figure 5.15: Linear - Test Signal - X_w

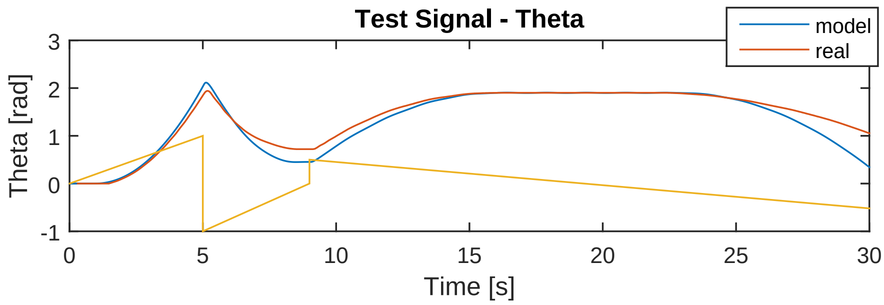

Figure 5.16: Linear - Test Signal - θ

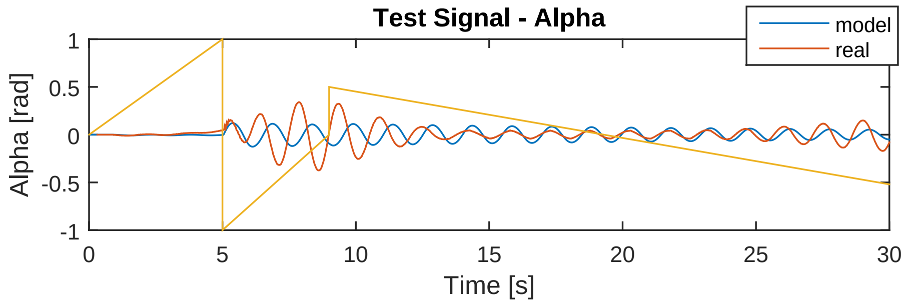

Figure 5.17: Linear - Test Signal - α

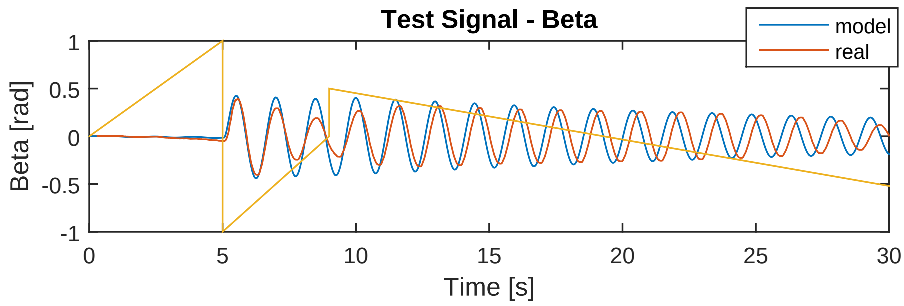

Figure 5.18: Linear - Test Signal - β

5.3 Linear State-Space Representation

In this section the simplified model is converted into a nonlinear state-space representation in the first step. In the second step, in order to linearize the model as well as to obtain a better computation time of the MPC algorithm, the nonlinear state-space representation is decoupled into three linear parts.

5.3.1 Conversion to State-Space Representation

The following state variables were selected for the conversion into a state-space representation.

$$\begin{array}{ccccc} x_1 = L & x_2 = \dot{L} & x_3 = X_w & x_4 = \dot{X}_w & x_5 = \alpha \\ x_6 = \dot{\alpha} & x_7 = \theta & x_8 = \dot{\theta} & x_9 = \beta & x_{10} = \dot{\beta} \end{array}$$

Furthermore the following inputs and outputs of the system were defined.

$$\begin{array}{ccccc} u_1 = U_c & u_2 = U_x & u_3 = U_\theta & & \\ y_1 = x_1 & y_2 = x_3 & y_3 = x_5 & y_4 = x_7 & y_5 = x_9 \end{array}$$

Using the introduced state variables, inputs and outputs of the system, the simplified model can be represented as a nonlinear state-space model as shown in (5.21). The input time delays are also included in this representation.

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} x_2 \\ -50x_2 + 6.48u_1(t-0.01) \\ x_4 \\ -107.83x_4 - 4.48x_5 + 20.09u_2(t-0.03) \\ x_6 \\ -\frac{107.83}{x_1}x_4 - \frac{14.29}{x_1}x_5 - \frac{0.0055}{x_1}x_6 + \frac{20.09}{x_1}u_2(t-0.03) \\ x_7 \\ \frac{1}{1+0.29x_3^2}(-10.37x_8 + 1.31x_3x_9 + 13u_3(t-0.06)) \\ x_{10} \\ \frac{1}{x_1}(-9.81x_9 - 0.018x_{10} - \frac{x_3}{1+0.29x_3^2}(-10.37x_8 + 1.31x_3x_9 + 13u_3(t-0.06))) \end{bmatrix}}_{f(x,u)} \tag{5.21}$$

Note that the cable dynamics are already linear. The trolley subsystem, row three to six, is nonlinear in terms of the cable length. The arm subsystem, row seven to ten, is nonlinear in terms of the cable length and the trolley position. This structure makes it possible to decouple the system and use the cable length or rather the cable length and the trolley position as parameters.

5.3.2 Decoupled State-Space Model

As mentioned before, the nonlinear state-space model can be linearized by splitting the system into three separate systems and using coupling terms as parameters. The cable length L is a parameter in the Trolley-Subsystem as well as in the Arm-Subsystem. Furthermore, the trolley position X_w is a parameter in the Arm-Subsystem. Linearizing the system thus brings the advantage of much less computation time of the MPC since the decoupled system is much less complex.

For easier notation of the Arm-Subsystem the new varying parameter $\Phi(X_w)$, dependent on X_w, is introduced. The three subsystems can be written as shown in (5.22), (5.23) and (5.24), respectively.

Cable-Subsystem:

$$\boxed{\begin{aligned} &\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{x}_C} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -50 \end{bmatrix}}_{A_C} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x_C} + \underbrace{\begin{bmatrix} 0 \\ 6.48 \end{bmatrix}}_{B_C} u_1 \\ &\text{Delay of Input } u_1\text{: } 0.01 \text{ s} \\ &\underbrace{\begin{bmatrix} y_1 \end{bmatrix}}_{\dot{y}_C} = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C_C} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned}} \tag{5.22}$$

Trolley-Subsystem:

$$\boxed{\begin{aligned} &\underbrace{\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix}}_{\dot{x}_T} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -107.83 & -4.48 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{107.83}{L} & -\frac{14.29}{L} & -\frac{0.0055}{L} \end{bmatrix}}_{A_T} \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}}_{x_T} + \underbrace{\begin{bmatrix} 0 \\ 20.09 \\ 0 \\ \frac{20.09}{L} \end{bmatrix}}_{B_T} u_1 \\ &\text{Delay of Input } u_1\text{: } 0.03 \text{ s} \\ &\underbrace{\begin{bmatrix} y_2 \\ y_3 \end{bmatrix}}_{\dot{y}_T} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{C_T} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \end{aligned}} \tag{5.23}$$

Arm-Subsystem:

$$\boxed{\begin{aligned}
&\Phi(X_w) = \frac{1}{1+0.29X_w^2}\\
&\underbrace{\begin{bmatrix}\dot{x}_7\\ \dot{x}_8\\ \dot{x}_9\\ \dot{x}_{10}\end{bmatrix}}_{\dot{x}_A} = \underbrace{\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & -10.37\Phi(X_w) & 1.31X_w\Phi(X_w) & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{10.37X_w\Phi(X_w)}{L} & \frac{-9.81+1.31X_w^2\Phi(X_w)}{L} & -\frac{0.018}{L}\end{bmatrix}}_{A_A}\underbrace{\begin{bmatrix}x_7\\ x_8\\ x_9\\ x_{10}\end{bmatrix}}_{x_A}\\
&\qquad + \underbrace{\begin{bmatrix}0\\ 13\Phi(X_w)\\ 0\\ \frac{-13X_w\Phi(X_w)}{L}\end{bmatrix}}_{B_A}u_2\\
&\text{Delay of Input } u_2\text{: 0.06 s}\\
&\underbrace{\begin{bmatrix}y_4\\ y_5\end{bmatrix}}_{\dot{y}_A} = \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}}_{C_A}\begin{bmatrix}x_7\\ x_8\\ x_9\\ x_{10}\end{bmatrix}
\end{aligned}} \tag{5.24}$$

6 General MPC

In this chapter the general structure of anMPC controller is stated. In section 6.1 different structures of Receding Horizon Control (RHC), their advantages and disadvantages are discussed. Furthermore, in section 6.2 an approach for offset free control is stated. In the last section a time-varying as well as a steady state Kalman filter, used as an observer, will be introduced.

As outlined in the introduction, the scope of this work is linear quadratic MPC. Linear models, as derived in the previous chapter, will be assumed. In general these models can be stated discretized as follows.

$$x(k+1) = Ax(k) + Bu(k) \tag{6.1}$$

$$y(k) = Cx(k) \tag{6.2}$$

Furthermore, a quadratic objective and affine constraints on x and u are assumed. With these assumptions, it is possible to formulate a quadratic programming problem later on.

6.1 Constrained RHC

The RHC is the centerpiece of the MPC structure. It minimizes a weighted cost function, subjected to constraints, over a given prediction horizon N. A sequence of optimal control signals $u(k)$ to $u(k+M)$, where M defines the control horizon, is the result of the optimization. This notation defines the value of u at the time instants k to $k+M$, respectively and is used further on for control and state variables.

Only the first control signal $u(k)$ is applied to the system, the others are discarded. After one time step, k is shifted forward, and the procedure is done again. This is done in order to control $x(k)$ to zero, where $x(k)$ is a column vector with the states in it. This fashion of generating an optimal sequence of control signals is shown in the Figure 6.1.

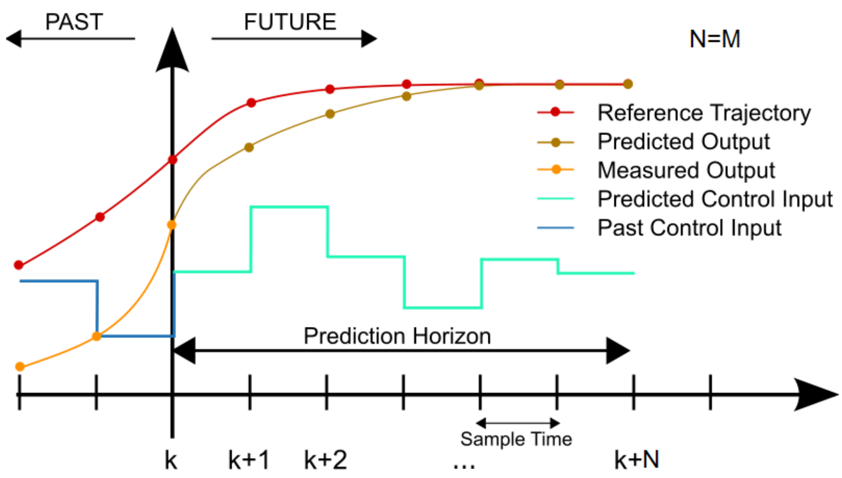

Figure 6.1: MPC - Scheme

There are the following possibilities of constructing the RHC:

1. Prediction & Control Horizon
   a) Prediction horizon N is equal to control horizon M.
   b) Prediction horizon N is bigger than control horizon M.
2. Optimization Variables
   a) All future states and control signals are considered as optimization variables.
   b) The future states are expressed a functions of the control signals (condensed RHC). Only control signals are considered as optimization variables.

Using the previous list, there are four possible combinations of constructing an RHC, choosing one of the alternatives in each of item 1 and 2 above. Each combination results in a different computation time of the RHC. The larger the prediction and control horizon is selected the better the stability behavior of the RHC gets. Assuming an unconstrained case this can be explained by the fact that an RHC with large N and $M = N$ gets closer to the infinite horizon LQ case for which stability is guaranteed, [12].

Since this thesis shall be used for a student laboratory, it is important to demonstrate the advantages and disadvantages of different combinations. The different combinations will be stated in the following subsections. In chapter 7 exact computation times are compared.

6.1.1 Control Horizon equal to Prediction Horizon

First of all an RHC, designed with prediction horizon N equal to control horizon M and future states as well as control signals considered as optimization variables, is discussed.

This structure gives the best overview about how an RHC in general works. The following definition describes the cost function for an RHC designed in this fashion.

Definition 6.1 *Cost function (N=M)*

$$V_N(x(k:k+N),u(k:k+N-1)) = x^T(k+N)Q_f x(k+N) + \sum_{i=k}^{k+N-1} x^T(i)Qx(i) + u^T(i)Ru(i) \tag{6.3}$$

where N denotes the prediction horizon, n the number of the states in vector x and m the number of system inputs. $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are weights to be chosen. Q is penalizing the state deviation and R the magnitude of the input. Q_f is the final weight which penalizes the final states.

The optimization problem for the RHC can be stated by using cost function 6.3 as follows.

Definition 6.2 *Optimization Problem (N=M)*

$$\min_{u(k:k+N-1),x(k:k+N)} V_N(x(k:k+N),u(k:k+N-1)) \tag{6.4}$$

$$\textit{subject to} \quad x(k+i) \in \mathcal{X}, \quad u(k+i) \in \mathcal{U}, \quad x(k+N) \in \mathcal{X}_f \subseteq \mathcal{X} \tag{6.5}$$

$$x(k+i+1) = Ax(k+i) + Bu(k+i) \quad \textit{for all } i \in (0,N-1) \tag{6.6}$$

The constraint sets $\mathcal{X}$, $\mathcal{X}_f$ and $\mathcal{U}$ are assumed to be affine.

The optimization problem stated by Definition 6.2 can be rewritten as a quadratic programming problem. The constraints (6.5) can be summarized in a inequality constraint. The system dynamics (6.6) are included as equality constraints. The cost function (6.4) can be transformed into a quadratic cost function as shown in the following definition.

Definition 6.3 *Quadratic Programming Problem (N=M)*

$$\min_z \frac{1}{2} z^T H z \tag{6.7}$$

$$\textit{subject to} \quad A_{eq} z = b_{eq} \tag{6.8}$$

$$A_{ineq} z \leq b_{ineq} \tag{6.9}$$

where the three equations can be written as stated in (6.10), (6.11) and (6.12), respectively. The notation x(k+i)$\rightarrow$ x(i) and u(k+i) $\rightarrow$ u(i) is used to short the equations.

$$\min_z \frac{1}{2} \underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N-1) \\ x(N) \\ u(0) \\ u(1) \\ \vdots \\ u(M-2) \\ u(M-1) \end{bmatrix}^T}_{z^T} \underbrace{\begin{bmatrix} Q & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & Q & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \ddots & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & Q & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & Q_f & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & R & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & R & \cdots & 0 & 0 \\ \vdots & 0 & \ddots & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & R & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & R \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N-1) \\ x(N) \\ u(0) \\ u(1) \\ \vdots \\ u(M-2) \\ u(M-1) \end{bmatrix}}_{z} \tag{6.10}$$

Note that $x(0) = x(k)$ is not part of the optimization since it is the actual real measured state.

$$\underbrace{\begin{bmatrix} I & 0 & \cdots & 0 & 0 & -B & 0 & \cdots & 0 & 0 \\ -A & I & \cdots & 0 & 0 & 0 & -B & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \ddots & I & 0 & 0 & 0 & \cdots & -B & 0 \\ 0 & 0 & \cdots & -A & I & 0 & 0 & \cdots & 0 & -B \end{bmatrix}}_{A_{eq}} \underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N-1) \\ x(N) \\ u(0) \\ u(1) \\ \vdots \\ u(M-2) \\ u(M-1) \end{bmatrix}}_{z} = \underbrace{\begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{b_{eq}} x(0) \tag{6.11}$$

By ordering the optimization variables in z chronologically, the matrix A_{eq} becomes banded, which is desirable for computational reasons, [5]. This is the reason why a chronological structured z should be implemented, instead of the one presented here.

$$\underbrace{\begin{bmatrix} I & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & I & 0 & \cdots & 0 \\ -I & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & -I & 0 & \cdots & 0 \\ 0 & \cdots & 0 & I & \cdots & 0 \\ \vdots & \cdots & 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & I \\ 0 & \cdots & 0 & -I & \cdots & 0 \\ \vdots & \cdots & 0 & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & -I \end{bmatrix}}_{A_{ineq}} \underbrace{\begin{bmatrix} x(1) \\ \vdots \\ x(N) \\ u(0) \\ \vdots \\ u(M-1) \end{bmatrix}}_{z} \leq \underbrace{\begin{bmatrix} x_{max} \\ \vdots \\ x_{max} \\ -x_{min} \\ \vdots \\ -x_{min} \\ u_{max} \\ \vdots \\ u_{max} \\ -u_{min} \\ \vdots \\ -u_{min} \end{bmatrix}}_{b_{ineq}} \tag{6.12}$$

The main disadvantage of an RHC constructed in the mentioned way is its computation time. This is because all future states as well as control signals are considered as optimization variables and furthermore, $M = N$ causes the largest number of optimization variables. On the other hand the structure is clear to understand and the stability behavior is better, compared to the case where $M < N$.

6.1.2 Control Horizon smaller than Prediction Horizon

As mentioned before, there is the possibility to construct the RHC with a smaller control horizon than prediction horizon. Compared with the previous case, $N = M$, this RHC has less computation time, because $(N - M)$ less control variables must be optimized. The structure of an RHC designed in this manner is discussed further on. The cost function for that case can be defined as follows.

Definition 6.4 *Cost function (M<N)*

$$V_N(x(k:k+N),u(k:k+M-1)) = x^T(k+N)Q_f x(k+N) + \sum_{i=k}^{k+N-1} x^T(i)Qx(i) + \sum_{i=k}^{k+M-1} u^T(i)Ru(i) \tag{6.13}$$

where N denotes the prediction horizon, M the control horizon, n the number of the states in vector x and m the number of system inputs. $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ are weights to be chosen. Q is penalizing the state deviation and R the magnitude of the input. Q_f is the final weight which penalizes the final states.

The basic structure of cost function (6.13) is the same as before for the $(N = M)$ case. The difference is that the sum is splitted up into two sums since $M < N$. The optimization problem for this cost function can be stated as follows.

Definition 6.5 *Optimization Problem (M<N)*

$$\min_{u(k:k+M-1),x(k:k+N)} V_N(x(k:k+N),u(k:k+M-1)) \tag{6.14}$$

$$\textit{subject to} \quad x(k+i) \in \mathcal{X}, \quad u(k+i) \in \mathcal{U}, \quad x(k+N) \in \mathcal{X}_f \subseteq \mathcal{X} \tag{6.15}$$

$$x(k+i+1) = Ax(k+i) + Bu(k+i) \quad \textit{for all } i \in (0,N-1) \tag{6.16}$$

$$u(k+M-1) = u(k+M) = ... = u(k+N-1) \tag{6.17}$$

The constraint sets $\mathcal{X}$, $\mathcal{X}_f$ and $\mathcal{U}$ are assumed to be affine.

Compared to Definition 6.2, Definition 6.5 introduces an additional constraint (6.17) for the $(M < N)$ case. This constraint "sets" all not optimized control signals to the value of the last optimized control signal $u(k+M-1)$.

The optimization problem can be reformulated as a quadratic programming problem in the same manner as before. Only the structure of the equality constraint, which describes the system dynamics, is different. This is because of the additional constraint (6.17). The definition of the quadratic programming problem as well as the new equality constraint can be found below.

Definition 6.6 *Quadratic Programming Problem (N<M)*

$$\min_z \frac{1}{2} z^T H z \tag{6.18}$$

$$\textit{subject to} \quad A_{eq} z = b_{eq} \tag{6.19}$$

$$A_{ineq} z \leq b_{ineq} \tag{6.20}$$

where the three equations can be written as stated in (6.10), (6.21) *and* (6.12), *respectively. The notation x(k+i)→ x(i) and u(k+i) → u(i) is used to short the equations.*

$$\underbrace{\begin{bmatrix} I & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & -B & 0 & \cdots & 0 & 0 \\ -A & I & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & -B & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & I & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & -B & 0 \\ 0 & 0 & \cdots & -A & I & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & -B \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \ddots & I & 0 & 0 & 0 & \cdots & 0 & -B \\ 0 & 0 & \cdots & 0 & 0 & \vdots & -A & I & 0 & 0 & \cdots & 0 & -B \end{bmatrix}}_{A_{eq}} \underbrace{\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(M-1) \\ x(M) \\ \vdots \\ x(N-1) \\ x(N) \\ u(0) \\ u(1) \\ \vdots \\ u(M-2) \\ u(M-1) \end{bmatrix}}_{z} = \underbrace{\begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{b_{eq}} x(0) \tag{6.21}$$

When designing an RHC with this structure ($M < N$), a compromise between computation time and stability must be found, since the computation time is less for a small control horizon but the stability suffers with a lower control horizon. How big the advantage of computation time is, compared to the ($N = M$) case, is shown in the next chapter.

6.1.3 Reduced Optimization Variables

It is possible to reduce the number of optimization variables by using a condensed version of the RHC constructions introduced before. The goal is to save computation time in order to be able to control systems with faster dynamics (small sampling time). The definition of the cost function for the condensed RHC can be stated as follows.

Definition 6.7 *Cost function* $(M \leq N)$ *(Condensed)*

$$V_N(x(k),u(k:k+M-1)) = x^T(k+N)Q_f x(k+N) + \sum_{i=k}^{k+N-1} x^T(i)Qx(i) + \sum_{i=k}^{k+M-1} u^T(i)Ru(i) \tag{6.22}$$

where N denotes the prediction horizon, M the control horizon, n the number of the states in vector x and m the number of system inputs. $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$ are weights to be chosen. Q is penalizing the state deviation and R the magnitude of the input. Q_f is the final weight which penalizes the final states.

$$V_N(x(k),u(k:k+M-1)) = x^T(k)Qx(k) + X^T\bar{Q}X + U^T\bar{R}U \tag{6.23}$$

$$\textit{where} \quad \bar{Q} = \begin{bmatrix} Q & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_f \end{bmatrix}, \bar{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix}, X = \begin{bmatrix} x(k) \\ \vdots \\ x(k+N) \end{bmatrix} \textit{ and}$$

$$U = \begin{bmatrix} u(k) \\ \vdots \\ u(k+M-1) \end{bmatrix}$$

The optimization problem for the condensed RHC structure can be defined as follows.

Definition 6.8 *Optimization Problem* $(M \leq N)$ *(Condensed)*

$$\min_{u(k:k+M-1)} V_N(x(k),u(k:k+M-1)) \tag{6.24}$$

$$\textit{subject to} \quad x(k+i) \in \mathcal{X},\ u(k+i) \in \mathcal{U}_N(x(k)) \quad for\ all\ i \in (0,N-1) \tag{6.25}$$

$$x(k+N) \in \mathcal{X}_f \subseteq \mathcal{X} \tag{6.26}$$

$$u(k+M-1) = u(k+M) = ... = u(k+N-1) \tag{6.27}$$

The constraint sets $\mathcal{X}$, $\mathcal{X}_f$ and $\mathcal{U}$ are assumed to be affine.

Since $x(k)$ is the actual measurement and therefore can not be optimized, the only considered optimization variables are the M control signals $u(k:k+M-1)$. Note that the optimization problem is not subjected to the dynamics of the system (equality constraints) anymore. These dynamics are included in the cost function by using a batched representation of the system dynamics.

$$\underbrace{\begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N) \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\psi} x(k) + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+M-1) \end{bmatrix}}_{U} \tag{6.28}$$

Using (6.28) in (6.23), the cost function can be rewritten as follows.

$$V_N(x(k),u(k:k+N-1)) = U^T \underbrace{(\Phi^T \bar{Q} \Phi + \bar{R})}_{H} U + \underbrace{2x^T(k)\psi^T \bar{Q} \phi}_{f} U + x(k)(Q + \psi^T \bar{Q} \psi) x(k) \tag{6.29}$$

Note that the last term is not part of the optimization since $x(k)$ is no optimization variable. With H and f, obtained from (6.29), the quadratic program problem for the condensed RHC with $M \leq N$ can be defined as follows.

Definition 6.9 *Quadratic Programming Problem (N<M) (Condensed)*

$$\min_z \frac{1}{2} U^T 2HU + f^T U \tag{6.30}$$

$$\textit{subject to} \quad A_{ineq} U \leq b_{ineq} \tag{6.31}$$

where the two equations can be written with the help of (6.29) and (6.32), respectively. Equation (6.32) uses the following new notation.

$$\bar{I} = \begin{bmatrix} I & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & I \end{bmatrix}, -\bar{I} = \begin{bmatrix} -I & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -I \end{bmatrix}$$

Note that the inequality constraints must be recalculated every sampling instant since they are dependent on x(0), A and B. This is because of the fact, that the state inequality constraints are included into the input inequality constraints by using (6.28).

$$\underbrace{\begin{bmatrix} \bar{I} \\ -\bar{I} \\ \Phi \\ -\Phi \end{bmatrix}}_{A_{ineq}} \underbrace{\begin{bmatrix} u(0) \\ \vdots \\ u(M-1) \end{bmatrix}}_{U} \leq \underbrace{\begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \\ -u_{min} \\ \vdots \\ -u_{min} \\ X_{max} - \psi x(0) \\ -X_{min} + \psi x(0) \end{bmatrix}}_{b_{ineq}} \tag{6.32}$$

Using a condensed version of the RHC allows to control systems with fast dynamics more efficiently. A comparison between all four different constructions of the RHC, using two different quadratic problem solvers will be given in chapter 7. Furthermore a code generation solver, which is finally used for controlling the system in real time, is compared regarding computation time.

6.2 Target & Offset Free Control

Since the system shall be controlled to a steady state target and not to the origin in general, the RHC works with deviation variables. These deviation variables express the difference relative to the desired steady state, fulfilling the condition for setpoint tracking, given in the following definition.

Proposition 6.1 *[5] Steady State Target Problem ($p \leq m$)*
It must be $p \leq m$ for the following equation to always have a solution.

$$\begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_{sp} \end{bmatrix}$$

where x_s is the steady state, u_s the steady state control signal and y_{sp} are the steady state set points.

Proposition 6.1 only holds if there are more control inputs than outputs to control ($p \leq m$). In chapter 5 the decoupled system dynamics for cable, trolley and arm dynamics are stated. For the trolley and arm dynamics it is obvious that only one input signal is available for controlling two outputs. There are mainly two possibilities to solve the steady state target problem for the condition $p > m$.

1. Define desired setpoints for both outputs and solve an optimization problem in order to find the best steady-state target.

2. Select controlled outputs z_{sp} as a subset of the measured outputs so that $p_z \leq m$.

Later on, the second possibility will be used to calculate the steady state target and so the discussion goes on with that approach. For detailed information about the first approach, consult paper [5]. Definition 6.1 can be reformulated for $p_z \leq m$ as follows.

Proposition 6.2 *[5] Steady State Target Problem ($p > m$)*
It must be $p_z \leq m$ for the following equation to always have a solution.

$$\begin{bmatrix} I - A & -B \\ C_z & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ z_{sp} \end{bmatrix}$$

where C_z is the reduced output matrix, in order to fulfill the condition $p_z \leq m$. z_{sp} are the steady state set points.

So far a perfect system model and no disturbances were assumed. For offset free control of the real system, an augmented disturbance model must be introduced. If the conditions of the following proposition are fulfilled then there is zero offset in the controlled outputs.

Proposition 6.3 *[13] Offset Free Control*
Assume that the steady state target problem is feasible and that an MPC with the following augmented model is used:

$$\begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix}$$

Further assume that $n_d = p_z$ and that B_d, C_d are chosen such that

$$rank \begin{bmatrix} I - A & -B_d \\ C_z & C_d \end{bmatrix} = n + n_d$$

Assume that the closed loop converges to a steady state with constraints inactive. Then there is zero offset in the controlled outputs, i.e.

$$z_s = z_{sp}$$

Amongst other things, the estimation of the introduced disturbance is discussed further on.

6.3 State & Disturbance Estimation

In the previous sections different RHC structures and how to achieve zero offset control was discussed. Non measurable states as well as the disturbance must be estimated by an observer using the augmented disturbance model of the systems, introduced in Proposition 6.3. Depending on, if the system dynamics are time variant or time invariant, a time-varying Kalman filter or a steady state Kalman filter can be implemented. Both structures will be discussed further on.

6.3.1 Time-Varying Kalman Filter

To simplify the notation all state space system matrices A,B,C and G are assumed to be disturbance augmented.

Consider the following plant state and measurement equations with additive Gaussian noise $w(k)$ on the input and $v(k)$ on the output. Note that matrix C is constant.

$$x(k+1) = A_k x(k) + B_k u(k) + G_k w(k) \tag{6.33}$$

$$y(k) = Cx(k) + v(k) \tag{6.34}$$

The time-varying Kalman filter is given by the following recursions, where L is the Kalman filter innovation gain and P the state estimation error covariance. Furthermore K is the process noise covariance and S the sensor noise covariance.

- Measurement update:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + L(k)[y(k) - C\hat{x}(k|k-1)] \tag{6.35}$$

$$L(k) = P(k|k-1)C^T[S(k) + CP(k|k-1)C^T]^{-1} \tag{6.36}$$

$$P(k|k) = (I - L(k)C)P(k|k-1) \tag{6.37}$$

- Time update:

$$\hat{x}(k+1|k) = A_k\hat{x}(k|k) + B_k u(k) \tag{6.38}$$

$$P(k+1|k) = A_k P(k|k) A_k^T + G_k K(k) G_k^T \tag{6.39}$$

- Assume the following initial conditions:

$$x(1|0) = 0 \tag{6.40}$$
$$P(1|0) = B_0 K B_0^T \tag{6.41}$$

The measurement and time update must be updated each sampling instant k of the MPC.

6.3.2 Steady State Kalman Filter

A steady state Kalman filter can be designed if the linear system is time-invariant. Therefore a steady state Kalman filter could be used to estimate the augmented system states for the cable system. The Kalman filter dynamics are obtained by the particularization of the general time varying dynamics for the time invariant situation.

Consider the following plant state and measurement equations with additive Gaussian noise $w(k)$ on the input and $v(k)$ on the output.

$$x(k+1) = Ax(k) + Bu(k) + Gw(k) \tag{6.42}$$
$$y(k) = Cx(k) + v(k) \tag{6.43}$$

Assume the following additional assumptions.

- $K = K^T > 0$
- $S = S^T > 0$
- The pair (A,G) is controllable
- The pair (C,A) is observable

If these conditions are fulfilled, the Kalman filter innovation gain L(k) and the prediction error covariance P(k) converge to the solution of the (filtering) algebraic Riccati equation.[5]

$$L = PC^T[CPC^T + S]^{-1} \tag{6.44}$$
$$P = APA^T - APC^T[CPC^T + R]^{-1}CPA^T + GKG^T \tag{6.45}$$

Since L and P converge to a steady-state solution, they can be calculated in the initialization of the controller and must not be calculated at each sampling instant. This saves computation time of the overall control structure.

7 Implemented MPC

In this chapter, the structure of the implemented MPC will be discussed first in section 7.1. It will be shown how the general MPC methods, discussed in the previous chapter, can be effectively used to control the tower crane. The implementation of the arm MPC Controller is shown step by step in section 7.2. It is stated which important methods must be respected when doing this. Furthermore all different possibilities of constructing an RHC will be compared regarding computation time in section 7.3. Also two different quadratic program solvers as well as one code generation solution are compared. Finally, real time control of the system will be discussed in section 7.4.

7.1 Structure

The tower crane system is controlled by three different MPC controllers. This is possible since a decoupled system was derived in chapter 5. In Figure 7.1 the global connection between these three decoupled and MPC controlled systems is shown. In section 6.2 it was discussed that for offset free control, the controlled outputs must be reduced to at least the number of control inputs. The targets of these controlled output are described by $z_{C_{sp}}$, $z_{T_{sp}}$ and $z_{A_{sp}}$. As introduced earlier in chapter 5, the system outputs are described by y_C, y_T and y_A. The cable MPC controller controls the position of the payload in z-direction. At each sampling instant the actual length of the cable is used as a parameter by the trolley MPC, which controls the trolley position in x-direction. Furthermore, the arm MPC controller uses the actual cable length as well as the trolley position at each sampling instant as parameters. The Arm MPC controller controls the position of the arm in θ-direction.

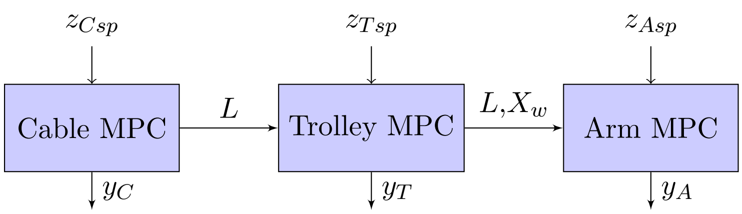

Figure 7.1: Global Control Structure

Figure 7.2 shows a deeper look into the structure of one of these MPC controlled subsystems shown in Figure 7.1.

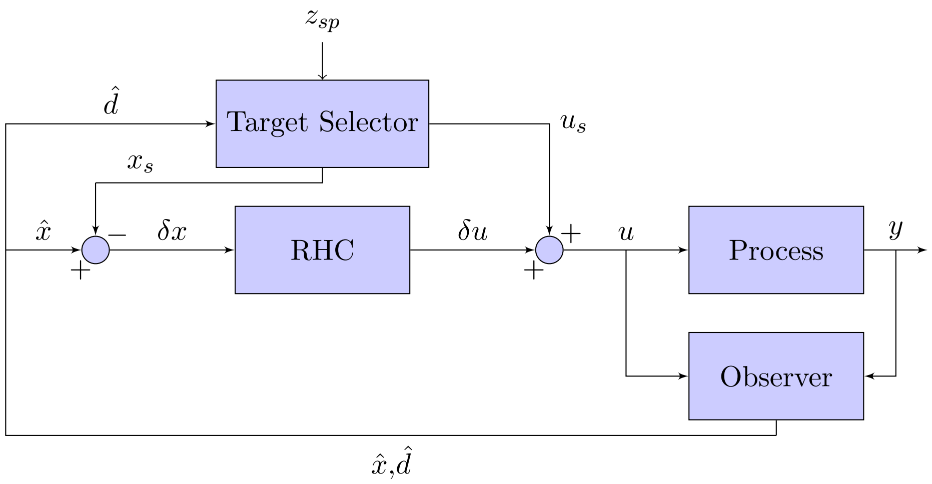

Figure 7.2: MPC Structure

<u>Receding Horizon Control</u>

The RHC can be seen as center of the MPC structure. Like stated in the previous chapter, future states are predicted by the RHC and a quadratic cost function, subjected to constraints, is minimized in order to control the states to zero by applying an optimal control signal. Since the states shall be controlled to a steady state target, the deviation variable δx must be calculated. The optimal control signal δu is the first part of the optimal control signal u which shall control δx to zero.

<u>Target Selector</u>

The Target Selector block calculates the steady state target x_s and the steady state target control signal u_s. The difference of the actual state and steady state target gives the above mentioned deviation variable δx. The process control signal u is the sum of δu and u_s.

In order to fulfill Proposition 6.2, the controlled outputs must be reduced to at least the number of available control inputs m. In general this is done by weighting the importance of the outputs and only use the important ones in the target selector calculation as controlled outputs. The trolley as well as the arm subsystem has two outputs but only one control input. But in this special case the steady state target of the second output of each subsystem is always zero. This is caused by the fact that the second outputs are the angles α and β, which must be zero when the system is in a steady state.

Offset Free Control

In the previous chapter, Proposition 6.3 was introduced. An MPC, using an augmented disturbance model, controls the system with zero offset if this proposition is satisfied. The disturbance model is used in the steady state target calculation as follows.

Definition 7.1 *Steady State Target Problem with Disturbance*

$$\begin{bmatrix} I-A & -B \\ C_z & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} B_d\hat{d}_s \\ z_{sp} - C_{z_d}\hat{d}_s \end{bmatrix}$$

where $\hat{d}_s = \hat{d}(k)$ is an estimate, obtained from the observer each sampling instant k.

For a control system obtained in this way, Proposition 6.3 holds, [13]. Using the disturbance model in the steady state target calculation, presupposes a recalculation of the steady state every sample instant. This is caused by the fact that the, assumed constant, disturbance varies in time, more precisely $\hat{d}_s = \hat{d}(k)$. However, since the trolley and arm subsystem are using parameters, which are changing over time, the system matrices A and B are time variant. Because of that, the steady state target problem must be solved every sampling instant anyway.

State & Disturbance Estimation

As stated in section 6.3, the disturbance $\hat{d}$ as well as the unmeasurable states are estimated by three different Kalman filters. For the cable subsystem a steady state Kalman filter was implemented. Since every time instant the length of the cable and the trolley position are used as parameters in the trolley and arm subsystem, these models are time variant. Therefore a time varying Kalman filter was implemented for the trolley and arm subsystem.

7.2 Implementation

In the previous section the structure of an MPC controller, used for this work, was described. The MPC consists of an RHC, a steady state target selector and a Kalman filter.. It also was stated how the three different MPC controllers are connected in order to control the whole system in an efficient way. In the sequel, it will be explained how to implement one of these three MPC controllers, exemplified by the arm MPC controller. Important methods and requirements, which must be considered when doing this, will be stated. Furthermore a Simulink-Model is presented in Figure 7.3, where the main blocks of the following steps are filled with colors. These colors are stated as well behind each step headline.

1. Initialization (Magenta & Dark Green)
   Note: The continuous system model of the arm must be discretized.
   a) Receding Horizon Controller (Condensed, M<N)
      - Define prediction horizon N and control horizon M.
      - Define the system state initial condition $x_A(0)$.
      - Define penalizing Q_A and R_A matrices for the cost function.
      - Define lower and upper input and output constraints.
        Note: The input signal is a PWM signal and no voltage signal.
      - Define the sampling time h_A of the RHC.
        Note: The sampling time must be chosen as an integer divisor of the system's time delay. If it would be chosen in another way, the discretized system matrices would change in dimension and the "kalman" command could not compute a convergent Kalman estimator. This is caused by the fact, that A would be not observable through C anymore.
   b) Augmented Disturbance Model
      - Define B_{A_d} and C_{A_d}.
        Note: The rank condition of proposition 6.3 is only fulfilled if $B_{A_d} \neq [0; 0; 0; 0]$.
   c) Steady State Target Selector
      - Define the steady state target $z_{A_{sp}}$.
   d) Kalman Filter
      - Define the process noise covariance K_A and the measurement noise covariance S_A.
        Note: The sensors used in the tower crane system are all digital encoders and no noise was observed.
      - Calculate the initial error covariance P_A.
2. Measure/Calculate Actual Outputs (Red)
   a) Calculate fitted angles α and β.
   b) Convert the measured and fitted system outputs into model values by using table 5.2.
3. Kalman Filter Measurement Update & Steady State Target Calculation (Green)
   Note: The discretized system model must be recalculated every time when either the cable length or the trolley position have changed.
   a) Use the actual cable length as well as the actual trolley position in order to calculate the Kalman filter measurement update. A new $\hat{x}_A$ and $\hat{d}_A$ is obtained.

b) Solve the steady state target problem, using the updated disturbance $\hat{d}_A$, in order to obtain the steady state target x_{A_s} and the steady state control signal u_{A_s}.

c) Calculate δx_A

4. Calculate "Deviation" Input and Output Constraints. (Cyan)
Note: In the initialization, input and output constraints were defined. But since the RHC deals with deviation variables, the "deviation constraints" must be calculated at each time instant.

a) Calculate the input constraints for the RHC by using the following equations.

$$\delta u_{A_{min}} = u_{A_{min}} - u_{A_s} \tag{7.1}$$
$$\delta u_{A_{max}} = u_{A_{max}} - u_{A_s} \tag{7.2}$$

b) Calculate the output constraints for the RHC by using the following equations.

$$\delta x_{A_{min}} = x_{A_{min}} - \hat{x}_A + \delta x_A = x_{A_{min}} - x_s \tag{7.3}$$
$$\delta x_{A_{max}} = x_{A_{max}} - \hat{x}_A + \delta x_A = x_{A_{max}} - x_s \tag{7.4}$$

5. Solve Quadratic Programming Problem (Yellow)

- By solving the quadratic programming problem with an optimizer a sequence of optimal deviation control signals δu_A are obtained.
Note: Only the first signal is used, the others are discarded.

6. Calculate Control Signal (Light Blue)

- The obtained optimal deviation control signal δu_A can be used to calculate the optimal control signal u_A with the following equation.

$$u_A = \delta u_A + u_{A_s} \tag{7.5}$$

7. Kalman Filter Time Update & Apply Control Signal (Orange)

a) Calculate the Kalman filter time update by using the control signal u_A.

b) Apply the obtained control signal u_A to the system.
Note: Use a saturation ($u_{A_{min}}$,$u_{A_{max}}$) before applying the signal to prevent the system from damage.

Varying System Model

Like mentioned in step three, the system model of the arm dynamics is varying over time and it must be recalculated and discretized at each sampling instant. But the calculation and discretization is too time consuming to be done at each sampling instant. Therefore the possible range of the cable length and the trolley position is divided into 100 small parts and the different system models are precalculated and saved in a tensor.

This tensor in the end has a dimension of 100×100 and for a specific cable length and trolley position the right model can be chosen. This tensor is saved in a data memory, colored dark green in Figure 7.3.

Varying System Model in the RHC

If one would implement the RHC 100% correct, the system dynamics would have to be varied over the prediction horizon as well. This would mean that the optimal states, calculated by the cable and trolley RHC, would have to be used to calculate future arm system matrices in order to use different A and B matrices in the equality constraints of the arm RHC. Since these calculations would need too much computation time, the system matrices A and B were assumed constant over the prediction horizon.

Kalman Filter Variables

For the measurement update as well as for the time update of the Kalman filter, variables need to be saved to be available at the next sampling instant. Therefore two data storages are needed to save the estimated state $\hat{x}$ and the error covariance P_A. These data memories are also colored dark green in the following Simulink model.

As mentioned before, Figure 7.3 shows the whole MPC implementation in a Simulink model. The MPC controllers for cable and trolley were combined in the two subsystems "Cable MPC" and "Trolley MPC", respectively. This was just done for displaying the whole model. The Cable MPC has a slightly different structure since a stationary Kalman filter was used.

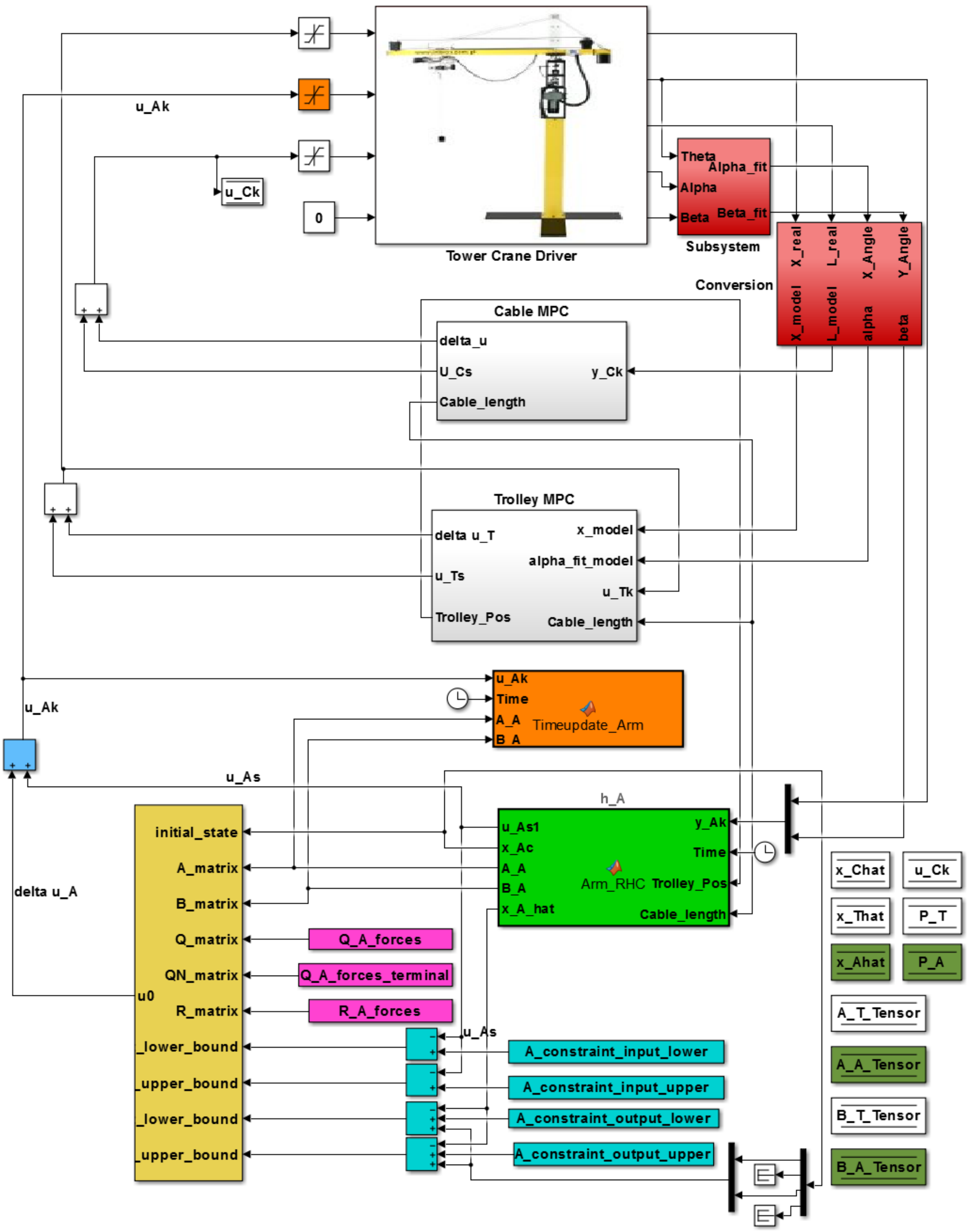

Figure 7.3: Real-Time-Control Simulink

7.3 Time Comparison

For demonstration, MPC controllers with different RHC structure were implemented. Furthermore, different solvers were used to minimize the cost function of the RHC. The used solvers are Quadprog, ECOS and FORCES Pro. Quadprog is a quadratic problem solver provided by the Optimization Toolbox in Matlab. ECOS and FORCES Pro are solvers, provided by the third-party embotech[14]. ECOS is an open-source solver for solving sparse second-order cone programs. For this purpose the quadratic programming problem is converted first, in order to solve it in a faster way than Quadprog does. FORCES Pro is a licensed code generation system for generating robust, high performance numerical optimization solvers. The next table shows time comparisons between different RHC structures and those mentioned solvers. The prediction horizon N=50 was the same for all four cases. The control horizon was either set to 50 or 6.

Table 7.1: Time Comparison of MPC Structures and Solvers

MPC Structure	Quadprog	ECOS	FORCES Pro
N=M	100%	30,9%	<0,01%
N>M	94,5%	28,4%	-
N=M (cond.)	34,9%	10,9%	-
N>M (cond.)	26,0%	5,8%	-

The largest time needed was set to 100% as a benchmark. All other percentages refer to this. It can be seen that ECOS is at least three times faster compared to Quadprog. In the case with a condensed RHC structure and prediction horizon smaller than control horizon it is nearly five times faster. Choosing different RHC structures can save up to 80% of the computation time. In the comparison of those three solvers, Forces Pro is unrivaled. The code generated solver solves the optimal programming problem in microseconds.

7.4 Real-Time Control

Not just because of the fast computation time of FORCES Pro, it was used for real-time control of the tower crane. Rather problems arise when trying to use Quadprog or ECOS for this purpose. The problem is that the Matlab code and the Simulink model must be compiled into C-Code in order to use it on the RT-DAC/USB2. Quadprog as well as ECOS are not supported by the C-Code "building" process of Matlab/Simulink. In order to be able to use either Quadrprog or ECOS, an S-Mex function must be generated. But this is not straight forward. Since FORCES Pro directly generates C-Code it can be used for real-time control. Another attractive solution would be to write a quadratic program solver. This is discussed later on.

8 Control Tasks

In this chapter position control of the trolley, arm and cable length will be demonstrated. This is done with and without penalization of the angle in the cost function. Furthermore, the reaction of the controller to a disturbance will be shown. General problems of controlling the system are discussed and approaches for solving these issues are presented. Since in "normal" control tasks the constraints on the outputs are mostly inactive a "special" control task with active constraints will be discussed. It will also be shown where the used control structure has its limits.

8.1 Cable, Trolley & Arm Control

Position control of trolley, arm and the cable can either be done by just penalizing the position or additionally penalizing the angles α and β. The output constraints are set to its maximum and are never active in the following presented control tasks.

8.1.1 Without Angle Penalization

Figure 8.1 to 8.4 are showing measurements of cable, trolley and arm control. Neither of them is penalizing the angles α and β. Mainly two problems, in controlling the crane, can be seen in Figure 8.2.

1. Overshoot of the Arm Position
   The position of the arm (θ) overshoots about 0.0165 radian. The arm motor has much faster dynamics than the other two motors of the system. This problem was solved by penalizing the control signal higher in the second measurement, see Figure 8.3.

2. Stiction - Slow Offset Free Control
   An examination of the control signal u_A shows a problem which was already addressed in the model chapter, namely the stiction. Although the control signal is not zero in between the fifth and eight second, the arm does not move. Nevertheless the offset free control, as described in the control structure chapter, works. The control signal u decreases. Since the arm moves only if the control signal is bigger than 0.2 or lower than -0.2, the offset free control works slowly.

In Figure 8.3 it can be seen that it is worth to prevent the arm from overshooting since the offset free control signal has not to pass the stiction area.

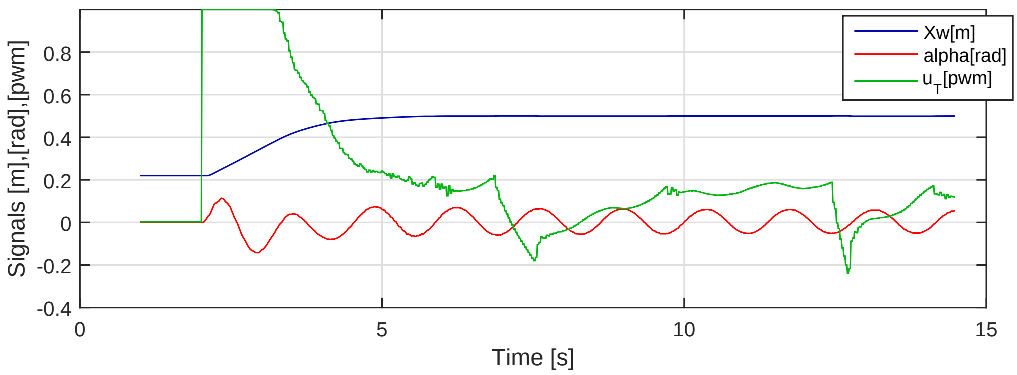

Figure 8.1: Trolley Control Without Angle Penalization

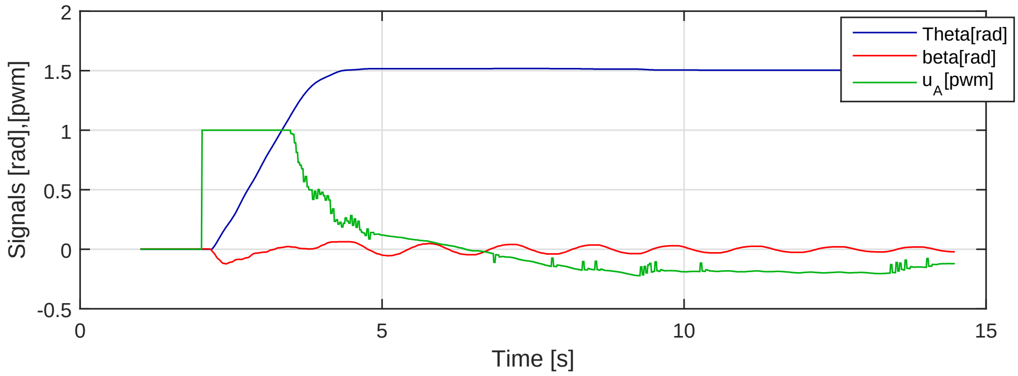

Figure 8.2: Arm Control Without Angle Penalization

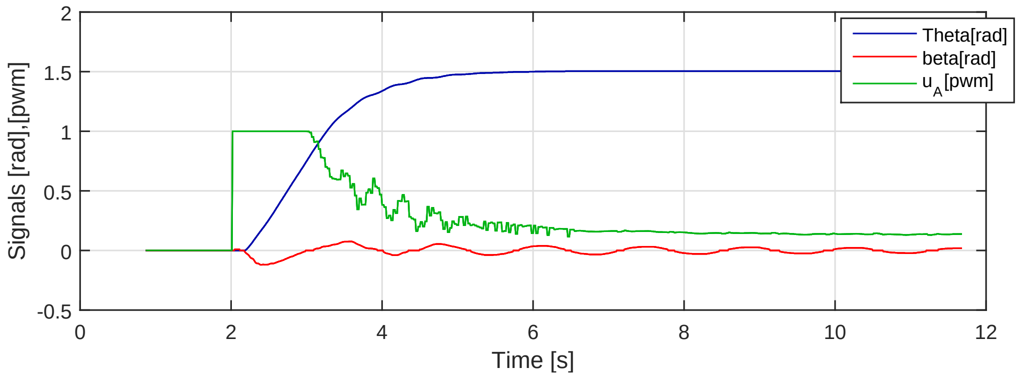

Figure 8.3: Arm Control Without Angle Penalization - Improved

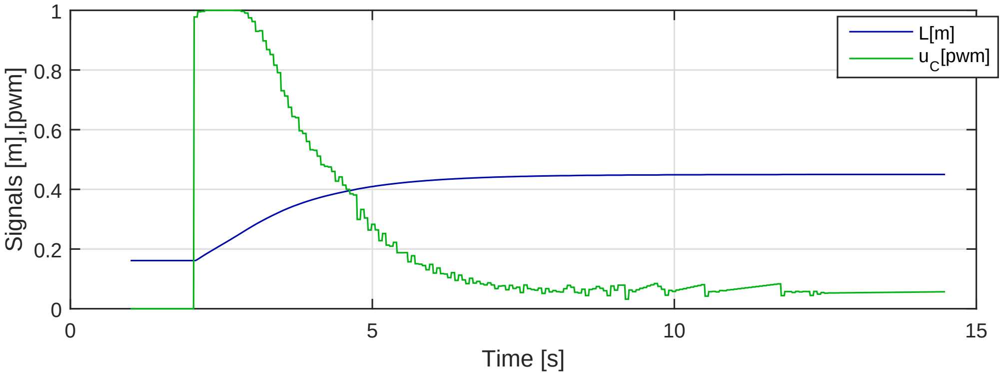

Figure 8.4: Cable Length Control

8.1.2 With Angle Penalization

Figure 8.5 to 8.7 are showing measurements of cable, trolley and arm control with additional penalization of the angles α and β. The controller controls the angles α and β to zero. The first measurements shows mainly one new problem which occurs when controlling the crane with penalization of the angles. The change of penalty, discussed in the previous subsection, is implemented in Figure 8.7, but not in the others.

In Figure 8.6 it is not clearly visible but the arm position overshoots like in the first measurement of the previous section. Nevertheless the offset gets even bigger at around 14 seconds. The arm moves into the wrong direction. This is caused by the fact that the measured angle β is constant but not zero. Since the angle is constant, the velocity is zero. This is an unnatural behavior of the system's model caused by the fact that the angles depend on the arm position. This problem was already discussed in 5.1.6. Even though the measured angles are fitted, they are not 100% correct. This results in non-zero steady state measured angles. To solve this problem the measured and fitted angles were set to zero if they are in the area between -0.008 and 0.008 radians, which results in discontinuous angle signals. This approach was implemented and tested successfully, see Figure 8.7.

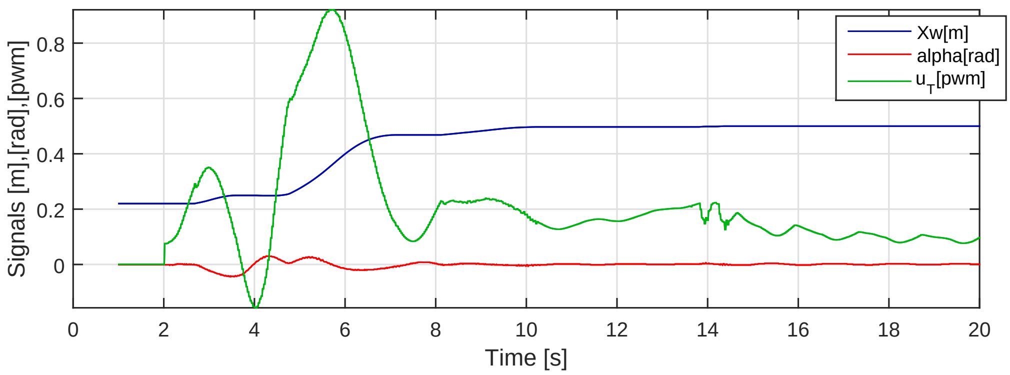

Figure 8.5: Trolley Control with Angle Penalization

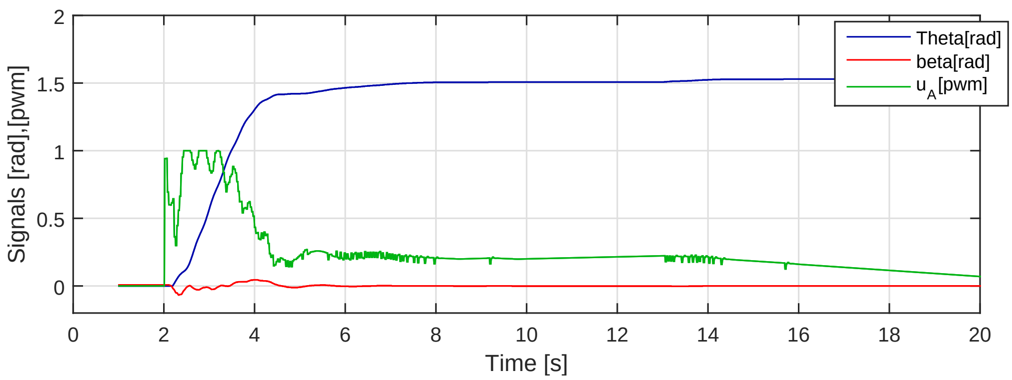

Figure 8.6: Arm Control with Angle Penalization

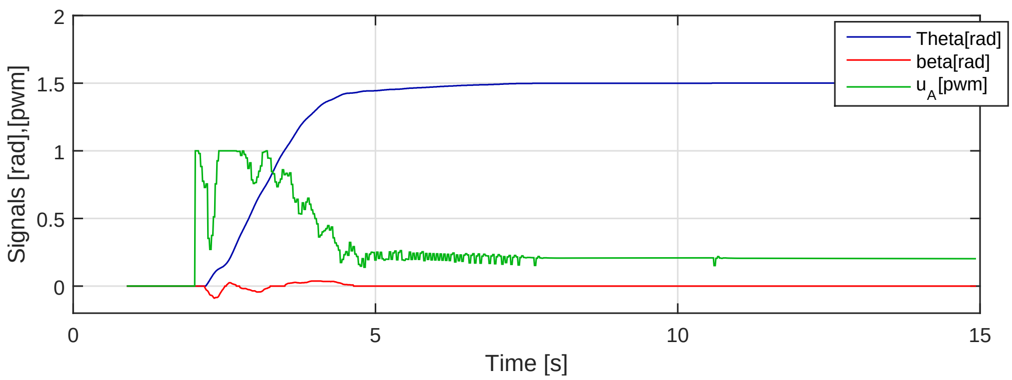

Figure 8.7: Arm Control with Angle Penalization - Improved

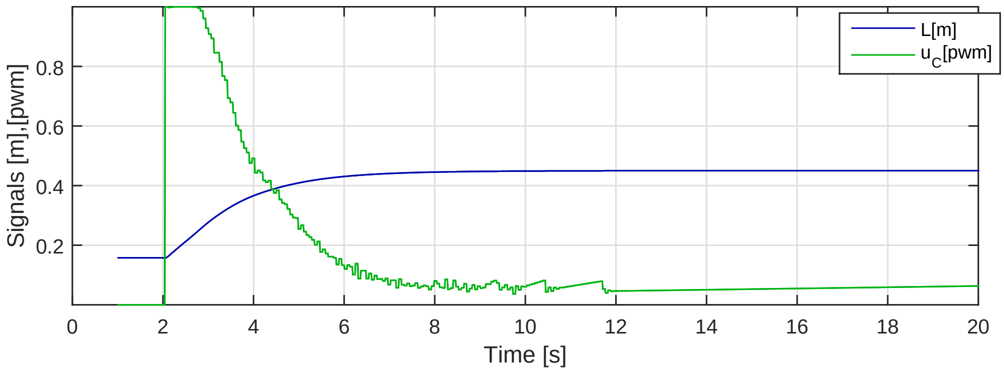

Figure 8.8: Cable Length Control

8.1.3 Disturbance Control

Figure 8.9 to 8.11 show the behavior of the controlled system when a disturbance occurs. The starting points of trolley and arm position are also the setpoints. The task is to reduce the angles α and β as fast as possible and return to the starting position.

In the first two measurement the control task is successfully accomplished for the trolley and arm subsystem, respectively. The swinging of the payload was controlled to zero and the trolley as well as the arm have returned to their starting positions. Only the slow offset free control, because of the long way through the stiction zone, is clearly visible.

In measurement 8.10 the disturbance to β is quite huge, but the swing is reduced fast. However, the disturbance to β in measurement 8.11 is smaller. It can be seen that the controller has no effect on the swinging behavior. This effect is caused again by the stiction of the system. The control signal is swinging in order to reduce the angle, but it is swinging in the area of -0.2 and 0.2, the stiction area. The control signal is not large enough to move the arm. The RHC, using the linearized model, is expecting the arm to move.

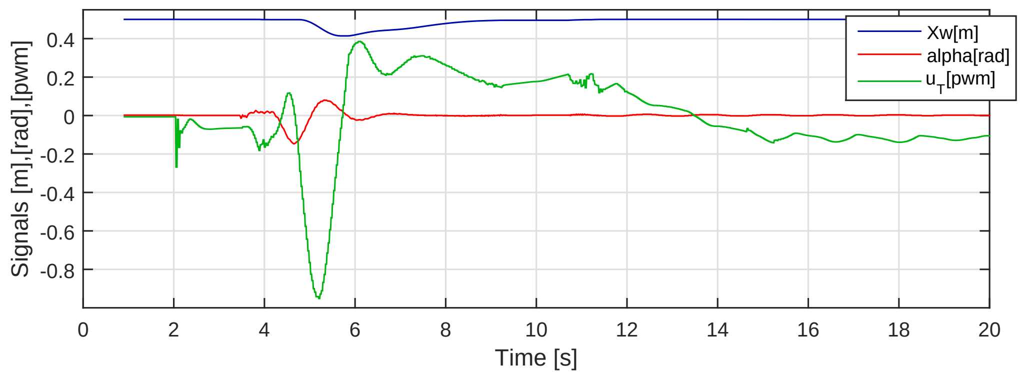

Figure 8.9: Disturbance Control Trolley with Angle Penalization

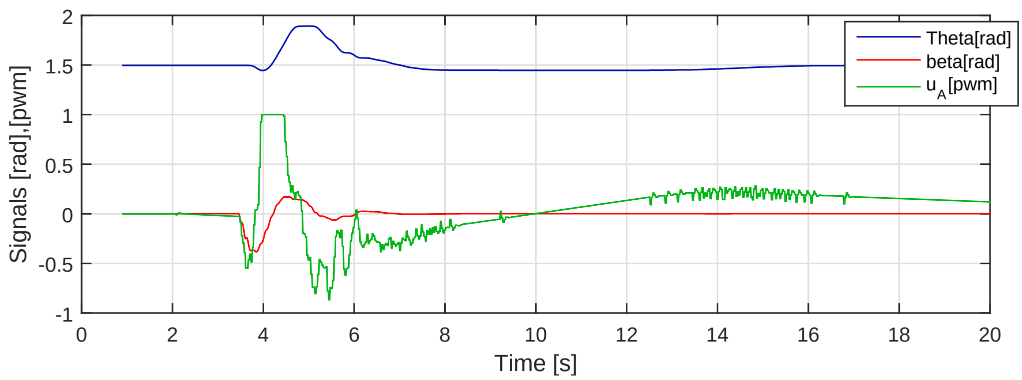

Figure 8.10: Disturbance Control Arm with Angle Penalization (1)

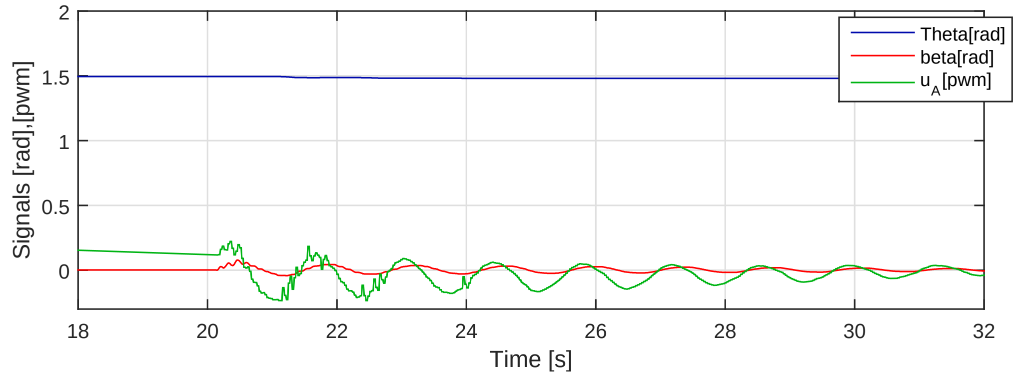

Figure 8.11: Disturbance Control Arm with Angle Penalization (2)

8.1.4 Active Constraints

In the control tasks, presented in the previous subsections, the constraints of cable, trolley and arm position were set to their maximum values and were never active. Only constraints in the control inputs were active. Furthermore, the constraints of the angles α and β were set to $\pm 2\pi$, which guarantees that these are not active while controlling as well. Hence,o an LQR controller could have been used for those control scenarios. In Figure 8.12, two measurements of the trolley subsystem are placed one above the other. Neither of them is penalizing α in the cost function. The difference is that in the second measurement (colored red), the constraints for α were set to ± 0.05. As can be seen in the figure, the controller keeps α successfully smaller than 0.05.

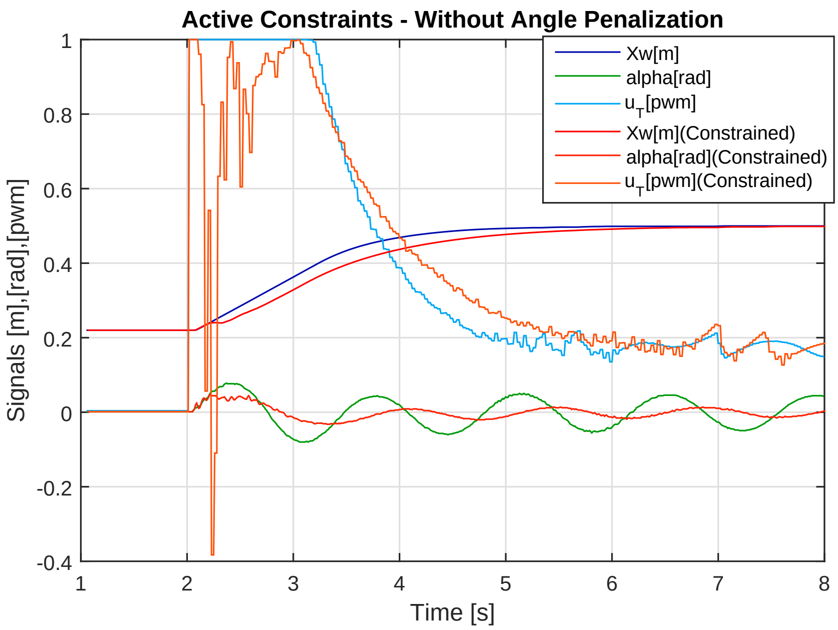

Figure 8.12: Constraints Active without Angle Penalization - Real

8.1.5 Limitations

Another control scenario of the trolley subsystem was investigated to show that the active constraints control works properly. The payload should be controlled near to a wall without hitting it. This was done by controlling the trolley position and recalculating the maximum allowed angle α at each sampling instant. The controller was not able to satisfy the constraints on α. The problem by using this approach is that over a prediction horizon of $N = 20$ the angle constraint was assumed to be constant. But clearly this is not true since the trolley keeps moving over the prediction horizon. Putting directly a constraint on a combination of the trolley position and α is not possible with FORCES Pro.

8.2 Payload Position Control

Another approach to control the tower crane system is to control the position of the payload instead of controlling cable length and trolley and arm position. For testing purposes, that approach was implemented only for the trolley subsystem. The payload position z, in x-direction, can be described by the following equation.

$$z = X_w - L\sin(\alpha) \tag{8.1}$$

Using small angle approximation, the following equation for the payload position can be stated.

$$z = c_z x = X_w - L\alpha \tag{8.2}$$

The matrix $c_z = [1 \; 0 \; -L \; 0]$ is designed to fulfill (8.2). If the steady state position of the payload is described by z_{sp}, the weighting matrix Q_T of the cost function can be obtained in the following way.

$$\begin{aligned}
(z - z_{sp})^2 &= (c_z x - z_{sp})^2 && (8.3)\\
&= (c_z(x - x_{sp}))^2 && (8.4)\\
&= (x - x_{sp})^T c_z^T c_z (x - x_{sp}) && (8.5)\\
&= (x - x_{sp})^T Q_T (x - x_{sp}) && (8.6)
\end{aligned}$$

Here, any x_{sp}, for which $z_{sp} = c_z x_{sp}$, can be used. In particular x_{sp} can be chosen $x_{sp} = x_s$, where x_s is the steady state target with $c_z x_s = z_{sp}$. The matrix $Q_T = c_z^T c_z$, shown in the following equation, is only positive semidefinite and only deviations in the final position are penalized.

$$Q_T = \begin{bmatrix} 1 & 0 & -L & 0 \\ 0 & 0 & 0 & 0 \\ -L & 0 & -L^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{8.7}$$

Implementing the described penalty approach brought no advantages compared to the earlier described control method. The problem is that it is not possible to put constraints directly on the payload position with FORCES Pro. Only the trolley position

X_w and the angle α can be penalized. Introducing a new state space model with changed variables could help. However, using Quadprog or ECOS could bring better results in simulations since the inequality constraints can be designed with more freedom. But as mentioned, these solvers are to slow for real time control.

Another idea would be to introduce two new states, namely the payload position x_{11} and its velocity x_{12} and a new output y_6 .

$$z = x_{11} = X_w - L\alpha = x_3 - Lx_5 \tag{8.8}$$

$$\dot{x}_{11} = x_{12} = \dot{x}_3 - L\dot{x}_5 = x_4 - Lx_6 \tag{8.9}$$

$$\dot{x}_{12} = \dot{x}_4 - L\dot{x}_6 = 9.81x_5 + 0.0055x_6 \tag{8.10}$$

$$y_6 = x_{11} \tag{8.11}$$

The trolley model 5.23 can be rewritten by using (8.9) to (8.11), in the following way.

Trolley-Subsystem:

$$\boxed{\begin{gathered}\underbrace{\begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix}}_{\dot{x}_T} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -107.83 & -4.48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{107.83}{L} & -\frac{14.29}{L} & -\frac{0.0055}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 9.81 & 0.0055 & 0 & 0 \end{bmatrix}}_{A_T} \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_{11} \\ x_{12} \end{bmatrix}}_{x_T} + \underbrace{\begin{bmatrix} 0 \\ 20.09 \\ 0 \\ \frac{20.09}{L} \\ 0 \\ 0 \end{bmatrix}}_{B_T} u_1 \\ \text{Delay of Input } u_1\text{: } 0.03 \text{ s} \\ \underbrace{\begin{bmatrix} y_6 \end{bmatrix}}_{\dot{y}_T} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{C_T} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_{11} \\ x_{12} \end{bmatrix}\end{gathered}} \tag{8.12}$$

Implementing model 8.12 would give the possibility to put weights on the final payload position as well as putting constraints to it. Using that approach would be possible by using FORCES Pro.

9 Assignment Guidance

In this chapter a rough proposal for an assignment guidance in the MPC course (SSY280) at the Chalmers University of Technology will be presented. Therefore, the results of the thesis will be reclaimed. Since the assignment guidance must be adjusted to the progress of the lecture it was decided that the detailed planing for the assignment must be done by the course/assignment supervisor. The supervisor will be provided with four fully operational and well commended Matlab/Simulink files.

1. Closed loop simulation with states and control signals as optimization variables. (N>=M) (Quadprog & ECOS)

2. Closed loop simulation with condensed RHC structure. (N>=M) (Quadprog & ECOS)

3. Closed loop simulation, using the code generation system FORCES Pro. (N=M) (Forces Pro)

4. Closed loop real-time control, using the code generation system FORCES Pro. (N=M) (Forces Pro)

Using this four files, the supervisor can built skeleton code for the course assignments.

9.1 Objectives

The purpose of the assignments is to control the 3D tower crane using MPC. The focus of the first part of the assignment is to get familiar with the details of a simple MPC algorithm by actually writing the Matlab code and validate the code by simulating the crane in closed loop. In the second part the objective is to get some insight and experience in obtaining zero offset when applying MPC control to a MIMO plant. The code produced in the first part must be extended. The requirement here is to achieve zero offset for the controlled outputs in the presence of constant but unknown disturbances. In the last part the code generation system FORCES Pro is used to control the tower crane in real-time. Therefore some modifications in the code must be done by the students.

9.2 Model

The tower crane system must be introduced to the students. They should be provided with the nonlinear state space representation of the full system. A first task could be to think about the global control structure, namely dividing the system into three subsystems in order to be able to use linear MPC techniques. Already at the beginning or later on the differences of the real system and the model must be discussed in the guidance. The relations between measured states and model states are important for controlling the system in real-time.

9.3 Simulations - Part 1

For this part of the assignment, the guidance should refresh MPC control theory from the lectures.

The students should implement a simple MPC structure with stats and control signals as optimization variables. They should think about controlled outputs in the steady state target selector. Namely, that the steady state targets for the angles α and β are always zero without calculating them. Using stationary and time-varying Kalman filters must be discussed.

The augmented model should not be used at this point. This gives the possibility to make the remaining steady state error visible. Why is there a remaining steady state error?

A comparison regarding computation time of different solvers and the possibility of using a smaller control horizon than prediction horizon should be done. Which advantages and disadvantages has a different RHC structure?

Furthermore let the students compare the MPC without constraints with LQ control techniques. Compare it to the constrained MPC case. How to calculate the constraints? They must be recalculated every sampling instant.

Overall for this part of the assignment a Matlab/Simulink skeleton of the first simulation file in the list above is used. The students should implement code for the generation of the different matrices used in the quadratic optimization. They have to think about witch matrices can be constructed in the initialization and which have to be recalculated every sampling instant because of the dependency of $x(k)$ or A and B.

9.4 Simulations - Part 2

For this part of the assignment, the guidance should refresh especially the use of an augmented disturbance model in the MPC structure. Witch condition/proposition must be fulfilled to reach zero offset control?

The students should change their code for using the condensed RHC structure. Which matrices are not needed anymore? Which ones must now be calculated every sampling instant? Are their still matrices which can be calculated in the initialization? A time comparison of the new RHC with different solvers should be done.

How to calculate the constraints? They must be recalculated every sampling instant, but in another way than with the earlier RHC structure.

The code must be changed for using the augmented disturbance model in the Kalman filter and the steady state target selector. For which matrices C_d and B_d does the system reaches zero offset control? Why? Validate it with simulations.

Overall for this part of the assignment a Matlab/Simulink skeleton of the second simulation file in the list above is used.

9.5 Real-Time Control

The students shall simulate the closed loop tower crane system by the use of FORCES Pro. That is not much work since the code generation software does all the work. No generation of optimization matrices is required. If the simulation works probably the fourth file can be used to control the real tower crane in real time.

Overall for this part of the assignment a Matlab/Simulink skeleton of the third and fourth simulation files in the list above is used. If the simulation works probably the students can transfer their settings to the real-time control file and tune the weighting parameters with the real system.

10 Summary and Conclusion

In this chapter a short summary about the thesis, results and conclusions will be given. Important findings and problems will be stated. Some ideas for further studies with the Tower Crane system will be given. Furthermore, the work will be compared to other studies and papers.

First of all it should be mentioned that all objectives of the thesis are fulfilled. Zero offset control of an educational system, by implementing an MPC structure, was achieved successfully. Advantages and disadvantages of different RHC structures were discussed. A time comparison of different solvers and code generation systems, regarding computation time, was done. The acquired knowledge was used to elaborate a laboratory guidance for the MPC-course (SSY280) at Chalmers University of Technology. This guidance shall give the students an impression of how to use an MPC structure for controlling a system.

After a short introduction and a discussion about the used methods, an educational system was selected successfully, in chapter 3. Table 3.1 shows the cost-utility analysis which was used to rank systems from different companies. In the end, the third ranked Tower Crane of the company INTECO was chosen. This was mainly done because of the company's location near to Sweden, its references and and the price of the system. The system was delivered in time and it is made of high quality. All expected system functions are working properly and the commissioning was quite easy.

A nonlinear model of the system, including actuators and friction terms, was derived successfully in chapter 5. The kinetic and potential energy of the system was used in a Lagrangian approach. Actuator terms were evaluated through physical properties. Some of the actuator parameters could not be obtained from data sheets. Therefore first estimates were taken from the studies [9] and [10] and were improved by applying step-response tests. The mathematically obtained nonlinear system model shows the same structure and parameter dimensions as the practically obtained model in paper [10].

Amongst other assumptions, small angle approximation was applied to the nonlinear model, in order to linearize it, in chapter 5. Two remaining, verified as not significant important, nonlinear coupling terms were neglected. Furthermore, the model was decoupled into subsystems, in order to finally get three linear state space representations of the models. These models are using inputs from the other subsystems as parameters. In order to speed up the real-time control, system models, for different input parameters, were saved in a look-up table. Constructing and discretization of a system model in Matlab would not be possible in real-time. The nonlinear as well as the linear decoupled model was verified successfully with step-response tests and applying a triangular input signal, in chapter 5. Small deviations of the linear model, compared to the real system, have not influenced the control quality noticeably.

Advantages and disadvantages of different RHC structures were stated in chapter 6. All structures were used in simulations successfully. Cost-functions, inequality as well as equality constraints structures were explained in detail. This detailed explanation was mainly done for the students, to use them as additional lecture notes.

The quadratic program solvers Quadprog and ECOS as well as the code generation system FORCES Pro were compared regarding computation time of different RHC structures, in chapter 7. Using different RHC structures and ECOS instead of Quadprog can speed up the optimization about 94%. Nevertheless, the quadratic program solvers are still to slow for using them for real-time control. Therefore the code generation system FORCES Pro was used. Further studies could contain the design of an own quadratic problem solver. This would open new possibilities for the students in the MPC course. Not to use a "black box" solver would provide a better understanding of how optimization code works. Furthermore, it would be possible to construct inequality constraints with more freedom than it is possible with FORCES Pro. This would solve the problems, mentioned in section 8.2.

A steady state target selector was used in the MPC structure, described in chapter 7. Thus the RHC works with deviation variables. By introducing an augmented disturbance model to the steady state target selector, zero offset control was achieved. Not measurable states as well as the disturbance were estimated by a time varying Kalman filter. Different control tasks were stated in chapter 8. Reliable real-time control of the system was achieved successfully. Further work could investigate the stiction behavior problem of the the trolley and the arm, mentioned in chapter 8. Some kind of stiction compensator must be introduced to the system. Maybe it would be just sufficient to add an offset to the control signal u, unequal zero, if the controlled system part is not moving, namely the velocity is zero. This would maybe bring the controlled system part into motion and afterwords the calculated, optimal, control signal could be applied.

Two studies, which have used the same tower crane system from INTECO, were referenced in this work. This thesis differs manly in the educational point of view from them. Special attention was paid to discuss different RHC structures as well as on explaining how to implement an MPC practically. Furthermore, Faisal Atlaf derived the system's model experimentally, whereas a mathematical approach was used in this thesis. Compared to the studies [9], zero offset control was achieved with methods mentioned earlier. Nevertheless, [9], focuses more on guaranteed stability of the MPC.

The work with the tower crane was fun. I have learned a lot more about control theory in general, but especially about MPC control. It was nice to see, how using different RHC structures and solvers can make the difference in real-time control. This shows that computation time is still worth to think about when implementing an MPC controller. Academical discussions with my supervisor and professor, Bo Egardt, have always opened new points of view to me.

Bibliography

[1] INTECO, Tower Crane User's Manual, 9.6 Edition.
URL http://www.inteco.com.pl/

[2] J. Maciejowski, Predictive Control with Constraints, Prentice Hall, 2001.

[3] S. Joe Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (7) (2003) 733–764.
URL http://www.sciencedirect.com/science/article/pii/S0967066102001867

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814.
URL http://www.sciencedirect.com/science/article/pii/S0005109899002149

[5] B. Egardt, Model predictive control, lecture notes (Chalmers University of Technology) (January 2015).

[6] Inteco, http://www.inteco.com.pl/, accessed: 2015-02-22.

[7] INTECO, RT-DAC/USB2, 1.0.3 Edition.
URL http://www.inteco.com.pl/wp-content/uploads/2014/05/RTDAC_USB2.pdf

[8] A. A. Al-Mousa, Control of rotary cranes using fuzzy logic and time-delayed position feedback control, Master's thesis, Virginia Polytechnic Institute and State University (2000).
URL http://scholar.lib.vt.edu/theses/available/etd-12092000-141829/unrestricted/thesis.pdf

[9] F. K. Sandor Iies, Jadranko Matusko, Real-time predictive control of 3D tower crane, Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium 23 (23) (2014) 224–230.
URL http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6864615&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6864615

[10] F. Altaf, Modeling and event-triggered control of multiple 3d tower cranes over wsns, Master's thesis, Kungliga Tekniska Högskolan (2010).
URL http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A570072&dswid=-2448

[11] H. M. Omar, Control of gantry and tower cranes, Ph.D. thesis, Virginia Polytechnic Institute and State University (2003).
URL http://scholar.lib.vt.edu/theses/available/etd-01262003-204800/unrestricted/Thesis.pdf

[12] G. C. Goodwin, M. M. Seron, J. A. de Doná, Constrained Control and Estimation, Springer, 2005.

[13] D. Q. Mayne, J. B. Rawlings, Model Predictive Control: Theory and Design, Nob Hill Publishing, LLC, 2009.

[14] embotech.
URL "https://www.embotech.com"