TALLINNA TEHNIKAÜLIKOOL

Elektrotehnika instituut

ATV70LT

Andrei Aksjonov

3D KRAANA JUHTIMISSÜSTEEM

Magistritöö

Instituudi direktor .... prof. Tõnu Lehtla

Juhendaja .... prof. Valery Vodovozov

Kaasjuhendaja .... dots. Eduard Petlenkov

Lõpetaja .... Andrei Aksjonov

Tallinn 2015

AUTORIDEKLARATSIOON

Kinnitan, et käesolev lõputöö on minu iseseisva töö tulemus. Kõik selle koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Varem ei ole selle alusel kutse- ega teaduskraadi ega inseneridiplomit taotletud. Töö on koostatud litsentseeritud tarkvara abil.

Tallinn, 26.05.2015.a. .... Andrei Aksjonov

ATV70LT

3D kraana juhtimissüsteem

Andrei Aksjonov, üliõpilaskood 132155AAAMM, mai 2015. – 78 lk.

TALLINNA TEHNIKAÜLIKOOL

Energeetikateaduskond

Elektrotehnika instituut, elektriajamite ja elektrivarustuse õppetool

Töö juhendaja: professor Valery Vodovozov

Töö kaasjuhendaja: dotsent Eduard Petlenkov

Võtmesõnad: Sildkraana, elektriajam, CNC, modelleerimine, süsteemijuhtimine, programmeerimine, simuleerimine, state-space

Referaat:

Lõputöö on 78 lehel, sisaldab 4 tabelit, 12 illustratsiooni, 24 joonist, 8 graafikut ning 4 lisa. Lõputöö on kirjutatud inglise keeles.

Lõputöö põhiülesandeks on 3D kraana matemaatiline modelleerimine ning mudeli simuleerimine MATLAB$^{®}$/Simulink$^{®}$ keskkonnas. Modelleeritud mudelit võrreldakse olemasoleva Inteco$^{®}$ mudeliga, mis põhineb kooderitel. Katsed ja simuleerimine on läbi viidud Tallinna Tehnikaülikooli Infotehnoloogia teaduskonnas asuvas automaatika laboratooriumis. 3D kraana süsteemi valmistaja on Poola ettevõtte Inteco$^{®}$.

3D kraana matemaatiliselt kirjeldatud mudel peab asendama kooderitest sõltuva mudeli. Kooderid teevad süsteemi kallimaks. Lisaks sellele nõuvada kooderid kalibreerimist ning püsivat hooldust. Samuti välistab automatiseeritud 3D kraana süsteemist operaatori. Seega vähendatakse õnnetuste riski ning täiustatakse 3D kraana positsioneerimise täpsust. Matemaatiliseks modelleerimiseks on teada 3D kraana komponentide massid.

Lõputöö tulemusena projekteeriti 3D kraana juhtimissüsteem nii, et matemaatilise mudeli käitumine oleks võimalikult sarnane süsteemi kooderite väljund signaalidega. Juhtimissüsteem omab kolme sisendit: telgedele x, y, ja z mõjuvad jõud. Süsteemi väljundis on kolm positsiooni kolmemõõtmelises ruumis x, y ja z ning kaks kraanal oleva koormuse kaldenurka. 3D kraana positsioon sõltub mõjuvatest jõududest ning on piiratud kraana konstruktsiooniga. Juhtimissüsteemi projekteerimiseks kasutati state-space meetodit.

ATV70LT

Моделирование и управление 3Д краном

Андрей Аксёнов**,** код студента 132155AAAMM, май 2015. – 78 стр.

ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ∗ Энергетический факультет

Электротехнический институт

Кафедра электротехники

Руководитель работы: профессор Валерий Водовозов

Соруководитель работы: доцент Эдуард Петленков

Ключевые слова: электропривод, контроллер, программирование, кран, моделирование, CNC, пространство состояний.

Реферат:

Объём диссертации составляет 78 страниц, содержит 4 таблицы, 12 иллюстраций, 8 графиков, 24 рисунка и 4 приложения. Диссертация написана на английском языке.

Главная задача, поставленная в диссертации - это математическое моделирование трехмерного крана и проектирование его управления. Полученная модель сравнивается с известными, в частности, с моделью, предоставленной производителем Inteco® и основанной на датчиках. Имеющаяся модель предоставлена производителем Inteco® и основана на кодерах. Система трехмерного крана предоставлена кафедрой информационных технологий Таллиннского технического университета.

Благодаря исключению датчиков, разработанная математическая модель способствует удешевлению конструкции благодаря исключению расходов на сервис, покупку датчиков и исключению оператора. Таким образом, снижается риск несчастных случаев и увеличивается точность позиционирования.

Система управляется по трем входным усилиям, трем позициям в трехмерном пространстве и двум углам раскачивающегося подвесного груза. Координаты крана определены в соответствии со входными усилиями и ограничены размером конструкции крана. Для реализации математической модели используется метод пространства состояний.

ATV70LT

The Modelling and Control of 3D Crane

Andrei Aksjonov, student code 132155AAAMM, May 2015. – 78 pages

TALLINN UNIVERSITY OF TECHNOLOGY ∗ Faculty of Power Engineering

Department of Electrical Engineering

Chair of Electrical Drives and Electricity Supply

Supervisor of the work: professor Valery Vodovozov

Co-supervisor of the work: assistant professor Eduard Petlenkov

Key words: electrical drive, controller, programming, crane, simulation, CNC, state-space

Summary:

The thesis is written on 78 pages, contains 4 tables, 12 illustrations, 8 graphics, 24 pictures and 4 annexes. The thesis is written in English language.

The main goal of the thesis is to derive the mathematical model of three-dimensional crane, to build the control system, to simulate and compare the obtained results with the model provided by the Inteco® 3DCrane manufacturer. The physical model of 3DCrane is provided by the Tallinn University of Technology Faculty of Informational Technology.

The derived model must exclude the encoders from the system. It will therefore decrease the cost of the services and the system itself. Moreover, the control system design may also be useful in industrial overhead cranes. It may replace the operator from the system, minimizing the human interface. Thus, it may reduce incidents and accuracy loss. Masses of the equipment of 3DCrane for modelling are provided by the manufacturer.

The study object is constructed by the Polish company Inteco®. It is known that the 3DCrane operates by three forces inputs. The signals are sent to the DC motors from PC. The system has also the feedback. The encoders send the position of the crane and angles of the payload back to the user. Moreover the positions are limited by the 3DCrane frame dimension.

The derived model must follow the position control. The idea is to project and tune the system output according to the input forces to each axis and, in general, to replace the same model that is based on the encoders with the same mathematically derived model. The control system method implemented to solve the problem is the state-space method.

TABLE OF CONTENTS

Table of contents--- 6

Magistritöö ülesanne (in estonian) --- 8

Acknowledgments--- 9

Preface--10

1. Introduction--11

2. Control systems ---15

2.1. Control system definition ---15

2.1.1. Open-loop system--16

2.1.2. Closed-loop system --16

2.2. Mathematical modelling---17

2.2.1. Simulation--19

2.2.2. MATLAB® and Simulink® --19

2.3. Linear systems ---20

2.4. Nonlinear systems---21

2.5. Nonlinear Multivariable (MIMO) systems --22

2.6. Chapter summary---23

3. Object description ---24

3.1. Industrial 3D crane--24

3.2. General description of the studied object ---25

3.3. Object overview ---30

3.4. Testing the 3DCrane ---32

3.5. Chapter summary--34

4. The model development and simulation --35

4.1. Dynamic model of the 3DCrane --35

4.1.1. Position of the payload equations---36

4.1.2. Dynamic equations--37

4.1.3. Approximation---38

4.2. Simulation of the dynamic model---39

4.3. State-space representation---41

4.4. State-space representation simulation ---45

4.5. Chapter summary--47

5. Models comparison--48

5.1. State-space representation and Inteco® model comparison ----------------------------------48

5.2. Chapter summary--52

6. The full model development and simulation ---53

6.1. Dynamic model of the 3DCrane using the Lagrangian approach ---------------------------53

6.2. Model simulation and comparison with the real 3DCrane system --------------------------56

6.3. Tuning the system--58

6.4. Chapter summary--60

7. Conclusion--61

References --65

Annexes ---67

Annex 1. MATLAB script for the simplified model of the cart ----------------------------------68

Annex 2. MATLAB script for the state-space representation-------------------------------------69

Annex 3. Full 3DCrane full dynamic model derivation--75

Annex 4. MATLAB script for the state-space representation of the full dynamic model------78

TALLINNA TEHNIKAÜLIKOOL

Elektrotehnika instituut

KOOSKÕLASTATUD

Prof. T. Lehtla...............................

.... 2015

MAGISTRITÖÖ ÜLESANNE

Andrei Aksjonov, üliõpilaskood 132155AAAMM

Magistritöö teema: 3D kraana juhtimissüsteem

Ülesanne: Modelleerida, projekteerida ning simuleerida 3D kraana juhtimissüsteemi

Lähteandmed:

1. 3D kraana kasutusjuhend
2. 3D kraana konstruktsiooni ning komponentide füüsilised parameetrid
3. Kooderitel põhinev 3D kraana Simulink® mudel
4. Koondplaan
5. Eeldatud mudeli tulemus

Lahendamisele kuuluvate probleemide loetelu:

Modelleerida Tallinna Tehnika Ülikooli Automaatika Instituudi 3D kraana juhtimissüsteemi. 3D kraana maketti valmistaja on Poola ettevõtte Inteco®.

1. Koostada 3D kraana dünaamiline mudel.
2. Koostada plokkskeem kasutades modelleerimiskeskkonda MATLAB® / Simulink®.
3. Modelleeritud mudeli simuleerida ning võrrelda olemas oleva Inteco® mudeliga, mis põhineb kooderitel.
4. Tulemused võrrelda ja analüüsida ning vajadusel juhtimissüsteem täiendada ja seadistada.
5. Järeldus ja uurimistöö tulevikuvisioon.

Magistritöö esitada inglise keeles kahes eksemplaris koos eesti- ja kahe võõrkeelse referaadiga hiljemalt 8.06.2015.

Juhendaja: Ülesande vastu võtnud:

Professor V. Vodovozov .... Üliõpilane A. Aksjonov ....

ACKNOWLEDGMENTS

The author would like to acknowledge the contributions of both universities Tallinn University of Technology (TUT) and Politecnico di Milano (Polimi) administrative and teaching personal, where the author has been studying during his Master of Science program. The author is grateful for being able to study in both universities. The education makes the dreams come true.

The author would like to thank all the professors and stuff of TUT for help, consulting and inspiring him to work hard and move forward, as well as people, who participated in review and controlling the results. Special thanks for their priceless time: Valery Vodovozov, Eduard Petlenkov, Zoja Raud, Aleksei Tepljakov, Kristina Vassiljeva, Tõnu Lehtla and Raivo Teemets.

My sincere appreciation is extended to Vertex Estonia AS company team, where I am currently employed: Aleksander Rulkov, Andri Märdikes, Artur Prees, Kristjan Kebja, Oleg Movko, Mikk Pärna, Tamur Mikk, Andrei Rõndin, Konstantin Zalutski, and Karin Ahven. Thank You for Your support and making my working hours flexible.

Next, I would like to acknowledge professors and engineers of TUT Faculty of Power Engineering for consulting the thesis topic. They are: Elmo Pettai, Margus Müür, Madis Lehtla. Additionally, special thanks to the TUT Faculty of Information Technology for providing the laboratory and equipment to study. Also thank You for letting me work until late night.

Finally, I want to express my infinite appreciation to my family and friends across the globe. The people, who always supporting me and care about the results in every activity I attend. The people, who motivates me the most.

PREFACE

The thesis is written in co-operation with two departments of Tallinn University of Technology[1] (TUT). The main supervisor is professor Valery Vodovozov, Tallinn University of Technology, Faculty of Power Engineering, Department of Electrical Engineering. The second supervisor is associate professor Eduard Petlenkov, Tallinn University of Technology, Faculty of Information Technology, Department of Compute, Chair of Automatic Control and System Analysis. The study equipment 3DCrane was provided by the Alpha Control Lab[2] in Tallinn University of Technology, Faculty of Informational Technology. The knowledge to complete the task was mainly used from the book references listed at the end of the thesis. Moreover, many doctoral students of Tallinn University of Technology gave the advises in solving the problems related to the thesis: assistant Zoja Raud, associate professor Kristina Vassiljeva, engineer Aleksei Tepljakov and many others, who are individually thanked in the acknowledgments.

The thesis topic was proposed by the author whose, main goal was to study the control systems engineering independently and more deeply. It has been therefore suggested by the professor Vodovozov to design the control system, which can exclude the encoders from the system. All the explanations introduced in thesis are enough to repeat the experiments independently.

The thesis is logically divided into chapters. Each chapter is then divided into clearly numbered and labelled sections. Sections are containing explanations, examples and author's conclusions based on the experiments and the knowledge gain. Each chapter completes with a brief summary. The thesis starts with the introduction stating the problem stressed in the thesis and therefore summarized with the conclusion. The computer experiments are held using the computer-aided software MATLAB®[3] and simulation software Simulink®[4]. The explanations are illustrated with Figures. The complete dynamic model derivation is placed in annexes. Moreover, all the MATLAB® scripts used for the simulation and computations are also in annexes. The annexes are listed at the end of the thesis.

[1] www.ttu.ee

[2] Alpha Control Lab (A-lab) – research laboratory within the Department of Computer Control, Tallinn University of Technology.

[3] MATLAB – registered trademark of the MathWorks, Inc.

[4] Simulink – registered trademark of the MathWorks, Inc.

1. INTRODUCTION

The topic of the thesis has been selected according to the author's interest in control engineering. Many courses have been passed related to this field. Most of the courses were passed during the Erasmus exchange study in Politecnico di Milano, Italy. The enrolled program was Automation Engineering. Thus most of the courses contents were related to the mathematical modelling and system control engineering. Moreover, it has been suggested the Alpha Control Lab in the TUT Faculty of Information Technology. The laboratory is perfectly equipped for the control engineering study.

The author, within his working experience, has dealt with few industrial three-dimensional cranes. Nowadays, cranes are operated mostly by human. This makes the system dangerous, because of the numerous accidents and accuracy loss. It has been therefore proposed to build the control system to replace the encoder in the real system by the mathematical model. It will make the system cheaper and more precise due to the fact that encoders provide and accumulate errors. Automation shall also remove the human factor from the system, what makes it safer.

The topicality of the thesis is due to the widely usage of 3D cranes in industry. The problem stressed in the present thesis is not new. Many researches have been made in order to find the best, cheapest and optimal solution to control 3D cranes. The AutoPilot control algorithm can remove the operator from tasks require repetitive motions, what is the cause of fatigue depressions and unwanted casualties.

Thus the main tasks intended to be solved in the present thesis are as follows:

- to derive the mathematical model for the 3DCrane[5]
- to find the optimal control system solution for the system based on mathematical model
- to simulate the model using Simulink environment and to compare it to the existing model provided by the crane manufacturer
- tune the derived model as close as possible to the existing model in order to be able to replace the encoder-based model with the mathematical model
- to use the optimal solution, physical laws and approximations if necessary for deriving the mathematical model

[5] "3DCrane" fused word here and in the whole thesis is related to the exact equipment.

- to use the suitable and cheapest control logic.

In short, provide the sensorless control logic and in the case of fail, to propose another solution for the future research.

Researchers from different universities around the world have proposed a lot of solutions to model the three-dimensional crane system as well as Cartesian coordinate robots. Some of the researchers were also using the small dimension model of 3D crane. Such a research are described in articles [1].

Another article [2], where the researchers used *Dymola*[6] model for the crane behaviour simulation.

Moreover article [3] where the authors used the same 3DCrane from Inteco® as in present thesis. In sum, many other and different control systems and approaches as well as other different mathematical models were submitted in past.

Here only a small part of them are listed. Moreover, Inteco® also provides their own mathematical model derivation. However none of the articles were found, where the authors use the state-space approach to build the control system for the 3D crane.

A-lab itself has made a sever researches related to the 3DCrane. The Bachelor of Science degree [4] thesis has been successfully defended. The research was to create the mobile phone application that allows to control the 3DCrane by the Wi-Fi network. The application is used indoor close to the controlled subject. Moreover the application should work on iOS system and should be uploaded to the Apple iTunes®[7]. The application displays the information of the subject, including the positions and deviation angles.

Another work recently done by Aleksei Tepljakov. The main task was to find an interface solution to control the 3DCrane with the joy pad device similar to the ones used in video games consoles. The idea is to move the crane along X, Y and Z axis using the dual analog-controller.

[6] Dymola – modelling and simulation environment based on the open Modelica modelling language

[7] iTunes – media store application by Apple Inc.

The analog-buttons send the signal and the controller moves the 3DCrane according the direction and axis was forwarded. The system works successfully.

Unlike the above studies, in my research the methods for solving the tasks are chosen as the most appropriate for the highly nonlinear system with multi inputs, multi outputs. First for derivation of the simplified model only the Newton’s second law of motion and geometric representation of the system was used. The approximation around the angles was also implemented.

After that the full model of the system is derived using Lagrange equation for kinetic and potential energies. Next, the Euler-Lagrange equations are used to build the final system of equations describing the 3DCrane dynamics. The system of equations were again approximated around the steady-state operational point. The state-space representation of a physical system is used to design the control in both simplified and full models. All the models are compared with the encoder-based model by Inteco®.

Obtained results have been analysed. The full model can replace the encoder-based model. They are not completely similar as they have errors in outputs between matching each other. However, has been suggested that the reason for errors could be the mistakes and approximations in modelling. As well as it can be the error accumulated in encoders, as they are no information about the calibration of the devices.

Chapter 2 introduces the Control System theory content. Different types of the systems are represented and shortly described based on the references. The connection between mathematical modelling, simulation and control systems design are also explained in Chapter 2.

Chapter 3 focuses on the study of the 3D crane small dimensional copy. The chapter consist of the description of the object and the components of the system such as encoders and chip for the computer interface. The model of the crane provided by the manufacturer is also presented. The chapter completes with the analysis and conclusions of the first simulation of the 3DCrane.

Chapter 4 is addressed to the mathematical model of 3DCrane. Two different models of the 3DCrane system are presented. First, the classical block scheme model in Simulink® is built. Second, the state-space approach is provided. Both solutions are simulated and analysed.

In Chapter 5 the derived model and the model provided by Inteco® are simulated together. For the derived model the state-space representation is used. The results are described and presented. Further conclusions and summarizations are given.

Chapter 6 describes the derivation of the full 3DCrane model taking into account the payload angle dynamics. Euler-Lagrange equation is used to derive the model. Again, the derived state-space representation is simulated together with 3DCrane manufacturer model. The obtained results are analysed and concluded. Finally, all the results gain in present thesis are shortly concluded in the last chapter.

Annexes 1, 2, 4 are containing the MATLAB scripts for the simulations of the derived dynamic models. Annex 3 in detail describes the Euler-Lagrange implementation to derive the full 3DCrane dynamic model.

The thesis is written using the Microsoft Corporation[8] licensed operating system Windows 7 Professional. Additionally, the next software has been used: Microsoft Word, Microsoft Paint, MATLAB/Simulink (used at the TUT Faculty of Information Technology laboratory).

[8] Microsoft Corporation – American multinational technology company specializing in develop, manufacture, license, support and sell computer software, consumer electronic, personal computers and services. Founded in 1975.

2. CONTROL SYSTEMS

Control systems are integrated part of our society [5]. Numerous applications are all around us: satellite antenna control, robotics, metallic part machined in CNC stage, etc. Many companies such as Apple Inc.®[9] and Google®[10] are now working on automated vehicles that can drive on the streets without a human interface. Automation is the technique of making a process or system work automatically. Nowadays, they are also called smart systems, where the system can automatically adapt to the environment giving the best possible solution. All this applications are created due to automatically controlled systems.

Human beings are not the only creators of automatically controlled systems, because these systems are also exist in nature long time ago. Within our own bodies the numerous control systems exist, such as pancreas that regulates the blood sugar [5]. We, however, take the examples, ideas and logic from nature. Even feedback control systems are older than humanity. Numerous biological control systems were built into the earliest inhabitants of our planet [5]. Feedback is the return of a portion of a process or system output to the input, especially when used to maintain performance or to control a system or process [6].

2.1. Control system definition

A control system consists of subsystems and processes (or plants) assembled for the purpose of obtaining a desired output with desired performance, given a specified input. The simplest form of control system is shown in Figure 2.1 [5].

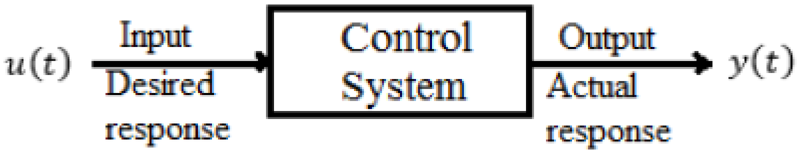


Fig. 2.1. Simplified description of a control system

[9] Apple Inc. – American multinational technology company specializing in design, develop and sell consumer electronics, computer software and personal computers. Founded in 1976.

[10] Google – American multinational technology company specializing in Internet-related services and product. Founded in 1998.

There are two major configurations of control systems: open-loop and closed-loop (feedback control) [5]. The system has the inputs, the signal provided by the designer. As an output the system has an expected behaviour of the system.

2.1.1. Open-loop system

Open-loop system (Figure 2.2) does not correct or monitor the output signal for disturbance. It starts with the input that is often called the controllable variable [5]. The input signal passes several blocks in order to be controlled.

Open-loop control systems are cheap and simple. However, they cannot compensate disturbances or noises. Thus, many corruptions affect the output. It is hard to predict disturbances. The output therefore can always vary, what makes the system unstable.

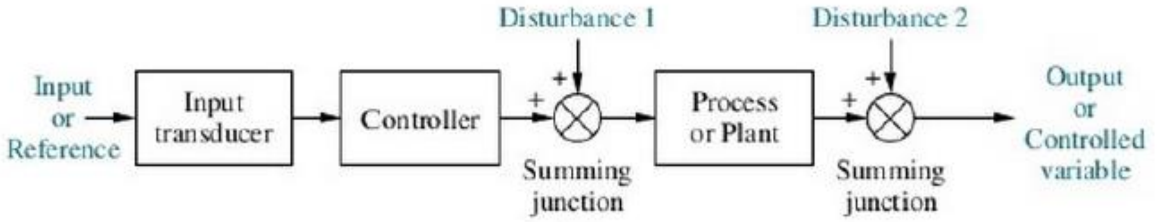


Fig. 2.2. Open-loop system

The disadvantage of this system is sensitivity to the disturbance. In order to protect the system against the unstable output, it is often added the feedback from the output to the input. This technique makes the system closed-loop.

2.1.2. Closed-loop system

The closed-loop system (Figure 2.3) monitors the output and compares it with the input [5]. This technique helps to take the disturbance into account and to compensate it. The closed-loop systems in comparison with the open-loop ones have a greater accuracy. They are less sensitive to noise, disturbance, and changes in the environment. On the other hand, closed-loop systems are more complex to design and expansive [5].

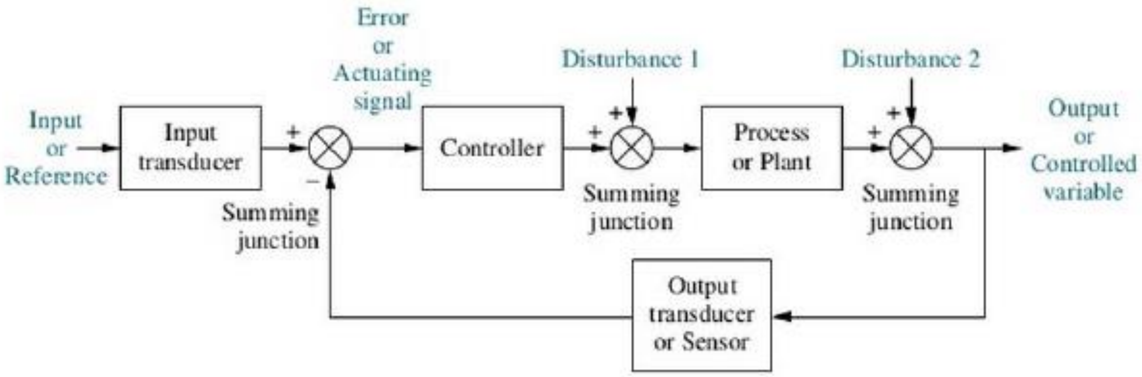


Fig. 2.3. Closed-loop system

The compensation of disturbance is not the only problem in control theory. The system must be stable in order to produce transient and steady-state response. Transient response is important because it affects the speed of the system and influences human patience and comfort, thus mitigating mechanical stresses. Steady-state response determines the accuracy of the control system. It governs how closely the output matches the desired response [5].

Moreover, it is often important to design the robust system. It means that the system is to be non-sensitive to parameter changes due to the fact that the relation between parameters changes and their effect on the performance is not linear.

In order to create a control system it is first necessary to create the mathematical model of the system to be controlled. In another words, designer must do the modelling task and describe the system using physical laws. Kirchhoff's laws are used in electrical network and Newton's law are used for mechanical systems, along with simplifying assumptions. The Newton's law will be used in current thesis.

2.2. Mathematical modelling

A models are an essential component of simulation [7]. Modelling is a critical issue in rapidly developing modern manufacture. Before multimillion-euro project will be build and tested in the environment, it is convenient to model and test the system using simulation tools and software. It helps to save the cost of the product and to avoid mistakes and accidents.

A meteorologist predicts the expected path of a tropical storm using weather models. An economist issues a quantitative forecast of the economy predicted on key economic variables. Health care professionals have access to a human patient simulator to receive training in the recognition and diagnosis of disease [7]. Every human activity can be modelled and therefore analysed in order to predict the behaviour.

The word "model" is a generic term referring to a conceptual or physical entity which resembles, mimics, describes, predicts, or conveys information about the behaviour of some process or system. The benefit from having a model is to be able to explore the intrinsic behaviour of a system in an economical and safe manner. In principle, the behaviour of dynamic system can be explained by mathematical equations and formulas, which embody either scientific principle or empirical observations related to the system [7].

The system's models consist of coupled algebraic and differential equations, when their parameters and variables change continuously over time or space. Partial differential equation models appear when a dependent variable is a function of two or more independent variables. The mathematical models of dynamic systems where the single independent variable is "time" are comprised of ordinary differential equations. The same applies to system with a single spatial independent variable [7].

Mathematical models can be categorized according their purpose: models to assist plant design and operation and models to assist control system design and operation [8]. In this work we are interested mainly in the second category as we are planning to control the system.

Ordinary differential equation models of dynamic systems are called "lumped parameters" (Figure 2.4) models. The spatial variation of the system parameters is negligible or it is being approximated by lumped sections with constant parameter values.

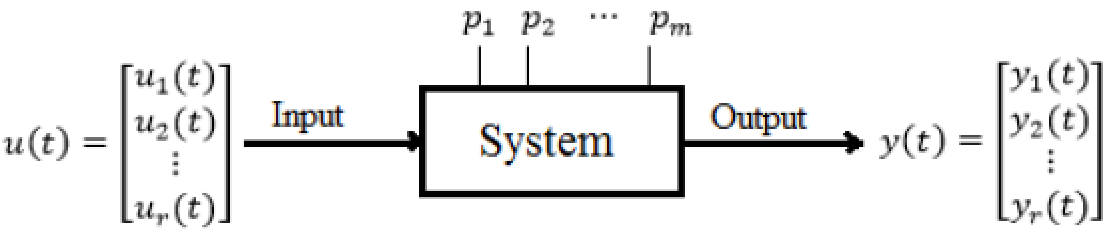

p_1 p_2 $\cdots$ p_m
$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}$
Input
System
Output
$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_r(t) \end{bmatrix}$

Fig. 2.4. A system with a lumped parameter model

The same system can be represented with different mathematical models. Regardless of the detail inherent in a mathematical model, it nevertheless represents an incomplete inexact depiction of the system [7]. The reason is refers to the simplifications and approximations used in modelling the system. Mathematical modelling relies on a combination of intuition, experience, empiricism, and the application of scientific laws of nature [7].

2.2.1. Simulation

The difference between a mathematical model and a simulation model is open interpretation [7]. Some in the simulation community is view the two as one and the same. Simulation refers to the solution of the model equation. Simulation is originated from the mathematical methods. However, simulation of dynamics system requires a simulation model, different in nature from mathematical model.

Moreover, it should be recognized that the "models" used in simulation are ideal and therefore may differ slightly from their physical implementations. These differences may lead in unstable systems to unexpected bias levels limit cycles [9].

Simulation models are commonly obtained from discrete-time approximations of continuous-time mathematical models. More than one simulation model can be developed from a single mathematical model of the system [7].

2.2.2. MATLAB® and Simulink®

MATLAB® (Matrix Laboratory) is the high-level language and interactive environment developed by MathWorks® for numerical computation. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python [10].

Simulink® is the most popular programming language tool for modelling, simulating and analysing dynamic systems developed by MathWorks®. It is integrated with MATLAB, what enables to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis. Simulink is a block diagram environment widely used in control theory and digital signal processing.

The Simulink library contains blocks for representing the mathematical models of commonly occurred components in dynamic systems. Building a Simulink model of a system consist of selecting the appropriate blocks and connecting them in a way that represents the mathematical model. Simulink offers a variety of numerical integrators to advance the continuous-time state vector over an integration step [7].

2.3. Linear systems

A linear system possesses two properties: superposition and homogeneity. The superposition states that the output response of a system to the sum of inputs is the sum of response to the individual inputs. Homogeneity property describes the responses of the system to a multiplication of the input by a scalar. Linearity can be visualized in Figure 2.5 [5].

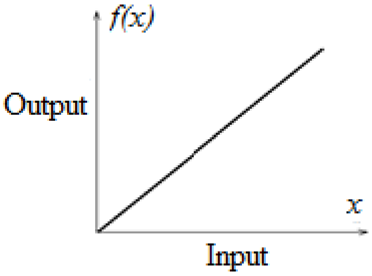


Fig. 2.5. Linear system

A common engineering practice is to assume that a system can be described by a set of linear differential equations for some operating range of interest as follows [11]:

$$\dot{x} = Ax + Bu, \tag{2.1}$$

where $x(t)$ – the state of system,

A, B – time invariant matrices,

$u(t)$ – the control input.

When the control input is presented, linear time invariant systems exhibit properties including superposition, asymptotic stability of the unforced system ensures bounded-input bounded-output stability, and a sinesoidal input leads to a sinesoidal output of the same frequency. The behaviour of a nonlinear system may be much more complex than a linear model can encapsulate [11].

2.4. Nonlinear systems

The definition of linearity cannot not applied to the relationship shown in Figure 2.6. Thus types of systems are called nonlinear. The designers can often make a linear approximation to a nonlinear system. Linearization is done for small-signal inputs about the steady-state solution when the small-signal input is equal to zero. This steady-state solution is called equilibrium [5].

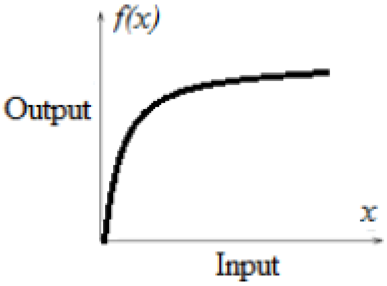


Fig. 2.6. Nonlinear system

The nonlinear features of the system may originate inherently or intentionally. Inherent nonlinearity is an inseparable characteristic of the laws governing the operation of the process to be controlled. Intentional nonlinearity is deliberately introduced into the design of the system as the best way to achieve the desired performance, based on both technical and economic criteria [11].

As the nonlinear systems are very hard to control, often they are approximated and replaced with a number of linear systems. These techniques are made in the interest of mathematical tractability of the model. Each motion in this case is described by linear differential equation.

The model of a system may contain one or more constants. Other coefficients may be constant over the whole range of operation of the model and for practical purpose they can also be regarded as constants. Some coefficients may be considered constant during any particular run of the model. Finally, there are coefficients that need to be modified during the running of the model. Current values are determined either from a stored formula or by interpolation in a stored table [8].

A dynamic model in the form of a vector-matrix differential equation has a natural classification of variables:

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du, \tag{2.2}$$

where x – n-dimensional vector of a state variable, where n is the order of the system,

y – m-dimensional vector of output variables,

u – p-dimensional vector of input variable,

A – a matrix of size $n \times n$,

B – a matrix of size $n \times p$,

C – a matrix of size $m \times n$,

D – a matrix of size m$\times$ p.

A state-space representation can be used for a mathematical model of a system with multiple inputs and outputs, denoted by MIMO. In this case A, B, C and D will be matrices of appropriate dimensions which accounts for the use of capital letters [12].

In practice each of the elements of y must be determined by observation and measurement. Also, u vector must contain elements representing changing plant inputs, all known external disturbances that are to be represented, and control variables [8].

2.5. Nonlinear Multivariable (MIMO) systems

Multi-Input Multi-Output control system (Figure 2.7) is the control theory system with more than one input and/or more than one output. This control system is also frequently known by the abbreviation MIMO. Another words, it is multivariable system, where each manipulator variable can depend on two or more of the controlled variables.

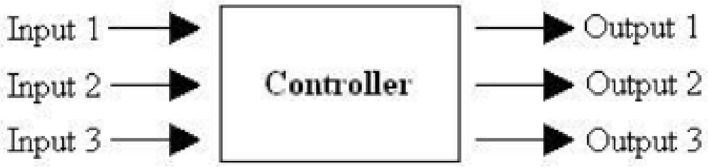


Fig. 2.7. MIMO control scheme

The control system that has only one input and only one output is called Single-Input Single-Output (SISO). Comparing to the SISO system, the MIMO is hard to design as it integrates multiple sensor data to coordinate multiple actuators [13].

The mathematical model is nonlinear due to the presence of the nonlinear differential equation and nonlinear algebraic equation [7]. Controller-loop interaction exists such that the action of one controller affects other loop in a multi-loop system. Depending upon the inter-relationship of the process variable, tuning each loop for maximum performance may result in system instability when operating in a closed-loop mode [14].

Multivariable control is not the only solution to control nonlinear multi-input multi-output system. There are several techniques exist to model the control: Predictive Constrain control, Model-Based Predictive Control, Dynamic Matrix Control, Statistical Process Control, etc.

2.6. Chapter summary

Control systems contribute to many aspects in our society and are also broadly represented in nature. A control system has the inputs, the system body and the outputs. A control system can be open loop or close loop. The close loop system monitors the outputs, being more complex and more expansive. It can also correct the input to avoid distortions.

The system first must be mathematically modelled, using the physical and mathematical rules. Most of the time, they are also approximated to simplify the solution. The approximation is necessary, because there are no linear system in nature. The second step is to assemble the model. There can be many solutions for the same model. Finally, the results must be simulated, studied and analysed.

Linear and nonlinear systems were described. The SISO and MIMO control system were stated and compared. The first has only one input and one output, the second respectively can have multiple inputs and multiple outputs.

The solution of the highly nonlinear MIMO system will be shown in Chapter 4. The state-space representation of the system will be derived, simulated and analysed.

3. OBJECT DESCRIPTION

3.1. Industrial 3D crane

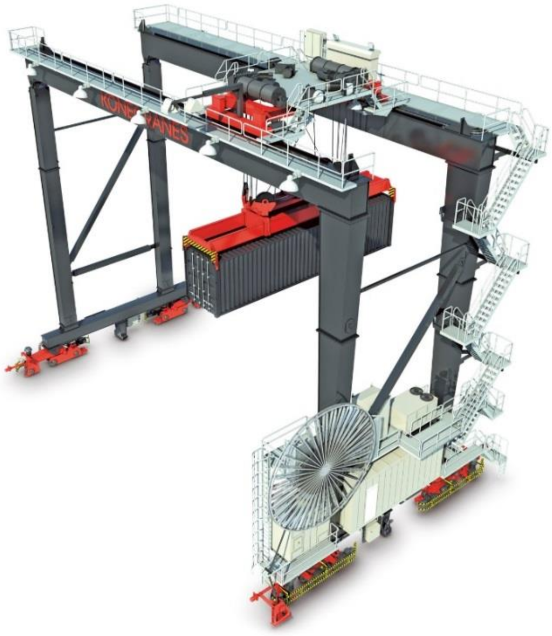


Fig. 3.1. Gantry crane by Konecranes®

3D cranes (Figure 3.1) are widely used in industry for vessel loading and unloading, to handle full or empty containers and lift up to 80 tons, automotive lines, ship and airplane building, power industry and other heavy manufacturing. Different 3D cranes are shown in Figures 3.2 and 3.3. As a rule, in ports 3D cranes are able to speed up the entire shipbuilding process. Nowadays cranes are being built with semi- or full-automation capability. Thus, many researches have been made in order to find the best, cheapest and optimal solution to control cranes. The AutoPilot control algorithm can remove the operator from tasks that require repetitive motions. Benefits include more precise handling, reduced component wear and greater availability [15].

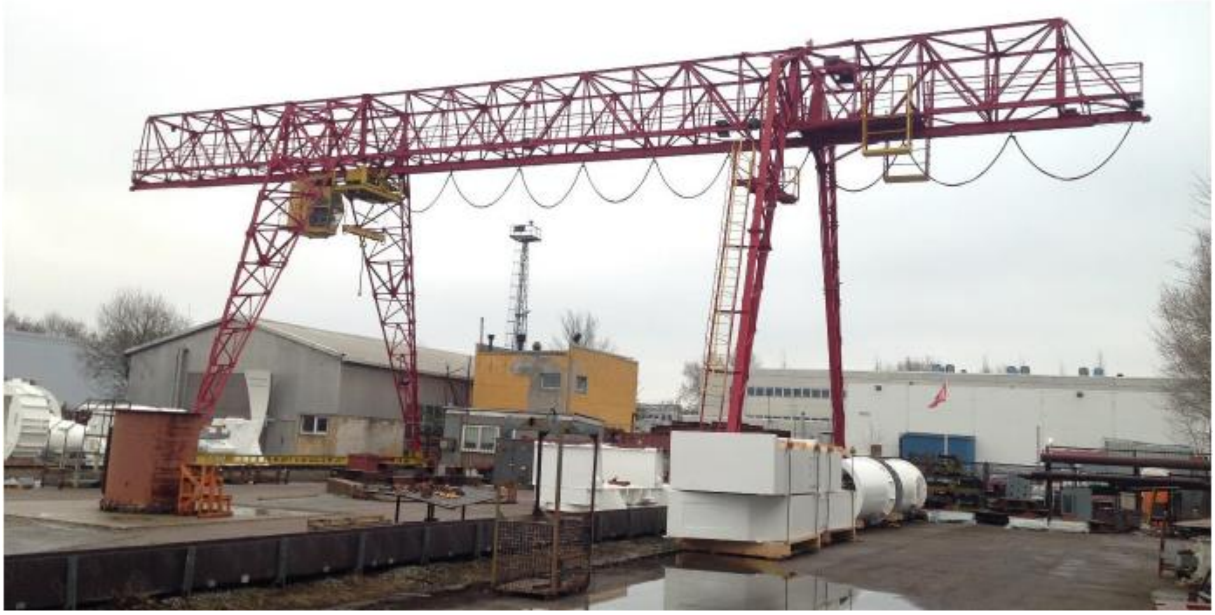


Fig. 3.2. Outdoor industrial 3D crane

Fig. 3.3. Indoor industrial 3D crane

In general, an operator moves the crane in the desired position. The acceleration or deceleration of the crane may cause undesirable load swing. The load swing can cause human accidents, can damage the crane or the load. To control the swing, the operators need experience in crane manoeuvring. Automated systems can help the operator to control the load swing or can even replace the operator. They can also achieve precise control and increased load speed positioning.

3.2. General description of the studied object

3D cranes are widely used in industry, construction works, ports and manufacturing. The three-dimensional crane (Figure 3.4) controlled from PC is studied in this research using is the laboratory model of industrial crane. The model is built for control education and research works mostly for automation control and information technology fields. The model is operated in real-time mode. 3DCrane is ideal for illustrating complex nonlinear MIMO (multi input multi output) control algorithms. The 3DCrane workspace can be viewed in Figure 3.5.

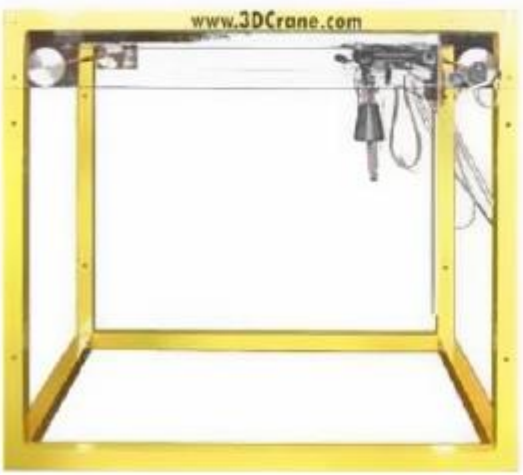


Fig. 3.4. Inteco® 3DCrane

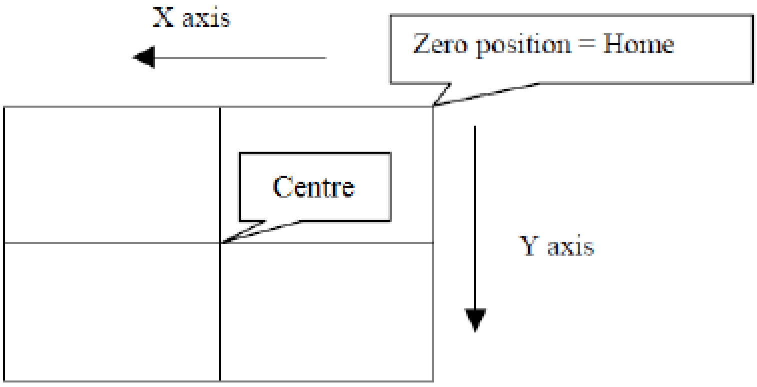


Fig. 3.5. View of the workspace of 3DCrane

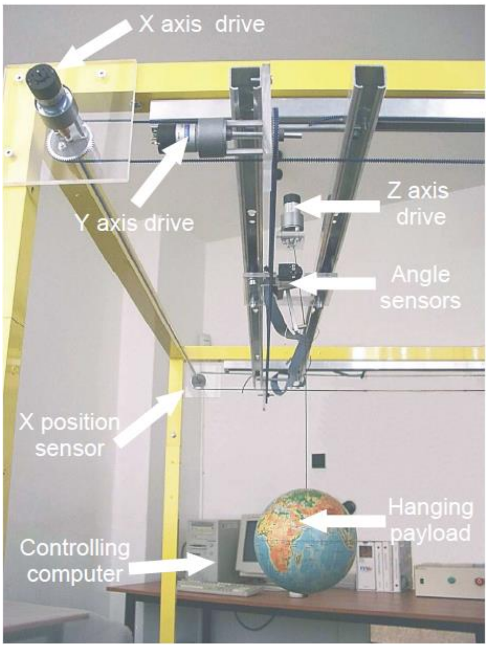


Fig. 3.6. The 3DCrane setup

The 3DCrane is a nonlinear MIMO electromechanical system having a complex behaviour and creating control problems. The system is controlled from PC. The mechanical unit is supplied with power and interfaced to a PC. The mechanical part is linked with A/D, D/A board configured in the Xilinx® technology[11]. The software operates under MS Windows® NT using MATLAB® /Simulink® environment and RTWT (Real-Time Window Target™) toolbox package[12]. The system, therefore, can be easy mounted and installed in a laboratory [16].

The 3DCrane setup (Figure 3.6) consists of a payload hanging on a pendulum-like lift-line wound by a motor mounted on a cart. The payload is lifted and lowered in the z direction. Both the rail and the cart are capable of horizontal motion in the x direction. The cart is capable of horizontal motion along the rail in the y direction. The payload attached to the end of the lift-line can be freely moved in three-dimensions. The 3DCrane is driven by the three DC motors [16]. Two motors

[11] Xilinx - American technology company producing and developing programmable logical products, including integrated circuits, software design tools, predefined system functions delivered as intellectual property cores and design services [17].

[12] Real-Time Window Target™ is a real-time engine for Simulink models and blocks connected to a range of I/O boards. It enables to create and control the system in real time.

are used to drive the crane in X and Y directions in working space and the last motor is used to lift the load.

The DC motors are produced by the Italian company Micro Motors S.R.L.[13] [18]. In our 3DCrane system three RH158.24.75 DC motors are used (Figure 3.7). The technical data of the motor is shown in Table 3.1. The direction of rotation depends on polarity. A motor can be mounted in any position. The dimensions of the motor are shown in Figure 3.8.

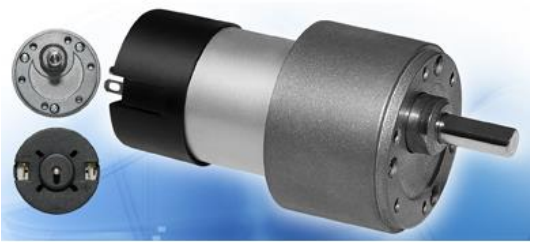


Fig. 3.7. RH158.24.75 DC motor

Table 3.1. RH158.24.75 DC motors technical data

Parameter (at ambient temperature +20% - tolerance +/-10%)		**Value**
Nominal Voltage $[V]$		24
L $[mm]$		66,5
Ration to:1		76,84
Nominal torque $[Ncm]$		50
Speed $[m/s]$	No load	81
	At nominal torque	55
Current $[mA]$	No load	<70
	At nominal torque	340
Maximum radial shaft load $[N]$		50
Maximum axial shaft load $[N]$		10
Temperature range $[°C]$		-20 / 60
Approximate weight $[g]$		190

[13] Micro Motors S.R.L - Italian company that produces small gear-motors with different specifications and applications.

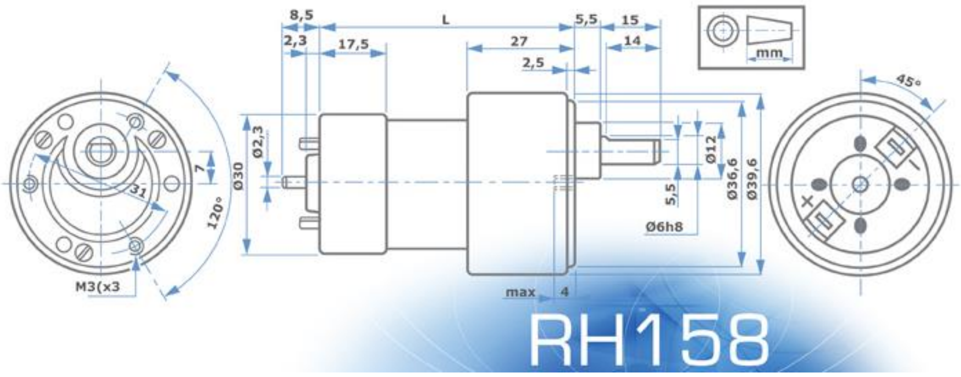


Fig. 3.8. The dimensions of the RH158.24.75 DC motor

Fig. 3.9. Two-phase Hall-effect 90° encoder

The Hall-effect switches are used in encoder (Figure 3.9). They are highly temperature stable and stress-resistant sensors mainly utilized in applications that provide steep magnetic slopes and low residual levels of magnetic flux density. Each device includes voltage regulator, quadric Hall voltage generator, temperature stability circuit, signal Schmitt chopper stabilized amplifier, Schmitt trigger and an open drain MOSFET on a single silicone chip. The on-board regulator permits operation with supply voltages of 3,5 to 24V. The output MOSFETs can sink up to 20 mA with suitable output pull up, they can be used directly with bipolar or MOS logic circuit [18]. The technical data of the encoder is provided in Table 3.2.

Table 3.2. Two-phase Hall-effect 90° encoder technical data

Absolute Maximum Ratings		
Parameter	**Symbol**	**Value**
Supply voltage $[V]$	VDD	28
Supply current $[mA]$	IDD	50
Output voltage $[V]$	VOUT	28
Output current $[mA]$	IOUT	50
Storage temperature range $[^\circ C]$	TS	-50 to 150
Maximum junction temperature $[^\circ C]$	TJ	165

The encoders measure movements with a high resolution equal to 4096 pulse per rotation, and together with the specialised mechanical solution create a unique measurement unit. The deviation of the load is measured with a high accuracy equal to 0,0015 rad [16].

The power interface amplifies the control signal, which is transmitted from the PC to the DC motor. It also converts the encoder pulse signals to the digital 16-bit form to be read by the PC [16]. The PC equipped with the RT-DAC/PCI multipurpose digital I/O board communicates with the power interface board. 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 programmable Xilinx® chip of the RT-DAC/PCI board. All functions of the board are accessed from the 3DCrane Toolbox, which operates directly in the MATLAB® /Simulink® environment [16].

The whole control logic is presented in Figure 3.10. The Xilinx® chip sends the appropriate sequence of pulses to control the voltage of the motor. The voltage therefore rotates the DC motor moving the cart to the appropriate position in the workspace. The encoders play the role of the feedback information. They show the location of the cart and rail as well as the workspace and the angles of the payload in real time.

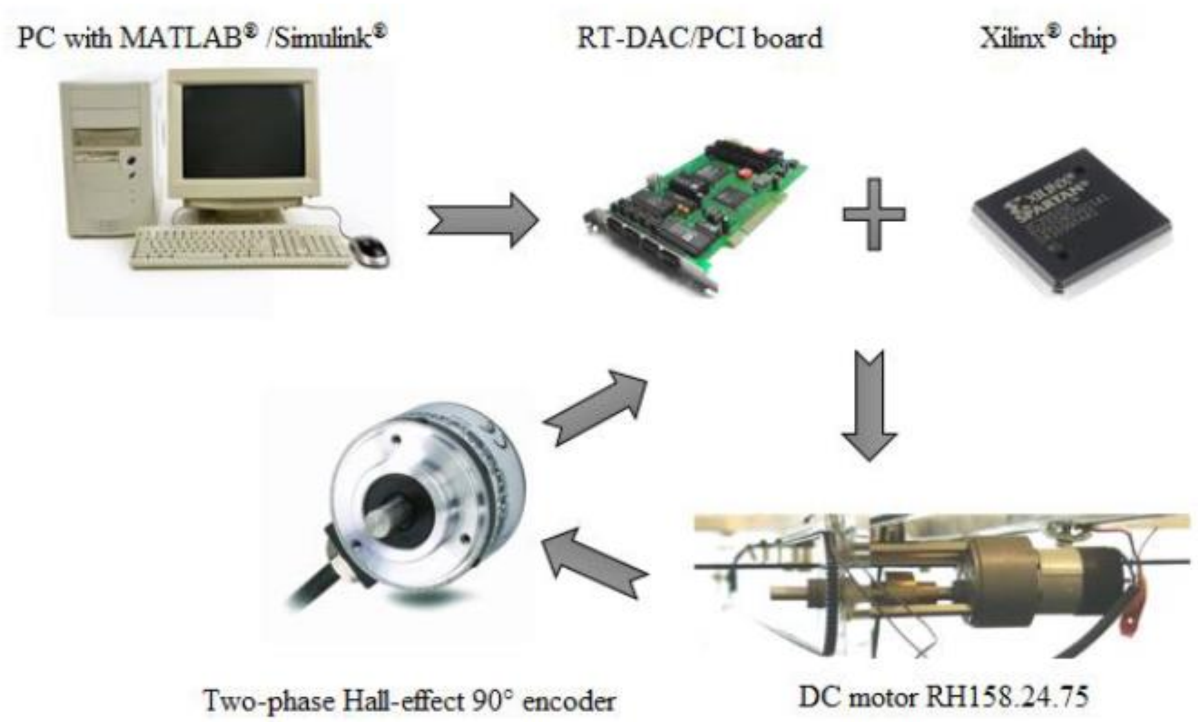


Fig. 3.10. The 3DCrane control components

3.3. Object overview

The crane is delivered in partially mounted form. The mounting frame makes a support and a flexible construction of the system (Figure 3.11). There is a rigid construction due to the reason it is fixed to the floor. The dimensions are both the width and length from 0,9 m to 2,5 m [16]. The system parameters of the 3DCrane are shown in Table 3.3.

Table 3.3. 3DCrane parameters

Parameter	**Value**
Maximum speed v_{max} $[m/s]$	0,3
Maximum acceleration a_{max} $[m/s^2]$	0,6
Workspace distance in X-axis direction $[mm]$	920
Workspace distance in Y-axis direction $[mm]$	920
Core length l $[mm]$	820
Payload mass m_p $[kg]$	1
Mass of the cart m_c $[kg]$	1,16
Mass of the moving rail m_r $[kg]$	2,2

The 3DCrane is also equipped with the safety button. The button is located on the table in front of the PC, which is used to control the system. In case of emergency, a user must press the button manually, what immediately will stop the current supply to the crane and the system stops. Moreover, the crane workspace is located at the corner of the laboratory (Figure 3.11). The location is marked with the black-yellow sticker line on the floor. It is strictly forbidden to cross the line while the 3DCrane is operated.

The software requirements are as follows:

- Pentium® or AMD based personal computer
- Microsoft Windows® XP/7x86
- MATLAB® version R2009a/b, R2010a/b or R2011a/b with appropriate version of Simulink® and RTW toolboxes.

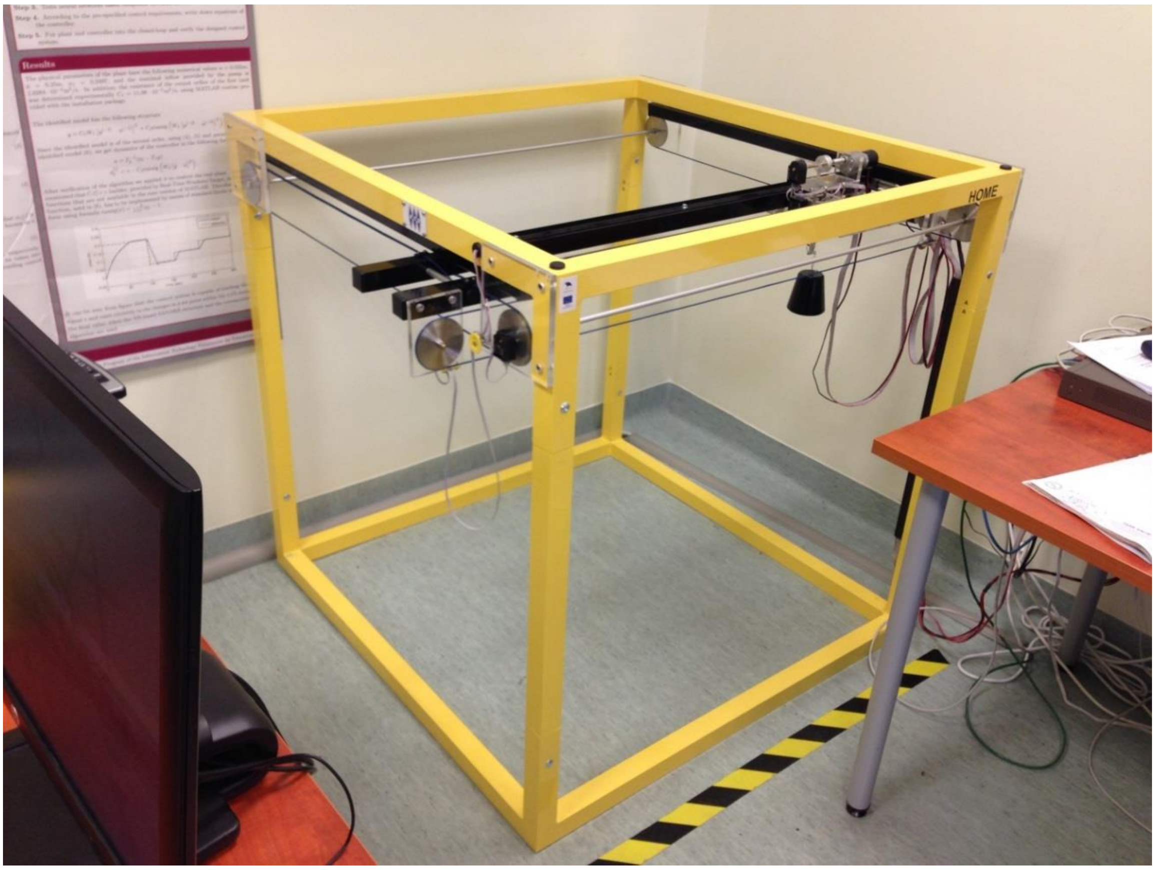


Fig. 3.11. 3DCrane workspace in the laboratory

The hardware consists of the following mechanical units: cart, 2D angle measuring unit (Figure 3.12), X axis drive (Figure 3.13). It is also consists of interface and Power Interface Unit and RT-DAC/PCI board.

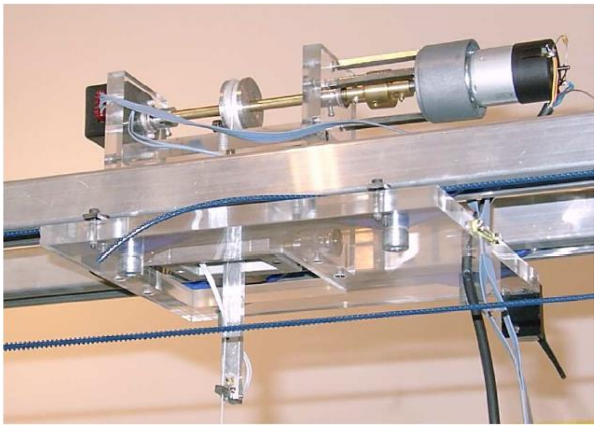


Fig. 3.12. Cart and 2D angle measuring unit

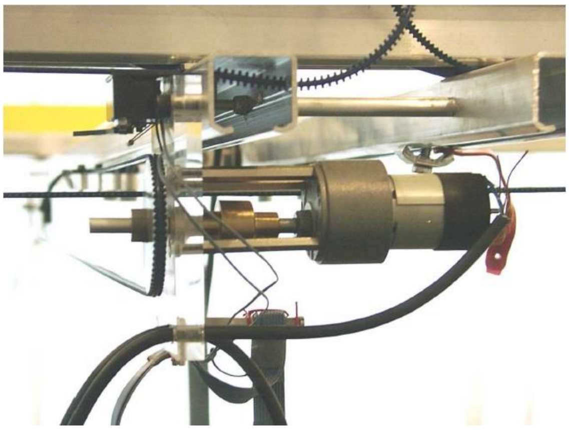


Fig. 3.13. X axis drive

3.4. Testing the 3DCrane

In order to understand the control system, it is necessary to test the model provided by the Inteco®. Testing may help to solve the problems appeared in assembling the system. In Figure 3.14 the model of 3DCrane is shown. It is important to know, that the outputs are the values measured by the encoders. In this type of system, the encoders must be calibrated accurately. The calibration must be provided at least every half a year. It is noted in encoder provider webpage: “Exposure to all absolute-maximum-rated conditions for extended period may affect device reliability” [18]. It is not known, if the 3DCrane was often used in absolute-maximum-rate.

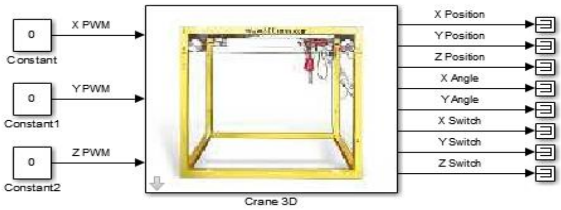


Fig. 3.14. Simulation model provided by Inteco®

X control, Y control, Z control inputs represent the control signals of the X, Y, Z DC drives that may be set. The control values may vary from -1.0 to +1.0. The value of -1, 0.0 and +1 mean respectively: the maximum control in a given direction defined by +1, zero control and the maximum control in the opposite direction to that defined by -1 [16].

3D Crane Simulink scheme is shown in Figure 3.15. It consist of the Parameters of the system, Measurement gain from the encoders and switches and control block that reads information from the model code, provided by Inteco®.

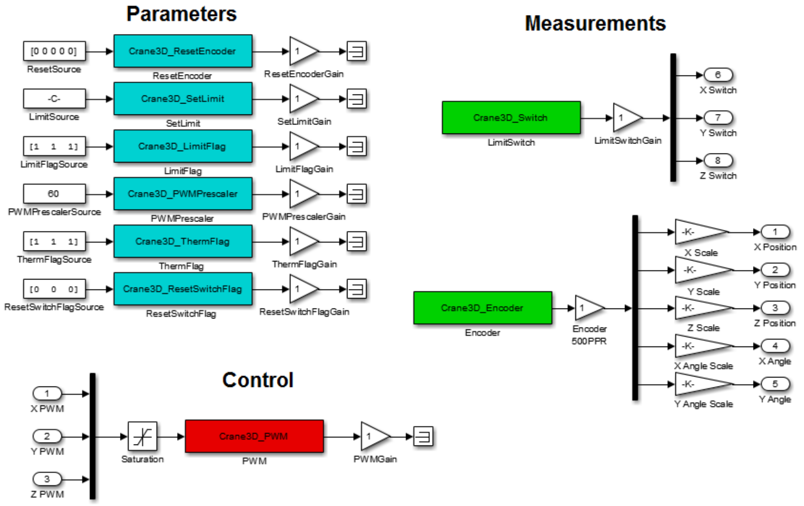


Fig. 3.15. Simulink Model of the 3D Crane by Inteco®

The simulation of the model control scheme is shown in Figure 3.16. The scopes are showing as the input signal and the position reading from the encoder for X, Y and Z axis in 3DCrane workspace. As it is described in [16] the cart of the crane must follow the position according the input impulse. Thus the range of the output is -1 to 1. The position must be limited by the physical workspace of the crane. The simulation results are shown in Figure 3.17.

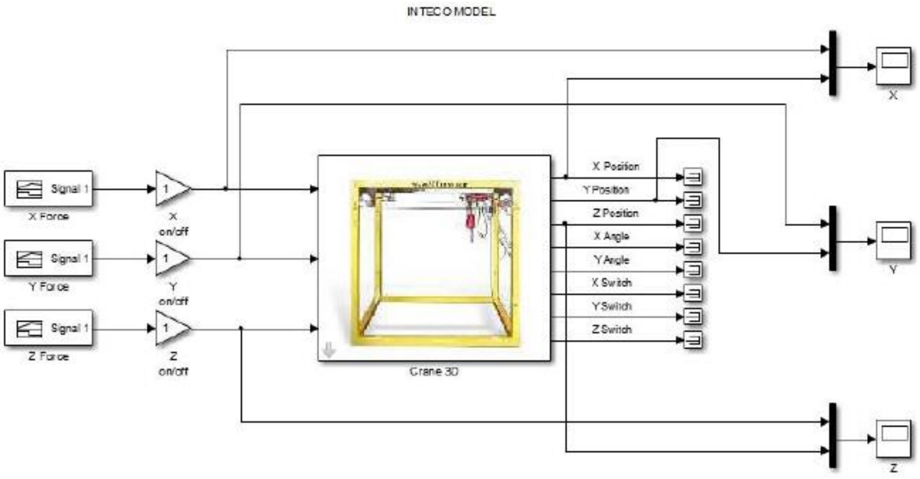
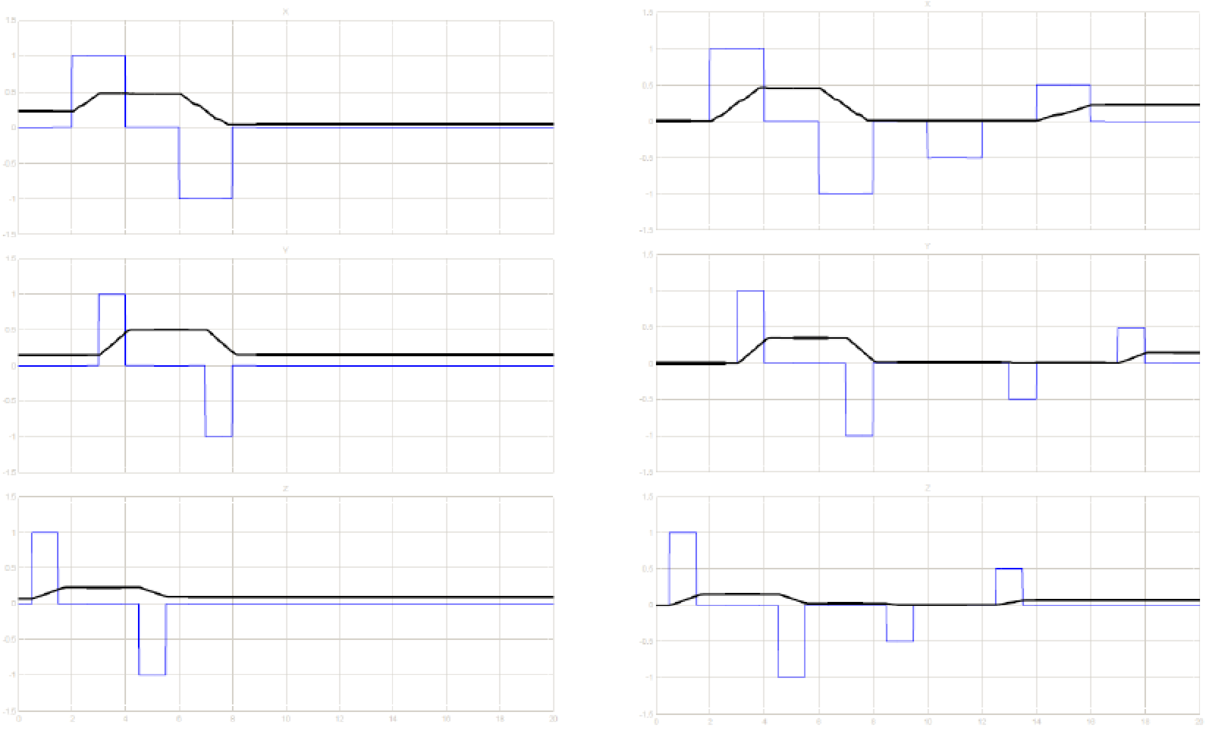


Fig. 3.16. Inteco® model control scheme



(a)

(b)

Fig. 3.17. The Inteco® model simulation inputs and outputs: (a) one position; (b) two different positions in workspace

According to the results of the simulation (Figure 3.17), it can be clearly seen that the crane does not move to exact position required by the input pulse. Moreover, the behaviour of the output looks like the velocity curve. It cannot be therefore conformed that the Inteco® provides the position control model.

3.5. Chapter summary

Nowadays, the three-dimensional cranes are widely used in industry. However, the control question plays the important role in crane system engineering. Many researches has been conducted aiming to improve the accuracy and minimize the system human interface. It is necessary to increase the work efficiency provided by the cranes, avoid accidents and minimize dependence on the system operator by involving automation control.

The studied object is the small-dimensional copy of the industrial 3D cranes. The control logic provides the signal to control three DC motors, moving along X, Y and Z axis. The position of the crane as well as the angles of the payload are measured by the encoders in real-time simulation. The force input must have an impulse curve in order to obtain required position.

The model control experimentation of the model provided by the manufacturer is shown in Figure 3.17. It has been concluded that, the 3DCrane control model does not behave as the position control. The output curve looks more like the speed control.

The main reason to build my own mathematical model is to obtain the position control. Because it is not easy to control the speed of the crane, comparing to the position in workspace. Moreover, the main idea is to avoid sensor feedback as the accuracy is decreasing due to the amortisation problem of the encoders. Another reason to avoid encoders is to decrease the cost of the service and the system itself.

4. THE MODEL DEVELOPMENT AND SIMULATION

As it was mentioned in Chapter 2 the modelling of the mechanical system is one of the most important parts in control engineering. Modelling helps to understand physical and theoretical behaviour of the system. The chapter outlines the dynamic model derivation of the 3DCrane. Derived model will be therefore approximated around the sines and cosines of the payload angles. Based on derived dynamic model, two solutions will be provided. Moreover the model outputs will be also tuned.

4.1. Dynamic model of the 3DCrane

The schematic drawing of the 3DCrane is shown in Figure 4.1. The schematic drawing of the 3DCrane workspace is shown in Figure 4.2. All the dynamic parameters presented in the figures are described in Table 4.1. There are 5 parameters measured by the encoders: x_w, y_w, l, α, β. This parameters are also the outputs of the controlled system.

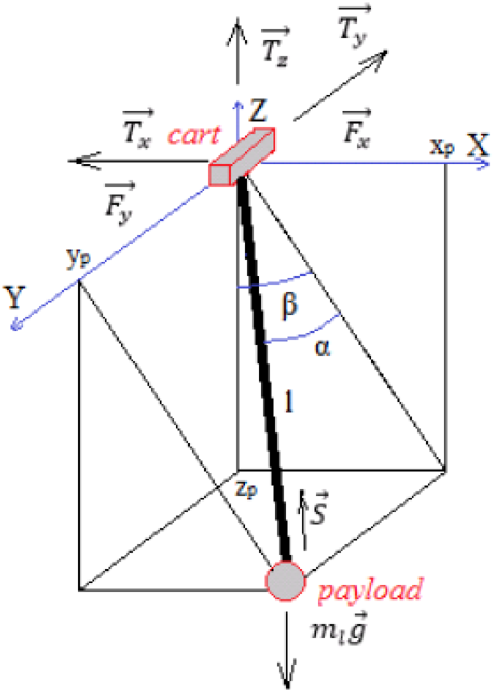


Fig. 4.1. 3DCrane schematic drawing

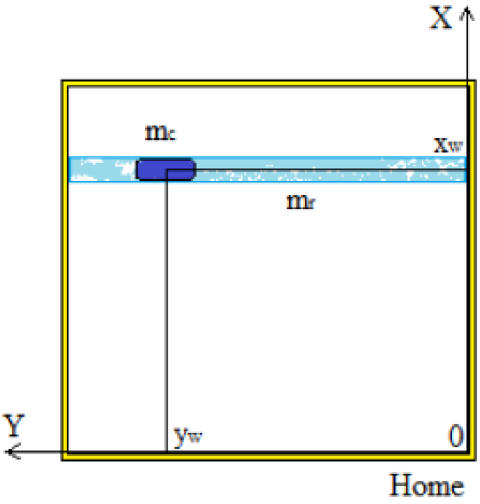


Fig. 4.2. 3DCrane Workspace schematic drawing

Table 4.1. 3DCrane dynamic parameters description

Symbol	Description	Symbol	Description
m_c	Mass of the cart	g	Gravitational constant
m_l	Mass of the payload	F_x	Force driving the rail with the cart
m_r	Mass of the moving rail	F_y	Force driving the cart along the rail
x_p	X-axis position of the payload	l	Length of the lifted load
y_p	Y-axis position of the payload	T_x	X-axis tension force
z_p	Z-axis position of the payload	T_y	Y-axis tension force
S	Reaction force acting while lift the load		
x_w	X-axis position of the cart in workspace		
y_w	Y-axis position of the cart in workspace		
α	Angle between the projection of the lift-line on XY plane and the lift-line		
β	Angle between the negative direction of the Z-axis and projection of the lift-line on XY plane		

4.1.1. Position of the payload equations

Using the simplified drawing of the position of the payload (Figure 4.3) the trigonometrical dependence is obtained.

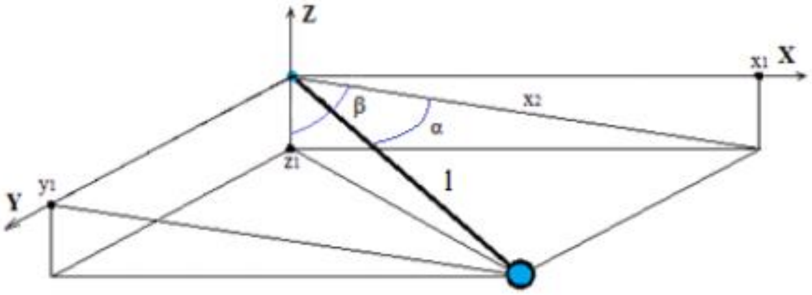


Fig. 4.3. The drawing of the position of the payload

$$\sin\beta = \frac{x_1}{x_2} \tag{4.1}$$

$$\sin\alpha = \frac{y_1}{l} \tag{4.2}$$

$$\cos\alpha = \frac{x_2}{l} \tag{4.3}$$

$$\cos\beta = \frac{z_1}{x_2} \tag{4.4}$$

$$x_1 = l\cos\alpha\sin\beta \tag{4.5}$$

$$y_1 = l\sin\alpha \tag{4.6}$$

$$z_1 = l\cos\alpha\cos\beta \tag{4.7}$$

The mathematical representation of the payload position can be therefore derived using equations (4.5), (4.6) and (4.7):

$$x_p = x_w + l\cos\alpha\sin\beta \tag{4.8}$$

$$y_p = y_w + l\sin\alpha \tag{4.9}$$

$$z_p = l\cos\alpha\cos\beta \tag{4.10}$$

4.1.2. Dynamic equations

In order to describe the dynamics of the 3DCrane system (Figure 4.1.) Newton's[14] second law of motion is used $\vec{F} = m\vec{a}$:

$$m_l\ddot{x}_p = S_x \tag{4.11}$$

$$m_l\ddot{y}_p = S_y \tag{4.12}$$

$$m_l\ddot{z}_p = -S_z - m_l g \tag{4.13}$$

$$(m_c + m_r)\ddot{x}_w = F_x - T_x + S_x \tag{4.14}$$

$$m_c\ddot{y}_w = F_y - T_y + S_y \tag{4.15}$$

where the axis components of the vector S are as follows:

$$S_x = S\sin\alpha\sin\beta \tag{4.16}$$

$$S_y = S\cos\alpha \tag{4.17}$$

$$S_z = S\sin\alpha\cos\beta \tag{4.18}$$

We now state that the total control forces supplied each axis are equal to N_x, N_y, N_z accordingly and take into account equations (4.16), (4.17) and (4.18). The input force is the total forces required for the control system in order to move the crane along the axis. 3DCrane dynamics can be therefore rewritten as follows:

[14] Isaac Newton – English physicist and mathematician (1642 – 1726/7). Widely recognised as one of the most influent scientists of all time.

$$\ddot{x}_p = \frac{N_z \cos\alpha \sin\beta}{m_l} \tag{4.19}$$

$$\ddot{y}_p = \frac{N_z \sin\alpha}{m_l} \tag{4.20}$$

$$\ddot{z}_p = \frac{-N_z \cos\alpha \cos\beta}{m_l} \tag{4.21}$$

$$\ddot{x}_w = \frac{N_x + N_z \cos\alpha \sin\beta}{(m_c + m_r)} \tag{4.22}$$

$$\ddot{y}_w = \frac{N_y + N_z \sin\alpha}{m_c} \tag{4.23}$$

The dynamic model obtained is highly nonlinear differential equations system of second order[15].

4.1.3. Approximation

The dynamic model obtained is highly nonlinear multivariable system. Its dynamical equations (4.19) to (4.23), no doubt, require the approximations. If assume that the deviation of the payload form the z-axis is small, the dynamic model can be then simplified. Assume both angles α and β are very small, almost negligible. The trigonometrical approximation rules are used:

$$\cos\alpha = \cos\left(\frac{\pi}{2} - (-\Delta\alpha)\right) = \cos\frac{\pi}{2} \cdot \cos(-\Delta\alpha) + \sin\frac{\pi}{2} \cdot \sin(-\Delta\alpha) = -\sin\Delta\alpha \cong -\Delta\alpha \tag{4.24}$$

$$\sin\alpha = \sin\left(\frac{\pi}{2} + \Delta\alpha\right) = \sin\frac{\pi}{2} \cdot \cos\Delta\alpha + \sin\frac{\pi}{2} \cdot \sin\Delta\alpha = \cos\Delta\alpha \cong 1 \tag{4.25}$$

Using the equations (4.24) and (4.25), the angles can be therefore expressed as:

$$\sin\alpha \cong \alpha \tag{4.26}$$

$$\cos\alpha \cong 1 \tag{4.27}$$

$$\cos\beta \cong 1 \tag{4.28}$$

$$\sin\beta \cong \beta \tag{4.29}$$

The simplified dynamic model of the 3DCrane is now obtained. The model depends on the total input forces on each axis N_x, N_y, N_z.

[15] Second order differential equation – equations containing the second derivative.

$$\ddot{x}_p = \frac{N_z}{m_l}\beta \tag{4.30}$$

$$\ddot{y}_p = \frac{N_z}{m_l}\alpha \tag{4.31}$$

$$\ddot{z}_p = -\frac{N_z}{m_l} \tag{4.32}$$

$$\ddot{x}_w = \frac{N_x+N_z\beta}{(m_c+m_r)} \tag{4.33}$$

$$\ddot{y}_w = \frac{N_y+N_z\alpha}{m_c} \tag{4.34}$$

Important remark. The model could be modified adding another dynamic effects, such as viscous damping coefficient, mass moment of inertia of the gantry and rail motors, gantry friction and other. Nevertheless, the addition dynamic effects can be directly cancelled by the controller. They are therefore not included and assumed inside of the control forces N_x, N_y, N_z. The next step is to assemble the Simulink model using the obtained results of the dynamics.

4.2. Simulation of the dynamic model

As it is said in Section 2.2 it is important to simulate the dynamic model. There might be several problems with the simulated model. If the model is assembled correctly, the approximations may cause an unexpected behaviour of the model. Subsequent study helps to understand the model and subject behaviour more precisely.

The Simulink block diagram of the model is presented in Figure 4.4.

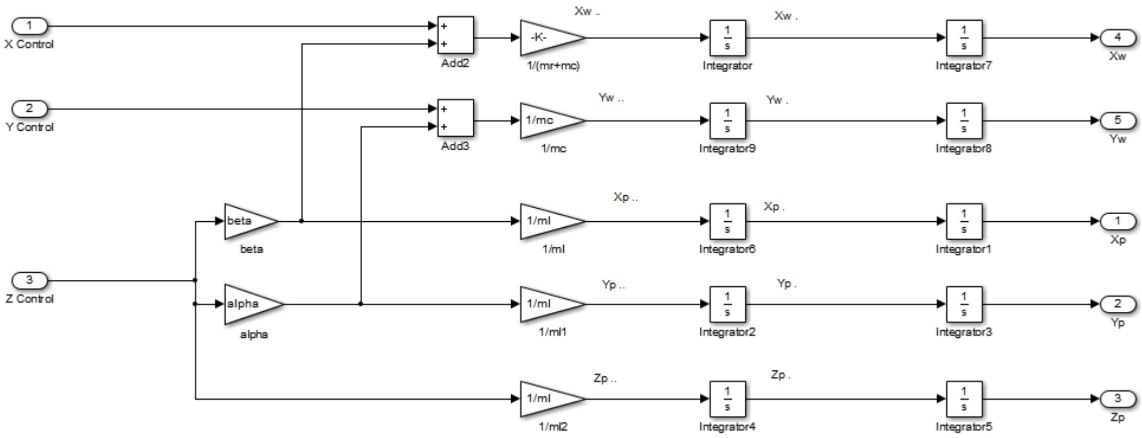


Fig. 4.4. The simplified dynamic model Simulink block scheme

After double integration of the input forces, the system must have the position as an output for each axis. In Figure 4.5 the control scheme of the model is shown. The control logic is to obtain the position giving the forces as the inputs. It has been stated in Section 2.4 that many variables are often considered as constant values. In this model both angles of the payload α and β are assumed constant (Annex 1 lines 8 and 9). However, they may not be completely neglected due to the reasons they present in the dynamic model.

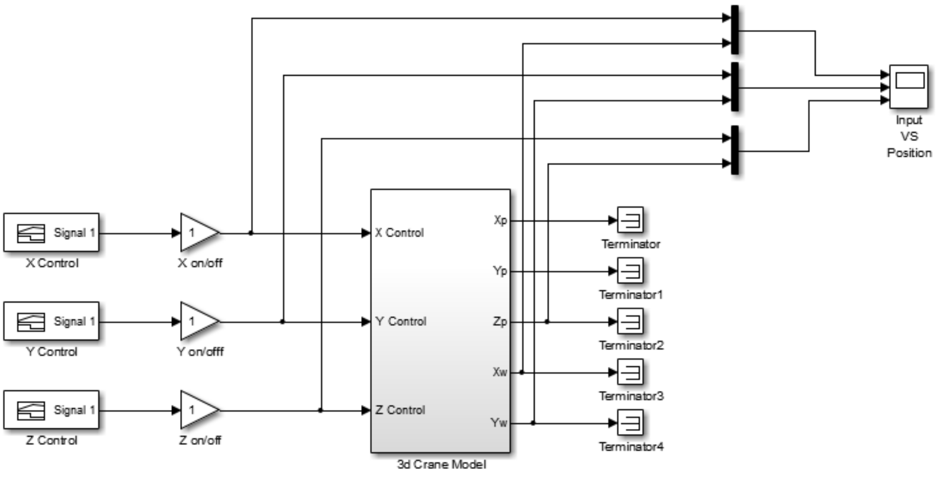
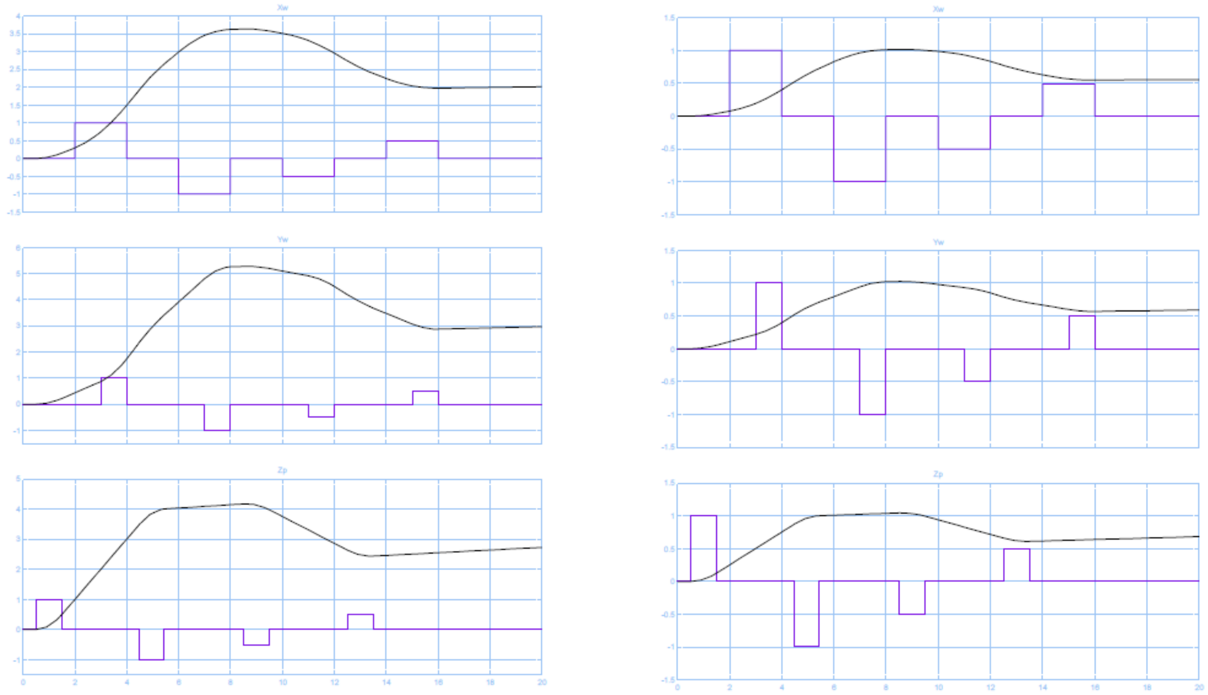


Fig. 4.5. The Simulink control scheme of the simplified dynamic model



(a) (b)

Fig. 4.6. Input (blue) and output (black) signals of the dynamic model: (a) no gain, (b) using the gains for each axis

The simulation results are presented in Figure 4.6 (a). The black curve represents the position output and the blue one shows the input force pulses. It can be clearly seen that the position does not go according to the input force (Figure 4.6 (b)). This problem can be solved by adding the gain to every axis input (Annex 1). They are also called amplification factors. Gain is the proportional value that shows the relation between the input and the output. Nevertheless, including the gain is a very dangerous step as it may cause the system instability.

Figure 4.6 (b) shows the correct solution for the model. Now it can be seen when the pulse wants the crane to move to 1 the crane goes to the axis position 1 relatively. In Figure 4.7 the modified Simulink block scheme is presented.

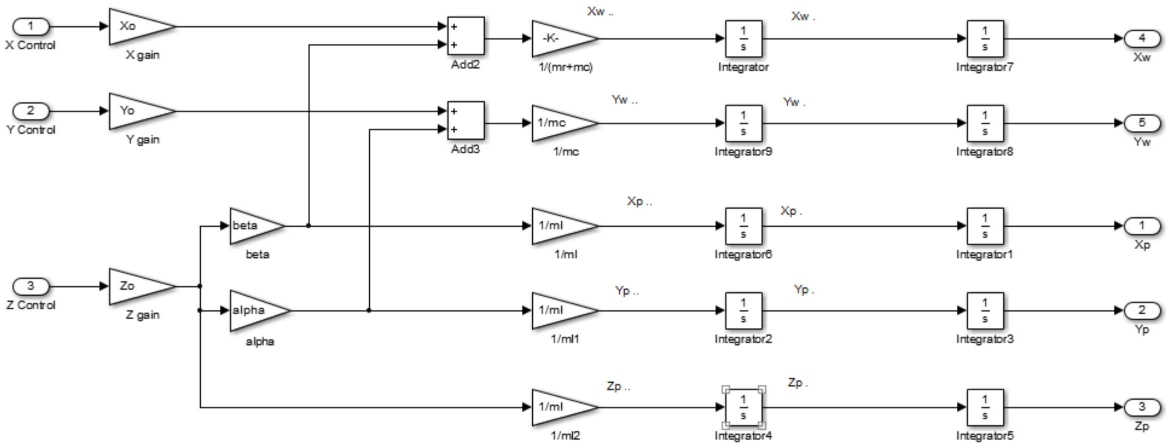


Fig. 4.7. The Simulink control scheme of the simplified dynamic model with the axis gains

Gains improve the system. Theoretically, the cart of the 3DCrane must now move to the position in workspace according to the input impulse.

4.3. State-space representation

In order to solve the 3DCrane approximated system, the state-space[16] representation is used. It is the best solution to highly complicated physical systems. Before deriving a linear combination of n variables, the dynamic model derived in previous chapter, must be modified. First of all, we denote the length of the lifted load l as the position of the Z-axis z_w. Using the

[16] State-Space – mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations.

approximations we made in previous chapter the position of the payload equations (4.8), (4.9) and (4.10) takes form:

$$x_p = x_w + z_w\beta \tag{4.35}$$

$$y_p = y_w + z_w\alpha \tag{4.36}$$

$$z_p = -z_w \tag{4.37}$$

The velocity of the payload can be easily found by taking the first derivative of the equations (4.35), (4.36) and (4.37):

$$\dot{x}_p = \dot{x}_w + \dot{z}_w\beta + z_w\dot{\beta} \tag{4.38}$$

$$\dot{y}_p = \dot{y}_w + \dot{z}_w\alpha + z_w\dot{\alpha} \tag{4.39}$$

$$\dot{z}_p = -\dot{z}_w \tag{4.40}$$

The acceleration of the payload is therefore the second derivative of (4.35), (4.36) and (4.37):

$$\ddot{x}_p = \ddot{x}_w + \ddot{z}_w\beta + 2\dot{z}_w\dot{\alpha} + z_w\ddot{\alpha} \tag{4.41}$$

$$\ddot{y}_p = \ddot{y}_w + \ddot{z}_w\alpha + 2\dot{z}_w\dot{\alpha} + z_w\ddot{\alpha} \tag{4.42}$$

$$\ddot{z}_p = -\ddot{z}_w \tag{4.43}$$

By substituting (4.41) – (4.43) to (4.19) – (4.23) we obtain five nonlinear differential equations describing the system. The variables x_w, y_w, z_w, α, β are unknown variables.

$$\ddot{x}_w + \ddot{z}_w\beta + 2\dot{z}_w\dot{\alpha} + z_w\ddot{\alpha} = \frac{N_z}{m_l}\beta \tag{4.44}$$

$$\ddot{y}_w + \ddot{z}_w\alpha + 2\dot{z}_w\dot{\alpha} + z_w\ddot{\alpha} = \frac{N_z}{m_l}\alpha \tag{4.45}$$

$$\ddot{z}_w = \frac{-N_z}{m_l} \tag{4.46}$$

$$\ddot{x}_w = \frac{N_x + N_z\beta}{(m_c + m_r)} \tag{4.47}$$

$$\ddot{y}_w = \frac{N_y + N_z\alpha}{m_c} \tag{4.48}$$

The smallest set of linearly independent system variables such that the values of the members of the set at time t_0 along with known forcing functions completely determine the value of all system variables for all $t \geq t_0$ are called state variables [5]. The linearization of nonlinear system in state variable form is presented as follows:

$$x_1 = x_w \tag{4.49}$$
$$x_2 = \dot{x}_w \tag{4.50}$$
$$x_3 = y_w \tag{4.51}$$
$$x_4 = \dot{y}_w \tag{4.52}$$
$$x_5 = \beta \tag{4.53}$$
$$x_6 = \dot{\beta} \tag{4.54}$$
$$x_7 = \alpha \tag{4.55}$$
$$x_8 = \dot{\alpha} \tag{4.56}$$
$$x_9 = z_w \tag{4.57}$$
$$x_{10} = \dot{z}_w \tag{4.58}$$

A vector whose elements are state variables is called state vector [5]. It is expressed as the $n \times 1$ vector $x = [x_1\ x_2\ ...\ x_n]^T$. The variables measured by the encoders of the 3DCrane form the outputs $m \times 1$ vector $y = [y_1\ y_2\ ...\ y_m]^T$. In our case the outputs are known as x_w, y_w, z_w, α, β. Taking into account equations (4.49) – (4.58) the output function can be expressed as followed:

$$y_1 = x_w = x_1 \tag{4.59}$$
$$y_2 = y_w = x_3 \tag{4.60}$$
$$y_3 = \beta = x_5 \tag{4.61}$$
$$y_4 = \alpha = x_7 \tag{4.62}$$
$$y_5 = z_w = x_9 \tag{4.63}$$

The input variables are control forces N_x, N_y, N_z. Together they form $p \times 1$ input vector $u = \left[u_1\ u_2\ ...\ u_p\right]^T$, where p is the number of inputs. A set of n simultaneous, first-order differential equations with n variables, where the n variables to be solved are the state variables called state equations [5]. The state derivatives are obtained taking into account (4.44) – (4.48). The simplified system of state equations for the 3DCrane is:

$$\dot{x}_1 = \dot{x}_w = x_2 \tag{4.64}$$

$$\dot{x}_2 = \ddot{x}_w = \frac{N_x + N_z x_5}{(m_c + m_r)} \tag{4.65}$$

$$\dot{x}_3 = \dot{y}_w = x_4 \tag{4.66}$$

$$\dot{x}_4 = \ddot{y}_w = \frac{N_y + N_z x_7}{m_c} \tag{4.67}$$

$$\dot{x}_5 = \dot{\beta} = x_6 \tag{4.68}$$

$$\dot{x}_6 = \ddot{\beta} = \frac{2\frac{N_z}{m_l}x_5 - \frac{N_x + N_z x_5}{(m_c + m_r)} - 2x_{10}x_6}{x_9} \tag{4.69}$$

$$\dot{x}_7 = \dot{\alpha} = x_8 \tag{4.70}$$

$$\dot{x}_8 = \ddot{\alpha} = \frac{2\frac{N_z}{m_l}x_7 - \frac{N_y + N_z x_7}{m_c} - 2x_{10}x_8}{x_9} \tag{4.71}$$

$$\dot{x}_9 = \dot{z}_w = x_{10} \tag{4.72}$$

$$\dot{x}_{10} = \ddot{z}_w = -\frac{N_z}{m_l} \tag{4.73}$$

A system is represented in state-space by the equations (2.2). It is therefore necessary to find state matrices A, B, C and D. The matrices components are formed with the components expressed as followed:

$$a_{ij} = \frac{\partial \dot{x}_i}{\partial x_j}, \qquad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n \tag{4.74}$$

$$b_{ij} = \frac{\partial \dot{x}_i}{\partial u_j}, \qquad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, p \tag{4.75}$$

$$c_{ij} = \frac{\partial y_i}{\partial x_j}, \qquad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \tag{4.76}$$

$$d_{ij} = \frac{\partial y_i}{\partial u_j}, \qquad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, p \tag{4.77}$$

All the differential equations for the state-space matrices elements can be easily calculated by the MATLAB operation *diff* (Annex 2). However obtaining the matrices elements does not provide the final solution. The A, B, C and D matrices must be represented taking into account the steady-state operation point $x^0 = \left[x^0\ \dot{x}^0\ y^0\ \dot{y}^0\ \beta^0\ \dot{\beta}^0\ \alpha^0\ \dot{\alpha}^0\ l^0\ \dot{l}^0\right]^T = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ and input $u^0 = [0\ 0\ 0]^T$, because all the derivatives of the state variables, measured variables and input forces must be equal to 0, when the expected position of the cart is obtained.

Note that the approximation states that the angles of the payload are very small. Their values are therefore taken equal to 0. In simulation they will show no oscillation in output. In real system however the output angles must show the nonlinear behaviour: here the oscillation is expected. In order to control and simulate the state-space representation of the 3DCrane, the obtained solution must be adequate.

4.4. State-space representation simulation

In order to simulate the state-space matrices in Simulink the *State-space* block can be used. Simulink® software converts a matrix containing zeros to a sparse matrix for efficient multiplication [19]. The specification of the state matrices inside of the block is necessary. The state-space matrix model is presented in Figure 4.8.

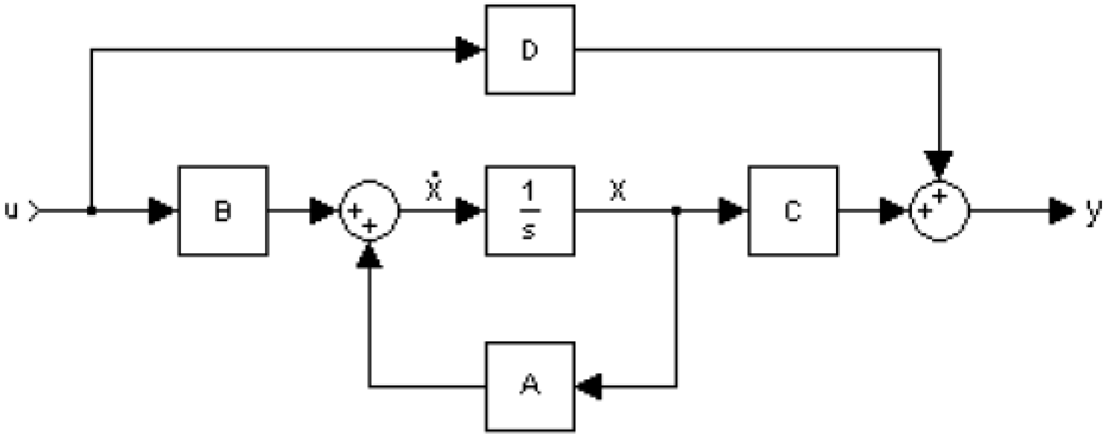


Fig. 4.8. State-space representation model

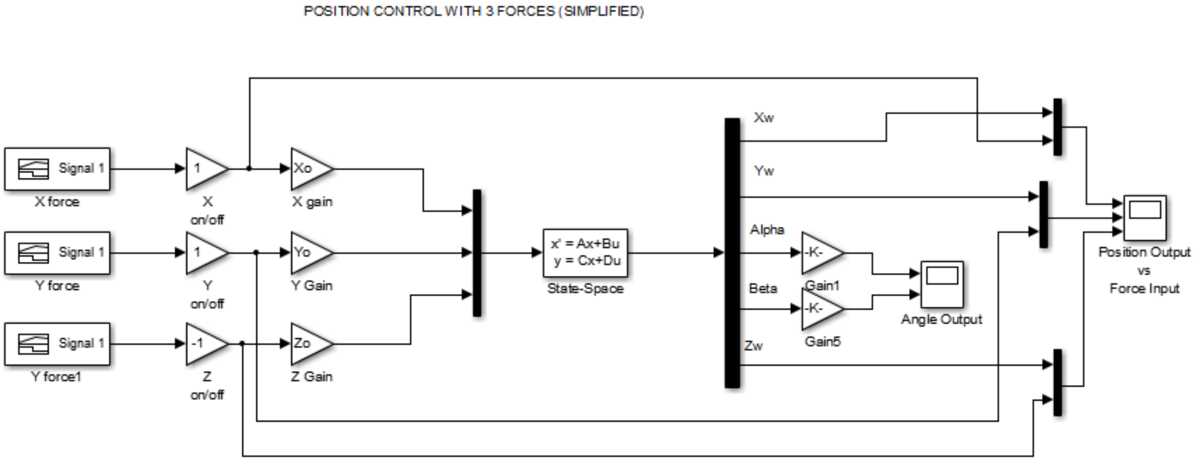
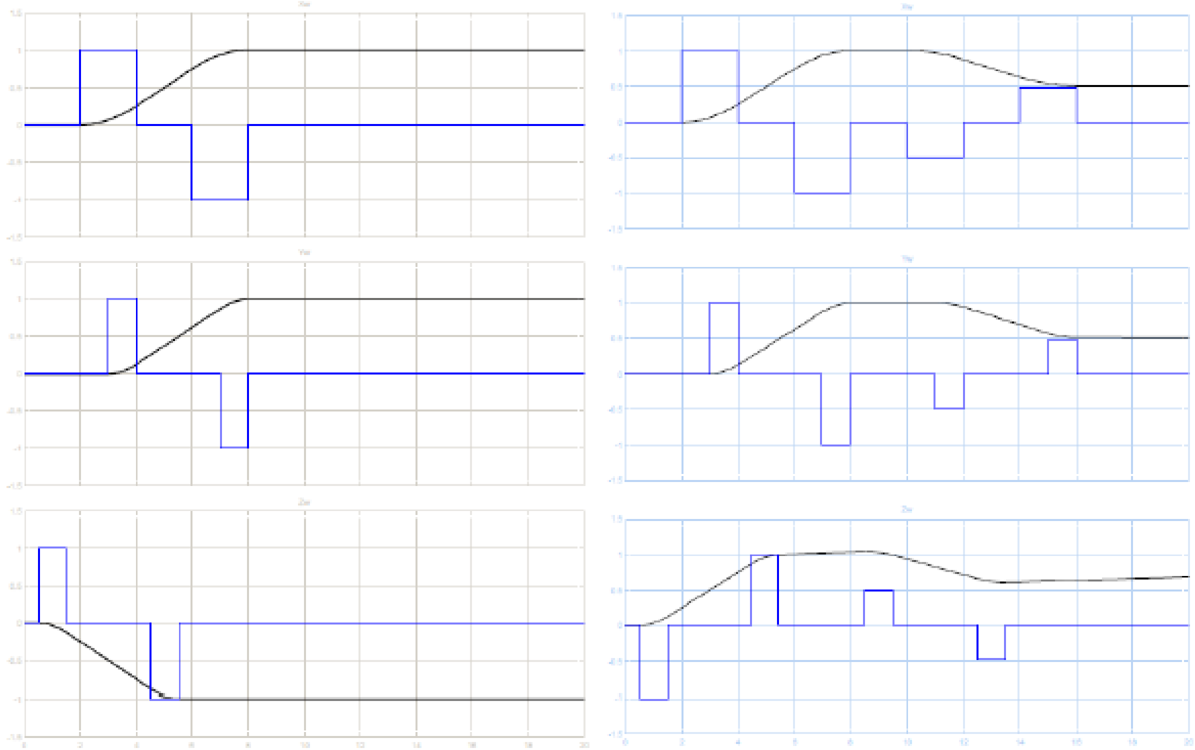


Fig. 4.9. State-space representation Simulink model

The state-space representation of the 3DCrane can be therefore simulated. The control scheme is shown in Figure 4.9. The input control forces are taken as described in Section 4.1.4. Important to note that there are also gains used in every axis force. However, the gains are slightly different from the ones obtained in Section 4.1.4. (Annex 2).

As an output we must have variables as we set in (4.59) – (4.63). The scope therefore compares input forces with the output positions. Two different input forces are presented in Figure 4.10. The black curve represents the position output and the blue one shows the input force impulse. The behaviour in Figure 4.10 (a) describes a single impulse. It is clearly seen that the model obtains the necessary position on every axis. Moreover, Z-axis goes in opposite direction, because the axis has been chosen up as the positive direction (Figure 4.1). For the simplicity the Z-axis gain was multiplied by -1.



(a) (b)

Fig. 4.10. Inputs (blue) and outputs (black) of the state-space representation: (a) single impulse, (b) double impulse

In Figure 4.10 (b) double impulse control is presented. According to this input, the cart must move to two different positions during the simulation time. Moreover, it is important to note, the output is absolutely identical to the simplified model simulation result (Figure 4.6 (b)) done in Section 4.2.

4.5. Chapter summary

The chapter describes the mathematical modelling of the 3DCrane system. The Newton's second law of motion is used to describe the dynamics and trigonometry representation used to state the position of the payload of the system.

Although the system is highly nonlinear MIMO, the trigonometrical approximation (4.26) – (4.29) can be used to simplify the model. The obtained dynamic model is a system of nonlinear differential equations of second order. Taking the derived equations and approximation into account, the Simulink model is therefore built. The simulation results are shown in Figure 4.6. In order to stabilize the system an amplification factors were added.

This chapter has also dealt with the state-space representation of the 3DCrane model. The dynamic equations and approximation described previously are used. The system has 10 state variables, 3 inputs and 5 outputs. The simulation results are presented in Figure 4.10. The provided solution makes the system behave as expected.

Both models developed are very similar. Indeed, they were built using the same dynamic equations and approximations. Nevertheless, the models cannot completely replace the real 3DCrane model. Due to the reason, the approximation excludes the angles of the payload. Any way, they can be used as the sensorless position control. The next chapter compares derived model and the Inteco® original 3DCrane model that depends on the encoders.

5. MODELS COMPARISON

The chapter deals with the mathematically obtained model comparison with the 3DCrane manufacturer model based on the encoders feedback. It is important to find out, how accurate was the model derived comparing to the real system. What is the difference between them and if true what cause the difference. The result can therefore exclude the encoders from the system. Also, it will provide the sensorless position control of the 3DCrane.

5.1. State-space representation and Inteco® model comparison

As soon as the 3DCrane modelled and the state-space solution are obtained, it is important to compare the theoretical model with the real 3DCrane model provided by the Inteco®. Both state-space and Inteco® models are placed in one Simulink® workspace window (Figure 5.1). The same input is given to both models. Three scopes are added to view the input, the Inteco® output of the model and the output of the state-space representation. The comparison of the results is presented in Figure 5.2. The blue line shows the state-space representation output, the black line - Inteco® model output and the red line shows the input force.

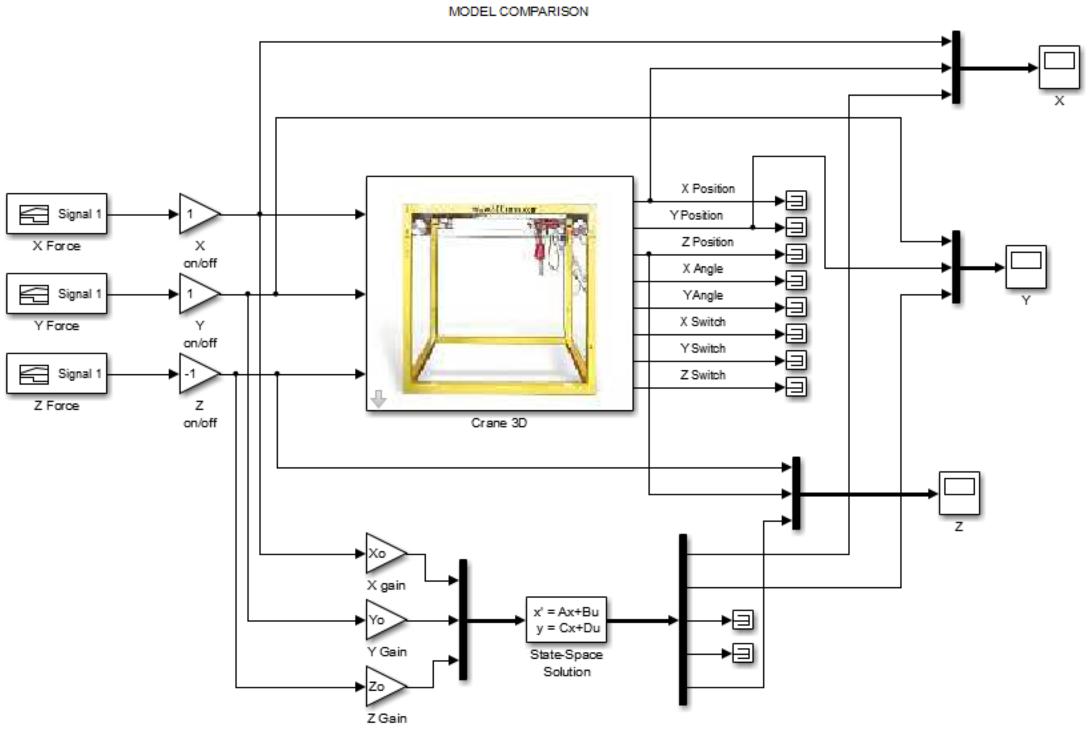


Fig. 5.1. Inteco® and state-space models in Simulink® workspace window

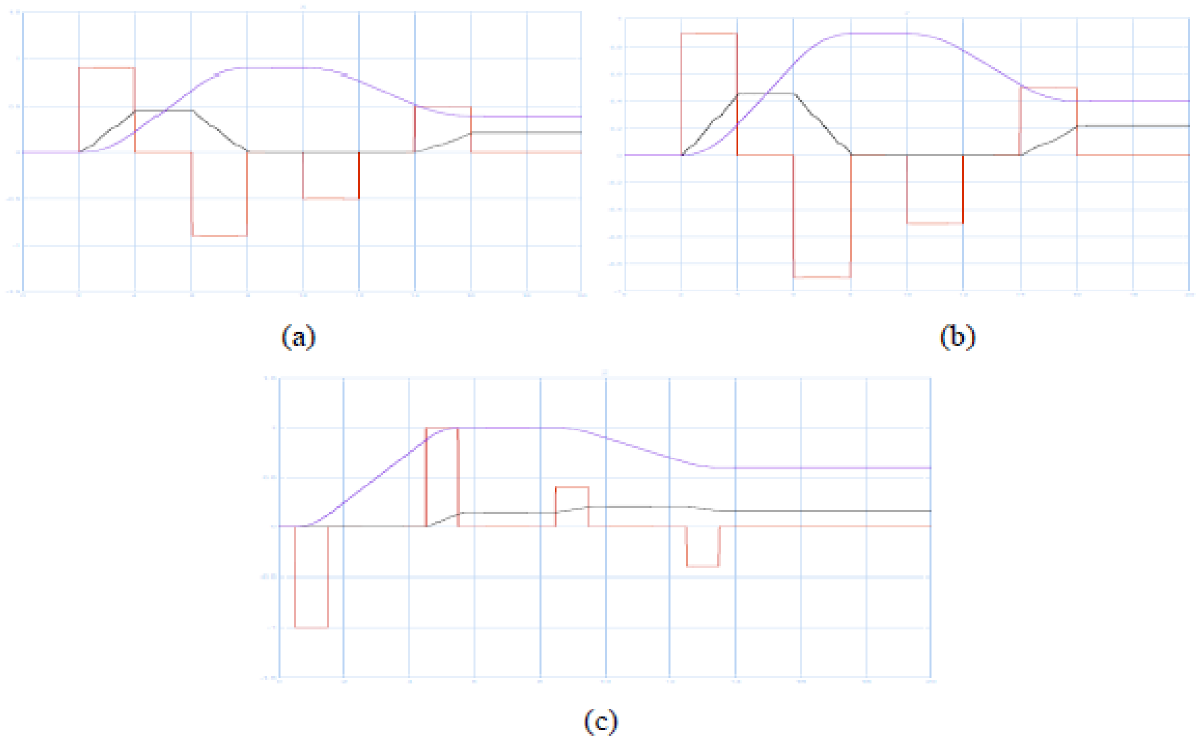


Fig. 5.2. Input forces (red), output of the Inteco® model (black) and stat-space output (blue): (a) X-axis, (b) Y-axis, (c) Z-axis

It can be clearly seen that the Inteco® model output slightly differs from the state-space representation output. First of all, it does not obtain an expected value of the workspace position. Secondly, the output scope in Inteco® model does not behave as the position output curve.

The Inteco® model output curve theoretically looks more like a speed curve. Moreover, during the experimentation, the 3DCrane trajectory also behaves as the impulse input. It makes the cart, rail and the load of the crane move with the speed in workspace, however, not to move to expected position. Therefore, it has been concluded that the 3DCrane model provided by the Inteco® does not have the position control at the output, as it is said in the model. More likely, it gives the speed of the movement along the according axis.

Thus, it is necessary to provide the next experimentations adding the *Integrator* block to the input of the Inteco® model (Figure 5.3). The block must integrate the input acceleration forming the speed as the input to the model block. The solution expects the position curve as the output of the model.

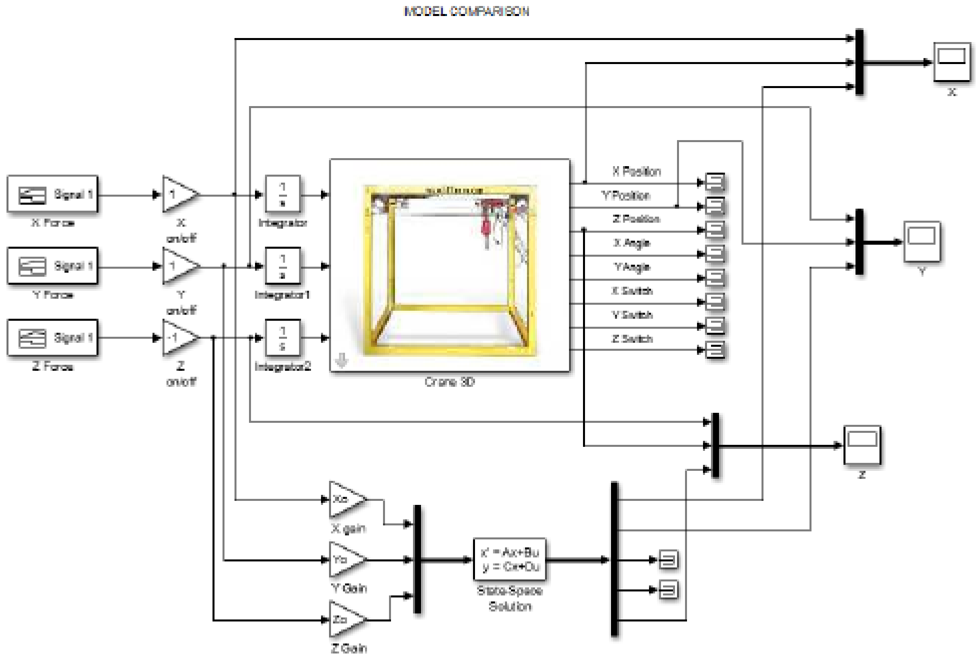
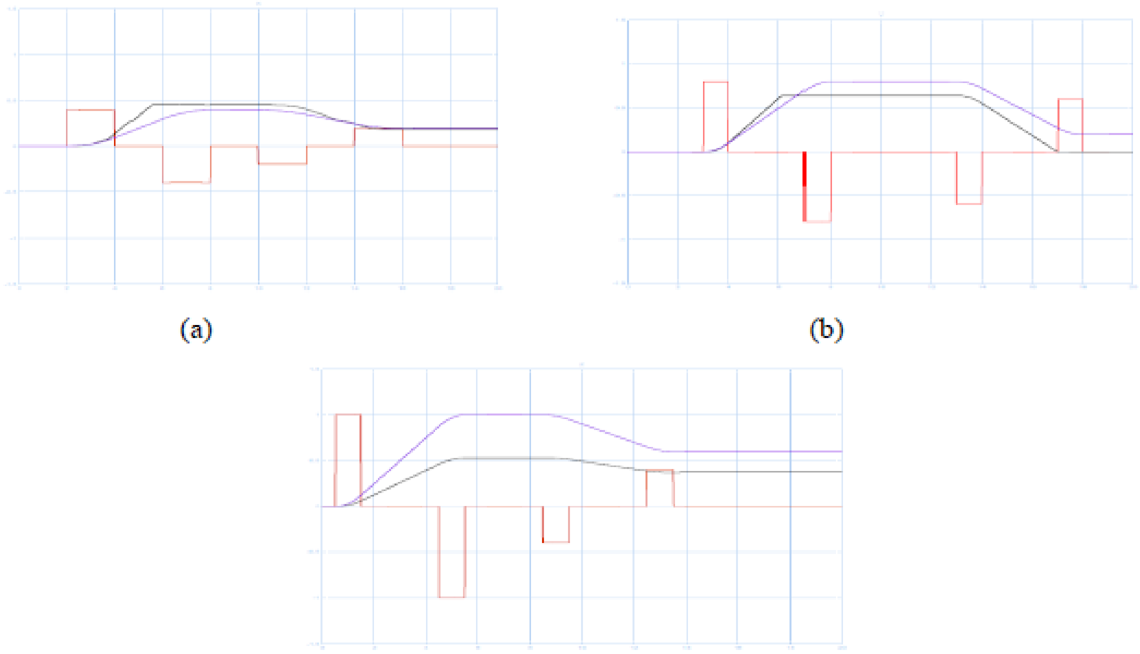


Fig. 5.3. Modified comparison model in Simulink® workspace window

The simulation results are provided in Figure 5.4. Again, the blue line represents the state-space representation output, the black line - Inteco® model output and the red line shows the input force. Moreover, it has been decided to reduce the input force value due to avoid the crane workspace position limitations.



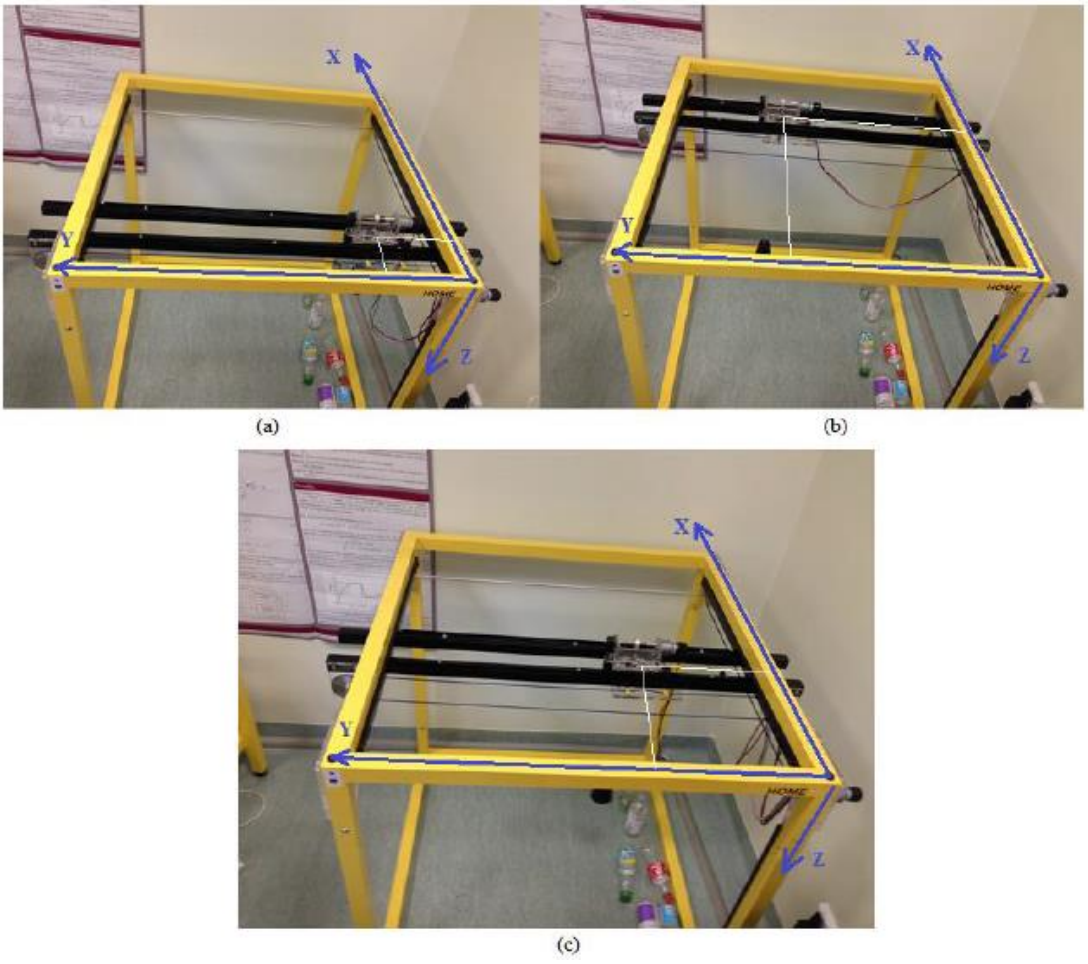
(a)

(b)

(c)

Fig. 5.4. The modified models comparison input force (red), output Inteco® model position (black) and output state-space (blue): (a) X-axis, (b) Y-axis, (c) Z-axis

According to the results obtained during the last simulation (Figure 5.4), the Inteco® 3DCrane model now has a position curve as an output. Moreover in Figure 5.5 it is shown the physical positions of the 3DCrane during the experimentation. On the picture (a) the crane cart is in the corner of the workspace frame (*Home* position). The Figure (b) shows the location of the cart in XY-axis. It can be seen that the positions are respectively located according to the pulse given at the input for X and Y axis. Finally, the Figure (c) shows the position of the crane closer to the *Home* position in both X and Y axis. The second pulses are also lower than the first ones, what explains this behaviour.



(a)

(b)

(c)

Fig. 5.5. The physical position of the 3DCrane in workspace during the experimentation: (a) home position of the crane, (b) first input impulse, (c) second input impulse

The control design logic is easier to obtain having the position as an output. The position is visionary easier to imagine comparing to the moving speed of the crane. However, state-space output does not match completely the output of the crane obtained from the encoders. Several reasons can cause this difference.

First of all the workspace is not exactly 1000x1000x1000 mm. According to measurements it is 920x920x820mm, the control is however designed in variation from -1 to 1. Secondly, the state-space model may not be accurate and causes the error. Nevertheless, it has been stated in Section 2.1 of Chapter 2 that the models are always ideal. Thus, it is logic that two model output differ one with another. Finally, the encoders may also accumulate the error. The encoder may not be properly calibrated by the time the experimentation was held or it has lost its accuracy during the exploitation.

5.2. Chapter summary

This chapter concludes the model derivation. The steady-state representation of the 3DCrane model was compared with the real model of the 3DCrane provided by the manufacturer Inteco®. As a result, the steady-state representation does not behave as a real 3DCrane model. Nevertheless, it has been concluded that manufacturer model does not provide the position as an output. The output looks more like the speed of the cart and payload along the axis (Figure 5.2). It has been also stated in Chapter 3.

The second experiment was carried adding the *Integrator* to every input of the Inteco® model. As a result, both models behaviour are similar to each other. It proves again, that manufacturer has a speed as an output of the model. Theoretically, by integrating the input force we must obtain the speed.

However, the state-space representation and Inteco® model still has the errors. Probable reason for the mismatch can be the state-space representation. The solution may be incorrect. The model is also ideal and differs from the real behaviour. Furthermore, the workspace is not exactly 1000x1000x1000 mm. Finally, it can be explained with the error accumulated in the encoders. Still the derived model can be used as a sensorless position control for the 3DCrane. As it was stated before, the model, unfortunately, cannot replace the encoder for the payload angles. Another solution has to be found due to model and simulate the payload angles respond and full dynamic model of the system.

6. THE FULL MODEL DEVELOPMENT AND SIMULATION

The full model of the 3DCrane is 5 degree of freedom system. 3 degrees of freedom are related to the X, Y and Z axis. Two more degrees of freedom are related to the angles of the payload lifting the line α and β (Figure 4.1).

The full model of the system is important in order to observe the behaviour of the payload. If the angles of the lift-line is possible to predict, before building the control system, the sensor measuring the angles can be removed. In order to obtain the dynamics equations of the system, the Lagrangian[17] approach for the dynamic system will be used in this chapter.

6.1. Dynamic model of the 3DCrane using the Lagrangian approach

Again as in Chapter 4.1, where the simplified model has been derived, we start with the mathematical representation of the payload position (4.8) – (4.10). The dynamic summary of the system will be obtained using the Lagrangian approach. In classical mechanics, the natural form of the Lagrangian is defined as the kinetic energy K of the system minus its gravitational potential energy P [20]:

$$L = K - P \tag{6.1}$$

$$K = \frac{1}{2}mv^2, \tag{6.2}$$

where m – mass,

v – speed.

$$P = mgl, \tag{6.3}$$

where m – mass,

g – acceleration of gravity,

l – length.

[17] Joseph-Louis Lagrange (1736 - 1813) – Italian mathematician and astronomer.

In this case K is the kinetic energy of the crane and its load. P defines the potential energy of the load. The precise derivation of the kinetic and potential energies as well as the full Lagrangian representation of the dynamic system is shown in Annex 3.

If the Lagrangian of a system is known, then the equation of motion of the system may be obtained by a direct substitution of the expression for the Lagrangian into the Euler[18]-Lagrange equation [20]. The Euler-Lagrange equation is a second-order partial differential equation, whose solutions are the function for which a given function is stationary [21].

The Euler-Lagrange formulation of the dynamic system is as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i, \tag{6.4}$$

where $i = 1,2,3,4,5$ – degree of freedom,

L – the Lagrangian of the dynamic system,

q_i – state variable,

F_i – force.

Derivation of the Euler-Lagrange formulas for each degree of freedom of the system is described as well in Annex 3. The final result of the dynamic system for the 3DCrane is as follows:

$$\begin{cases} (m_c + m_r + m_l + m_l)\ddot{x} + m_l\ddot{l}\cos\alpha\sin\beta - 2m_l\dot{l}\dot{\alpha}\sin\alpha\sin\beta + 2m_l\dot{l}\dot{\beta}\cos\alpha\cos\beta - m_l l\ddot{\alpha}\sin\alpha\sin\beta - \\ -m_l l\dot{\alpha}^2\cos\alpha\sin\beta - 2m_l\dot{l}\dot{\alpha}\dot{\beta}\sin\alpha\cos\beta + m_l l\ddot{\beta}\cos\alpha\cos\beta - m_l l\dot{\beta}^2\cos\alpha\sin\beta = u_x \\ (m_c + m_l + m_l)\ddot{y} + m_l\ddot{l}\sin\alpha + 2m_l\dot{l}\dot{\alpha}\cos\alpha + m_l l\ddot{\alpha}\cos\alpha - m_l l\dot{\alpha}^2\sin\alpha \quad = u_y \\ 2m_l\ddot{l} + m_l\dot{x}\cos\alpha\sin\beta + m_l\ddot{y}\sin\alpha - m_l\dot{l}\dot{\beta}cos^2\alpha - m_l\dot{l}\dot{\alpha}^2 - m_l g\cos\alpha\cos\beta = u_z \\ -m_l\ddot{x}l\sin\alpha\sin\beta + m_l\ddot{y}l\cos\alpha + m_l l^2\ddot{\alpha} + 2m_l l\dot{l}\dot{\alpha} + m_l gl\cos\alpha\sin\beta = 0 \\ m_l\ddot{x}l\cos\alpha\cos\beta - 2m_l l^2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha + 2m_l l^2\ddot{\beta}cos^2\alpha + 2m_l l\dot{l}\dot{\beta}cos^2\alpha + m_l gl\sin\alpha\cos\beta = 0 \end{cases} \tag{6.5}$$

Here the input forces for the 3DCrane are stated as u_x, u_y and u_z. Character u is related to the input in control system theory.

[18] Leonhard Euler (1707 - 1783) – Swiss mathematician and physicist.

State-space representation will be used to build the model of the system. Again, the A, B, C and D matrices must be represented. In order to diminish the dynamic system (6.5), the steady-state operation point $x^0 = \left[x^0\ \dot{x}^0\ y^0\ \dot{y}^0\ \beta^0\ \dot{\beta}^0\ \alpha^0\ \dot{\alpha}^0\ l^0\ \dot{l}^0\right]^T = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ can be used.

In Annex 3 nonlinear differential equations of second order describing the dynamics of the system is found. The final model taking into account the steady-state operational point is as follows:

$$\begin{cases} \ddot{x} = \frac{1}{m_c+m_r+m_l}u_x + \frac{m_l g}{m_c+m_r+m_l}\beta \\ \ddot{y} = \frac{1}{m_c+m_l}u_y + \frac{m_l g}{m_c+m_l}\alpha \\ \ddot{\beta} = -\frac{1}{l(m_c+m_r+m_l)}u_x - \frac{g(m_c+m_r+m_l+m_l)}{l(m_c+m_r+m_l)}\beta \\ \ddot{\alpha} = -\frac{1}{l(m_c+m_l)}u_y - \frac{g(m_c+m_l+m_l)}{l(m_c+m_l)}\alpha \\ \ddot{l} = \frac{1}{2m_l}u_z \end{cases} \tag{6.6}$$

The linearization of nonlinear system in state variable form is presented in (4.49) – (4.58). The output is presented in (4.59) – (4.63). The input of the system is presented as forces vector $u = \left[u_x\ u_y\ u_z\right]^T$. Finally, the state derivatives are given by:

$$\dot{x}_1 = \dot{x}_w = x_2 \tag{6.7}$$

$$\dot{x}_2 = \ddot{x}_w = \frac{1}{m_c+m_r+m_l}u_x + \frac{m_l g}{m_c+m_r+m_l}\beta \tag{6.8}$$

$$\dot{x}_3 = \dot{y}_w = x_4 \tag{6.9}$$

$$\dot{x}_4 = \ddot{y}_w = \frac{1}{m_c+m_l}u_y + \frac{m_l g}{m_c+m_l}\alpha \tag{6.10}$$

$$\dot{x}_5 = \dot{\beta} = x_6 \tag{6.11}$$

$$\dot{x}_6 = \ddot{\beta} = -\frac{1}{l(m_c+m_r+m_l)}u_x - \frac{g(m_c+m_r+m_l+m_l)}{l(m_c+m_r+m_l)}\beta \tag{6.12}$$

$$\dot{x}_7 = \dot{\alpha} = x_8 \tag{6.13}$$

$$\dot{x}_8 = \ddot{\alpha} = -\frac{1}{l(m_c+m_l)}u_y - \frac{g(m_c+m_l+m_l)}{l(m_c+m_l)}\alpha \tag{6.14}$$

$$\dot{x}_9 = \dot{z}_w = x_{10} \tag{6.15}$$

$$\dot{x}_{10} = \ddot{z}_w = \frac{u_z}{2m_l} \tag{6.16}$$

Together with the simulation script the state-space A, B, C and D matrices are found in Annex 4. This model allows observing and studying the full model of the 3DCrane. The angles of the payload can be simulated and analysed.

6.2. Model simulation and comparison with the real 3DCrane system

Again as in Chapter 5 Figure 5.1 both the real 3DCrane and state-space representation models will be connected in Simulink®. However, this time we are also interested in α and β angles. Thus, the additional scopes will be added to observe and study the inputs and outputs (Figure 6.1).

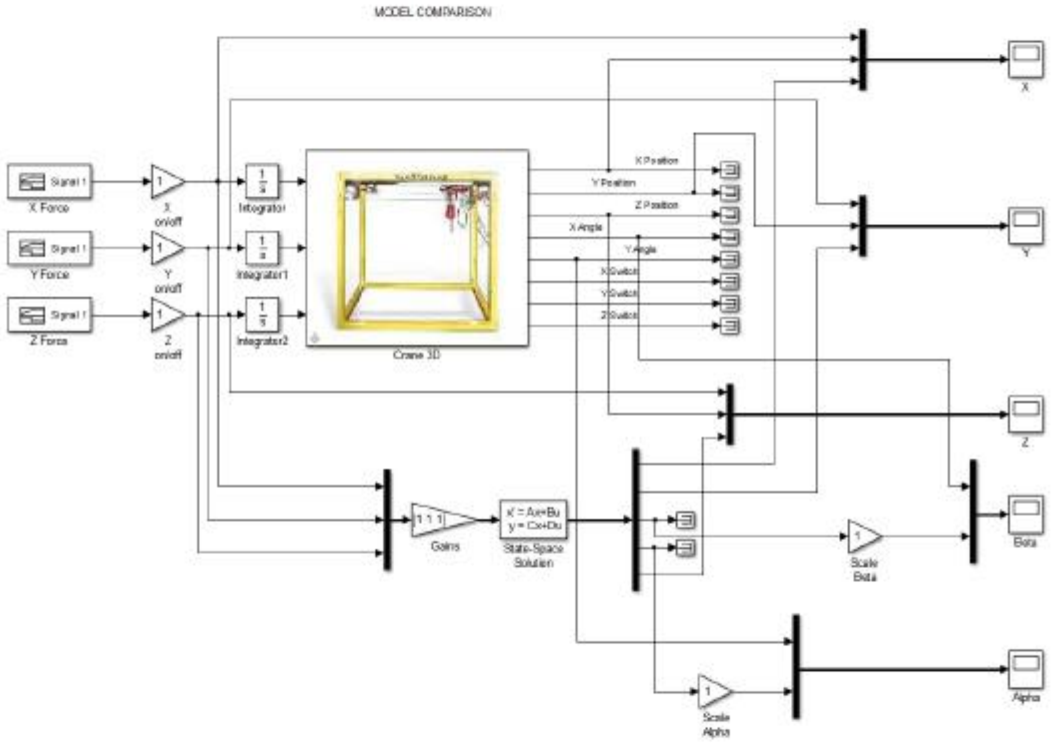
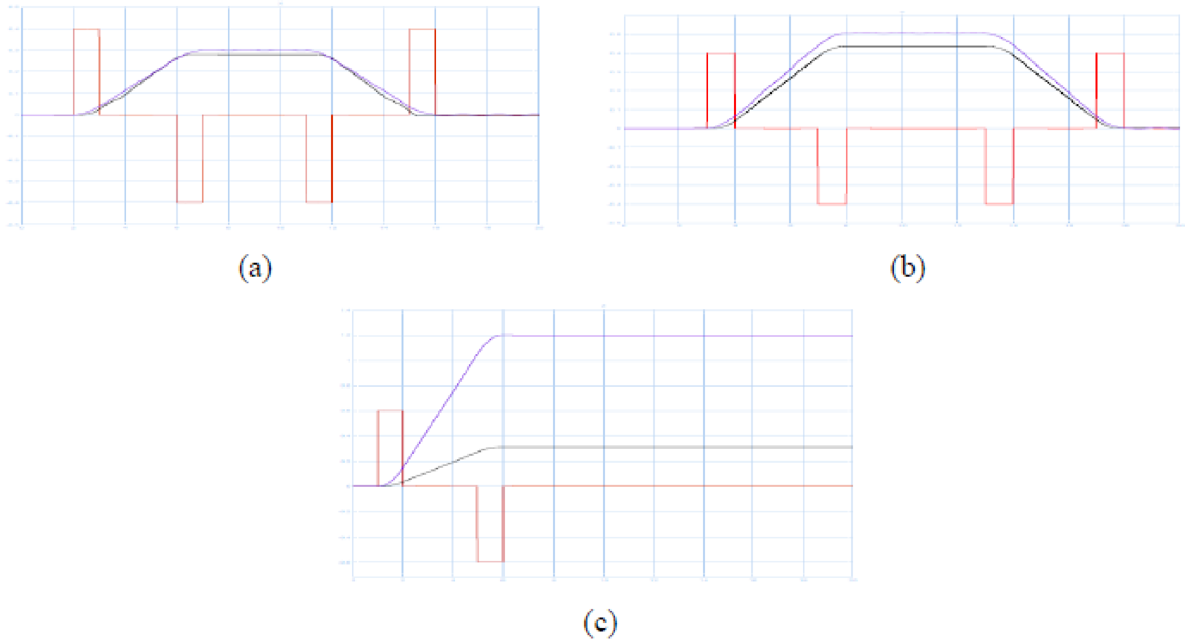


Fig. 6.1. Inteco® and state-space models in Simulink® workspace window

However, it is important to state that the length of the lift-line l must be entered as a constant value (Annex 4 line 7). Even if the position of the Z-axis is controlled variable, the other dynamic model variables still depend on l (6.6). In simulation of the length of the lift-line $l = 0.6\ mm$. The value is expected according to the control input. It is assumed that if the input force is equal to $0.6\ N$, the position of the load along the Z-axis should be 0.6 mm as well. Thus the length is also 6 mm.

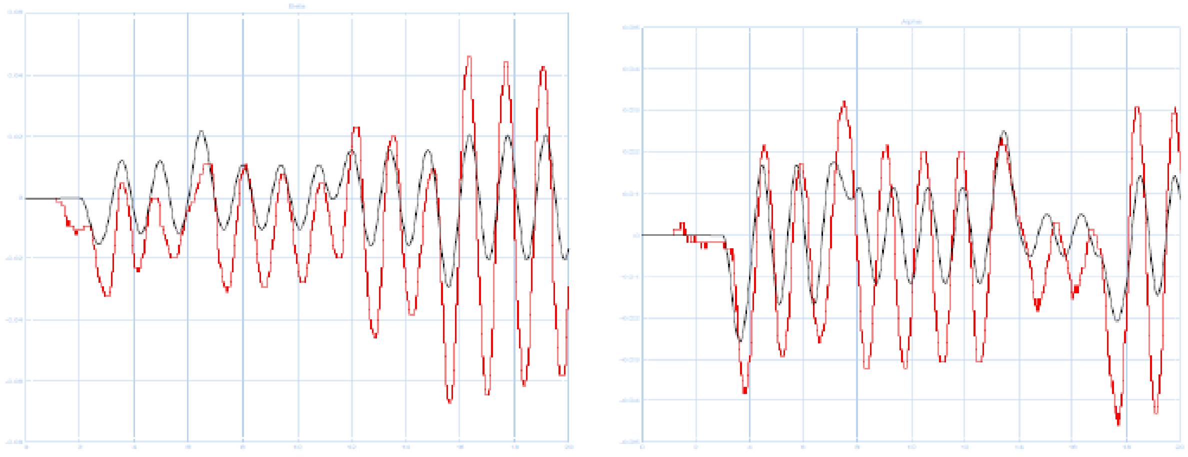
The simulation results are provided in Figures 6.2. and 6.3. Figure 6.2 presents the position control along the X, Y and Z-axis. Figure 6.3 shows the angles of the payload encoders and model output. It can be clearly seen that the axis control (Figure 6.2) has some small error along the X and Y-axis comparing to the encoders output. Z-axis however has a very big error between two models. Moreover, it is clear that Z-axis encoder is far from the expected output.



(a) (b)

(c)

Fig. 6.2. Input forces (red), output of the Inteco® model (black) and stat-space output (blue): (a) X-axis, (b) Y-axis, (c) Z-axis



(a) (b)

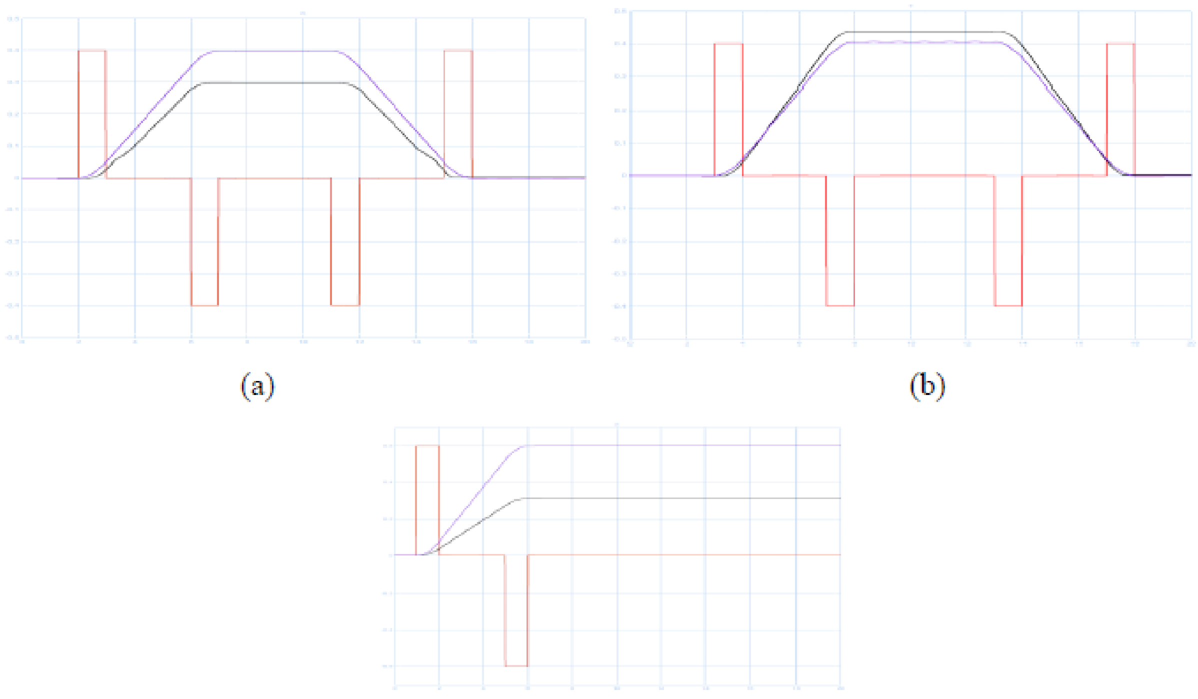
Fig. 6.3. Output of the Inteco® model (red) and stat-space output (black): (a) β (b) α

In Figure 6.3 α and β outputs are presented. It can be concluded that the state-space model is not precise. The periods of load oscillations almost match the encoder values. However, the amplitude is different between the practical and theoretical results.

It can be presumed that many dynamics such as viscose friction coefficient and encoder dynamic are neglected while modelling the system. Thus, the derived model of the 3DCrane is ideal. Moreover, the 3DCrane frame can also be tilted, what may cause the model output errors. Finally, it must be also considered the encoder error or the mistakes and approximation in modelling the state-space representation.

6.3. Tuning the system

According to the results presented in Figures 6.2 and 6.3 the output respond still has some errors. While the output position respond is easy to tune simply dividing the output steady-state value by the input force value, the correction gain for the angles is hard to find. Annex 4 line 9 shows the input matrix calculation. In Figure 6.4 the tuned position responds are presented.



(a) (b)

(c)

Fig. 6.4. Input forces (red), output of the Inteco® model (black) and stat-space output (blue) with the correction gain: (a) X-axis, (b) Y-axis, (c) Z-axis

The positions along the X, Y and Z-axes are designed to follow according the input force. The state-space model is also tuned according to the design. However, it can be seen (Figure 6.4) that there is still error between the encoders output and the state-space representation of the model.

As for the angles, the input force does not influence the angles outputs. The respond is the same as it is shown in Figure 6.3. Nevertheless, as the frequencies of the angles are the same, there can be several reasons why the encoders and the state-space representation do not match each other in amplitude:

1. The dynamic model is simplified.
2. Derived model is ideal and exclude much dynamics.
3. The influence of other components such as encoder attached to the side of the cart moving along the Y-axes is neglected in model (Figure 6.5).
4. The 3DCrane frame can be tilted.
5. Encoder error.
6. Other problems.

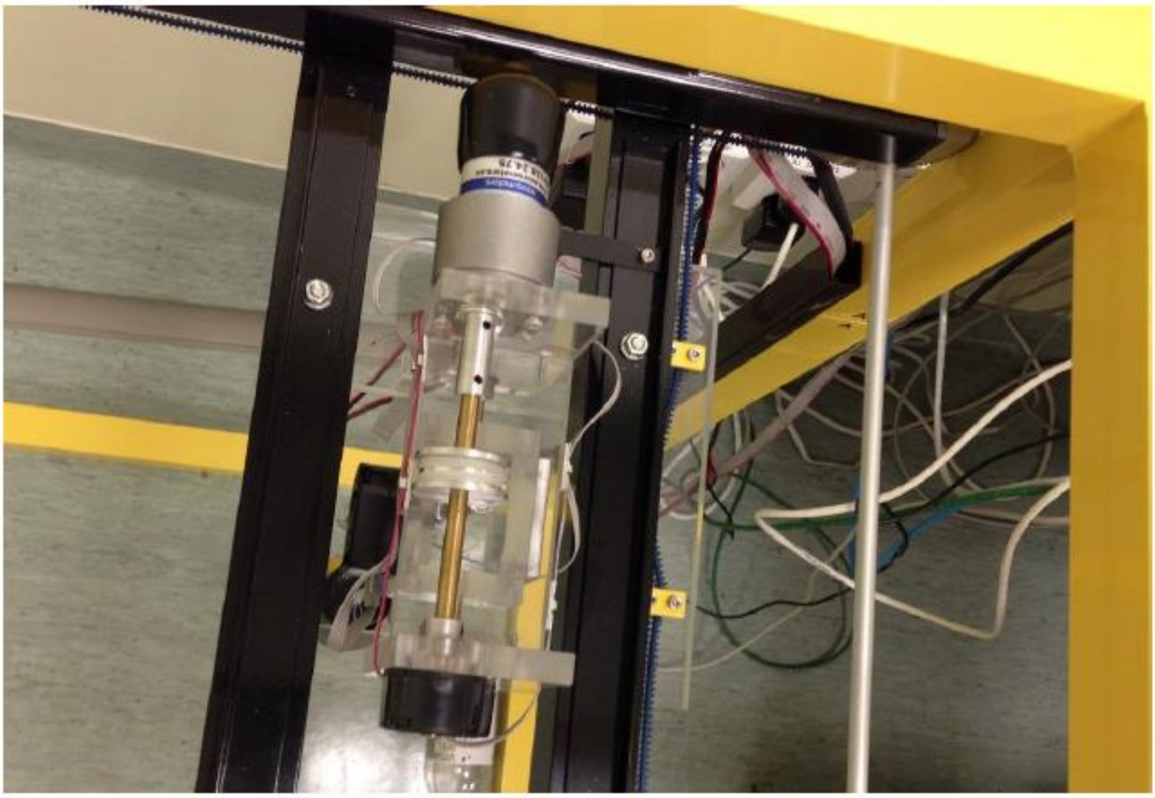


Fig. 6.5. The cart components that might cause an additional dynamic

6.4. Chapter summary

The chapter outlines here the full modelling and simulation of the 3DCrane. The Euler-Lagrange approach was used in order to derive the dynamic model for the system. The full derivation is presented in Annex 3.

The state-space representation of gain dynamics are presented and simulated in this chapter. The state-space matrices are compared to the real 3D crane behaviour. The state-space model can completely replace all the encoders in the system. However, the practical results do not completely match the state-space representation output.

We found that there is still small error between derived model and encoder results. There can be several reasons for this error. Again the model has been approximated, many dynamics has been neglected, and the encoder may not be calibrated by the moment of experimentation. Nevertheless, the control system can be still used for the operation of the 3DCrane.

The correction gain used to match the output position respond. The correction gain allows to compensate the approximated and neglected dynamics. The correction gain is obtained dividing the output respond by the expected respond value. The positioning results are presented in Figure 6.4. The respond of the state-space representation obtains the expected output with the correction gain.

Notwithstanding, the modelled angles are far from the encode results. Although the amplitude is not the same comparing the 3DCrane model and state-space representation, the frequencies of the oscillation for both α and β angels are very similar.

7. CONCLUSION

Automation industry is not the field, what human has built. We find many automated systems around us: in our body, in weather, in nature in general. We try to understand and use the knowledge of automation and control design in our everyday life. We want to make our life more safe, easy and cheap. Automation and control engineering is all around us. We use devices and robots to replace the routine and dangerous work. We cannot imagine our life without a number of systems, what are the products of automation.

The study object is small dimension model of 3D crane (also called overhead crane). The 3D cranes are widely used in the industry for vessel loading and unloading, to handle full or empty containers and lift up to 80 tons, automotive lines, ship and airplane building, power industry and other heavy manufacturing. Many researches has been made in order to automate the 3D cranes. It will help to avoid accidents and maybe replace the operator.

The provided model of 3D crane is based on the encoders feedback. Encoders due to their specification may accumulate errors within exploitation. Encoders must not only be calibrated regularly, but also they increase the cost of the system. It has been therefore decided to derive the model of 3DCrane in order to replace the encoders with the mathematical model. Several advantages may be named with successful solution: the mathematical model is cheaper, it does not require calibrations, it is easy to build the automated control system and the system does not depend on the feedback.

Brief introduction to the control systems engineering opens the thesis. It presents the theoretical implementation of the control and automation engineering. The explanation, how the mathematical modelling, simulation and control design are connected between each other in automation field is shown. Different control systems are shortly described.

The new solution for modelling the 3D crane presented in thesis is the state-space representation of dynamic model of the system. State variables are taken into account while modelling the system. Dynamic system is calculated using the Euler-Lagrange formulation for the dynamic system. Lagrangian equation is calculated balancing the kinetic and potential energies.

Moreover, the final dynamic equations are therefore approximated around the cosines and sines of the payload angles. The system of second order differential equations is obtained.

State-space representation in particular consists of specifying the system's desired poles locations, what are represented in matrices depending on the state variables. The controller consists of state-variables feeding back to the input (Figure 4.8). One of the biggest advantages of applying state-space matrices is that it allows computer simulation. Comparing with many other control designs, state-space allows to design automation using the computer.

In present thesis, state-space has three inputs as control forces for each axis and five outputs: the position of the crane in three-dimensional space x, y, z and angles of the payload. First, the angle between the projection of the lift-line on XY plane and the lift-line α. Second, the angle between the negative direction of the Z-axis and projection of the lift-line on XY plane β. The z coordinate of the payload was also expressed as the length of the lift-line l.

First of all, the dynamic model provided by the 3DCrane manufacturer was simulated and analysed. It has been concluded regarding the results (Figure 3.17) that the Inteco® model has the velocity as an output. Next, the integrator has been added to the input of the system. The behaviour of the crane after that was similar to the position scope. When the derived model and the model provided by the manufacturer were compared, the last model was simulated together with the additional integrator block.

Both derived model outputs are similar to each other due to the reason they are built using the same mathematical model. Anyway, these models cannot completely replace the Inteco® model, because the angles dynamic has been approximated. The output therefore has no changes in the angles output signal. However, the control system may still replace sensorless position control in case angle dynamics are not important.

Next, theoretical and practical models were compared. In Figure 5.4 both Inteco® and derived models output signals versus the input forces are presented. It has been concluded that model does not behave the same, however very close to each other. There can be several reasons for these results. Probable reason for the mismatch can be the rough approximation of the mathematical model, the workspace of the 3DCrane is not exact. Finally, it could be explained by the error accumulated in the encoders.

The next task was to derive the full model of the 3DCrane system in order to observe the dynamics of the payload angles. Here the Euler-Lagrange approach was used. The full explanation of model derivation is outlined in Annex 3.

The simulation results in Figure 6.2 has proven that the full model is close to the real encoder output. However, it can be clearly seen that Z-axis has a large error between outputs (Figure 6.2 (c)). Moreover, the amplitude of the payload angles is not the same comparing to the 3DCrane model and state-space representation (Figure 6.3). The frequency of the oscillation for both α and β angels nevertheless is very similar.

If the angle dynamics is hard to tune, the output position can be tuned with the correction gain for every axis. The results are presented in Figure 6.4. With the correction gain the state-space behaves as expected: the positions moves according to the value of input force. However, the position is limited with the frame dimension and lift-line length.

Several proposals are made to explain the wrong behaviour of the angles dynamics. First, the derived model is ideal and simplified around the payload oscillation. Secondly, some other components may influence the system. For instance, the encoder attached to the side of the cart moving along the Y-axes (Figure 6.5) was neglected in model. The component has the mass and theoretically should not be neglected. Third reason, the 3DCrane frame can be tilted. Finally, encoder error may appear. Moreover, many other reasons can trouble the correct dynamics.

Despite the fact, the real encoder model output is not absolutely the same as in state-space representation, the derived model can still be used as the sensorless control logic. Anyway, in order to implement the theory in real life, in industrial 3D crane some additional devices must be financed. The MATLAB® model must be interfaced with the system due to through the hardware.

In industry the mathematical model of 3D crane may have a very useful impact. Even the solution presented in this thesis may be very beneficial. It can be used to replace the operator in routine tasks.

Furthermore, the derived model can be useful in future study. Different controls can be used in 3DCrane control system before testing it on a real crane. It is always cheaper to predict the behaviour of the system before implement it in the real system. For that is important to have the mathematical representation as accurate as possible.

The control engineer do not have to design the control system in the laboratory. It can be done everywhere, where it is comfortable using only the laptop. Later it can be tested in the laboratory.

This model can also be used to improve it even more precise. Other approach is to use it for the students' education. It can be a very good example of modelling of a highly nonlinear multivariable system using the state-space representation.

In future author would like to continue working in automation control systems design. There are lot of interesting tasks to complete related to the 3DCrane and other multivariable dynamic systems. For instance, the fuzzy logic or PID control theory can be tested on 3DCrane to reduce the payload oscillations. As for the 3DCrane, the safe control logic can be built in Simulink® or any other modelling and simulation environment, if know the exact requirements.

In my opinion, I have done well with the task. Moreover, I have learnt a lot about the modelling and control systems design. The language skills have been improved due to the reading many references in English language to submit the task. However, there could be another solutions simulated, for example, Laplace[19] transform or model prediction control based on state-space and many more.

[19] Pierre-Simon, marquis de Laplace (1749 - 1827) – French mathematician, physicist and astronomer.

References

1. Raja Ismail, R. M. T.; Ahmad, M. A.; Ramli, M.S.; Rashidi, F. R. M. Nonlinear Dynamic Modelling and Analysis of a 3-D Overhead Gantry Crane System with System Parameters Variation. – Pekan: University Malaysia Pahang, Faculty of Electrical and Electronics Engineering. – 8 p.
2. Trajković, D. M.; Antić, D. S.; Nikolić, S. S.; Perić, S. Lj.; Milovanović, M. B. Fuzzy Logic-Based Control of Three-Dimensional Crane System. – Niš: University of Niš, Faculty of Mechanical Engineering, Republic of Serbia, 2013. – 12 p.
3. Jovanović, Z; Perić, A.; Nikolić, S.; Milojković, M.; Milosević, M. Anti-Swing Fuzzy Controller Applied in a 3D Crane System. – Niš: University of Niš, Faculty of Electronic Engineering, Republic of Serbia 2012. – 5 p.
4. Preobrazhenskiy, D. 3D kraana juhtimise mobiilne rakendus iOS süsteemi jaoks. Bachelor of Science thesis. – Tallinn: TUT Chair of Automatic Control and System Analysis, 2013. – 25 p.
5. Nise, N. S. Control Systems Engineering: Sixth Edition International Student Version. – Pomona: California State Polytechnic University, John Wiley & Sons, Inc., 2011. – 968 p.
6. Feedback encyclopaedic definition: http://www.thefreedictionary.com Accessed 23.02.2015.
7. Klee, H. Simulation of Dynamic Systems with MATLAB® and Simulink®. – Orland: University of Central Florida, 2007. – 762 p.
8. Nicholson, H. Modelling of Dynamical Systems vol.1: IEE Control Engineering Series 12. – London: Peter Peregrinus Ltd., UK, 1980. – 227 p.
9. Billings, S. A.; Gray, J. O.; Owens, D. H. Nonlinear system design: IEE Control Engineering Series 25. – London: Peter Peregrinus Ltd., UK, 1984. – 192 p.
10. MATLAB Wikipedia webpage: http://en.wikipedia.org/wiki/MATLAB Accessed 22.02.2015.
11. Dixon, W. E.; Behal, A.; Dawson, D.M.; Nagarkatti, S. P. Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach. – College Park, university of Maryland, Department of Electrical and Computer Engineering, 2003. – 394 p.
12. Artherton, D. Control Engineering 2nd Edition: An introduction with the use of Matlab. – www.bookboon.com, 2013. – 150 p.

13. Woolf, P. University of Michigan Multiple Input, Multiple Output I: Numerical Decoupling. https://controls.engin.umich.edu/wiki/images/e/e9/Lecture.19.pdf Accessed 16.02.2015.
14. IDC Technologies industrial Automation. – www.bookboon.com, 2012. – 205 p.
15. KONECRANES webpage: http://www.konecranes.com. Accessed 15.02.2015.
16. Inteco Limited 3DCrane User's Manual: MATLAB R2009a/b, R2010a/b, R2011a/b PCI version: http://www.inteco.com.pl/products/3d-crane/
17. Xilinx® Wikipedia® webpage: http://en.wikipedia.org/wiki/Xilinx. Accessed 3.03.2015.
18. Micro Motors S.R.L. webpage: http://www.micromotors.eu/. Accessed 25.03.2015.
19. MathWorks® webpage: http://www.mathworks.com/. Accessed 30.03.2015.
20. Lagrangian equation definition: http://en.wikipedia.org/wiki/Lagrangian. Accessed 10.05.2015.
21. Euler-Lagrangian equation definition: http://en.wikipedia.org/wiki/Euler%E2%80%93Lagrange_equation. Accessed 10.05.2015.

Annexes

Annex 1

MATLAB script for the simplified model of the cart

```
%%   SIMPLIFIED MODEL OF THE CART   %%

clear all;          % Clear the Workspace
clc;                % Clear the Command Window

%%      Constants

mc = 1.155;         % Mass of the cart
ml = 1;             % Mass of the load
mr = 2.2;           % Mass of the rail

alpha = 0.5;        % Angle alpha
beta = 1;           % Angle beta

%%      Axis Gains

Xo = (mc+mr)/11.4;  % Gain for X-axis
Yo = mc/7;          % Gain for Y-axis
Zo = ml/4;          % Gain for Z-axis

sim ('Model3');     % Simulate the model
```

Annex 2

MATLAB script for the state-space representation

```
  %%        STATE-SPACE REPRESENTATION         %%

clear all;          % Clear the Workspace
clc;                % Clear the Command Window

%%      Constants

mc = 1.155;         % Mass of the cart
ml = 1;             % Mass of the load
mr = 2.2;           % Mass of the rail
g = 9.81;           % Gravitational acceleration

%%      Axis Gains

Xo = (mc+mr)/8;     % Gain for X-axis
Yo = mc/4;          % Gain for Y-axis
Zo = ml/4;          % Gain for Z-axis

%%      Symbols

syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 Nx Ny Nz;   % Short-cut for constructing
                                                    % symbolic objects

%%      Nonlinear Differential Equations

%   State Variables

f1 = x2;
f2 = (Nx+Nz*x5)/(mc+mr);
f3 = x4;
f4 = (Ny+Nz*x7)/mc;
f5 = x6;
f6 = ( 2*Nz*x5/ml-(Nx+Nz*x5)/(mc+mr)-2*x10*x6 )/x9;
f7 = x8;
f8 = ( 2*Nz*x7/ml-(Ny+Nz*x7)/mc-2*x10*x8 )/x9;
f9 = x10;
f10 = (-1)*Nz/ml;

%   Outputs

y1 = x1;
y2 = x3;
y3 = x5;
y4 = x7;
y5 = x9;

%%          A 10x10 matrix

a11 = diff(f1,x1);          % Difference and approximate derivative
a12 = diff(f1,x2);
a13 = diff(f1,x3);
a14 = diff(f1,x4);
a15 = diff(f1,x5);
a16 = diff(f1,x6);
a17 = diff(f1,x7);
a18 = diff(f1,x8);
```

```
a19 = diff(f1,x9);
a110 = diff(f1,x10);

a21 = diff(f2,x1);
a22 = diff(f2,x2);
a23 = diff(f2,x3);
a24 = diff(f2,x4);
a25 = diff(f2,x5);
a26 = diff(f2,x6);
a27 = diff(f2,x7);
a28 = diff(f2,x8);
a29 = diff(f2,x9);
a210 = diff(f2,x10);

a31 = diff(f3,x1);
a32 = diff(f3,x2);
a33 = diff(f3,x3);
a34 = diff(f3,x4);
a35 = diff(f3,x5);
a36 = diff(f3,x6);
a37 = diff(f3,x7);
a38 = diff(f3,x8);
a39 = diff(f3,x9);
a310 = diff(f3,x10);

a41 = diff(f4,x1);
a42 = diff(f4,x2);
a43 = diff(f4,x3);
a44 = diff(f4,x4);
a45 = diff(f4,x5);
a46 = diff(f4,x6);
a47 = diff(f4,x7);
a48 = diff(f4,x8);
a49 = diff(f4,x9);
a410 = diff(f4,x10);

a51 = diff(f5,x1);
a52 = diff(f5,x2);
a53 = diff(f5,x3);
a54 = diff(f5,x4);
a55 = diff(f5,x5);
a56 = diff(f5,x6);
a57 = diff(f5,x7);
a58 = diff(f5,x8);
a59 = diff(f5,x9);
a510 = diff(f5,x10);

a61 = diff(f6,x1);
a62 = diff(f6,x2);
a63 = diff(f6,x3);
a64 = diff(f6,x4);
a65 = diff(f6,x5);
a66 = diff(f6,x6);
a67 = diff(f6,x7);
a68 = diff(f6,x8);
a69 = diff(f6,x9);
a610 = diff(f6,x10);

a71 = diff(f7,x1);
a72 = diff(f7,x2);
a73 = diff(f7,x3);
a74 = diff(f7,x4);
a75 = diff(f7,x5);
a76 = diff(f7,x6);
a77 = diff(f7,x7);
a78 = diff(f7,x8);
```

```
a79 = diff(f7,x9);
a710 = diff(f7,x10);

a81 = diff(f8,x1);
a82 = diff(f8,x2);
a83 = diff(f8,x3);
a84 = diff(f8,x4);
a85 = diff(f8,x5);
a86 = diff(f8,x6);
a87 = diff(f8,x7);
a88 = diff(f8,x8);
a89 = diff(f8,x9);
a810 = diff(f8,x10);

a91 = diff(f9,x1);
a92 = diff(f9,x2);
a93 = diff(f9,x3);
a94 = diff(f9,x4);
a95 = diff(f9,x5);
a96 = diff(f9,x6);
a97 = diff(f9,x7);
a98 = diff(f9,x8);
a99 = diff(f9,x9);
a910 = diff(f9,x10);

a101 = diff(f10,x1);
a102 = diff(f10,x2);
a103 = diff(f10,x3);
a104 = diff(f10,x4);
a105 = diff(f10,x5);
a106 = diff(f10,x6);
a107 = diff(f10,x7);
a108 = diff(f10,x8);
a109 = diff(f10,x9);
a1010 = diff(f10,x10);

%%      State Space A Matrix Calculation

A = [a11 a12 a13 a14 a15 a16 a17 a18 a19 a110;
a21 a22 a23 a24 a25 a26 a27 a28 a29 a210;
a31 a32 a33 a34 a35 a36 a37 a38 a39 a310;
a41 a42 a43 a44 a45 a46 a47 a48 a49 a410;
a51 a52 a53 a54 a55 a56 a57 a58 a59 a510;
a61 a62 a63 a64 a65 a66 a67 a68 a69 a610;
a71 a72 a73 a74 a75 a76 a77 a78 a79 a710;
a81 a82 a83 a84 a85 a86 a87 a88 a89 a810;
a91 a92 a93 a94 a95 a96 a97 a98 a99 a910;
a101 a102 a103 a104 a105 a106 a107 a108 a109 a1010];

%%          B 10x3 matrix

b11 = diff(f1,Nx);
b12 = diff(f1,Ny);
b13 = diff(f1,Nz);

b21 = diff(f2,Nx);
b22 = diff(f2,Ny);
b23 = diff(f2,Nz);

b31 = diff(f3,Nx);
b32 = diff(f3,Ny);
b33 = diff(f3,Nz);

b41 = diff(f4,Nx);
b42 = diff(f4,Ny);
```

```
b43 = diff(f4,Nz);

b51 = diff(f5,Nx);
b52 = diff(f5,Ny);
b53 = diff(f5,Nz);

b61 = diff(f6,Nx);
b62 = diff(f6,Ny);
b63 = diff(f6,Nz);

b71 = diff(f7,Nx);
b72 = diff(f7,Ny);
b73 = diff(f7,Nz);

b81 = diff(f8,Nx);
b82 = diff(f8,Ny);
b83 = diff(f8,Nz);

b91 = diff(f9,Nx);
b92 = diff(f9,Ny);
b93 = diff(f9,Nz);

b101 = diff(f10,Nx);
b102 = diff(f10,Ny);
b103 = diff(f10,Nz);

%%          State Space B Matrix Calculation

B = [b11 b12 b13;
b21 b22 b23;
b31 b32 b33;
b41 b42 b43;
b51 b52 b53;
b61 b62 b63;
b71 b72 b73;
b81 b82 b83;
b91 b92 b93;
b101 b102 b103];

%%          C 5x10 matrix

c11 = diff(y1,x1);
c12 = diff(y1,x2);
c13 = diff(y1,x3);
c14 = diff(y1,x4);
c15 = diff(y1,x5);
c16 = diff(y1,x6);
c17 = diff(y1,x7);
c18 = diff(y1,x8);
c19 = diff(y1,x9);
c110 = diff(y1,x10);

c21 = diff(y2,x1);
c22 = diff(y2,x2);
c23 = diff(y2,x3);
c24 = diff(y2,x4);
c25 = diff(y2,x5);
c26 = diff(y2,x6);
c27 = diff(y2,x7);
c28 = diff(y2,x8);
c29 = diff(y2,x9);
c210 = diff(y2,x10);

c31 = diff(y3,x1);
```

```
c32 = diff(y3,x2);
c33 = diff(y3,x3);
c34 = diff(y3,x4);
c35 = diff(y3,x5);
c36 = diff(y3,x6);
c37 = diff(y3,x7);
c38 = diff(y3,x8);
c39 = diff(y3,x9);
c310 = diff(y3,x10);

c41 = diff(y4,x1);
c42 = diff(y4,x2);
c43 = diff(y4,x3);
c44 = diff(y4,x4);
c45 = diff(y4,x5);
c46 = diff(y4,x6);
c47 = diff(y4,x7);
c48 = diff(y4,x8);
c49 = diff(y4,x9);
c410 = diff(y4,x10);

c51 = diff(y5,x1);
c52 = diff(y5,x2);
c53 = diff(y5,x3);
c54 = diff(y5,x4);
c55 = diff(y5,x5);
c56 = diff(y5,x6);
c57 = diff(y5,x7);
c58 = diff(y5,x8);
c59 = diff(y5,x9);
c510 = diff(y5,x10);

%%      State Space C Matrix Calculation

C = [c11 c12 c13 c14 c15 c16 c17 c18 c19 c110;
c21 c22 c23 c24 c25 c26 c27 c28 c29 c210;
c31 c32 c33 c34 c35 c36 c37 c38 c39 c310;
c41 c42 c43 c44 c45 c46 c47 c48 c49 c410;
c51 c52 c53 c54 c55 c56 c57 c58 c59 c510;];

%%          D 5x3 matrix

d11 = diff(y1,Nx);
d12 = diff(y1,Ny);
d13 = diff(y1,Nz);

d21 = diff(y2,Nx);
d22 = diff(y2,Ny);
d23 = diff(y2,Nz);

d31 = diff(y3,Nx);
d32 = diff(y3,Ny);
d33 = diff(y3,Nz);

d41 = diff(y4,Nx);
d42 = diff(y4,Ny);
d43 = diff(y4,Nz);

d51 = diff(y5,Nx);
d52 = diff(y5,Ny);
d53 = diff(y5,Nz);

%%          State Space D Matrix Calculation
```

```
D = [d11 d12 d13;
d21 d22 d23;
d31 d32 d33;
d41 d42 d43;
d51 d52 d53];


%%          Steady State Solution

A = [0 1 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0;
0 0 0 1 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 1 0 0 0 0;
0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 1 0 0;
0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 1;
0 0 0 0 0 0 0 0 0 0];

B = [0 0 0;
200/671 0 0;
0 0 0;
0 200/231 0;
0 0 0;
0 0 0;
0 0 0;
0 0 0;
0 0 0;
0 0 -1];

C = [1 0 0 0 0 0 0 0 0 0;
0 0 1 0 0 0 0 0 0 0;
0 0 0 0 1 0 0 0 0 0;
0 0 0 0 0 0 1 0 0 0;
0 0 0 0 0 0 0 0 1 0];

D = zeros(5,3);

isstable (ss)                   % Stability check

sim ('StateSpace4sim')          % SimuLink Simulation

%%          Transfer Function

[num, den] = ss2tf (A,B,C,D,1);   % State-space  to transfer function
                                      % conversion: numerator &
                                      % denominator
F= ss(A, B, C, D);                % Converts model to state space
T = tf (F);                       % Transfer Function
I = sym(eye(10));
I1= C*inv(sym('s')*I-A)*B;        % Laplace transform
pretty (I1);                      % Pretty print a symbolic expression
```

Full 3DCrane full dynamic model derivation

$$x_w = x$$

$$y_w = y$$

$$x_p = x + l\cos\alpha\sin\beta$$

$$y_p = y + l\sin\alpha$$

$$z_p = -l\cos\alpha\cos\beta$$

$$\dot{x}_p = \dot{x} + \dot{l}\cos\alpha\sin\beta - l\dot{\alpha}\sin\alpha\sin\beta + l\dot{\beta}\cos\alpha\cos\beta$$

$$\dot{y}_p = \dot{y} + \dot{l}\sin\alpha - l\dot{\alpha}\cos\alpha$$

$$\dot{z}_p = -\dot{l}\cos\alpha\cos\beta + l\dot{\alpha}\sin\alpha\cos\beta + l\dot{\beta}\cos\alpha\sin\beta$$

The speed of the load:

$$v^2 = \dot{x}_p^2 + \dot{y}_p^2 + \dot{z}_p^2$$

$$v^2 = \dot{x}^2 + 2\dot{x}\dot{l}\cos\alpha\sin\beta - 2l\dot{x}\dot{\alpha}\sin\alpha\sin\beta + 2l\dot{x}\dot{\beta}\cos\alpha\cos\beta + \dot{l}^2cos^2\alpha\ sin^2\beta - 2l\dot{l}\dot{\alpha}\sin\alpha\cos\alpha\ sin^2\beta + 2l\dot{l}\dot{\beta}cos^2\alpha\sin\beta\cos\beta + l^2\dot{\alpha}^2sin^2\alpha\ sin^2\beta - 2l^2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha\sin\beta\cos\beta + l^2\dot{\beta}^2cos^2\alpha\ cos^2\beta + \dot{y}^2 + 2\dot{y}\dot{l}\sin\alpha + 2l\dot{y}\dot{\alpha}\cos\alpha + \dot{l}^2sin^2\alpha + 2l\dot{l}\dot{\alpha}\sin\alpha\cos\alpha + l^2\dot{\alpha}^2cos^2\alpha + \dot{l}^2cos^2\alpha\ cos^2\beta - 2l\dot{l}\dot{\alpha}\sin\alpha\cos\alpha\ cos^2\beta - 2l\dot{l}\dot{\beta}cos^2\alpha\sin\beta\cos\beta + l^2\dot{\alpha}^2sin^2\alpha\ cos^2\beta + 2l^2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha\sin\beta\cos\beta + l^2\dot{\beta}^2cos^2\alpha\ sin^2\beta$$

$$v^2 = \dot{x}^2 + \dot{y}^2 + \dot{l}^2 + 2\dot{x}\left(\dot{l}\cos\alpha\sin\beta - l\dot{\alpha}\sin\alpha\sin\beta + l\dot{\beta}\cos\alpha\cos\beta\right) + 2\dot{y}\left(\dot{l}\sin\alpha + l\dot{\alpha}\cos\alpha\right) + l^2\dot{\alpha}^2 + l^2\dot{\beta}^2cos^2\alpha$$

Total kinetic energy of the crane:

$$K = \frac{1}{2}\left((m_c + m_r + m_l)\dot{x}^2 + (m_c + m_l)\dot{y}^2 + m_l\dot{l}^2\right) + \frac{1}{2}m_lv^2$$

$$K = \frac{1}{2}\left((m_c + m_r + m_l + m_l)\dot{x}^2 + (m_c + m_l + m_l)\dot{y}^2 + 2m_l\dot{l}^2\right) + m_l\dot{x}\left(\dot{l}\cos\alpha\sin\beta - l\dot{\alpha}\sin\alpha\sin\beta + l\dot{\beta}\cos\alpha\cos\beta\right) + m_l\dot{y}\left(\dot{l}\sin\alpha + l\dot{\alpha}\cos\alpha\right) + \frac{1}{2}m_ll^2\dot{\alpha}^2 + \frac{1}{2}m_ll^2\dot{\beta}^2cos^2\alpha$$

Total potential energy of the system:

$P = m_l g l + m_l g l \cos\alpha \cos\beta$

The Lagrangian of the dynamic system (6.1):

$L = K - P$

$L = \frac{1}{2}\Big((m_c + m_r + m_l + m_l)\dot{x}^2 + (m_c + m_l + m_l)\dot{y}^2 + 2m_l\dot{l}^2\Big) + m_l\dot{x}\big(\dot{l}\cos\alpha\sin\beta -$

$l\dot{\alpha}\sin\alpha\sin\beta + l\dot{\beta}\cos\alpha\cos\beta\big) + m_l\dot{y}\big(\dot{l}\sin\alpha + l\dot{\alpha}\cos\alpha\big) + \frac{1}{2}m_l l^2\dot{\alpha}^2 + \frac{1}{2}m_l l^2\dot{\beta}^2 cos^2\alpha -$

$m_l g l - m_l g l \cos\alpha\cos\beta$

The Euler-Lagrange formulation of the dynamic system (6.4):

$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = F_i,$ where $i = 1{,}2{,}3{,}4{,}5$

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x_i} = u_x \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}_i}\right) - \frac{\partial L}{\partial y_i} = u_y \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}_i}\right) - \frac{\partial L}{\partial z_i} = u_z \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}_i}\right) - \frac{\partial L}{\partial \alpha_i} = 0 \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\beta}_i}\right) - \frac{\partial L}{\partial \beta_i} = 0 \end{cases}$$

$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = (m_c + m_r + m_l + m_l)\ddot{x} + m_l\ddot{l}\cos\alpha\sin\beta - m_l\dot{l}\dot{\alpha}\sin\alpha\sin\beta +$

$m_l\dot{l}\dot{\beta}\cos\alpha\cos\beta - m_l\dot{l}\dot{\alpha}\sin\alpha\sin\beta - m_l l\ddot{\alpha}\sin\alpha\sin\beta - m_l l\dot{\alpha}^2\cos\alpha\sin\beta -$

$m_l\dot{l}\dot{\alpha}\dot{\beta}\sin\alpha\cos\beta + m_l\dot{l}\dot{\beta}\cos\alpha\cos\beta + m_l l\ddot{\beta}\cos\alpha\cos\beta - m_l l\dot{\beta}\dot{\alpha}\sin\alpha\cos\beta -$

$m_l l\dot{\beta}^2\cos\alpha\sin\beta$

$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) - \frac{\partial L}{\partial y} = (m_c + m_l + m_l)\ddot{y} + m_l\ddot{l}\sin\alpha + m_l\dot{l}\dot{\alpha}\cos\alpha + m_l\dot{l}\dot{\alpha}\cos\alpha + m_l l\ddot{\alpha}\cos\alpha -$

$m_l l\dot{\alpha}^2\sin\alpha$

$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{l}}\right) - \frac{\partial L}{\partial l} = 2m_l\ddot{l} + m_l\dot{x}\cos\alpha\sin\beta - m_l\dot{x}\dot{\alpha}\sin\alpha\sin\beta + m_l\dot{x}\dot{\beta}\cos\alpha\cos\beta +$

$m_l\ddot{y}\sin\alpha + m_l\dot{y}\dot{\alpha}\cos\alpha - \big(-m_l\dot{x}\dot{\alpha}\sin\alpha\sin\beta + m_l\dot{x}\dot{\beta}\cos\alpha\cos\beta + m_l\dot{y}\dot{\alpha}\cos\alpha +$

$m_l\dot{l}\dot{\beta}cos^2\alpha + m_l\dot{l}\dot{\alpha}^2\big) - m_l g\cos\alpha\cos\beta$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}}\right)-\frac{\partial L}{\partial \alpha}=-m_l\ddot{x}l\sin\alpha\sin\beta-m_l\dot{x}\dot{l}\sin\alpha\sin\beta-m_l l\dot{x}\dot{\alpha}\cos\alpha\sin\beta-$$
$$m_l l\dot{x}\dot{\beta}\sin\alpha\cos\beta+m_l\ddot{y}l\cos\alpha+m_l\dot{y}\dot{l}\cos\alpha-m_l l\dot{y}\dot{\alpha}\sin\alpha+m_l l^2\ddot{\alpha}+2m_l l\dot{l}\dot{\alpha}-$$
$$\left(-m_l\dot{x}\dot{l}\sin\alpha\sin\beta-m_l l\dot{x}\dot{\alpha}\cos\alpha\sin\beta-m_l l\dot{x}\dot{\beta}\sin\alpha\cos\beta+m_l\dot{y}\dot{l}\cos\alpha-m_l l\dot{y}\dot{\alpha}\sin\alpha-\right.$$
$$\left.m_l l^2\dot{\beta}^2\sin\alpha\cos\alpha\right)+m_l gl\cos\alpha\sin\beta$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\beta}}\right)-\frac{\partial L}{\partial \beta}=m_l\ddot{x}l\cos\alpha\cos\beta+m_l\dot{x}\dot{l}\cos\alpha\cos\beta-m_l l\dot{x}\dot{\alpha}\sin\alpha\cos\beta-$$
$$m_l l\dot{x}\dot{\beta}\cos\alpha\sin\beta+m_l\dot{l}^2\dot{\beta}cos^2\alpha-2m_l l^2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha+2m_l l^2\ddot{\beta}cos^2\alpha+$$
$$2m_l l\dot{l}\dot{\beta}cos^2\alpha-\left(m_l\dot{x}\dot{l}\cos\alpha\cos\beta-m_l l\dot{x}\dot{\alpha}\sin\alpha\cos\beta-m_l l\dot{x}\dot{\beta}\cos\alpha\sin\beta\right)+$$
$$m_l gl\sin\alpha\cos\beta$$

$$\boldsymbol{\begin{cases}(m_c+m_r+m_l+m_l)\ddot{x}+m_l\ddot{l}\cos\alpha\sin\beta-2m_l\dot{l}\dot{\alpha}\sin\alpha\sin\beta+2m_l\dot{l}\dot{\beta}\cos\alpha\cos\beta-m_l l\ddot{\alpha}\sin\alpha\sin\beta- \\ -m_l l\dot{\alpha}^2\cos\alpha\sin\beta-2m_l\dot{l}\dot{\alpha}\dot{\beta}\sin\alpha\cos\beta+m_l l\ddot{\beta}\cos\alpha\cos\beta-m_l l\dot{\beta}^2\cos\alpha\sin\beta=u_x \\ (m_c+m_l+m_l)\ddot{y}+m_l\ddot{l}\sin\alpha+2m_l\dot{l}\dot{\alpha}\cos\alpha+m_l l\ddot{\alpha}\cos\alpha-m_l l\dot{\alpha}^2\sin\alpha\quad=u_y \\ 2m_l\ddot{l}+m_l\dot{x}\cos\alpha\sin\beta+m_l\ddot{y}\sin\alpha-m_l\dot{l}\dot{\beta}cos^2\alpha-m_l\dot{l}\dot{\alpha}^2-m_l g\cos\alpha\cos\beta=u_z \\ -m_l\ddot{x}l\sin\alpha\sin\beta+m_l\ddot{y}l\cos\alpha+m_l l^2\ddot{\alpha}+2m_l l\dot{l}\dot{\alpha}+m_l gl\cos\alpha\sin\beta=0 \\ m_l\ddot{x}l\cos\alpha\cos\beta-2m_l l^2\dot{\alpha}\dot{\beta}\sin\alpha\cos\alpha+2m_l l^2\ddot{\beta}cos^2\alpha+2m_l l\dot{l}\dot{\beta}cos^2\alpha+m_l gl\sin\alpha\cos\beta=0\end{cases}}$$

$$\begin{cases}(m_c+m_r+m_l+m_l)\ddot{x}+m_l l\ddot{\beta}-u_x=0 \\ (m_c+m_l+m_l)\ddot{y}+m_l l\ddot{\alpha}-u_y=0 \\ 2m_l\ddot{l}-m_l g-u_z=0 \\ \ddot{y}+l\ddot{\alpha}+g\alpha=0 \\ \ddot{x}+l\ddot{\beta}+g\beta=0\end{cases}$$

The steady-state operation point $x^0=\left[x^0\ \dot{x}^0\ y^0\ \dot{y}^0\ \beta^0\ \dot{\beta}^0\ \alpha^0\ \dot{\alpha}^0\ l^0\ \dot{l}^0\right]^T=$ $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$ gives the final result for the full dynamic model of the 3DCrane:

$$\boldsymbol{\begin{cases}\ddot{x}=\frac{1}{m_c+m_r+m_l}u_x+\frac{m_l g}{m_c+m_r+m_l}\beta \\ \ddot{y}=\frac{1}{m_c+m_l}u_y+\frac{m_l g}{m_c+m_l}\alpha \\ \ddot{\beta}=-\frac{1}{l(m_c+m_r+m_l)}u_x-\frac{g(m_c+m_r+m_l+m_l)}{l(m_c+m_r+m_l)}\beta \\ \ddot{\alpha}=-\frac{1}{l(m_c+m_l)}u_y-\frac{g(m_c+m_l+m_l)}{l(m_c+m_l)}\alpha \\ \ddot{l}=\frac{1}{2m_l}u_z\end{cases}}$$

MATLAB script for the state-space representation of the full dynamic model

```
clear all;          % Clear the Workspace
clc;                % Clear the Command Wind0w

%%      Constants

mc = 1.155;         % Mass of the cart
ml = 1;             % Mass of the load
mr = 2.2;           % Mass of the rail

l = 0.6;            % Length of lift-line

g = 9.81;           % Gravitational acceleration

Gain = [4/3 4/5 6/12];     % Input Correction Gain

%%        State-State representation of the full dynamic model

A = [0 1 0 0 0 0 0 0 0 0;
     0 0 0 0 ml*g/(mc+mr+ml) 0 0 0 0 0;
     0 0 0 1 0 0 0 0 0 0;
     0 0 0 0 0 0 ml*g/(mc+ml) 0 0 0;
     0 0 0 0 0 1 0 0 0 0;
     0 0 0 0 -g*(ml+mc+mr+ml)/((mc+mr+ml)*l) 0 0 0 0 0;
     0 0 0 0 0 0 0 1 0 0;
     0 0 0 0 0 0 -g*(ml+mc+ml)/((mc+ml)*l) 0 0 0;
     0 0 0 0 0 0 0 0 0 1;
     0 0 0 0 0 0 0 0 0 0];

B = [0 0 0;
     1/(mc+mr+ml) 0 0;
     0 0 0;
     0 1/(mc+ml) 0;
     0 0 0;
     -1/((mc+mr+ml)*l) 0 0;
     0 0 0;
     0 -1/((mc+ml)*l) 0;
     0 0 0;
     0 0 1/(ml+ml)];

C = [1 0 0 0 0 0 0 0 0 0;
     0 0 1 0 0 0 0 0 0 0;
     0 0 0 0 1 0 0 0 0 0;
     0 0 0 0 0 0 1 0 0 0;
     0 0 0 0 0 0 0 0 1 0];

D = zeros(5,3);

isstable (ss)           % Stability check

sim ('StateSpaceFull')  % SimuLink Simulation
```