



CONTROL OF CRANES WITH A SWINGING LOAD, AN INVESTIGATION OF
PRAGMATIC CONTROL

A Thesis submitted by

Bilal Hamid Abduljabbar

BSc, MSc

For the award of

Doctor of Philosophy

2021

Abstract

In modern industrial systems, cranes are widely used for the transference of heavy loads. There are different types of cranes which are used, such as tower cranes and gantry cranes. During routine operations, it is difficult for crane operators to lift and manoeuvre heavy payloads without causing significant load swing. To ensure the safety of workers and to prevent damage on site, it is essential to suppress the swinging load during operation.

Systems used in the past to control movement and load swing have used complex control algorithms requiring massive computations. This research investigates the application of 'pragmatic' control techniques to crane systems which represents a new method for controlling crane movement have been applied to crane systems for the first time. A new design of a pragmatic control such as a cascaded constrained loop is proposed for crane systems which are not currently being used in such systems. A simple and robust pragmatic controller has been designed to control a crane system so that the controller could move a swinging load from one point to another in minimum time without generating a large swing.

For the purposes of this research, two physical models have been constructed, in addition to considerable simulation studies. One model is applicable for a tower crane with a hoist that travels along a jib that rotates on a vertical axis. The other is a single axis model, with one axis operating on a gantry system that has been used to investigate the finer points of the control.

This work presents a state-space control method that can be implemented on a simple microcontroller. It used computer vision to monitor the crane load position and then used that position as feedback in a control loop. It has been proposed that a camera, mounted on the hoist and looking vertically downwards, can detect a corner-reflector mounted on the hook to give the essential signals for stabilisation. The development of control techniques, in particular 'pragmatic control', can then be central to autonomous control or auto-stabilised manual control.

There is considerable discussion of the generation of the 'pragmatic' control algorithm and its simplification. Also, many hardware-based techniques are developed, including software for extracting state-space data from the camera and for reading the two-phase incremental encoders of the motors at high speed.

The short time-constants associated with the smaller laboratory models caused a problem because of the slow frame-rate of the camera, thereby requiring a more sophisticated discrete-time method was required for them. Practical testing validated the control strategy and successful and robust performance was achieved. Loads were transported and positioned successfully, with little swing evident during operation of the crane.

Mathematical models of a crane in state-space are presented. A ‘pragmatic’ controller was successfully used to control the crane in simulation and then in a laboratory. Results were evaluated and compared to published benchmarks.

Certification of Thesis

This thesis is entirely the work of **Bilal Hamid Abduljabbar** except where otherwise acknowledged. The work is original and has not previously been submitted for any other award, except where acknowledged.

Bilal Hamid Abduljabbar

Principal Supervisor: Professor John Billingsley

Associate Supervisor: Professor Paul Wen

Student and supervisors signatures of endorsement are held at the University.

List of Publications

- 1) Bilal Abduljabbar, John, Billingsley, Control of a Tower Crane with Pragmatic Hierarchical Algorithm. *Journal of Applied Nonlinear Dynamics*. Accepted 8 Mar 2020.
- 2) Abduljabbar, B., Billingsley, J. and Wen, P., 2021, February. A pragmatic controller for a gantry crane. In *Journal of Physics: Conference Series* (Vol. 1780, No. 1, p. 012007). IOP Publishing.

Acknowledgement

IN THE NAME OF ALLAH, THE MOST BENEFICENT, THE MOST MERCIFUL

First and foremost, I would like to thank God for his blessing, kindness and for giving me the strength to finish my doctorate.

I should like to thank the many people who made this thesis possible. I would like to express my sincere thanks and gratitude to the principal supervisor, Professor John Billingsley for his continued support and encouragement during my PhD journey. Professor John Billingsley has been a father-like figure and advisor to me. John has extended my thinking in many directions while at the same time helping me to stay on, or at least near the path of my thesis journey. Honestly speaking, without his guide, help, advice, expertise, and support this thesis would not be accomplished. His support extended beyond my academic endeavours. He helped me to overcome many difficulties and problems during the research journey.

I would also like to extend my thanks and gratitude to my Associate supervisor, Professor Paul Wen. He helped me through his valuable feedback and comments. Also, I would like to thank my colleague Dr Sarang Ghude who helped me at the beginning of my PhD journey.

My special thanks are extended to the staff of School of electrical and mechanical engineering, the University of Southern Queensland who supported me during my PhD journey, Especially Mr Brian Aston (Coordinating Technical Officer), Mr Oliver Kinder (Coordinating Technical Officer) and Mr Terry Byrne (Senior Coordinating Technical Officer), who gave me great help and support during the fabrication of both crane systems. Also, my thanks to Mrs Libby Collett for my thesis English editing (Proofreading).

I would like to express my highest appreciation to the Ministry of Higher Education Iraq, which sponsored my studies. Through their financial and logistical support, I was able to achieve my goal. Furthermore, this research has been supported by an Australian Government Research Training Program Scholarship (RTP).

I would like to thank my father and mother. They have supported me through their support, prayers and blessings for me. I wish them a life full of health and happiness. This thesis is dedicated to them.

I greatly owe my wife for her endless patience and support during my PhD journey. Without my lovely wife, this work will never be accomplished. Special thanks to my beautiful daughters Sally and Leen for her beautiful smiles.

I would also like to thank my brothers, sister, for their love, help and constant support throughout the study. I wish them a life full of health and happiness.

Table of Contents

Abstract.....	i
Certification of Thesis.....	iii
List of Publications	iv
Acknowledgement	v
List of Figures	xii
List of Tables	xvi
Abbreviations.....	xvii
1 CHAPTER ONE: - INTRODUCTION	1
1.1 Overview	1
1.2 Research Background.....	1
1.3 Research Aim	2
1.4 Specific Research Objectives	3
1.5 Hypothesis.....	3
1.6 Research Contribution.....	4
1.7 Thesis Structure.....	4
2 CHAPTER TWO: - LITERATURE REVIEW	6
2.1 Introduction	6
2.2 Pragmatic Technique.....	7
2.3 System Modelling	8
2.3.1 Gantry Crane Modelling	9

2.3.2	Tower Crane Modelling	16
2.4	Control Techniques Used in Crane System.....	20
2.4.1	Open-Loop Techniques.....	20
2.4.2	Closed-Loop Techniques	23
2.5	Conclusion.....	31
3	CHAPTER THREE: - CRANE SYSTEM MODELLING.....	33
3.1	Introduction	33
3.2	Gantry Crane Model Description	33
3.3	Tower Crane Model Description.....	41
3.4	Tower Crane Model Design	42
3.4.1	Tower Crane Model Based on State Space Representation.....	43
3.4.2	Tower Crane Model Based on Practical Point of View	50
3.5	The Simplicity of Pragmatic Technique.....	65
3.6	Conclusion.....	66
4	CHAPTER FOUR: - PRAGMATIC CONTROL TECHNIQUE- SIMULATION STUDIES	67
4.1	Introduction	67
4.2	Simulation of Gantry Crane Model.....	67
4.3	Simulation of Tower Crane based on State Space Representation	75
4.4	Simulation of Tower Crane based on Practical Point of View	81
4.5	Conclusion.....	87

5	CHAPTER FIVE: - EXPERIMENTAL DESIGN OF GANTRY CRANE SYSTEM....	88
5.1	Mechanical Design Considerations	88
5.2	Experimental Components and Setup	92
5.2.1	Hoist and Cable Motion	92
5.2.2	L298N Dual Motor Controller Module.....	94
5.2.3	Arduino	94
5.2.4	Gantry Crane’s Motor Wiring Connection	95
5.3	Computer Vision Strategy in Gantry Crane	96
5.3.1	Object Representation	97
5.3.2	Segmentation.....	97
5.3.3	Processing Software.....	98
5.3.4	Load Position Tracking by The Camera	98
5.4	Encoder Reading	102
5.4.1	Encoder Measurement	103
5.4.2	Reading the Encoder Algorithm	104
5.5	Positioning Control by Arduino	106
5.6	Serial Communication Between Arduino and Processing Software.....	108
5.6.1	Arduino Section	108
5.6.2	Processing Section	110
5.7	Control algorithm via Processing.....	111
5.8	Experimental Results and Discussion	112

5.9	Conclusion.....	119
6	CHAPTER SIX: - EXPERIMENTAL DESIGN OF TOWER CRANE SYSTEM	120
6.1	Mechanical Design Considerations	120
6.2	Experimental Components and Setup	124
6.2.1	Hoist and Cable Motion	124
6.2.2	Jib Angular Motion	125
6.3	The Electrical Setup of Tower Crane	126
6.4	Computer Vision Strategy for Tower Crane	127
6.5	The Control Algorithm of the Tower Crane.....	128
6.5.1	Reading the Encoders by Arduino	128
6.5.2	Estimating Velocity by Arduino	129
6.5.3	Frame Rate and Timing	130
6.5.4	Control Algorithm via Processing for Tower Crane.....	131
6.6	Experimental Results and Discussion	133
6.7	Conclusion.....	141
7	CHAPTER SEVEN: - CONCLUSION AND FUTURE WORKS.....	142
7.1	Introduction	142
7.2	System Models	143
7.3	Hardware Model.....	144
7.4	Limitations of The Experimental Set-Up.....	145
7.4.1	The Problem of Quantization.....	145

7.4.2	Reading the Encoders	146
7.4.3	Estimating Velocity	147
7.4.4	Frame Rate and Timing	148
7.5	Recommendations and Future Work.....	149
REFERENCES		150

List of Figures

Figure 2-1 The Structure of Planar Gantry crane (Singhose et al., 2000)	9
Figure 2-2 The scheme of a double-pendulum overhead crane by (Liu et al., 2005).....	14
Figure 3-1 The structure of the gantry crane system	34
Figure 3-2 Block diagram of the system controller	37
Figure 3-3 3D Tower Crane Model	42
Figure 3-4 Tower Crane Model; jib, hoist and load.....	45
Figure 4-1 Top view of the gantry crane system in MATLAB	68
Figure 4-2 Position response of 5-meter cable length (a) movement in axis only (b) movement of both $x - y$ axis.....	69
Figure 4-3 Position response of 10-meter cable length (a) movement in x-axis only (b) movement of both $x - y$ axis.....	70
Figure 4-4 Crane movement with 45° slew motion by using JavaScript.....	71
Figure 4-5 Hoist velocity response (a) with 5 m cable length (b) with 10 m cable length.....	72
Figure 4-6 Load velocity response (a) with 5 m cable length (b) with 10 m cable length	73
Figure 4-7 Schematic of swing angle with 5-meter cable length.....	74
Figure 4-8 Schematic of swing angle with 10 m cable length.....	75
Figure 4-9 Top view of the tower crane system in MATLAB	76
Figure 4-10 Jib velocity response	77
Figure 4-11 Hoist velocity response	78
Figure 4-12 Load velocity response.....	78

Figure 4-13 Positions of load and hoist during travelling with five-meter cable length	79
Figure 4-14 Swing angles of the system with five-meter cable length.....	80
Figure 4-15 Positions of load and hoist during travelling with ten-meter cable length.....	80
Figure 4-16 Swing angles of the system with ten-meter cable length	81
Figure 4-17 Top view of the tower crane system in JavaScript.....	83
Figure 4-18 Hoist velocity response	84
Figure 4-19 Position response of the hoist.....	84
Figure 4-20 Position response of the jib angle.....	85
Figure 4-21 Load velocity in x-y coordinates.....	86
Figure 4-22 Swing angle of the system.....	86
Figure 5-1 T-slot System 40 Aluminum extrusion profile 4040	88
Figure 5-2 Aluminum plate of the hoist.....	89
Figure 5-3 The hoist system design	90
Figure 5-4 The fabricated hoist system.....	90
Figure 5-5 Final design of one-axis gantry crane	91
Figure 5-6 The fabricated one axis gantry crane	91
Figure 5-7 Metal Geared motor 200 RPM 31D*30L mm with 64 CPR Encoder	92
Figure 5-8 L298N Dual Motor Controller Module.....	94
Figure 5-9 Arduino Uno Board (Sparkfun RedBoard)	95
Figure 5-10 Initial image of the load	100
Figure 5-11 Detected load by the camera	101

Figure 5-12 Camera calculation set-up	101
Figure 5-13 Quadrature Encoder A and B Output Signals	103
Figure 5-14 Position response of hoist and load with no weight.....	113
Figure 5-15 Position response of hoist and load with no weight with disturbance	114
Figure 5-16 Position response of hoist and load with 1 kg weight.....	114
Figure 5-17 Velocity response of hoist and load	115
Figure 5-18 Velocity response of hoist and load with disturbance.....	115
Figure 5-19 Velocity response of hoist and load with 1kg weight	116
Figure 5-20 Swing angle response with no weight.....	117
Figure 5-21 Swing angle response with no weight with disturbance	117
Figure 5-22 Swing angle response with 1 kg weight.....	118
Figure 5-23 swing angle response with 1 kg weight with disturbance	119
Figure 6-1 Crane jib (side and top view)	121
Figure 6-2 Jib angle driver structure	122
Figure 6-3 Crane hoist system	122
Figure 6-4 The proposed design of the tower crane system	123
Figure 6-5 Final design of tower crane	123
Figure 6-6 Metal Geared motor 133 RPM 37D*57L mm with 64 CPR Encoder	124
Figure 6-7 Position response of tower hoist with no weight	134
Figure 6-8 Position response of tower jib with no weight.....	135
Figure 6-9 Velocity response of tower hoist.....	135

Figure 6-10 Velocity response of tower hoist.....	136
Figure 6-11 The swing angle “Beta” response with no weight	136
Figure 6-12 The swing angle “theta” response with no weight.....	137
Figure 6-13 Position response of tower hoist with 1 kg weight	137
Figure 6-14 Position response of tower jib with 1 kg weight	138
Figure 6-15 Velocity response of tower jib	138
Figure 6-16 Velocity response of tower hoist.....	139
Figure 6-17 Position response of tower hoist with disturbance.....	140
Figure 6-18 Position response of tower jib with disturbance	140
Figure 6-19 The velocity response of hoist and jib with disturbance	141

List of Tables

Table 3-1 Nomenclature variables of gantry crane model.....	35
Table 3-2 Nomenclature variables of tower crane model.....	43
Table 3-3 Nomenclature variable of second tower crane model	51

Abbreviations

l_x, l_y	Position of the load in x, y direction
l_{x_c}, l_{y_c}	The load position is seen in camera with relative to the hoist
ol_{x_c}, ol_{y_c}	The load position is seen in camera with relative to the hoist in the prior frame
v_{lx_c}, v_{ly_c}	The load velocity is seen in camera with relative to the hoist
$d\alpha$	The angle change
dr	The change in hoist radius
x_{world}, y_{world}	x, y world coordinates with respect to crane
x_{crane}, y_{crane}	x, y crane coordinates
s	Sine of angle
c	Cosine of angle
CPR	Count per revolution
ka	Steps per radian of jib motor
kr	Steps per radian of the hoist motor
kc	The factor of the camera (pixel/radian)
$dframe$	Camera frame interval
α_p	The angle perceived by processing

α_p	The angle perceived by processing in the prior frame
α_A	The angle of jib read by Arduino (steps)
rh_p	Distance travel by the hoist which observed by processing
orh_p	Distance travel by the hoist which observed by processing in the prior frame
rh_A	Distance travel by the hoist which observed by Arduino (steps)
vjx_p, vjy_p	The velocity of the jib in x, y directions measured by Processing
l_{xp}, l_{yp}	The load position in x, y direction read by Processing
v_{lxp}, v_{lyp}	The load velocity in x, y direction read by Processing
v_{lx}, v_{ly}	Velocities of the load in x, y direction
x_{target}, y_{target}	The demanded position of the load in x, y direction
v_{lxdem}, v_{lydem}	The demanded velocities of the load in x, y direction
v_{max}	The maximum velocity
dem	Positioning period constant
a_{dlx}, a_{dly}	The demanded acceleration of the load
h_{dx}, h_{dy}	The demanded positions of the hoist in x, y direction
$v\alpha_A$	Jib motor velocity calculated by Arduino

v_{rhA}	Hoist motor velocity calculated by Arduino
α	The angle of the jib rotates perpendicularly on the mast
α_d	The demanded angle of the jib
rh	The distance of the hoist moving along the jib
v_h	The velocity of the hoist along the jib
r_{hd}	the demanded distance of the hoist along the jib
v_{hd}	the demanded velocity of the hoist
$v_{\alpha d}$	the demanded angular velocity of the jib
u_α	The input of the jib motor
u_h	The input of the hoist motor
g	gravitational acceleration ($981cm/s^2$)
l	The length of the rope
dt	Change of time=0.005s

1 CHAPTER ONE: - INTRODUCTION

1.1 Overview

Though the term ‘Pragmatic’ is relatively new, the paradigm of pragmatic control is at least half a century old. ‘Pragmatic’ control can be understood to involve a more straightforward strategy that can be understood intuitively. Examples include ‘constrained nested control loops’ and ‘Logical Predictive Control’. These techniques can be given the label of ‘pragmatic’ strategies, being easy to understand. The term ‘pragmatic’ can also imply that the strategy can be implemented within a simple microcontroller.

In modern industrial systems, cranes are widely used for heavy load transfer. They have many applications in industries, shipyard, factories and construction sites. There are different types of cranes which are used for various applications including tower cranes, gantry cranes, boom cranes and mobile cranes. Different types of cranes are used under different circumstances and environments. The main focus in this thesis will be on gantry and tower cranes as they are the most common types of cranes in use in the construction industry-. The main emphasis of this research is on making crane systems work as fast and accurately as possible without severe swing. The required crane acceleration for the motion causes an undesirable load swing, and this has negative consequences for system control and the crane’s performance. Moreover, it reduces the efficiency of the crane and its operating speed. Besides the load position, for load swinging minimization, it is necessary to control the load swing angle.

1.2 Research Background

In many industrial applications, the operation of the crane system must depend on a skilled operator using visual feedback of the crane’s operation. The pendulum motion of the load presents a severe risk of damage or injury to the material and personnel on site. In recent decades, the control of cranes has led researchers to apply a variety of control strategies. The bridge crane operates using Cartesian (x - y) axes, while a tower crane uses polar (r - θ) axes. Both tower and gantry cranes have the configuration of three degrees of freedom, for instance, the gantry crane, the first one is horizontal movement of the hoist (trolley) in x direction while

the second one is the horizontal movement in y direction. The third degree of freedom is the vertical movement of the load. However, the tower crane has the first degree of freedom of rotational motion of the horizontal jib. And the other two are the horizontal movement of the hoist (trolley) along the jib and the vertical movement of the load. Normally, the crane is controlled by a skilled human operator. In case of gantry the operator is using a button pendant to lift the load and moves it to the desired position, while in tower crane, the operator usually sitting in a cab at the top of the crane.

The essential factors in crane operation are speed of operation, position accuracy and site safety. These factors play a vital role in working sites. For example, execute the duty quickly will keep the cost down, some duties need an accurate position of the load to be lowered such fixing a massive part into another, and the safety is required in the workplace to avoid the collisions between lifted load with surrounds and people.

1.3 Research Aim

The overall aim of this research is to investigate the application of pragmatic control techniques to crane systems. It is difficult for crane operators to lift and manoeuvre payloads without causing significant swing as evidenced by the uncontrolled motion of the load, especially the residual load swing. It is essential to control crane movement and suppress the swinging of the load during the operation for the safety of workers and persons and property nearby. To address these concerns, such systems attract research papers that often feature complex control algorithms requiring massive computation. This study aims to design a controller which controls the position of the crane load and suppresses the load swing. Computer vision can be used to monitor the crane load position and obtain real-time loads measurement as a reliable measurement of swing feedback in the controller loop. In addition, the proposed strategy can be implemented within a simple microcontroller.

1.4 Specific Research Objectives

The specific objectives of this research are:

1. To apply practical techniques on crane systems and ascertain whether a ‘pragmatic’ controller can be successfully applied, with ‘success’ determined by comparison with published benchmarks.
2. To derive mathematical models in the state-space form of crane systems for which solutions do not exist.
3. To make further refinements to the pragmatic control techniques, seeking to control such systems.
4. To implement the controller experimentally, embedding the implemented control techniques into a simple microcontroller.
5. To evaluate and analyse the reliability of the system and the performance of the implemented control strategies.

The main goal of this research is to design and implement a simple and robust pragmatic control technique to control crane systems. An example is a gantry or tower crane system where the swinging load of the crane must be moved from one point to another point in minimal time without generating a large swing.

1.5 Hypothesis

There are systems which are regarded as challenging to control such as cranes, and this risk means this issue has received the attention of many researchers who wish to apply some complicated mathematical analysis to them. The contention is that there are pragmatic methods that can achieve control, which is equally as good, but that is based on a more intuitive level.

This research will involve developing a model for a gantry and tower crane systems, based on mathematical state-space representation. The model will then be the basis on which to devise

a suitable control technique. Analysis of the performance of the suggested control approach will help to evaluate its feasibility. Finally, implementation of the proposed controller via an embedded microcontroller will enable a practical evaluation of the performance.

1.6 Research Contribution

When we compare this research with the previous work, the main contributions are the following:

1. A new design of a pragmatic control such as a cascaded constrained loop is proposed in Chapter three for crane systems which are not currently being used in such systems. The control target for these systems is to drive the systems as fast and accurately as possible without generating severe swinging.
2. The new design of the pragmatic control strategies will be implemented on a real system, and its performance will be evaluated through experiments and then compared with previous work. The proposed technique will be embedded in a simple microcontroller (Arduino).
3. Typically, studies in the literature have developed a variety of control techniques to control crane systems. In this thesis, a “pragmatic control” technique will be developed and applied to crane systems for the first time.
4. The most important application of this research, however, may be beneficial for educational purposes because it will help students to understand the real behaviour of these systems.

1.7 Thesis Structure

This thesis is divided into eight chapters, as follows:

The first chapter introduces the conducted research. It gives an introduction to the pragmatic technique and cranes. Furthermore, it outlines the research aims, objectives, thesis contribution and the structure of the thesis.

The second chapter introduces the literature review. It reviews the pragmatic control technique and where it has been used. Then, the previous works that have been done on gantry and tower

cranes which include deriving the system's model, methods they have been used, and the control techniques used to control the crane.

The third chapter demonstrates system modelling. State variables of the systems will be identified. Gantry and tower crane mathematical modelling will be derived using state-space equations.

The fourth chapter introduces the simulation studies. It states the design of the proposed technique and applies it to the gantry and tower crane operations by simulation. Control of the position and anti-swing of gantry and tower crane were simulated using JavaScript Language and MATLAB software.

The fifth chapter introduces the experimental study of gantry crane, the mechanical design of single-axis gantry crane will be demonstrated. It will represent the experimental setup of the system which include choosing motors and encoders, wiring and microcontroller. A camera has been used in crane systems to collect data of the load such as position and swing angle. In addition, the experimental results will be demonstrated in that chapter.

The sixth chapter demonstrates the experimental setup of the tower crane system. A practical experiment of tower crane has been designed. Pragmatic control technique has been used to control the position of the system and suppress if any load swing might occur during the operation. Also, it introduces the using of embedded microcontroller "Arduino" and applies it to control the system.

The seventh chapter presents the conclusion and scope for future work. The conclusion of this research will be presented, and the suggestion of future work improvement outlined.

2 CHAPTER TWO: - LITERATURE REVIEW

2.1 Introduction

The control systems are in use everywhere, from home to vehicles, from serious to trivial applications (Lumkes Jr, 2001) as the development of engineered control systems have been in existence since the latter half of the 20th century. The control relies on the dynamic and the modelling of the system. The term ‘dynamical system’ refers to how the states or the variables of any system develop or change over time. The state variables of a system can be defined as the minimum number of variables that uniquely define the dynamical states of a system.

A variety of mathematical models represents dynamical systems. Such models are often described by (ordinary or partial) differential equations, systems often represented by state-space or transfer function. The model of a system allows us to understand the physical behaviour of the system and how a system will behave, to analyse it and make an adequate prediction about it. Using the chosen model of any system will help to design a controller which will overcome the uncertainties of the system and force it to the desired behaviour. Systems range from simple such as a simple pendulum to complicated systems which have unpredictable behaviour. Systems can be regarded as ‘difficult’ if they are in high order, non-minimum phase, multi-input multi-output and constrained states giving nonlinearity. Cranes are considered to be a “difficult system” because they exhibit complex dynamic and non-holonomic behaviour (Duong et al., 2012).

Most of the existing research was conducted by using numerical and simulation only to control different systems. System simulation is represented as the running of the system's model. It may not always produce accurate results and it works on logical manipulation, whereas analytical and experimental results are accurate as these are obtained by proven mathematical manipulation. However, both simulation and experimental are complementary rather than competing, which both of them both can give significant insights of different nature. In this research, the intention is to develop robust pragmatic control techniques that are capable of controlling a two-type of crane system. Therefore, the practical solutions for controlling any dynamic system will demonstrate theoretical techniques experimentally showing both realistic success and failure and give an insight into methods of instruction.

2.2 Pragmatic Technique

The term “pragmatic” refers to the way of dealing based on practical rather than theoretical issues. Also, it means based on the Cambridge dictionary “solving problems in a sensible way that suits the conditions that exist now, rather than obeying fixed theories, ideas, or rules”. Most researchers prefer to carry out their research by simulation rather than experiment due to the low cost and simplicity. However, conducting an experiment especially in dynamic systems will solve the problems that exist in the systems as a designer facing many issues which affect the system, and definitely will give a broad picture about the demeanour of the conducting experiment. The essence of control of any dynamic system is to understand its dynamic behaviour (Billingsley, 2006). ‘Pragmatic’ control can be understood to involve a more straightforward strategy that can be understood intuitively. Examples include ‘constrained nested control loops’ and ‘Logical Predictive Control’, produced in the 1950s by (Coales and Noton, 1956, Chestnut and Wetmore, 1959). The research on this technique had been continued for several years in Cambridge (Adey et al., 1963, Dodds, 1984).

Classical proportional–integral–derivative PID control is often an inadequate, but computationally simple technique, such as Logical Predictive Control (LPC) or the method of ‘cascaded constrained loops’ that can address many such problems. These techniques can be given the label of ‘pragmatic’ strategies, being easy to understand. The term ‘pragmatic’ can also imply that the strategy can be implemented within a simple microcontroller which gives very fast response in real-time to the system (Billingsley and Ghude, 2015). This strategy uses a fast model of the plant that can predict the system’s behaviour through a straightforward simulation of plant dynamics.

There are various dynamic systems. Such systems attract research papers, often featuring complex control algorithms requiring massive computation. Different control strategies have been used to control systems ranging from classical PID control to logical techniques such as LPC or cascaded constrained loops. Further refinements of these techniques will be sought as one of the research objectives. These can be given the label of ‘pragmatic’ strategies.

(Falahian et al., 2014) used pragmatic modelling for a dynamic biological system through an artificial neural network and the experiments they conducted showed the simplicity of the modelling compared with existing models. (Huang, 2003) used a pragmatic approach to

identify the process and assess the performance of the control loop performance by applying linear quadratic Gaussian (LQR) and proportional-integrator-derivative PI/PID controllers. Both simulation and practical solutions have their capabilities and limitations, but they play a role in designing and controlling systems. However, the simulation of any parameter of system gives an optimal value of that component, compared with a practical solution which gives a real measurement. However, we cannot neglect the benefits of simulation studies as it is essential, for instance, to design a dynamic system and control it, but the practical solution is essential too. Moreover, the actual design of any system provides a real answer to what the system behaviour is. Therefore, the practical solution linking with simulation will provide better linkage between the diverse aspects of designing and controlling a dynamic system.

2.3 System Modelling

There are many methods for generating the dynamic equations of a mechanical system such as cranes. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation or analysis. Cranes can be divided into several types based on usages such as gantry crane, bridge crane and boom crane which have been used in the industry. However, tower crane and mobile boom crane are commonly used in construction sites. Nevertheless, it can be classified based on the degrees of freedom (DOF) of each system.

The modelling of the cranes divided into a single-pendulum and a double-pendulum cranes. Researchers have derived several mathematical models. In this thesis, the review of the previous work will be focused on the modelling and approaches that have been used by researchers.

2.3.1 Gantry Crane Modelling

The gantry crane is used in factories widely. It consists of a trolley that moves along a girder while the girder moves in perpendicular with respect of trolley. The modelling of the cranes is divided into a single-pendulum and a double-pendulum crane. The following section will review each one elaborately.

2.3.1.1 Single-Pendulum Modelling

Previous researchers have derived various mathematical models of different cranes, and most of the approaches were based on the lumped mass methods. The modelling of a gantry crane was by using the Lagrange method. The structure of the gantry crane consists of two subsystems: the trolley (cart) and the payload. Figure 2-1 shows the gantry crane system with its payload.

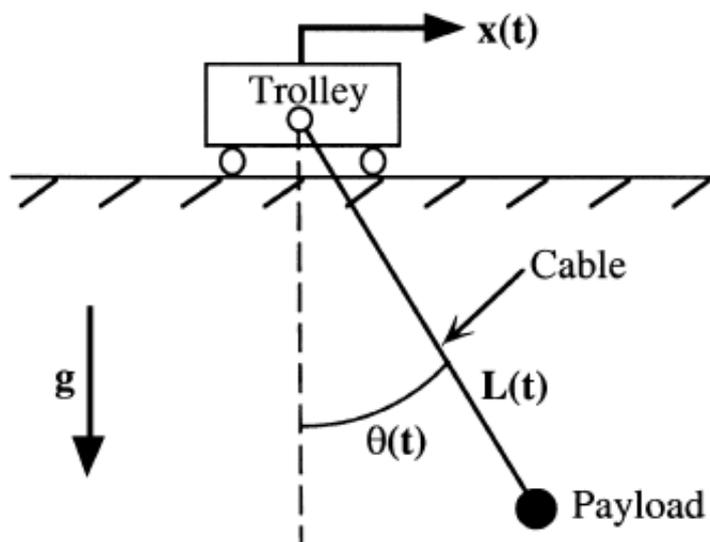


Figure 2-1 The Structure of Planar Gantry crane (Singhose et al., 2000)

Usually, the cart is driven by a force and the payload suspended from the trolley by a rope. In Figure 2-1, the configuration of the system is specified by the horizontal position of the trolley x , the length of the cable is L , and the swing angle of the cable is θ .

Most of the researcher used Lagrangian method for modelling the crane and designed the controller using equation 2.1 below:

$$L = T - V \quad (2.1)$$

Where T is total kinetic energy and V is the potential energy.

The method involves finding the kinetic and potential energies of the system and solving the Lagrange equation to obtain the mathematical equations representing the system.

(Omar, 2003) used the Lagrangian approach to derive the equations of motion. He assumed that the load and trolley position vectors are given by:

$$r_L = \{x + L\sin(\theta) - L\cos(\theta)\} \text{ And } r_T = \{x, 0\} \quad (2.2)$$

Then derived the kinetic and potential energies of the system. Also, he assumed that the cable length is fixed, and it may get small changes when it needed.

In addition, (Park et al., 2007) derived the model of a gantry crane using the Lagrange method. Their assumptions were: The payload and trolley are connected by a massless rigid rod, the trolley mass and the position of the trolley are known, all frictional elements in the trolley and hoist motions can be neglected, and the rod elongation is negligible.

While (Sun et al., 2014a) represented the model of gantry crane by using Lagrange's modelling technique. With considering the rope is rigid, the payload swing angle is within the scope of $(-\pi/2, \pi/2)$.

Another paper (Maghsoudi et al., 2016) used the mathematical model which is obtained based on the given characteristics of the crane by the manufacturer and the study by (Pauluk et al., 2001) who is, in turn, derived a 3D model of gantry crane using ten state equations describing the dynamics of the crane. The spherical system has been adopted to the Cartesian system to obtain the swing angle in x and y direction.

Meanwhile, (Abe, 2011) modelled an overhead crane which consists of trolley, payload and rigid rod, the rod is constrained to move in X - Z plane only like a pendulum. He derived the kinetic energy and the potential energy of the system and used the Lagrange method to model the equation of motion.

Furthermore (Nguyen et al., 2017) designed an overhead crane which moves on the track, with ignoring cable hardness, trolley friction, air resistance and plastic deformation. Their model was based on the model of (Yang and Yang, 2007). While (Park et al., 2008) used exactly the same modelling of (Yang and Yang, 2007). The model was based on Lagrangian formulation with assuming that, a massless rigid cable between load and gantry and the angular position and the velocity of the load is measurable. The equations of motion were as:

$$\gamma\ddot{x} + \beta\cos\theta\ddot{\theta} - \beta\sin\theta\dot{\theta}^2 = u \quad (2.3)$$

$$\alpha\ddot{\theta} + \beta\cos\theta\ddot{x} - \eta\sin\theta = 0 \quad (2.4)$$

Where $\gamma = M + m$, $\beta = ml$, $\alpha = ml^2$, $\eta = -mgl$ are constant system parameters.

Other research (Bruins, 2010) presumed that the crane system consists of four parts, the crane (M) is moved by a transport belt which is connected to a motor controlled by a frequency converter. During the movement, the load (m) will oscillate relative to the crane. He derived a state-space model for the motor, model of the mass and the load and assumed belt damping neglected.

Work conducted by (Yu et al., 2014) designed a two-dimensional overhead crane which has been derived from (Sun et al., 2011), who modelled a two-dimensional overhead crane with assuming that the rope is massless and inflexible, moreover, the payload during the process always beneath the trolley. The kinematic equation for the system was derived, which describes the coupling behaviour between trolley acceleration and the payload swing angle.

Other researchers (Vukosavljev and Broucke, 2014) modelled a gantry crane which described by a trolley moving along track carrying a load by fixed and rigid chord. The state model of the system was trolley position, trolley velocity, swing angle and angular velocity of the load. The output of the system is the position is trolley position and swing angle of the load.

Meanwhile (Huang et al., 2014) modelled a bridge crane with distributed mass payload, the equations of motion for this model were derived using the dynamics software package, MotionGenesis. The inputs to the model are the acceleration of the bridge and the acceleration of the trolley. The outputs are the swing angles of the suspension cable in x and y direction and

the payload swing angles relative to the suspension cable in x , y direction and the payload twist angle. It assumed that the trolley is significantly more massive than the hook and payload and the cable is rigid and massless.

In their work, (Tang and Huang, 2016) designed a bridge crane under windy condition. The system inputs are the acceleration of the trolley, the suspension cable length, and the wind force. The output is the swing angle of the suspension cable. Kane's method was used to derive the equation motion of the system.

Adding to our understanding, (He and Ge, 2016) assumed a hybrid partial differential equation–ordinary differential equation system that describes a non-uniform gantry crane system with constrained tension. A bottom payload hangs from the top gantry by connecting a flexible cable. The flexible cable is a non-uniform due to the spatiotemporally varying tension applied to the system. Dynamic equations of the non-uniform gantry crane (kinetic and potential equations) derived by using Hamilton's principle.

In 2011, (Shebel et al., 2011) derived the equation of motion using the Lagrange approach. The model consists of two equations which are the trolley and the load angle acceleration. Then the kinetic and potential energies. It has been assumed that the swing angle kept small to simplify the system model.

In their work, (Jaafar et al., 2013) preferred to use the Lagrange method to express the mathematical model of the crane. It has been considered that the crane system has two independent coordinates which are trolley displacement and payload oscillation. Kinetic and potential energies have been derived. Moreover, the dynamic of the DC motor is included in the complete form of the differential equation of the crane system.

Later, (Annur et al., 2018) designed a 3D model of an overhead crane which consists of three subsystems which are rail, cart and payload. The equations of motion were derived using the Lagrange method. The total kinetic energy is the sum of the kinetic energies of rail, cart and payload. The payload's potential energy was considered.

Another approach was used by (Singhose et al., 2000) who used a planar model of gantry crane. It has been assumed that the cable is inflexible, massless and pinned to the trolley.

The acceleration of the trolley and the hoisting velocity considered as the input of the system. However, the trolley position and the swing angle are the outputs.

Modelling of a two-dimensional overhead crane was done by (Zhang et al., 2017). In their work, a fixed-length rope was assumed. Nonlinear friction model has considered in the system modelling. A summation of the inertia matrix, the centripetal-Coriolis matrix, and gravity vector have been assumed that it is equal to the control input vector to express a dynamic form of the system. An assumption has been made in modelling which limits the swing angle between π and $-\pi$.

In their research, (Tumari et al., 2013) used a state-space form to represent the gantry crane. The trolley and the load considered as a point masses and assumed to move in two-dimensional, x - y plane. The trolley connected with payload by a rigid bar. The trolley mass, payload mass, gravitational acceleration, trolley position, trolley velocity, bar angle and bar angle rate considered as the parameters of the system.

In another research paper, (Almutairi and Zribi, 2016) designed a gantry crane system which consists of a cart moving on a one-dimensional track with a pendulum attached to it. Their model was based on commercial apparatus from Quanser. The mathematical model was provided with the system. It assumed that the system consists of cart position, payload position in x and y direction, masses of the cart and payload and the swing angle. Moreover, the acting force on the cart was calculated by subtracting the velocity of the cart and the applied voltage to drive the cart. The nonlinear equation of the system was based on Quanser pendulum model.

Also (Wu et al., 2015) designed a two-dimensional overhead crane. It assumed the trolley position, the payload swing angle and the overall force applied to the system is the system's parameters. Furthermore, bridge deformation, air resistance as well as stiffness and mass of the rope is neglected, and the load is considered as a point mass. The dynamics consist of the actuated part (trolley) and underactuated (payload), which defines the coupling behaviour between the trolley acceleration and the payload swing angle.

By contrast, (Fatehi et al., 2014) modelled an overhead crane. The dynamical model includes both transverse vibrations of the flexible cable and large swing angles of cable while the trolley is moving horizontally. Rayleigh-Ritz's discretization method was used to obtain the ordinary

differential equation model for transverse deflection of the cable. Also, Euler–Lagrange formulation was used to obtain a nonlinear dynamic model of the crane system.

There are various works established in single-pendulum modelling. However, there did not represent much of an improvement (Ramli et al., 2017). As mentioned, most of the previous researcher’s models used Lagrange method and involved the kinetic and potential energies in their models with slight differences between each other.

2.3.1.2 Double-Pendulum Modelling

Some researchers had considered a double-pendulum model when they designed a gantry crane system. It has been assumed that with the trolley being driven on a girder, the hook connected to the trolley by cable and the hook connected to the load by a cable as well. So it will be considered as double-pendulum mechanism if the mass of the hook is considered as shown in Figure (2-2).

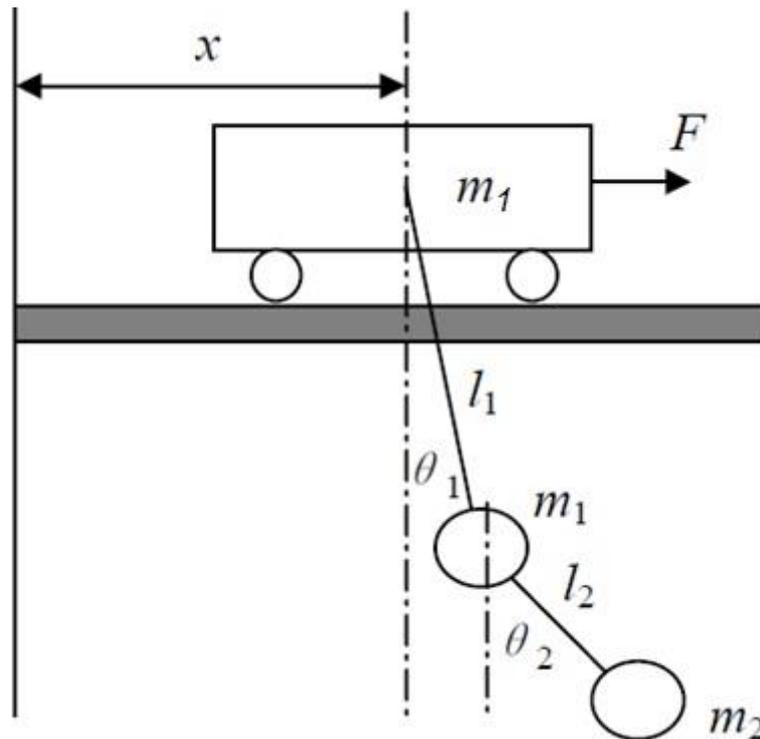


Figure 2-2 The scheme of a double-pendulum overhead crane by (Liu et al., 2005)

In Figure 2-2, the symbol m_1 describes the trolley mass, L the length of the rope, m_2 the mass of the payload, θ_1 the swing angle of the payload to the vertical line, while θ_2 is the swing angle

of the payload to the m_l . The position of the trolley is represented by x , F the applied force to the trolley by motor and g the gravitational acceleration. Some researchers like (Liu et al., 2004, Weiping et al., 2004, Liu et al., 2005) designed a double-pendulum overhead crane. All of them used the same model. For simplicity, it assumed that the trolley and the load regarded as a point mass, the friction force is neglected, the trolley, the hook and the payload is only moving in the x - y plane, and finally, the rope is non-extensional. The dynamic equation in x - y plane derived using the Lagrangian method. The system has two different natural frequencies due to the design of a double pendulum mechanism, and it depends on the cable length and the masses of payload and hook.

Researchers (Kim and Singhose, 2006) modelled a bridge crane with a double pendulum dynamic. The trolley moves by applying a force, two masses represent the hook and the payload connected by two cables which generate two pendulum angles. The frequencies depend on the two cables and the mass ratio (payload mass, hook mass). The model was limited to specific tasks because they assumed that the payload remains near the ground, and the sum of the two cables stays constant. However, in 2010, Kim and Singhose used the same model (Kim and Singhose, 2006) and applied a different control strategy (Kim and Singhose, 2010).

While (Ahmad et al., 2009) modelled an overhead crane with a double pendulum mechanism. The Euler-Lagrange formulation is considered in characterizing the dynamic behaviour of the crane system to incorporate payload. The complete model was the same as in (Liu et al., 2004, Weiping et al., 2004, Liu et al., 2005).

In their research, (Masoud et al., 2014) designed a double pendulum crane. Two mathematical models were derived. The first model is a fully nonlinear model of the double-pendulum crane derived using Lagrange approach, which is used for numerical simulation of the controller performance. The second is a linear decoupled model of the double-pendulum crane.

Meanwhile (Lee, 2013) considered a double pendulum model of an overhead crane. Based on Lagrange equations, the model of the system had been derived the same as the models mentioned previously. Three generalised coordinates correspond to the three degrees of freedom which are θ denotes swing angle of hook, ϕ is swing angle of the payload, and x expresses trolley displacement.

In their paper, (Huang et al., 2015) represented the double pendulum bridge crane transporting a distributed-mass beam. The payload attached to the hook by two massless rigging cables. The input to the model is the acceleration of the trolley. The outputs are the swing angle of the suspension cable, θ , and the payload swing angle relative to the suspension cable, β . The nonlinear equation of the motion was derived using Kane method, which is also a form of Lagrange method. Furthermore, the natural frequencies were derived, and it depends on cable length, rigging cable length payload length and mass ratio. However, (Zhang et al., 2016) used the dynamic model of (Weiping et al., 2004, Huang et al., 2015) and controlled the system.

In their research paper, (Qian et al., 2016) modelled an overhead crane system which consists of three subsystems: trolley, hook and payload. In Cartesian space, there exist three generalized coordinate variables to describe the system. Assumptions have been made in their models such as no friction, massless and rigid cables, mass-point hook and mass-point payload. The equations of motion have been obtained by the Lagrange method.

Alternatively, (Sun et al., 2016a) designed a crane with double pendulum dynamics. The equations of motion have been derived by Lagrange approach. They assume that only the position and swing-angles are measurable, while the velocities of the cart, hook and payload are unavailable. Furthermore, due to physical constraints, merely limited forces/torques can be produced by the actuating motors.

In their 2017 paper, (Jian and Mohamed, 2017) modelled an overhead crane system. They assumed that the hook and the payload are assumed to be a mass-point, the cable for the hook and payload is massless and inflexible, and the elongation of the cables during the motion of the cable is neglected. The dynamic model of the double-pendulum overhead system is obtained through the Euler-Lagrange method. The Euler-Lagrange method involves the total kinetic and potential energies of the system.

2.3.2 Tower Crane Modelling

A tower crane is widely used in construction sites. It consists of a trolley that moves along a rotating jib. The rotation of the jib will be in a horizontal plane. These movements of the jib and the trolley enable positioning of the trolley and the load over the demanded position in the

workplace. Many researchers have been designed and modelled a tower crane. It will be reviewed as follows:

(Golafshani, 1999) in his thesis mentioned that, there are no previous academic work on motion control of tower crane has been recorded. He derived a simplified speed-controlled model for the tower crane. The model was divided into two parts which are the trolley moving along the jib and the suspended load hanging from the trolley. He assumed that the crane's members are rigid, hook and load, considered as a point mass, no friction and non-flexible cable. The kinetic and potential energies equations have been derived, then the Lagrange method was used to obtain the tower crane model. Massive computations have been done to obtain the final simplified model.

In their work, (Al-Mousa, 2000, Al-Mousa et al., 2003) modelled a tower crane. Firstly, they derived the equation of the load position to the trolley. Then, use it to obtain the kinetic and potential energies equations. The equations of motion were derived using Lagrange approach. Also, they reformulated the equations of motion in a state-space form with ten equations which represented the system.

(Omar and Nayfeh, 2003) derived and modelled the tower crane using the Lagrange method. They simplified the model by assuming a small swing angle and fixed cable length due to the nonlinearity and complexity of the equations. Also, it has been assumed the accelerations of the trolley and jib have the same order of magnitude to simplify their model further which result of two groups of equations: translation motion equations and rotational motion equations.

Taking a different approach, (Lew and Khalil, 2000, Lew and Halder, 2003) modelled a robotic crane with two revolute joints carries a suspended load. The dynamics of the system can be obtained using the Lagrange equation. Assuming the swing motion is very small, and the load is a point mass. Then they linearized the model an expressed in a state-space form.

(Sadati and Hooshmand, 2006) derived the equations of motion of tower crane using the Lagrange approach, in the same way as other researchers (Omar and Nayfeh, 2001, Al-Mousa et al., 2003) by assuming the friction is ignored.

(Ju et al., 2006) studied the dynamic response of the tower crane and pendulum motion of the payload. Massive computation involves describing the system. The finite element method has

been used to obtain the model. The pendulum motion is represented as a multi-body system. Furthermore, integrated governing equations for the coupled dynamics problem are derived based on Lagrange's equations including the dissipation function.

(Singhose and Kim, 2007) represented a tower crane as of a planar double-pendulum crane. Assuming that the cable and rigging lengths do not change during the motion. The linearized equations of motion are representing the rate of change in two angles which are between the trolley-hook and hook-payload. Assuming the frequencies depend on the two cable lengths and the mass ratio.

In another research paper (Rubio-Avila et al., 2007, Avila et al., 2008) a design was created for a tower crane that was "self-balancing", which considered the lifted load and the horizontal displacement of the counterweight to maintain balance. Lagrange method was used to obtain the equations of motion.

A further contribution to the research (Vaughan, 2008, Vaughan et al., 2010) examined the dynamic of mobile tower crane and a tower crane with double-pendulum dynamic. The inputs are based on translation, base rotation, jib rotation relative to the base, the trolley position along the jib, and the cable length. The outputs are the hook deflection angle in the radial direction and the tangential direction. Assuming the crane is more massive than the load and the cable is massless and inelastic. The equations of motion were obtained using a commercial dynamic package.

Another work (Graichen et al., 2010, Böck and Kugi, 2013) described a laboratory tower crane system with five degrees-of-freedom, which are: the translation motion of the trolley, vertical motion of rope, rotation angle of the jib, swing angles the radial direction and in the tangential direction. The equations of motion of the crane derived via the Lagrange method. The second-order differential equations describe the dynamics of the system.

Meanwhile, Duong and his collaborators (Duong et al., 2012) considered a rotary tower crane which been studied by other researchers. The system is five-degree-of-freedom which is larger than the number of control inputs (3 inputs). Thus, the system has been considered an under-actuated system. The dynamic equations were transformed using dependent coordinates. The coordinates of the systems are the jib angle, trolley position, cable length and two swing angles of the load. The equations of the motion were derived using Lagrange approach.

Other researchers (Samin et al., 2013a, Samin et al., 2013b) modelled a three degree-of-freedom rotary crane. It has been assumed that the trolley position is fixed, the cable length is fixed and considered one swing angle which perpendicular to the jib. The rotary crane system is modelled in state-space form by considering the nonlinear equation of motion that obtained using Euler-Lagrange technique.

A different design was posited (Bariša et al., 2014, Matuško et al., 2015) featuring a three degree-of-freedom tower crane. The generalized coordinates of the system are jib angular position, swing angle, trolley position, pendulum swing angle, and cable length. Mathematical equations of the system motion were derived via Lagrange equations by defining total potential and kinetic energy of the system. Complicated computations have been done to obtain the model.

Other researchers (Perig et al., 2014) have introduced a model of three degree-of-freedom crane boom. Complex computation represents the system has been done. By using Lagrange equations and a geometric constraint equation, the nonlinear crane system model has been derived.

In his thesis, (Breuning, 2015) introduced the model of tower crane. A nonlinear mathematical model of the system without cable dynamics is derived by using the Lagrangian approach to obtain the equations of motion. Electrical and mechanical properties of the motors are used to model the actuators of the system. The friction of the trolley and the arm is included. For simplicity, it is assumed that the cable length to be constant.

Wu's research team (Wu et al., 2016) described their model of tower crane system which consists of tower, jib, trolley, steel cable, payload, and other mechanical and electrical components. The system is driven by three inputs which are: rotating torque of the jib, pulling force of the trolley and vertical force which control the cable length. It assumed that the cable length is fixed during operation. By using Lagrange equations and after a complex series of calculation, the tower crane system model was described.

A tower crane system was modelled by (Sun et al., 2016b) in their work. A four degree-of-freedom system has been considered. Massive computations have been done to obtain the dynamical equations which consist of the jib slew angle, the trolley translation displacement, the payload's swing. Also, the moment of inertia of the jib the trolley mass the payload mass,

the suspension rope length, the gravitational constant, the slew control torque, the translation control force, the friction torque and force with assuming the payload swing angles $\pi/2$ to $-\pi/2$.

Finally, (Alhassan et al., 2018) represented the modelling of a rotary crane. Some assumptions were made such fixed cable length, trolley movement and jib rotation are frictionless, and no external disturbances to reduce the complexity of the model. The kinetic and potential energies have been obtained. Lagrange approach was used to derive the dynamical model.

2.4 Control Techniques Used in Crane System

In general, crane presents a control problem where the undamped swinging of the load has led earlier researchers to apply a cumbersome control technique. Crane control is a well-studied problem. Several papers and thesis have been written on the subject. The next sections will describe the major contributions in this field. The previously used techniques can mainly be divided into the open loop and closed loop techniques. These techniques for both gantry and tower cranes will be reviewed and discussed accordingly.

2.4.1 Open-Loop Techniques

This technique is a type of continuous control system in which the output does not affect the control action of the input signal. Thus, it does not inform the output condition of the system. This technique has been used widely in crane system by many researchers because it is easy to implement. However, the disadvantage of this technique is sensitive to external disturbance, such as wind or ocean waves (Omar, 2003). Input shaping is one of the open-loop techniques. Input shaping is a command shaping method that has been used to limit the crane load oscillation. A series of impulses is convolved in real-time with the original reference command to create a shaped command.

In their research, (Singhose et al., 2000) studied the effects of hoisting on the input shaping method control of gantry crane. Several types of input shaping are evaluated and compared with time-optimal rigid-body commands. Input shaping showed that it does not yield exactly

zero residual vibration; however, when the hoist distance is small, the method yields essentially zero residual vibration.

Also (Park et al., 2000) used input shaping technique to control the container crane system. Time-efficient feedforward controls of input shaping are proposed for reducing the residual vibrations. It modelled as linear time-varying systems which do not yield zero vibration.

In their work (Huh and Hong, 2002) have investigated a modified input shaping control to suppress the residual swing angle in container crane. The conventional input shapers are enhanced by adding one more constraint to limit the sway angle of the load. The proposed method increases the travelling time of the crane.

Another paper by (Masoud and Daqaq, 2006) investigated a new approach of input shaping technique in container crane. The new approach was based on the graphical representation of the phase portrait that describes the response of a container crane payload to a double-step acceleration profile.

(Gürleyük et al., 2008) represented a three-step input shaping technique. It extended the zero-vibration derivative ZVD shaping technique into a generalized three-step shaper method. This technique was applied to the crane system, which consists of one motor, belt and a pendulum hanging from the belt.

Conversely, (Yanyang et al., 2011) proposed a control technique based feedforward input shaping technique for anti-swing control of the crane system. The used anti-swing technique mainly focuses on horizontal motion of the crane system. Furthermore, the motion is constrained in the vertical plane.

(Ho et al., 2014) investigated three input shaping control schemes of an overhead crane, which are zero vibration, zero vibration derivative and zero vibration derivative-derivative are applied to reduce sway motions of the load. It is assumed that the trolley of the crane moves along x -axis only.

The tower crane is unlike a gantry crane because of the tower rotation movement, which is a result of nonlinearity behaviour. Therefore, some researchers put assumptions to reduce these nonlinearities or apply multi-command to suppress the load oscillation. (Vaughan et al., 2010)

used input shaping method to limit the oscillation of a simple model of a tower crane. A multimode specified input shaper by utilising a technique that suppressed the two frequencies.

(Samin et al., 2013b) represented a Linear Quadratic Regulator (LQR) control then extended to an input shaping technique to reduce the swing angle of the load of three degree-of-freedom rotary crane. Positive and modified specified negative amplitude input shapers with the derivative effects are designed based on the properties of the system. It has been assumed that one swing angle which perpendicular to the jib and applies the technique to reduce the swing of it.

Some of the other researchers used a combination of several feedback control schemes with an input shaper such as Proportional integral derivative PID with zero vibration shaper as in (Maghsoudi et al., 2016).

In their paper, (Alhassan et al., 2018) represented an input shaping technique and shows the performance for sway control of the rotary crane. They used zero vibration, zero vibration derivative and zero vibration derivative-derivative and made a comparison between them.

Various attempts to control the crane using filters such as (Ahmad et al., 2010), they investigated the using of nominal characteristics following trajectory following and PI compensator for position control of cart movement. They were incorporating Infinite Impulse Response filter schemes for anti-swaying control of the system. This technique was applied for double pendulum gantry crane. They represented an infinite impulse response low pass filter technique for anti-sway control of a gantry crane system. This technique was applied to a one-dimensional gantry crane. In addition, they have implemented infinite impulse response filter techniques in hybrid control schemes of a rotary crane system. Linear Quadratic Regulator (LQR) control was developed for horizontal angle position control of the rotary crane and then extended to incorporate filtering techniques for anti-swaying control of the system. The system was consisting of an arm and a pendulum with a rigid rod connected to it.

The command smoothing technique was used to control the gantry crane. In their paper, (Huang et al., 2015, Huang et al., 2014) controlled a bridge crane with distributed-mass payloads using a command smoothing technique to suppress the payload oscillation. The designed smoother was a combination of low-pass, and multi-notch filter eliminates the first and the second mode frequencies of the payload swing. This technique was applied on one-dimensional bridge crane.

To conclude, open-loop techniques have been used by many researchers in case of crane system as it is potentially cheap. However, it is inadequate due to variations or disturbances and may become ineffectual to these systems.

2.4.2 Closed-Loop Techniques

The goal of any control system is to measure, monitor and control the system into specific requirements. A closed-loop control system is also known as a feedback control system. This technique uses feedback to maintain the demanded output condition by comparing it with the actual condition. The difference between output and reference input called “error signal” which returns and compares with demand reference to achieve the demanded action. This technique is commonly used to control the crane system. There are several types of these techniques which can be divided into a linear control (such as proportional integrator derivative PID, Linear quadratic regulator LQR), an intelligent control (fuzzy logic control, neural network), sliding mode control, an optimal control (such as Model predictive control MPC, linear quadratic Gaussian LQG), an adaptive control, and other control techniques. In this section, a brief review of each technique that has been used in crane system will be demonstrated.

One of the linear technique is proportional integrator derivative PID which is applied to control the cranes. (Wahyudi, 2007) have designed a sensorless control of the overhead crane. Also, classical PID controllers were designed to evaluate the proposed technique. A PID controller is adopted to control the trolley position while PD controller has been used for anti-swing control. (Cakan and Onen, 2016) have designed two PD controllers for trolley position and sway control of the crane. In the literature, it has been found that most of the PID controllers were designed with the aid of other techniques, or by using two PID controllers for control of the position and the load’s swing. (Solihin et al., 2008) proposed a method for tuning PID controller of automatic gantry crane control using particle swarm optimization (PSO). This method applied to find the optimal PID gains of the system. It was further proposed (Solihin et al., 2009) that a PID anti-swing control of crane could be based on Kharitonov’s Stability. The method uses a Genetic Algorithm (GA) in min-max optimization to find a stable, robust PID. In the optimization, the robustness of the controller is tested using Kharitonov’s stability. In their work, (Majid et al., 2013) made a comparison between PID and PD controller with input

shaping technique for gantry crane, while others (Tumari et al., 2013) proposed a PID control scheme for active sway control of a gantry crane system. Another research team (Diep and Khoa, 2014) designed three controllers for anti-swing, rope length and position control of a gantry crane. Particle swarm optimization was used to obtain the parameters tuning. The controllers of the system consist of a PID controller for position control of trolley, PI controller for length control of hoist rope and PD for anti-swing control. (Hussien et al., 2015) have used PID and PD controllers to control a gantry crane. A combination of the Priority Fitness Scheme and Particle Swarm Optimization (PFPSO) is used to optimize the parameters of PID and PD controller. Hai et al. (2017) have proposed a PID – Fuzzy Sliding Mode Control (PID-FSMC) algorithm for overhead crane system to reduce the sway angle of an overhead crane system.

The linear quadratic regulator has been used for control crane system, as proposed by (Kim et al., 2011) who have used the LQR method for the anti-sway control of a mobile harbour crane, while (Santhi and LB, 2014) employed a linear quadratic regulator for controlling the trolley position and swing motion which give better performance than simple PD and PID control.

For the tower crane, various attempts of using linear control were conducted to control the position and the swing angle of the load. (Rubio-Avila et al., 2007, Avila et al., 2008) have presented modelling and construction of “self-balancing” tower crane, also proposed theoretically a P without anti-sway control with velocity feedback, PID without anti-sway control, and PID with anti-sway control. In addition, (Arvin et al., 2014) designed Fuzzy-PID controllers for the rotary crane in MATLAB Simulink.

A Linear Quadratic Regulator (LQR) control was developed by (Samin et al., 2013a) for the tower rotation angle of the rotary crane. The used technique was incorporated with input shaping technique, while (Win, 2016) in his thesis has used LQR method to minimize the payload swing of a tower crane.

Adaptive control technique has been used by many researchers to control cranes. Based on the nonlinear model of the crane, adaptive control has the capability of estimating the parameter uncertainties and the external disturbances of the system. Several researchers (Hua and Shing, 2005, Yang and Yang, 2007) have proposed a nonlinear adaptive control scheme for an overhead system. Stability proof has been given of the overall system in terms of Lyapunov concept. (Sun et al., 2014b) presented an adaptive coupling control approach for under-actuated

cranes with load hoisting/lowering subject to unknown plant parameters. (Nguyen and Kim, 2015) have developed a nonlinear adaptive control of a 3D overhead crane the nonlinear control scheme employs adaptation laws that estimate unknown system parameters, friction forces and the mass of the load. The estimated values are used to compute control forces applied to the trolley of the crane. (Zhang et al., 2016) developed an adaptive tracking controller for double-pendulum overhead cranes subject to parametric uncertainties and external disturbances.

In case of tower crane, (Sun et al., 2016a) proposed an adaptive control scheme for under-actuated tower cranes to achieve simultaneous slew/translation positioning and swing suppression which was the first work for a tower crane.

Sliding mode control technique attracted researchers to be applied to the crane system. Many researchers have used this technique due to its capability to reject the uncertainties and the nonlinearities of the system. (Liu et al., 2004) proposed an adaptive sliding mode control method. This technique was applied on a double pendulum overhead crane system. It has been found that the sliding mode control technique was adopted with other techniques such as fuzzy control. A further proposal (Liu et al., 2005) was made for a composite sliding mode fuzzy control (CSMFC) approach for a double pendulum overhead crane system. The composite sliding mode function was consists of three sliding mode functions that are used to decouple the complex system. (Park et al., 2008) developed an adaptive fuzzy sliding-mode control (AFSMC) for the robust anti-sway trajectory tracking of overhead cranes subject to both system uncertainty and actuator nonlinearity. Their technique was consisting of fuzzy sliding-mode control law for anti-sway and fuzzy uncertainty observer to cope with system uncertainty as well as actuator nonlinearity.

In his research, (Lee, 2013) designed nonlinear controllers based on both conventional and hierarchical sliding mode techniques for double-pendulum overhead crane systems. Two controllers were designed for both tracking and anti-swing control.

By contrast, (Sun et al., 2014a) proposed a new anti-swing control scheme for under-actuated gantry crane subject to non-modelled uncertainties. Mainly, they constructed an elaborate manifold and then presented a nonlinear control law that keeps the system state always staying on the manifold.

Optimal control is to deal with the process of determining control and state trajectories for a dynamic system during a period of time to reduce the performance index. In another term, it is a set of the differential equation which describing the path of control variable to minimize the cost function. A numerous amount of research has been carried out by using this technique to control the crane system. For instance, (Vukov et al., 2012) have used a model predictive control technique to control an overhead crane system.

In their work, (Wu et al., 2015) have applied a model predictive control to an overhead crane. It has been used to minimize an objective function that is formulated as the integration of energy consumption and swing angle. This technique utilised the information of current displacement, velocity and swing angle for predicting the following control sequence (acceleration or force).

Other studies (Santhi and LB, 2014, Santhi et al., 2018) have used linear quadratic Gaussian LQG to control the position and the swing of the overhead crane. It is a combination of a Kalman filter with the linear quadratic regulator.

Also, the model predictive control technique has been applied to the tower crane. Such as (Graichen et al., 2010) have presented a model predictive controller to control a laboratory crane. This technique accounts for control constraints and based on the gradient projection method that allows for a time and efficient memory computation of the single iterations.

(Bariša et al., 2014) proposed a nonlinear model predictive control of tower crane based on reference shaping which was used to calculate optimal reference for the inner control loop of the tower crane.

(Böck and Kugi, 2013) have adopted a model predictive control scheme with a special focus on real-time feasibility with small sampling times. This technique was applied to the laboratory tower crane. The proposed controller aims at tracking the zero-path error manifold while additionally determining the time evolution along the path.

In his thesis (Breuning, 2015) used a linear-quadratic model predictive control to control a tower crane. Three controllers based on model predictive control were designed to control arm, trolley and cable.

Intelligent control techniques have been used to control the crane system, which can be divided into neural network control and fuzzy logic control. Neural network control involves two steps which are system identification and control. It involves a network of simple processing elements (artificial neurons) which can exhibit complex global behaviour, determined by the connections between the processing elements and element parameters. Some researchers used this technique to control the crane system, such as (Abe, 2011), which presented the use of a neural network to control the swing of an overhead crane. Radial basis function networks were employed to generate the desired trolley position. Thus, a particle swarm optimization was used as a learning algorithm in which a maximum swing angle after positioning is adopted as the objective function to be optimized.

(Tinkir et al., 2011) proposed a hierarchical artificial neural network-based adaptive fuzzy logic control of flexible link carrying pendulum system which was assumed as scaled a tower crane system. The proposed controller has two subsystem controllers such as fast and slow artificial neural-network-based fuzzy logic controllers which were sorted according to their importance in the control scheme.

Also (Duong et al., 2012) have proposed a recurrent neural network and developed by the evolutionary algorithm which uses the operators of a constricted particle swarm optimization and a binary-coded genetic algorithm. This technique was applied to three-dimensional tower crane system.

In their paper, (Matuško et al., 2015) proposed a control scheme for tower crane which consisted of a tensor product model transformation based nonlinear feedback controller, with additional neural network-based friction compensator. Neural network parameters adaptation law is derived using Lyapunov stability analysis.

Fuzzy logic control has been used to control many different systems. Many researchers have been used fuzzy logic control to control the crane system due to its capability to control the system without an accurate model of it. (Omar and Nayfeh, 2003) used a fuzzy logic controller to control the swing angle of the payload which consisted of a set of linguistic rules.

(Jalani, 2006) also adopted a fuzzy logic controller for gantry crane to control the payload position and the swing angle. Two fuzzy controllers were used for both position and swing control. The design of the controller was based on a heuristic approach.

(Smoczek and Szpytko, 2008) designed a fuzzy controller with Takagi-Sugeno-Kang fuzzy inference system for an overhead crane. Their controller design has consisted of 27 rules which expressed the control strategy based on information from three input signals: error of bridge position, the velocity of the bridge and swing angle of the load.

(Bruins, 2010) proposed a fuzzy logic controller for gantry crane. It was dependant on a rule base which based on the angle of the rod and the deviation in the position.

In their paper, (Shebel et al., 2011) proposed a fuzzy PD strategy to control an overhead system. It allows approximate adjustment of the controller's parameters according to methods for PD controller synthesis. Also (Kaur et al., 2014) have proposed a fuzzy controller to control the position of an overhead crane.

Furthering research in the field, (Almutairi and Zribi, 2016) have presented two fuzzy controllers to control the position of the cart of a gantry crane and suppressing the swing angle of the payload. A dual PD fuzzy controller was proposed. Two fuzzy logic controllers have been designed. The first fuzzy was designed to tune the gains of the first PD controller, which is used for regulating the position of the cart. The second fuzzy system was used to tune the

gains of the second PD controller, which is used for suppressing the swing of the payload during the movement of the crane.

Meanwhile, (Qian et al., 2016) developed a single input rule model-based fuzzy controller for control of the double-pendulum-type overhead crane. The controller includes six single input rule model that are dynamically weighted. The genetic algorithm was adopted to tune some parameters of the controller.

The implementation of fuzzy logic control also attracted researchers to apply it in rotatory/tower crane systems. Other research from (Al-Mousa, 2000, Al-Mousa et al., 2003) used fuzzy logic control which was applied to the rotary crane. The fuzzy logic controller was proposed first with the idea of split horizon. It used some fuzzy engines for tracking position and others for damping load swing. Each of these controllers has two fuzzy inference engines. Two input signals were used, which are the desired position of the trolley and rotational angle of the jib. The controller also receives four other inputs from feedback which are the actual position of the trolley, actual rotational angle of the jib, and two angles of the payload.

(Sadati and Hooshmand, 2006) proposed an anti-swing controller by using fuzzy clustering technique. An automated procedure by using fuzzy clustering for the determination of the number of operating points and their locations using.

(Wu et al., 2016) have proposed H infinity based adaptive fuzzy control technique to control the swing motion of tower crane. The control law was based on a variable structure adaptive fuzzy scheme. The proposed control scheme is based on Lyapunov stability to suppress the influence of the external disturbances and eliminate fuzzy approximation errors.

Several other controllers have been used to control the crane system such as (Weiping et al., 2004) who have proposed a passivity –based control method to control an overhead crane. This method was able to reduce the number of system states that need to be measured.

(Karkoub and Zribi, 2001) proposed a variable structure controller in conjunction with a state feedback control scheme, and a μ -synthesis control scheme is proposed to control the overhead crane.

(Tumari et al., 2012) in their research investigated the use of H_∞ controller with pole clustering based on linear matrix inequalities LMI techniques to control the payload positioning of gantry crane. The graphical LMI region profile gives flexibility in choosing the specific parameter of pole placement constraint.

(Hilhorst et al., 2013) proposed a reduced-order multi-objective H_∞ controller design approach for an overhead crane. A discrete-time linear time-invariant model was identified.

(Zhang et al., 2017) proposed an error tracking control method for overhead crane systems for which the error trajectories for the trolley and the payload swing can be pre-specified. Lyapunov techniques and LaSalle's invariance theorem are utilized to prove the convergence and stability of the closed-loop system.

Finally, (Rauh et al., 2017) have presented a gain-scheduled linear feedback control approach which was developed with the help of linear matrix inequalities LMI. An extended linearization technique for the oscillation suppression in crane systems, where the feedback gains are determined by using a robust optimization procedure which employs a formulation of the control task in terms of LMI.

2.5 Conclusion

In conclusion, the control of crane is essential to increase productivity and safety. Usually the most time-consuming task, when using a crane, is the movement of hoist and the load. The load needs to be transported as fast as possible without large swing. As reviewed, many works have been done previously to control the position of the crane as well as reducing the load swing. Regarding the dynamical model of the crane, most researchers have been deriving the model by using the Lagrange approach. Open-loop techniques have been shown to be effective. However, these are extremely sensitive to parameter variations, changing conditions and external disturbances, decreasing the reliability of the system's performance. , the previous techniques that have been used to control the crane such as PID, it showed that the PID controllers were developed with aid of other technique like neural network or use two PID controllers. Also, fuzzy logic controllers were considered complex due to the computational cost like fuzzification, fuzzy operator and defuzzification. In addition, fuzzy set and the fuzzy rules are difficult to determine. Furthermore, model predictive control has been used to control crane. It had its advantage to deal with constraints but more computational involved and it has poor performance with external disturbance. It also found that the work that has been done related to rotary/ tower cranes was quite low compared to those related to gantry/overhead cranes. Most of the solutions that have been conducted to control the crane system based on simulation and seldom tests on laboratory models (Smoczek and Szpytko, 2014). The practical of these control strategies remain to be further tested (Sun et al., 2014a, Sun et al., 2014b). In this thesis, A new design of a pragmatic control of a cascaded constrained loop is proposed for crane systems which are not currently being used in such systems. Moreover, the model of the system presents a state-space control method that can be implemented on a simple microcontroller. In addition, it uses computer vision to monitor the crane load position and then

used that position as feedback in a control loop. It has been proposed that a camera, mounted on the hoist and looking vertically downwards, can detect a corner-reflector mounted on the hook to give the essential signals for stabilisation. The significance of using a camera is to give an accurate coordinate of the load during the transportation.

3 CHAPTER THREE: - CRANE SYSTEM MODELLING

3.1 Introduction

It is necessary to have an equivalent representation of any system to study and examine the controllability of it. Whether solving a dynamic system analytically or preparing to simulate it, we need a mathematical representation of its variable. Usually, the intuitive way is by the application of Newton's Law. However, there are some constraints which lead to more general methods. However, these methods lack the simplicity of Newton's law. Most of the control systems contain mechanical or electrical or both types of components. It is necessary to convert these components into mathematical models to determine and analyse such systems. In the mathematical model, the diverse functionalities in the system are represented by the mathematical equations.

As mentioned in Chapter two, most of the previous crane systems models have been modelled using the Lagrange method. The method involves finding the kinetic and potential energies of the system and solving the Lagrange equation to obtain the mathematical equations representing the system. However, in this thesis, a state-space modelling method will be used to derive the system equation. A state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations.

In this chapter, we demonstrate a complete description of both gantry and tower cranes models, a derivation of the equations of motion will be explained theoretically and pragmatically.

3.2 Gantry Crane Model Description

In this section, a dynamical model of a gantry crane will be described. The model for the gantry crane consists of a trolley moves on a horizontal girder. The trolley consists of a hoist system (rope and hook) to lift and lower the load. Also, the girder moves perpendicularly to the trolley

movement. It is essential to know what part of the crane dynamics should be included in the control design and what part can be neglected. Figure 1 shows the gantry crane structure.

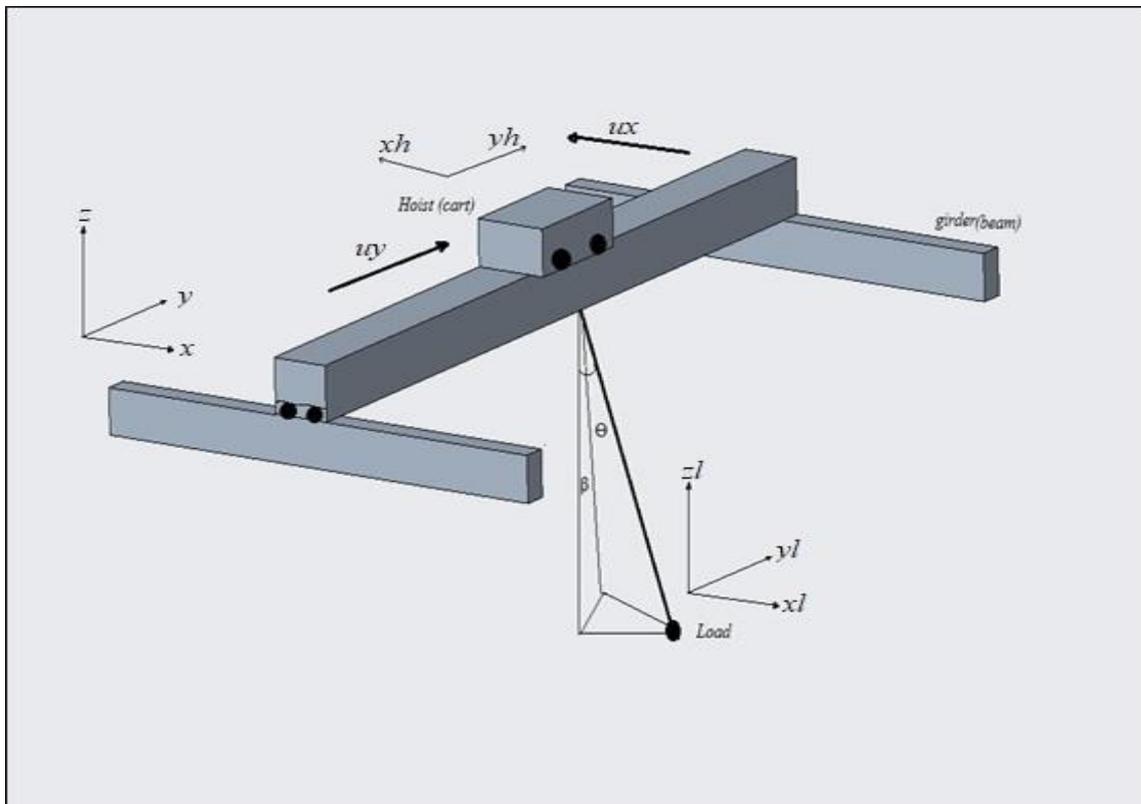


Figure 3-1 The structure of the gantry crane system

As shown in Figure 3-1, the structure of the gantry crane consists of

- (1) Two girders are fixed either in-wall or on four columns with tyres.
- (2) A beam or rail that is mounted to the girder.
- (3) A trolley that slides over the beam in a translation motion.
- (4) The second movement is either because the movement of the girders it was fixed on (columns with tyres or the beam) moves along the girder.
- (5) A suspension system of cables and hook. In the very general case, the length of the cable can be changed during load transport or at least at the pickup and endpoints.

The combination of trolley and girder (beam) movement allows the load to reach the desired destination. Therefore, the acceleration of the cart x and acceleration of the girder y are

considered the inputs of the crane system. In this chapter, the state-space equation method used to derive the system model, which makes the mathematical model of the gantry crane much realistic. Table 3.1 shows the used parameter terms of the gantry crane.

Table 3-1 Nomenclature variables of gantry crane model

Symbols	Description
l_x, l_y	Position of the load in x and y direction
h_x, h_y	Position of the hoist (trolley) in x and y direction
v_{lx}, v_{ly}	Velocities of the load in x and y direction
v_{hx}, v_{hy}	Velocities of the hoist (trolley)
x_{target}, y_{target}	The demanded position of the load in x and y direction
v_{xldem}, v_{yldem}	The demanded velocities of the load in x and y direction
h_{xdem}, h_{yldem}	The demanded positions of the hoist in x and y direction
u_x, u_y	Input in x -direction Input in y -direction
v_{error_x}, v_{error_y}	Velocity error in the x -direction

	Velocity error in the y-direction
g	gravitational acceleration ($9.81m/s^2$)
l	The length of the rope
dt	Change of time=0.01s
θ	The swing angle of the load with respect to x- axis
β	The swing angle of the load with respect to y- axis

Although there are many possible configurations, we consider that the beam is lying in the y-direction, with the rails conveying it in the x-direction. The cable raises or lowers the load in the vertical z-direction. Therefore, we define variables l_x, l_y and l_z for the three coordinates of the load, ignoring its tipping and twisting, h_x, h_y for the hoist, taking its z coordinate as zero. For the corresponding velocities of load and hoist, we take v_{lx}, v_{ly}, v_{hx} and v_{hy} . We take the cable length as l , and it's derivative as \dot{l} . The two motor inputs will be u_x and u_y , while the pairs of parameters that define their second-order response are a_x, b_x, a_y and b_y .

The proposed method is concerned with velocities and positions of system, and as a result of that, the mass of beam, hoist, rope and load will not be included in the system model. The essence of the pragmatic law is designed as a set of 'nested loops', where the innermost loop takes the form of a velocity loop wrapped around a motor to give smooth velocity control. Target values are derived from states in the outer loops, to be applied to a succession of inner loops, as shown in Figure 3-2.

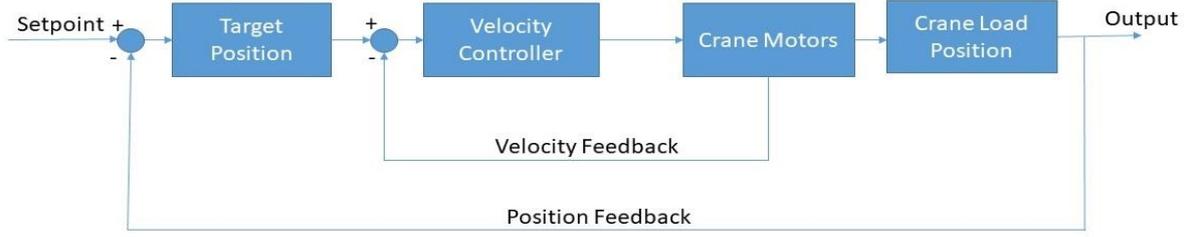


Figure 3-2 Block diagram of the system controller

These target values are subjected to constraints. The data on the hoist position and velocity in both axes, load position and velocity, the demanded positions of the hoist, the demanded position of the load, hoisting rope length and its time rate of change are assumed to be the main parameters of the design.

At first, we assume that the cable length is constant and that angles are small. We also assume that the inertia of the gantry motors is such that the effect of the swinging cable tension on their dynamics can be ignored.

We then have the following eight state equations:

$$\frac{dl_x}{dt} = v_{lx} \quad (3.1)$$

$$\frac{dl_y}{dt} = v_{ly} \quad (3.2)$$

$$\frac{dh_x}{dt} = v_{hx} \quad (3.3)$$

$$\frac{dh_y}{dt} = v_{hy} \quad (3.4)$$

$$\frac{dv_{lx}}{dt} = (h_x - l_x) \frac{g}{c} \quad (3.5)$$

$$\frac{dv_{ly}}{dt} = (h_y - l_y) \frac{g}{c} \quad (3.6)$$

$$\frac{dv_{hx}}{dt} = a_x u_x - b_x v_{hx} \quad (3.7)$$

$$\frac{dv_{hy}}{dt} = a_y u_y - b_y v_{hy} \quad (3.8)$$

We then have the following eight state equations of the system, which can be expressed in the form of a matrix:

$$\frac{d}{dt} \begin{bmatrix} l_x \\ l_y \\ h_x \\ h_y \\ v_{lx} \\ v_{ly} \\ v_{hx} \\ v_{hy} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -g/c & 0 & g/c & 0 & 0 & 0 & 0 & 0 \\ 0 & -g/c & 0 & g/c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -bx & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -by \end{bmatrix} \begin{bmatrix} l_x \\ l_y \\ h_x \\ h_y \\ v_{lx} \\ v_{ly} \\ v_{hx} \\ v_{hy} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ ax * u_x \\ ay * u_y \end{bmatrix} \quad (3.9)$$

Later we can add two more equations to these, relating to the cable length, and also include the effects of the cable tension on the hoist, we can simulate the solution to the equations with simple Euler integration, with a time step of dt , such as

$$l_x = l_x + v_{lx} * dt \quad (3.10)$$

where any desired accuracy can be achieved by making dt sufficiently small.

Now to complete the strategy, we need to define the demanded load velocities and the demanded hoist position v_{xldem} , v_{yldem} , h_{xdem} , and h_{ydem} that will be used to reach the target at x_{target} and y_{target} .

Velocity demand of the load is determined based on the difference between the target position r and the position of the load, which described below:

$$v_{lxdem} = (x_{target} - l_x) \quad (3.11)$$

$$v_{lydem} = (y_{target} - l_y) \quad (3.12)$$

Which is limited by

$$\begin{aligned} & \text{if } (v_{lxdem} > v_{max}) \{ v_{lxdem} = v_{max} \} \\ & \text{if } (v_{lxdem} < -v_{max}) \{ v_{lxdem} = -v_{max} \} \\ & \text{if } (v_{lydem} > v_{max}) \{ v_{lydem} = v_{max} \} \\ & \text{if } (v_{lydem} < -v_{max}) \{ v_{lydem} = -v_{max} \} \end{aligned} \quad (3.13)$$

Then, velocity error is the variable for determining the difference between the demanded velocity and the actual speed. Value for Velocity Error decays as the hoist is approaching the target position. Using velocity-demand control, the hoist attempts to reach the entire distance at the highest speed possible. Eventually, the speed demand approaches zero as the target position became closer. So, the acceleration of the load as described in equation 3.14, 3.15 below:

$$v_{error_x} = v_{lxdem} - v_{lx} \quad (3.14)$$

$$v_{error_y} = v_{lydem} - v_{ly} \quad (3.15)$$

The demanded hoist position is defined based on the summation of velocity error of the load and the actual load with adding the load position which presented in equations 3.16, 3.17 below:

$$h_{xdem} = l_x + kh * v_{error_x} \quad (3.16)$$

$$h_{ydem} = l_y + kh * v_{error_y} \quad (3.17)$$

Where later the gain parameter kh will be made to depend on ζ , which is a value of 2.5 after the tuning process through the model simulation.

The drive u_x , u_y are made to be the hoist position error in both axes (Equation 3.18, 3.19). Since the actual drive of electric motors was limited to the rated design, a limiter was set at the output variable that constraint the maximum drive to within limited range in both directions. A plus sign represents movement in the forward direction, while a minus sign is a movement in reverse (Equation 3.20, 3.21).

$$u_x = (h_{xdem} - h_x) \quad (3.18)$$

$$u_y = (h_{ydem} - h_y) \quad (3.19)$$

$$u_x = \begin{cases} \text{if } u_x > 1 & (u_x = 1) \\ \text{if } u_x < -1 & (u_x = -1) \end{cases} \quad (3.20)$$

$$u_y = \begin{cases} \text{if } u_y > 1 & (u_y = 1) \\ \text{if } u_y < -1 & (u_y = -1) \end{cases} \quad (3.21)$$

Where u_x , u_y are then limited to the maximum proportion of full motor drive of one. If necessary additional velocity feedback can be applied to the motors to augment the self-damping b_x , and b_y that is assumed.

Now having chosen the various control parameters, the dynamics of the system can be computed by such as the load velocities in x and y direction. The new velocity of the load will be computed instantaneously in both direction (Equation 3.22, 3.23)

$$v_{lx} = v_{lx} + (h_x - l_x) * g * \frac{dt}{c} \quad (3.22)$$

$$v_{ly} = v_{ly} + (h_y - l_y) * g * \frac{dt}{c} \quad (3.23)$$

Where v_{lx} and v_{ly} are the velocities of the load in x and y respectively with adding the updating velocity of the load at each instant.

While the new positions of the load in each instant computed by updating the load position in both directions during the movement (Equation 3.24, 3.25).

$$l_x = l_x + v_{lx} * dt \quad (3.24)$$

$$l_y = l_y + v_{ly} * dt \quad (3.25)$$

The essence of this law is designed as a set of ‘nested loops’, where the innermost loop might take the form of a velocity loop wrapped around a motor to give smooth velocity control. Target values are derived from states in the outer loops to be applied to a succession of inner loops. These target values are subjected to constraints. The data on the swing angle, hoist position and velocity in both axes, load position and velocity, hoisting rope length and its time rate of change, are assumed to be known.

3.3 Tower Crane Model Description

In modern industrial systems, tower cranes are widely used for heavy loads transfer. It has many applications, mainly in construction sites. Generally, cranes can be regarded as under-actuated system due to their independent control actuator being less than degrees of freedom to be controlled. Mostly, tower crane consists of a supporting mechanism which is part of its structure and hoisting system. The hanging load, which is part of the hoisting system, often exhibits an oscillatory behaviour due to the under-actuation of the system. Therefore, it is essential to keep the oscillation as less as the operator can to avoid the collision and to meet safety requirement.

3.4 Tower Crane Model Design

The tower crane consists of trolley moving along a jib in translation motion in x and y directions, and the jib rotating in a horizontal plane around the z -axis. The trolley consists of a hoist system (rope and hook) to lift and lower the load. The combination of trolley and jib movement allows the load to reach the desired destination. To make it simple, a few assumptions were considered in this work.

- 1) The crane is considered as a rigid body. The crane body consists of metal parts robustly joined together, which allows slight flexibility under normal working conditions. Furthermore, we neglect frictional torques in the mechanism.
- 2) The load and the hook were considered as a point mass.
- 3) The cable does not flex or stretch under the load.
- 4) The crane is driven by motors.

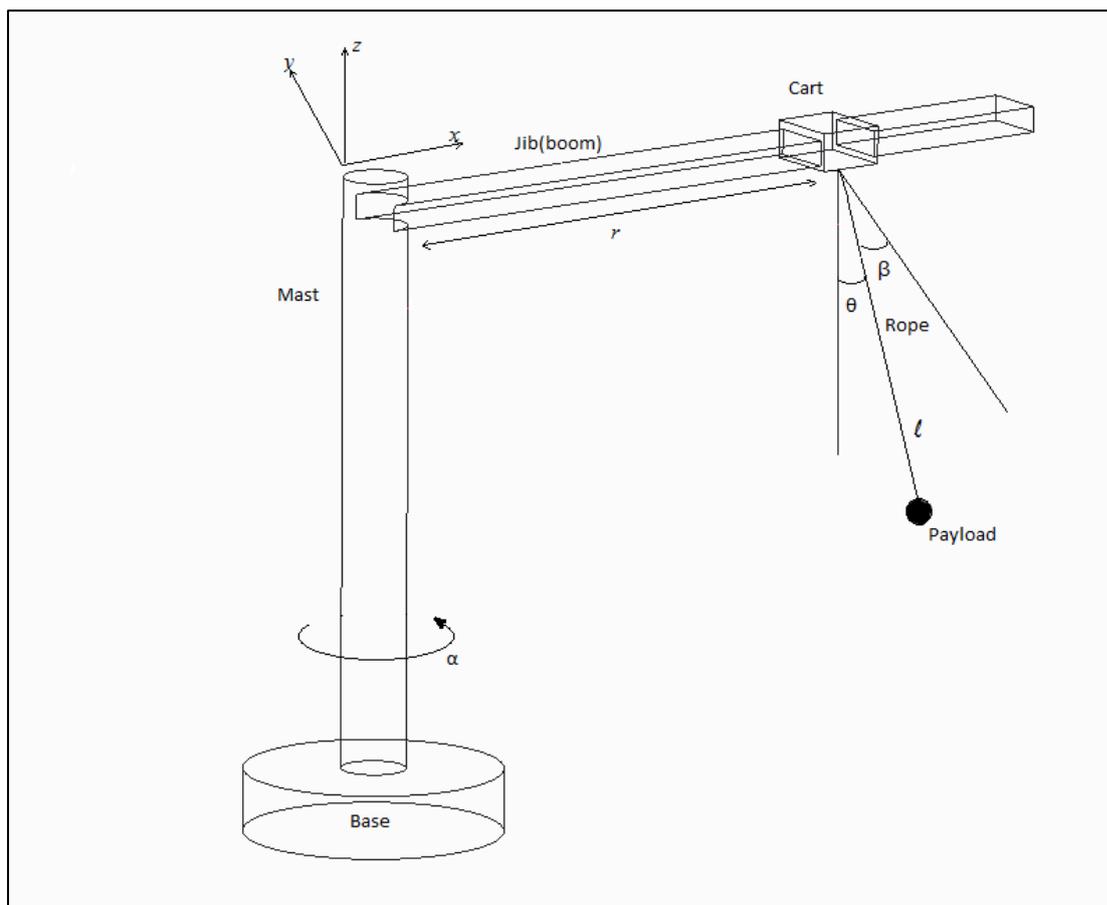


Figure 3-3 3D Tower Crane Model

As we can see in Figure 3-3 the jib rotates perpendicularly on the mast with angle α . In addition, the cart moves in translation motion along the jib with r distance. Therefore, the angle α , position of the cart r and changing cable length l are considered the inputs of the crane system.

Various possible models of tower crane can be obtained, which depend on the used parameters, limitations and assumptions. Therefore, in this thesis, two models of tower crane have been derived. The first model is based on state-space representation with nested feedback. The second model also based on the state-space model and the practical point of view. However, the idea is that the whole model configuration is driven by the view from the crane.

3.4.1 Tower Crane Model Based on State Space Representation

In this section, we present the model of tower crane based on state-space representation. It follows from Figure 3.3 that the angle α is the jib rotation of the crane and r is the distance which the hoist moves along the jib. In essence, a state-space representation is used to derive the system model, which makes the mathematical model of the tower crane much realistic. Table 3-2 shows the used parameter terms of the crane.

Table 3-2 Nomenclature variables of tower crane model

Symbols	Description
l_x, l_y	Position of the load in x and y direction
h_x, h_y	Position of the hoist (trolley) in x and y direction
v_{lx}, v_{ly}	Velocities of the load in x and y direction
v_{hx}, v_{hy}	Velocities of the hoist (cart) in x and y direction

t_x, t_y	The target position in x and y direction
v_{dx}, v_{dy}	The demanded velocities of the load in x and y direction
h_{dx}, h_{dy}	The demanded positions of the hoist in x and y direction
α	The angle of the jib rotates perpendicularly on the mast
v_α	The rate of change of the angle α
h	The distance of the hoist moving along the jib
v_h	The velocity of the hoist along the jib
a_{dx}, a_{dy}	the demanded acceleration in x and y direction
α_d	The demanded angle of the jib
h_d	The demanded hoist distance
v_{\max}	The maximum velocity
u_α	The input of the jib rotation motor
u_h	The input of the hoist motor
g	gravitational acceleration ($9.81m/s^2$)

C	The length of the rope
dt	Change of time=0.01s

Figure (3-4) below shows the top view of the tower crane, which consists of jib, hoist and load model.

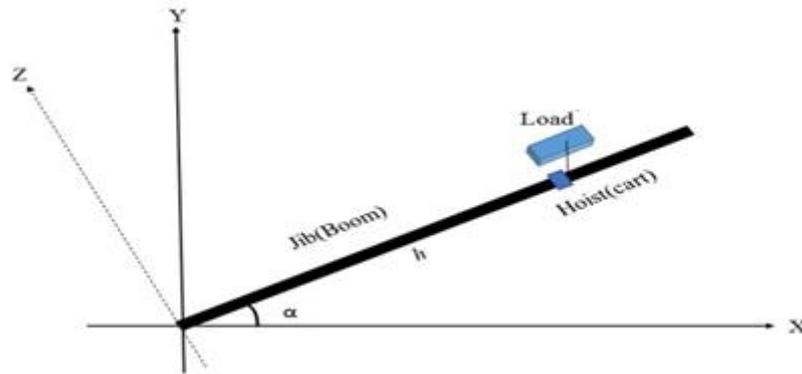


Figure 3-4 Tower Crane Model; jib, hoist and load

To derive the equations, it has been assumed the rotation of the jib will be between $x - y$ plane and perpendicular on the z axis. To define the system inputs, as figure (3.4) shows that the rotation angle of the jib is α . So the rate of angle change is the angular velocity of the jib is $\dot{\alpha} = \omega = v\alpha$ while the rate of change of the angular velocity is the acceleration of motor that rotates the jib is $\dot{\omega} = k_1 u_1 - k_2 \omega$ where k_1, k_2 are constant, u_1 is an input for the motor and ω is the angular velocity.

As we mentioned previously, the cart (trolley) moves in translation motion along with the jib. The velocity of the hoist depends on the acceleration of the hoist's motor drive. The state equation of the cart velocity defined as the rate of change of the travelled distance of the trolley, which yields equation as $\dot{h} = v h$. Similarly to the first input of the system, the derivation of the cart velocity gives us the acceleration of the drive that represented in the equation $\dot{v} h = k_3 u_2 - k_4 v$ where k_3, k_4 are constant, u_2 is the input for the motor and v is the trolley velocity.

Now we define the load position. Two angles characterize the load swings, θ and β . The angle θ is the angle of load's rope with respect of zx the plane while the angle β is the angle of rope with respect of zy plane as shown figure (3.3). Therefore, the main purpose of the controller is to move the load for one place to another with minimal load swing.

At first, we assume that cable is fixed, and the angles are small. We ignored the inertia of the driving motors due to the effect of the swinging cable on their dynamics is insignificant. The two motor inputs will be driving the system, which is the angle drive and the hoist drive while the pairs of parameters that define their second-order response are ax, bx, ay and by .

To model the crane, we have to define a set of state variables, choosing names for them that will be compatible with code. The position of the load will be defined in three Cartesian directions, which we define as l_x, l_y and l_z . This will allow us to define the three components of the load velocity as v_{lx}, v_{ly} and v_{lz} . Similarly, we can represent the three coordinates of the hoist position as h_x, h_y and h_z , and those of its velocity as v_{hx}, v_{hy} and v_{hz} . The Cartesian coordinates of the hoist are computed from the jib angle α and radius h with rates $v\alpha$ and vh . Basically, a state space modelling is used to derive the system model which makes the mathematical model of the tower crane much pragmatic. Therefore, when we simulate the dynamics with simple Euler integration with time-step dt , the obvious equation. We then have the following state equations:

$$\frac{dl_x}{dt} = v_{lx}, \frac{dl_y}{dt} = v_{ly}, \frac{dh_x}{dt} = v_{hx}, \frac{dh_y}{dt} = v_{hy} \quad (3.26)$$

Transformation from tower base to pivot of boom is just a matter of adding tower height to the z coordinate. In the simulation, the load velocity is transformed at each step. For small deflections, we can assume that the load height remains constant if the cable length has a constant value cable. The cable will apply a lateral acceleration to the load.

$$\begin{aligned} \frac{dv_{l_x}}{dt} &= (h_x - l_x) \frac{g}{c}, \\ \frac{dv_{l_y}}{dt} &= (h_y - l_y) \frac{g}{c}, \end{aligned} \quad (3.27)$$

Where $g = 9.81m/s^2$ the gravitational acceleration constant

Furthermore, the hoist acceleration and jib angular acceleration will be calculated as follows:

$$\begin{aligned}\frac{dv_h}{dt} &= au_h - bv_h, \\ \frac{dv_\alpha}{dt} &= a u_\alpha - bv_\alpha\end{aligned}\tag{3.28}$$

We need to define the hoist coordinates which are derived from:

$$h_x = h \cos \alpha, \quad h_y = h \sin \alpha\tag{3.29}$$

When we start to consider control, we must start with a target for the load t_x and t_y . The algorithm computes demanded velocities v_{dx} , v_{dy} which are proportional to the errors between load and target positions as follows:

$$v_{dx} = kv(t_x - l_x), \quad v_{dy} = kv(t_y - l_y)\tag{3.30}$$

But the vector demand is then limited in magnitude while maintaining its direction.

$$\begin{aligned}if(v_{dx} > v_{\max})\{v_{dx} = v_{\max}\} \\ if(v_{dx} < -v_{\max})\{v_{dx} = -v_{\max}\} \\ if(v_{dy} > v_{\max})\{v_{dy} = v_{\max}\} \\ if(v_{dy} < -v_{\max})\{v_{dy} = -v_{\max}\}\end{aligned}\tag{3.31}$$

Where v_{\max} is the maximum velocity which is limited to 2 m/s

Now we have a load velocity error and seek to apply an acceleration to correct it. We calculate a demanded acceleration a_{dx} , a_{dy} which are proportional to the velocity error,

$$\begin{aligned}a_{dx} &= kc(v_{dx} - v_{lx}) \\ a_{dy} &= kc(v_{dy} - v_{ly})\end{aligned}\tag{3.32}$$

Where the parameter kc is set to be 2

Nevertheless, we limit the demanded acceleration in both x and y axes in magnitude as shown below:

$$\begin{aligned}
 & \text{if } (a_{dx} > 1) \{a_{dx} = 1\} \\
 & \text{if } (a_{dx} < -1) \{a_{dx} = -1\} \\
 & \text{if } (a_{dy} > 1) \{a_{dy} = 1\} \\
 & \text{if } (a_{dy} < -1) \{a_{dy} = -1\}
 \end{aligned} \tag{3.33}$$

Now the acceleration is proportional to the horizontal displacement of the hoist relative to the load so that we can compute a demanded position of the hoist from:

$$\begin{aligned}
 h_{dx} &= l_x + a_{dx} \times \frac{c}{g} \\
 h_{dy} &= l_y + a_{dy} \times \frac{c}{g}
 \end{aligned} \tag{3.34}$$

Tower crane system has two inputs which are the angle α and the hoist move along the jib h . For the angle drive u_α , is made based on the difference between the demanded angle and the actual angle of the jib.

$$u_\alpha = ka(\alpha_d - \alpha) \tag{3.35}$$

Where later the gain parameter ka will be made to depend on cable length, which is a value of 20 after tuning. In addition, for the hoist drive u_h , is made based on the difference between the demanded distance of the hoist and the actual hoist distance along the jib.

$$u_h = kh(h_d - h) \tag{3.36}$$

Then, the gain parameter kh is deduced after tuning, which is 50.

Since the actual drive of electric motors was limited to the rated design, a limiter was set at the output variable that constraint the maximum drive to within limited range in both directions. A plus sign represents movement in a forward direction, while a minus sign is a movement in reverse.

$$\begin{aligned}
& \text{if } (u_\alpha > 0.4)\{u_\alpha = 0.4\} \\
& \text{if } (u_\alpha < -0.4)\{u_\alpha = -0.4\} \\
& \text{if } (u_h > 4)\{u_h = 4\} \\
& \text{if } (u_h < -4)\{u_h = -4\}
\end{aligned} \tag{3.37}$$

The limit ranges are set between 0.4 to -0.4 radian/sec² for the jib and 4 to -4 cm/sec² for the hoist which been deduced by the simulation to obtain the optimum value. The rate of change of the angle v_α has been computed based on the state equation (3.28):

$$\frac{dv_\alpha}{dt} = a u_\alpha - b v_\alpha$$

We were able to add a demand signal to the feedback with k taking a large value, the motor will accelerate rapidly to reach the demanded value, applying full drive for any substantial error. When the control is discrete time, however, as in computer control, the requirement for a stable response will put a limit on the feedback value that will depend on the sample rate. Therefore, the value of a_y is 80 and b_y is 20, which are deduced by tuning the parameters of the equation throughout the experiment. we were able to obtain a response without overshoot.

So now the v_α equation will be calculated as shown below:

$$v_{\alpha t+1} = v_{\alpha t} + 20(4u_\alpha - v_{\alpha t})\Delta t \tag{3.38}$$

t and $t+1$ subscript have been used to indicate current and previous variable values.

Moreover, the new angle of the jib will be computed instantaneously as shown below:

$$\alpha = \alpha + v_\alpha \Delta t \tag{3.39}$$

The velocity of the hoist will be updated instantaneously during travelling, and it computed based on the state equation (3.28):

$$\frac{dv_h}{dt} = a u_h - b v_h$$

Where ax is 100 and bx is 20, which are deduced by tuning the parameters of the equation. This tuning protocol has been set pragmatically which can use the command input in a similar way like the experiment.

So now the vh equation will be calculated as below:

$$v_{ht+1} = v_{ht} + 20(4u_h - v_{ht})\Delta t \quad (3.40)$$

where t and $t+1$ subscript have been used to indicate current and previous variable values.

Furthermore, the distance of the hoist along the jib will be updated and computed as below:

$$h = h + v_h \Delta t \quad (3.41)$$

Finally, to calculate the swing angles of the load β and θ , the formulas below describe the calculation of the load angle:

$$\beta = \sin(h_x - l_x) / c \quad (3.42)$$

$$\theta = \sin(h_y - l_y) / c \quad (3.43)$$

In conclusion, a simple mathematical model was derived for the tower crane system using state-space representation. A practical controller for position and anti-swing of the tower crane is designed to transfer the load as fast as possible with suppressing the load swing.

3.4.2 Tower Crane Model Based on Practical Point of View

In this section, we present the model of tower crane based on state-space representation. It follows from figure 3.2 that the angle α is the jib rotation of the crane and the rh is the distance which the hoist moves along the jib. However, this model will mimic the real system by using a virtual camera, and the whole system is driven by the view from the crane. Basically, the crane hoist will be in the centre of the graphical user interface, and the camera view will rotate as the crane moves. This model was mainly observed based on the tower crane experiment. The tower crane experiment details will be explained thoroughly in Chapter six, however, now

a brief explanation about the essence of the experiment and model will be conducted to give a concise picture of how the model was derived.

Initially, two DC motors have been used to operate the tower crane. One motor was to rotate the jib while the second was to drive the hoist along the jib. These motors were controlled using the Arduino microcontroller board. A camera was installed underneath the cart to observe the position and velocity of the payload. The camera was operated using processing language.

The speed of the motor which rotates the jib is 14 rotation per minute (RPM) with a gear ratio of 270:1. While the speed of the motor which drives the hoist along the jib is 133 RPM with a gear ratio of 30:1, both motors were with a built-in encoder with 64 counts per revolution CPR. The CPR refers to the number of quadrature decoded states that exist between the two outputs A and B signals. With both outputs A and B switching between high and low, there exist 2 bits of information represented as four distinct states. The term quadrature decoding describes the method of using both outputs A and B together to count each state change. This results in four times the amount of counts that exist for each pulse or period.

Before we start to derive the modelling equation, we need to define the variable of the tower crane model. Table 3.3 shows the used parameter terms of the crane.

Table 3-3 Nomenclature variable of second tower crane model

Symbols	Description
l_x, l_y	Position of the load in x, y direction
l_{x_c}, l_{y_c}	The load position is seen in camera with relative to the hoist
ol_{x_c}, ol_{y_c}	The load position is seen in camera with relative to the hoist in the prior frame
$v_{l_{x_c}}, v_{l_{y_c}}$	The load velocity is seen in camera with relative to the hoist
$d\alpha$	The angle change

dr	The change in hoist radius
x_{world}, y_{world}	x, y world coordinates with respect to crane
x_{crane}, y_{crane}	x, y crane coordinates
s	Sine of angle
c	Cosine of angle
CPR	Count per revolution
ka	Steps per radian of jib motor
kr	Steps per radian of the hoist motor
kc	The factor of the camera (pixel/radian)
$dframe$	Camera frame interval
α_p	The angle perceived by Processing
$o\alpha_p$	The angle perceived by processing in the prior frame
α_A	The angle of jib read by Arduino (steps)
rh_p	Distance travel by the hoist which observed by Processing
orh_p	Distance travel by the hoist which observed by Processing in the prior frame
rh_A	Distance travel by the hoist which observed by Arduino (steps)
vjx_p, vjy_p	The velocity of the jib in x, y directions measured by Processing

l_{xP}, l_{yP}	The load position in x, y direction read by Processing
v_{lxP}, v_{lyP}	The load velocity in x, y direction read by Processing
v_{lx}, v_{ly}	Velocities of the load in x, y direction
x_{target}, y_{target}	The demanded position of the load in x, y direction
v_{lxdem}, v_{lydem}	The demanded velocities of the load in x, y direction
v_{max}	The maximum velocity
dem	Positioning period constant
a_{dix}, a_{diy}	The demanded acceleration of the load
h_{dx}, h_{dy}	The demanded positions of the hoist in x, y direction
$v\alpha_A$	Jib motor velocity calculated by Arduino
v_{rhA}	Hoist motor velocity calculated by Arduino
α	The angle of the jib rotates perpendicularly on the mast
α_d	The demanded angle of the jib
rh	The distance of the hoist moving along the jib
v_h	The velocity of the hoist along the jib
r_{hd}	the demanded distance of the hoist along the jib

v_{hd}	the demanded velocity of the hoist
v_{ad}	the demanded angular velocity of the jib
u_{α}	The input of the jib motor
u_h	The input of the hoist motor
g	gravitational acceleration (981cm/s^2)
c	The length of the rope
dt	Change of time=0.005s

Firstly, as the whole system is driven by the view from the crane, the crane hoist will be in the centre of the graphical user interface, and the camera view will rotate as the crane moves. So, the transformations of the mathematical model have been sorted out to describe the relationship between the crane and world points. Global axis remains the same for the whole structural system. It is an axis whose X, Y and Z directions are the same for the whole structural system. world axes are convenient for specifying the location of each node, the orientation of each element, the boundary conditions and the loads for the entire structural system. The solution of the structural system is also represented in the world coordinate system. The need for a separate set of axis arises because it is difficult to write the equilibrium equation for all the structural elements in a global axis.

Firstly, we have one global frame of reference. The global origin is called world origin. There are local coordinate systems which called crane coordinates, Thus, the function to crane set as the equations below:

$$s = \sin \alpha$$

$$c = \cos \alpha$$

$$x_{crane} = x_{world} * c - y_{world} * s \quad (3.44)$$

$$y_{crane} = x_{world} * s - y_{world} * c - rh \quad (3.45)$$

Where s and c is a shortcut of sine and cosine respectively, x_{crane} and y_{crane} are the coordinates of the crane on the $x y$ axes, x_{world} and y_{world} are the coordinate of the world and rh is the distance of the hoist along the jib. However, the function to the world will be as the equations below:

$$x_{world} = x_{crane} * c - y_{world} * s \quad (3.46)$$

$$y_{world} = -(x_{crane} + rh) * s + y_{crane} * c \quad (3.47)$$

The simulation has been rescaled to represent the model. The distance units are centimetres while the angle is measured using radian. To calculate the counts or steps of the encoder's motor. The equation below is used to obtain the steps of the encoder of each motor:

$$\frac{gearratio * CPR}{(2\pi)} \quad (3.48)$$

Where CPR stands for counts per revolution of the motor's encoder.

Firstly, for the jib motor, the steps per radian will be obtained as below:

$$Jib\ motor(ka) = \frac{(270 * 64)}{(2\pi)} = 2751\ counts\ per\ radian$$

Secondly, for the hoist motor, the steps per centimetre will be obtained as below:

$$hoist\ motor = \frac{(gear\ ratio * CPR)}{(2\pi)} / (motor's\ pulley\ radius)$$

$$hoist\ motor(kr) = \frac{(30 * 64)}{(2\pi)} / 0.5 = 611\ counts\ per\ centemetre$$

To calculate the real movement with the camera, we need to know the camera properties. The real measurement of the camera with 640*480-pixel window, the used camera has a span of 22° and coverage of 9*8 cm. The coverage area of the camera at a certain distance is obtained as in the following equation below:

$$kC_{(radian/pixel)} = \frac{\text{camera length}}{\text{cable length} * \text{length in pixel}} \quad (3.49)$$

Using the equation above, the variable kc will be:

$$kC_{(radian/pixel)} = \frac{9}{22 * 640} = 0.0064$$

$$kc = 1564 \text{ pixel per radian}$$

From the image, the coordinates of the load relative to the hoist have been measured, which are l_{x_c} and l_{y_c} . Now we have to calculate the actual coordinates of load and convert to centimetre, by multiplying the coordinates of load seen in camera, kc and cable length as the formula below:

$$l_x = \frac{l_{x_c} * c}{kc} \quad (3.50)$$

$$l_y = \frac{l_{y_c} * c}{kc} \quad (3.51)$$

Suppose the camera frame arrives at the 30 millisecond interval, so the frame interval $dframe$ at dt of 5 millisecond will be calculated as the following:

$$dframe = 6 * dt \quad (3.52)$$

To estimate the velocity of the load seen in the camera, simply divide the difference in position per the frame interval as the formula below:

$$v_{lxc} = \frac{l_{x_c} - ol_{x_c}}{dframe} \quad (3.53)$$

Where v_{lxc} is the velocity of load seen in camera in the x -direction. l_{xc} is the position of load seen in camera in the x -direction, ol_{xc} is the previous load position in the x -direction in the prior frame and $dframe$ is the frame interval. Moreover, the y -axis will use a similar formula as following:

$$v_{lyc} = \frac{l_{yc} - ol_{yc}}{dframe} \quad (3.54)$$

Where v_{lyc} is the velocity of load seen in camera in the y -direction. l_{yc} The position of load seen in camera in the y -direction, ol_{yc} is the previous load position in the y -direction in the prior frame and $dframe$ is the frame interval.

As the essence of the modelling of this system based on the experiment view, where variables prefixed with the letter A are coming from Arduino. However, the variables prefixed with the letter P are as perceived by Processing. Now to calculate the angle of the jib, which is measured by radian, we need to divide the measured rotation of the jib motor over the steps per radian factor ka as the following formula:

$$\alpha_p = \frac{\alpha_A}{ka} \quad (3.55)$$

Where α_p the angle is perceived by processing, α_A is the steps of the jib motor and ka is the factor of steps per radian measured by Arduino. A similar process goes with the distance of the hoist along the jib as the following formula:

$$rh_p = rh = \frac{rh_A}{kr} \quad (3.56)$$

Where rh_p equal rh , which is the distance travelled by the hoist perceived by processing, rh_A is the steps of the hoist motor and kr is the factor of hoist motor steps per radian measured by Arduino.

To measure the jib velocities in x and y direction with related to hoist, the formula below used to measure the velocity of the jib by processing:

$$vjx_p = (\alpha_p - o\alpha_p) * rh_p / dframe \quad (3.57)$$

Where vjx_p is the velocity of the jib in x-direction measured by processing, α_p is the angle of the jib measured in processing, $o\alpha_p$ is the previous measured angle by processing, rh_p is the distance travelled by the hoist perceived by processing and $dframe$ is the frame interval.

Using the following equation will calculate the jib velocity in y-direction:

$$vjy_p = (rh_p - orh_p) / dframe \quad (3.58)$$

Where vjy_p is the velocity of the jib in y-direction measured by processing, orh_p is the previous measured distance of the hoist in processing, rh_p is the distance travelled by the hoist perceived by processing and $dframe$ is the frame interval.

For the observed load position, it needs to be converted to centimetres unit by the following equations:

$$l_{xp} = \frac{(l_{xc} * c)}{kc} \quad (3.59)$$

$$l_{yp} = \frac{(l_{yc} * c)}{kc} \quad (3.60)$$

Where l_{xp} and l_{yp} is the load position in x, y direction measured by processing, l_{xc} and l_{yc} are the position of load seen in camera in x, y direction respectively and kc which is the factor of the conversion captured the picture of the load by the camera to reality measured by pixel per radian.

Then the jib velocities have been added to get observed load velocity by using the following equations:

$$v_{lxP} = vj\dot{x}_P + (v_{lxc} * c) / kc \quad (3.61)$$

$$v_{lyP} = vj\dot{y}_P + (v_{lyc} * c) / kc \quad (3.62)$$

Now in the processing section must include an equation to transform the target location. We can calculate the demanded velocity of the load, which represents the error between the target position and the actual load position. So, the demanded velocity of the load in x and y direction will be represented by the equations below:

$$v_{lxdem} = x_{target} - l_{xP} \quad (3.63)$$

$$v_{lydem} = y_{target} - l_{yP} \quad (3.64)$$

Which is limited by

$$\begin{aligned} & f(v_{lxdem} > v_{max}) \{ v_{lxdem} = v_{max} \} \\ & if(v_{lxdem} < -v_{max}) \{ v_{lxdem} = -v_{max} \} \\ & if(v_{lydem} > v_{max}) \{ v_{lydem} = v_{max} \} \\ & if(v_{lydem} < -v_{max}) \{ v_{lydem} = -v_{max} \} \end{aligned}$$

Where v_{max} is the maximum velocity which is limited to 6 cm/sec.

The difference between the demanded velocities of the load in both direction and the actual load velocities which have been perceived by processing will give us the acceleration of the load as described in the equations below:

$$a_{dlx} = (v_{lxdem} - v_{lxP}) * 5 \quad (3.65)$$

$$a_{dly} = (v_{lydem} - v_{lyP}) * 5 \quad (3.66)$$

Where 5 is a feedback gain, the acceleration needs to be limited as well to avoid the overshoot as shown below:

$$\begin{aligned}
& \text{if}(a_{dx} > 10)\{a_{dx} = 10\} \\
& \text{if}(a_{dx} < -10)\{a_{dx} = -10\} \\
& \text{if}(a_{dy} > 10)\{a_{dy} = 10\} \\
& \text{if}(a_{dy} < -10)\{a_{dy} = -10\}
\end{aligned}$$

The demanded hoist position is defined based on the summation of the demanded acceleration of the load multiplying cable length divided by gravity and the actual load with adding the load position:

$$h_{dx} = l_{xP} + (a_{dx} * c / g) \quad (3.67)$$

$$h_{dy} = l_{yP} + (a_{dy} * c / g) \quad (3.68)$$

Now we need to calculate the velocity demand to send to Arduino. The following equation is the velocity demand of the hoist:

$$v_{hd} = h_{dy} * 20 * kr \quad (3.69)$$

Where v_{hd} is the velocity demand of the hoist, h_{dy} is the demand hoist position in the y-direction and kr is the factor of hoist motor steps per radian measured by Arduino. However, by using trigonometry, the following equation is the velocity demand of the jib:

$$v_{ad} = -h_{dx} * 20 / (rh * ka) \quad (3.70)$$

Where v_{ad} is the angular velocity demand of the jib, h_{dx} is the demand hoist position in the x-direction, ka is the factor of jib motor steps per radian measured by Arduino, rh is the travelling distance of the hoist and the gain 20 will give a time constant of 50 milliseconds.

For the Arduino part. It has been considered that the motor control in the Arduino at 5-millisecond intervals. Firstly, we need to calculate the velocities of the motors. So, for the jib motor velocity, we have used the following formula to calculate its velocity as shown below:

$$v\alpha_A = (\alpha - o\alpha) / dt \quad (3.71)$$

Where

$$o\alpha = \alpha * ka \quad (3.72)$$

Where v_{α_A} the velocity of the jib motor, the angle is the jib angle, $o\alpha$ is the previous angle of the jib at the prior sample, ka is the factor of steps per radian and dt is the time interval. To calculate the velocity of the hoist motor, we used the following formula below:

$$v_{rhA} = (rh - orh_A) / dt \quad (3.73)$$

Where:

$$orh_A = rh * kr \quad (3.74)$$

Where v_{rhA} the velocity of the hoist motor, rh is the distance which travelled by hoist, orh_A is the previous distance of the hoist at the prior sample, kr is the factor of steps per centimetre and dt is the time interval.

As it has been explained, tower crane has two inputs which are the angle and hoist. For the angle drive u_{α} , is made based on the difference between the demanded velocity and the actual velocity of the jib. However, we need drives as radians per sec² so that it will be divide by ka . Equation (3.77) describes the input of the jib angle:

$$u_{\alpha} = kj * (v_{\alpha d} - v_{\alpha_A}) / ka \quad (3.75)$$

Where later the gain parameter kj will be made to depend on cable length, which is a value of 10.

In addition, for the hoist drive u_h , is made based on the difference between the demanded velocity of the hoist and the actual hoist velocity along the jib. However, we need to drive as centimetre per sec², so it divided by kr as the following formula below:

$$u_h = kh * (v_{hd} - v_{rhA}) / kr \quad (3.76)$$

Then, the gain parameter kh is deduced after tuning, which is 10.

Since the actual drive of electric motors was limited to the rated design, a limiter was set at the output variable that constraint the maximum drive to within limited range in both directions. A plus sign represents movement in the forward direction, while a minus sign is a movement in reverse.

$$\begin{aligned} & \text{if}(u_\alpha > 5)\{u_\alpha = 5\} \\ & \text{if}(u_\alpha < -5)\{u_\alpha = -5\} \\ & \text{if}(u_h > 10)\{u_h = 10\} \\ & \text{if}(u_h < -10)\{u_h = -10\} \end{aligned}$$

After calculating the motor drive as radians or centimetre per sec², Now with the motor drives, we can work out the dynamics. The rate of change of the angle v_α can be computed based on the state equation below:

$$\frac{dv_\alpha}{dt} = a_y u_\alpha - b_y v_\alpha \quad (3.77)$$

Where a_y is 3 and b_y is 30 which are deduced by tuning the parameters of the equation. So now the v_α equation will be calculated as below:

$$v_\alpha = v_\alpha + (u_\alpha * 3 - 30 * v_\alpha) dt \quad (3.78)$$

This will give a speed lag of 250 milliseconds and a top speed of 0.1 radian per second.

Moreover, the new angle of the jib will be computed instantaneously as below:

$$\alpha = \alpha + v_\alpha dt \quad (3.79)$$

The velocity of the hoist will be updated instantaneously during travelling, and it computed based on the state equation below:

$$\frac{dv_h}{dt} = a_x u_h - b_x v_h \quad (3.80)$$

Where a_x is 30 and b_x is 30, which are deduced by tuning the parameters of the equation. So now the v_h equation will be calculated as below:

$$v_h = v_h + (30 * u_h - 30 * v_h) dt \quad (3.81)$$

Furthermore, the distance of the hoist along the jib will be updated and computed as below:

$$rh = rh + v_h dt \quad (3.82)$$

Now, we need to set up the geometry calculations to rotate view from the crane. Let us assume that the jib rotates in a specific angle α and the hoist moves a certain distance rh so from the view of the crane, the hoist movements will be as the following equations:

$$s = \sin(v_\alpha * dt)$$

$$c = \cos(v_\alpha * dt)$$

$$dr = vh * dt \quad (3.83)$$

These will be the hoist movements. By trigonometry calculation, with sine of the angle approximately equal to 1 the perpendicular distance will be:

$$dx = rh * v_\alpha * dt \quad (3.84)$$

Where dr is the hoist movement along the jib and dx is the perpendicular distance with respect to the view

In addition, the velocity of the travelling load depends on the velocity of the moving hoist along the crane's jib, then, let us assume that the load's velocity coordinate as in x direction v_{lx} and y -direction v_{ly} . So the derived equation for the load velocity in x and y direction as follows:

$$v_{lx} = v_{lx} + g * \frac{l_x}{c} * dt \quad (3.85)$$

Moreover, the velocity in the y -direction is:

$$v_{ly} = v_{ly} + g * \frac{l_y}{c} * dt \quad (3.86)$$

Where l_x and l_y is the position of the load in x and y axes, C is the rope's length and g the gravity force, which is 981 cm/s^2 . v_{lx} , v_{ly} are the velocities of the load in x and y respectively with adding the updating velocity of the load at each instant.

Furthermore, the new positions of the load in each instant computed by updating the load position in both directions during the movement as described below:

$$l_x = l_x + v_{lx} * dt \quad (3.87)$$

$$l_y = l_y + v_{ly} * dt \quad (3.88)$$

Finally, to move the target position with the yard, the following equations derived from moving the background of the graphical user interface to mimic the reality of the crane operation:

$$temp = x_{target} * c + (y_{target} + rh) * s \quad (3.89)$$

$$y_{target} = -x_{target} * s + (y_{target} + rh) * c - rh - dr \quad (3.90)$$

$$x_{target} = temp$$

Where $temp$ a temporary target position x_{target} y_{target} are the target position of the load in x, y coordinates.

Moreover, the load needs to be rotated based on Coriolis which the ordinary Newtonian laws of motion of bodies are to be used in a rotating frame of reference, an inertial force acting to the right of the direction of body motion for counter clockwise rotation of the reference frame or to the left for clockwise rotation. So, the velocities of the load in both directions will act as the following formula:

$$tempv = v_{lx} * c + v_{ly} * s \quad (3.91)$$

$$v_{ly} = -v_{lx} * s + v_{ly} * c \quad (3.92)$$

$$v_{lx} = temp$$

Where $tempv$ a temporary load velocity.

However, the load-displacement need to be rotated as well as below:

$$templ = l_x * c + (l_y + rh) * s \quad (3.93)$$

$$l_y = -l_x * s + (l_y + rh) * c - rh - dr \quad (3.94)$$

$$l_x = temp$$

Where $templ$ a temporary load position.

3.5 The Simplicity of Pragmatic Technique

A simple mathematical model was derived for a gantry and tower crane systems based on state-space representation. A crane model was derived as a set of inputs, outputs and state of variables that represent the system. This method provides an appropriate and concise way to model and analyse systems with multiple inputs and outputs. The models have been designed from the physical relationship between positions, velocities and accelerations.

The proposed method is claimed to be simple because it can control the crane system, which is equally as good, but that is based on a more intuitive level. As mentioned in previous chapters, most of the previous works on the crane system were made based on Lagrange Method which needs massive computations.

The essence of the pragmatic law is designed as a set of ‘nested loops’, where the innermost loop takes the form of a velocity loop wrapped around a motor to give smooth velocity control. Target values are derived from states in the outer loops, to be applied to a succession of inner loops as described below:

- From the difference between the load position and its target, a velocity demand is calculated, proportional to the distance. This is limited in magnitude to a safe travel velocity.

- From the difference between the velocity demand and the actual load velocity a demanded load acceleration is calculated. This is also limited in magnitude representing a limit on the demanded cable angles.
- To achieve this demanded acceleration, an offset is demanded between the lateral hoist position and that of the load, finally giving signals that can be injected into the motor control loops as demand for motor velocity.

As it will be mentioned in the results chapter, the performance of the simulated system will show that the acceleration of the drives increased rapidly to pull up the hanging load and moves steadily and finally, as the target position is approached, the desired velocity falls and increases quickly, at the final stage of stoppage the hoist ‘hangs back’ to decelerate the load and make it stop at the demanded target.

For the tower-crane simulation, the strategy was made easier to implement by working in coordinates that rotated with the jib angle and were centered on the hoist. As the jib rotated the target coordinates were rotated and moved in the opposite sense. In this way, all calculations were aligned with the coordinates of the downward-looking camera.

3.6 Conclusion

In conclusion, a simple mathematical model was derived for the tower crane system based on the real system by using a virtual camera and the whole system is driven by the view from the crane. In addition, practical controllers for position and anti-swing of the tower crane were developed. This model is simulated with the aid of JavaScript language.

4 CHAPTER FOUR: - PRAGMATIC CONTROL TECHNIQUE- SIMULATION STUDIES

4.1 Introduction

This chapter investigates the use of the pragmatic control technique in crane systems which includes the gantry crane and tower crane as examples. The controller is designed for swing control and hoist positioning control. This chapter will include simulations of a model of a gantry crane and two models of the tower crane. All those models were derived using state-space representation. The simulation of the crane models is carried out to verify the performance of the proposed technique. MATLAB software and JavaScript programming language were used to perform the system model and controller implementation. In addition, the simulation results will be demonstrated throughout the following sections.

4.2 Simulation of Gantry Crane Model

The model for the gantry crane consists of a trolley moves on a horizontal girder. The trolley consists of a hoist system (rope and hook) to lift and lower the load. In this section, the proposed pragmatic control technique is implemented and tested with the aid of JavaScript and MATLAB. The simulation of the gantry crane is carried out to verify the performance of the proposed technique. The parameters that have been used to simulate the gantry crane model as well the simulation is based on a state matrix which has been shown in the previous chapter. The simulation of the system dynamics is executed with control inputs of u_x and u_y equations. Several simulation studies have been performed to study the behaviours of the proposed technique. The hoist (trolley) runs across the horizontal beam, and we can consider it is lying on the y-direction, with the rails conveying it in the x direction. Figure 4-1 shows a top view of the system. The brown line represents the crane's jib (girder) while the blue circle represents the hoist of the crane. In addition, the red circle, which is underneath the hoist is the representation of the load. The whole system is laying on a space of 40 *40 metre.

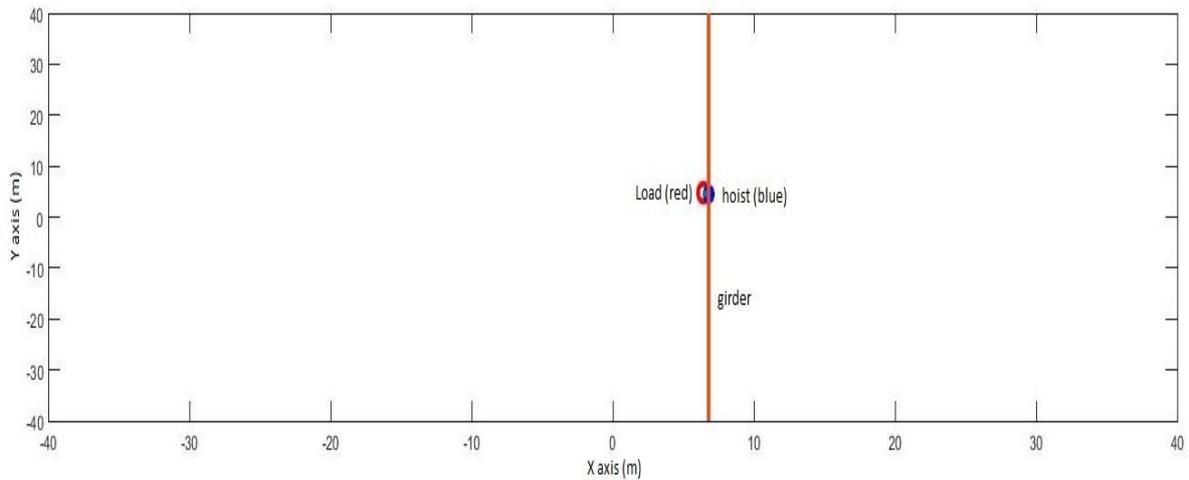
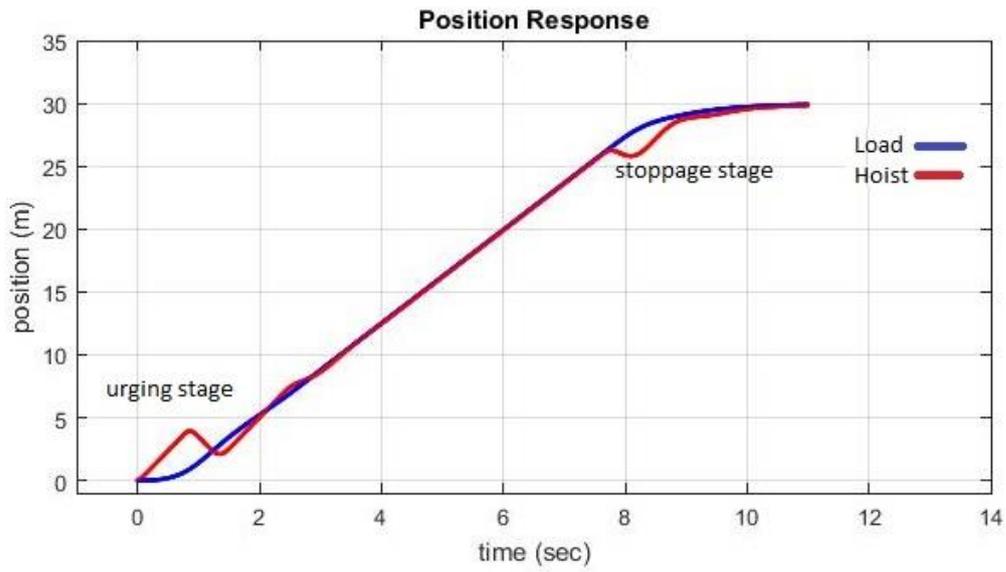


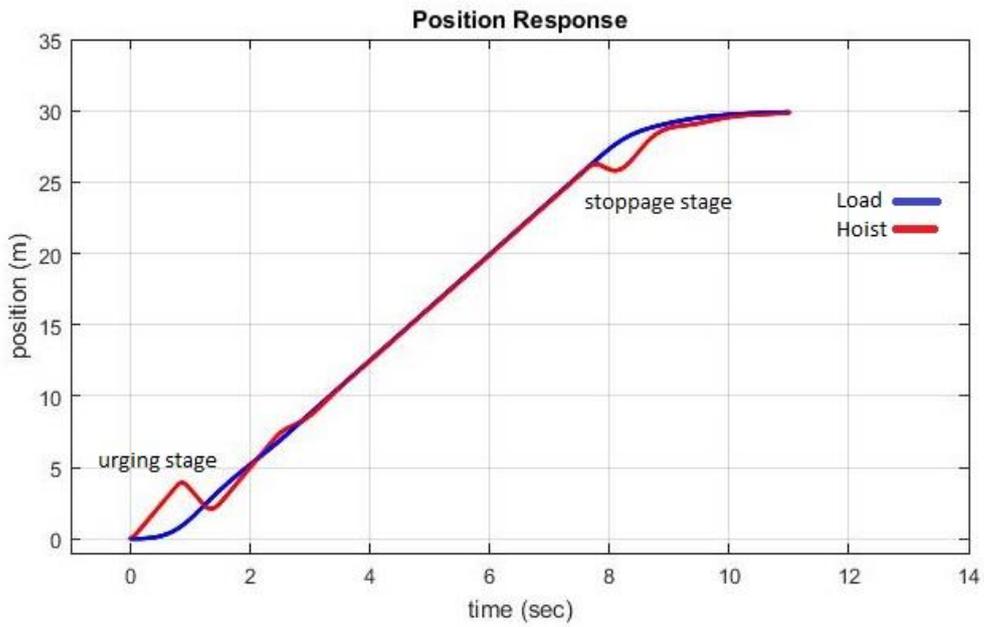
Figure 4-1 Top view of the gantry crane system in MATLAB

The pragmatic controller developed in the previous section has been evaluated using MATLAB. Sampling time of the controller $dt = 0.01$ sec is being used. The controller was tested with a fixed cable length of five and ten meters separately. With the used sampling time, it was possible to perform a real-time simulation of the complete control system on a standard computer.

Figure 4-2 shows the position response of load and hoist during travelling with a 5-metre cable length with different scenarios. The first scenario in Figure 4-2 a show the position response of one axis which being executed in x -axis only of 30m distance. However, the second scenario was moving the load in both axes of 30 m in x and 20 m in y -axes. As the load is accelerated, the hoist moves ahead to give a horizontal component of the cable tension. When the load has reached the desired velocity, the hoist is moving immediately above the load, only deviating to compensate for disturbances such as wind gusts. As the target position is approached, the desired velocity falls, so the hoist ‘hangs back’ to decelerate the load and finally brings it to a halt; where once again it takes up a position that is immediately overhead.



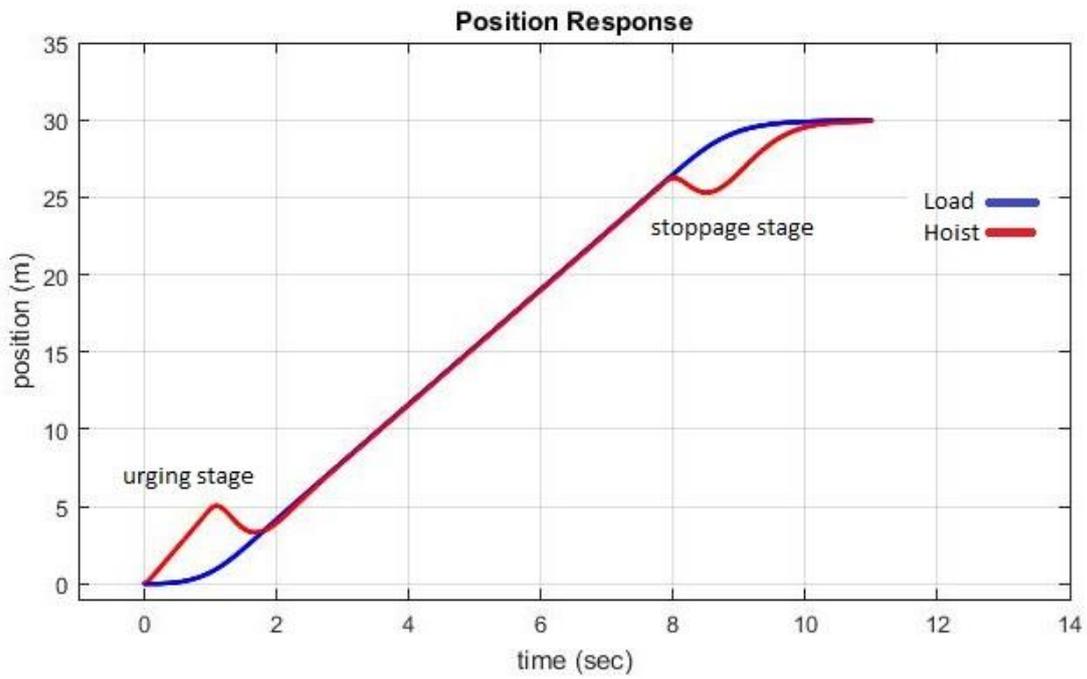
(a)



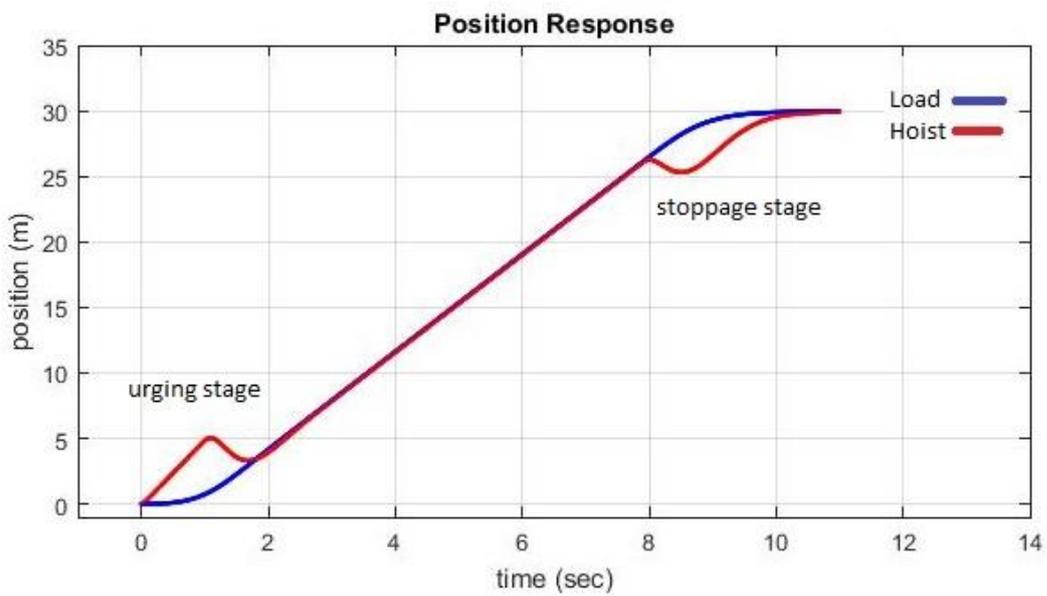
(b)

Figure 4-2 Position response of 5-meter cable length (a) movement in axis only (b) movement of both $x - y$ axis

Further tests have been undertaken to validate the reliability of the proposed technique. A ten-metre cable length was chosen to test the controller reliability, as shown in Figure 4-3.



(a)



(b)

Figure 4-3 Position response of 10-meter cable length (a) movement in x -axis only (b) movement of both $x - y$ axis

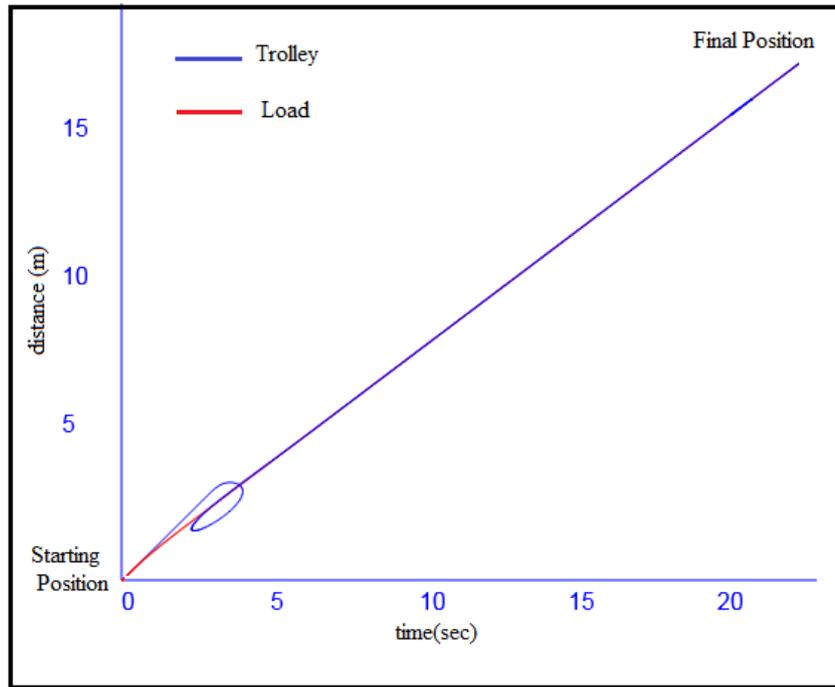
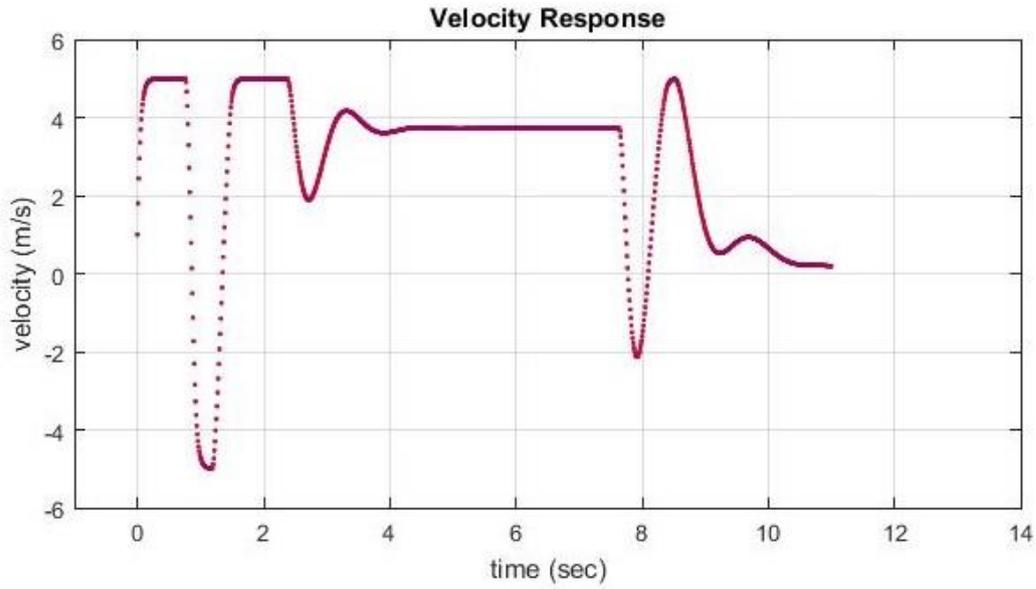


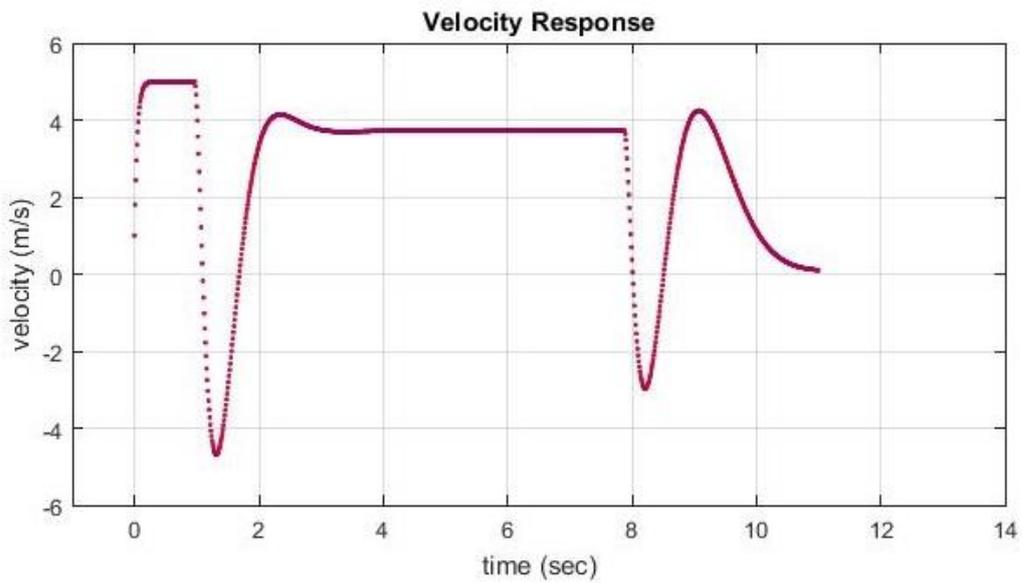
Figure 4-4 Crane movement with 45° slew motion by using JavaScript

Figure 4-4 shows a 2D motion of the crane system using JavaScript. Commands were generated to move the hoist from its position to the demanded position through 45° of slew from the original position of the crane. At the starting position, acceleration causes the trolley to move to the demanded position which results in an elliptic or circular shaped movement to cope with an undesired swing of the load, causing the loop to become excited. Usually, the crane's acceleration generates a conical swing of the load. However, the controllers tried to correct the position of the crane's trolley to reduce the swing that was generated due to acceleration.

Figures 4-2 and 4-3 show almost the same results except for a slightly larger deviation of the trolley when the cable length is longer, with the load swing being noticeable.



(a)

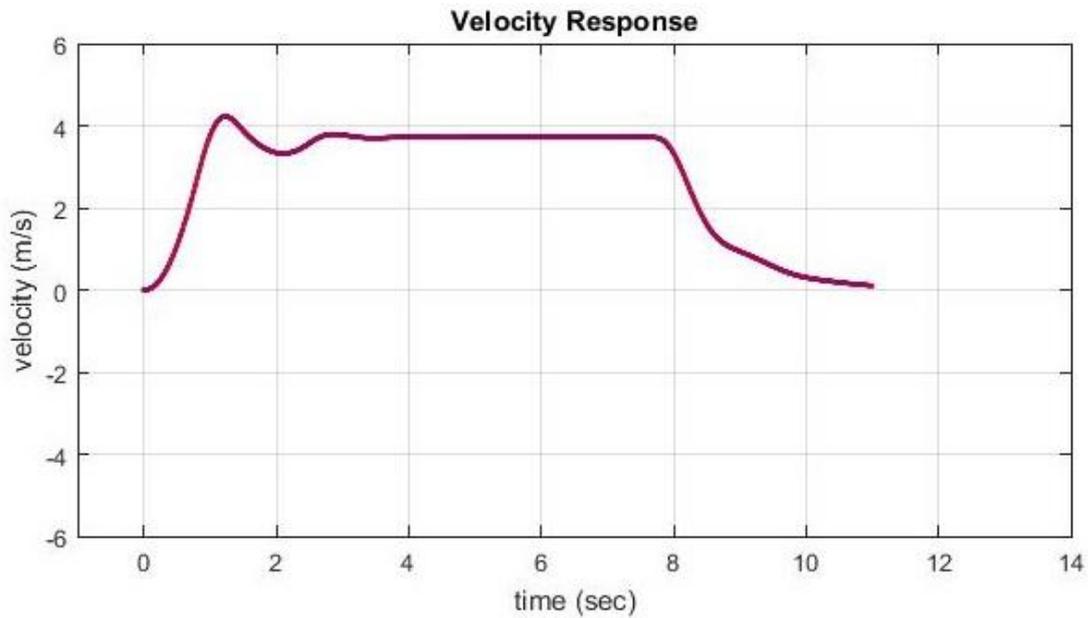


(b)

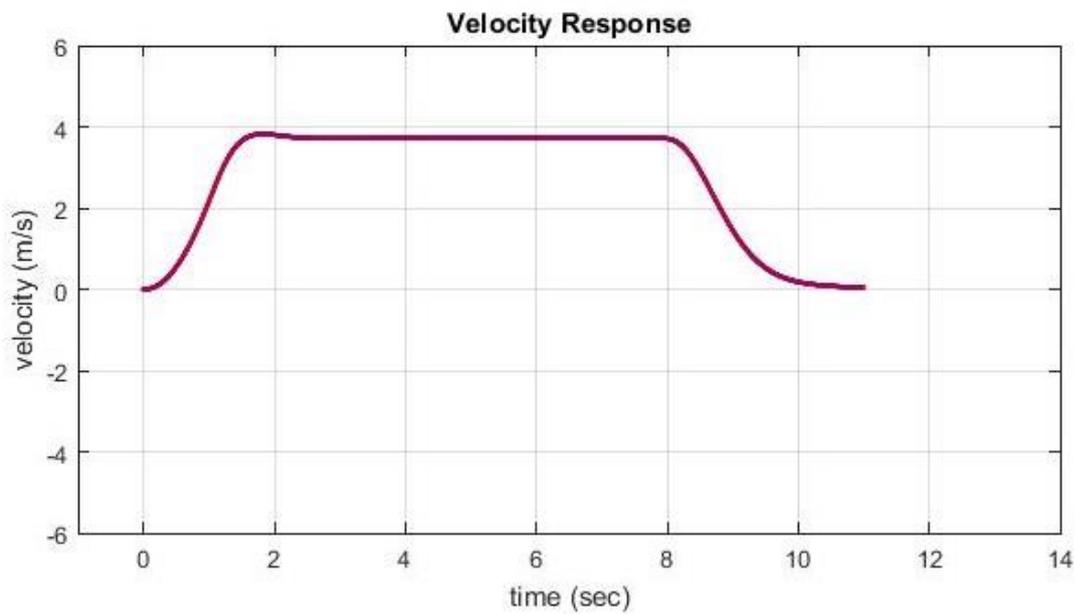
Figure 4-5 Hoist velocity response (a) with 5 m cable length (b) with 10 m cable length

Figure (4-5 a) apparently indicates the hoist velocity profile. For the cable of 5 m length, the velocity goes up to 5 m/s to pull up the load while accelerating. Then went to -5 m/s rapidly to make the hoist over the load again then move the hoist with a velocity of 4 m/s until the load reaches the demanded position, so the hoist decelerates back again to stop in the final position

with no swing. However, with less disruption than the 5 m cable length, figure (4-5 b) shows the velocity profile of the hoist with 10 m cable length.



(a)



(b)

Figure 4-6 Load velocity response (a) with 5 m cable length (b) with 10 m cable length

Figure 4-6 a indicates the load velocity profile. For the cable of 5 m length the velocity goes up to over 4 m/s then decreased to around 3.5 m/s then goes up again around 4 m/s until

reaching the final position. While with 10 m cable length the velocity goes up to 4 m/s smoothly then decreased to zero gradually when reached the demanded position.

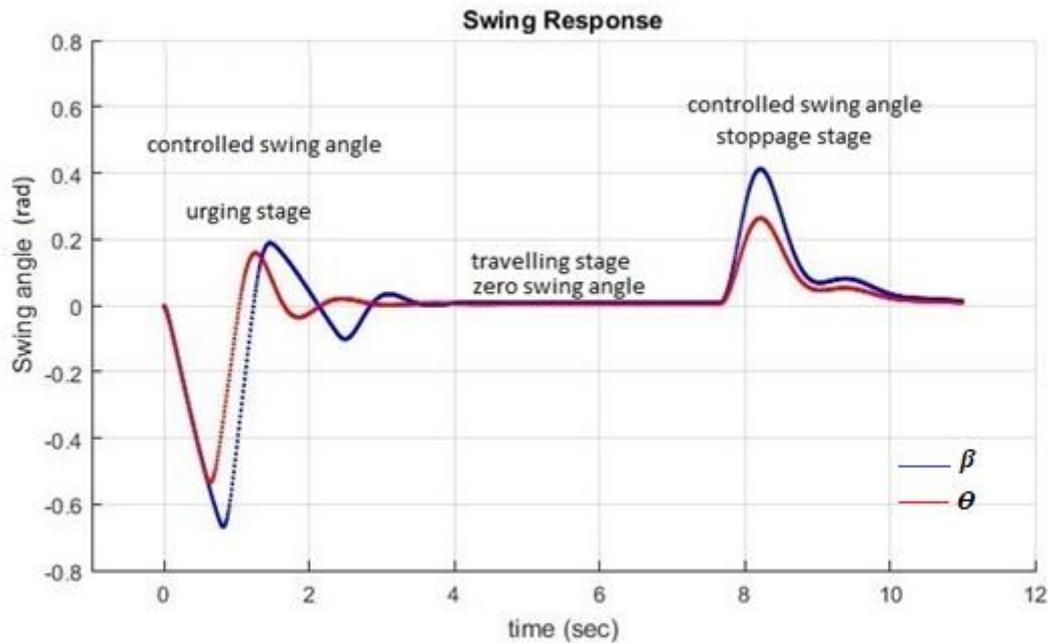


Figure 4-7 Schematic of swing angle with 5-meter cable length

Figure 4-7 shows the swing angle between the hoist and the load with 5 m cable length. Two axes movement were ordered in both x - y plane. In the acceleration stage (urging stage) when the hoist tries to pull up the load, the angle is around -0.6 radian then changed to around 0.2 radian to compensate the position of the hoist being over the load. No swing during the movement until reach the demanded position there is a swing of 0.4 radian due to the hoist deceleration and goes to zero in the final position. Figure 4.8 shows the swing angle between the hoist and the load with 10 m cable length. Two axes movement were ordered in both x - y plane. In the acceleration stage (urging stage) when the hoist tries to pull up the load, the angle is around -0.4 radian to compensate the position of the hoist being over the load. No swing during the movement until reach the demanded position there is a swing of 0.3 and 0.2 radian due to the hoist deceleration and goes to zero in the final position.

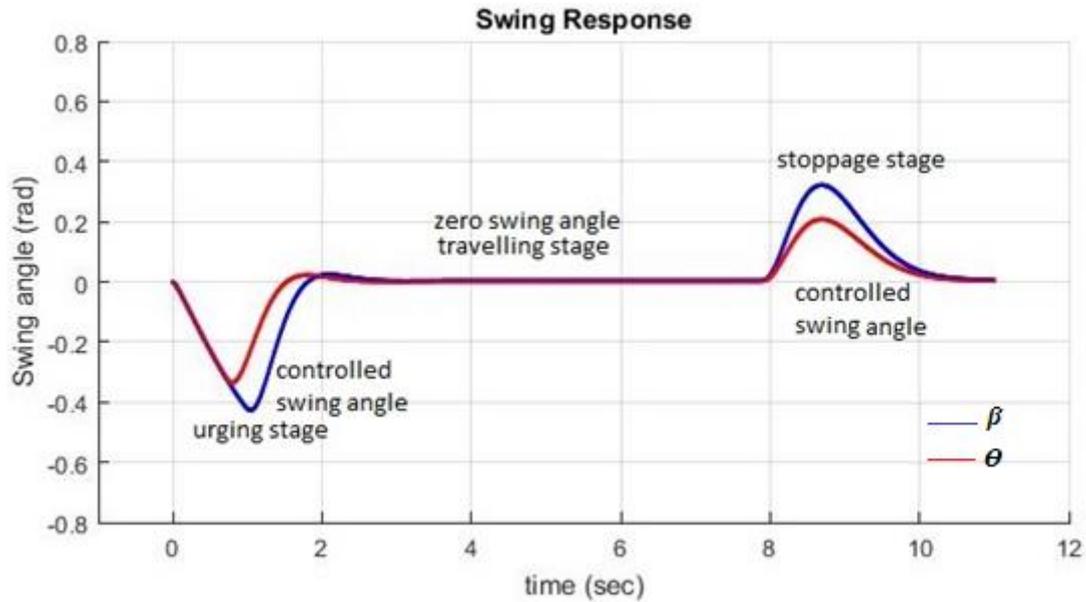


Figure 4-8 Schematic of swing angle with 10 m cable length

To conclude, these results showed the simplicity of the proposed strategy compared with the complexity of the published alternative. Thus, the results show the robustness of the proposed strategy with no load swing during the movement and no sensitivity with longer cable length or external disturbance.

4.3 Simulation of Tower Crane based on State Space Representation

In this section, the proposed pragmatic control technique is developed in the previous section has been evaluated with a test case simulation. Simulation of tower crane has been designed to verify the performance of the proposed technique. The simulation was developed based on the state space modelling of the system. The simulation of the system dynamics is executed with a control input of u_h and u_a equations. Several simulation studies have been performed to study the behaviours of the proposed technique. The tower crane model representation was performed

using MATLAB software. Figure 4-9 shows a top view of the system. The blue line represents the crane's jib while the blue circle represents the hoist of the crane. In addition, the red circle, which is underneath the hoist is the representation of the load. The whole system is laying on a space of 40 *40 metre.

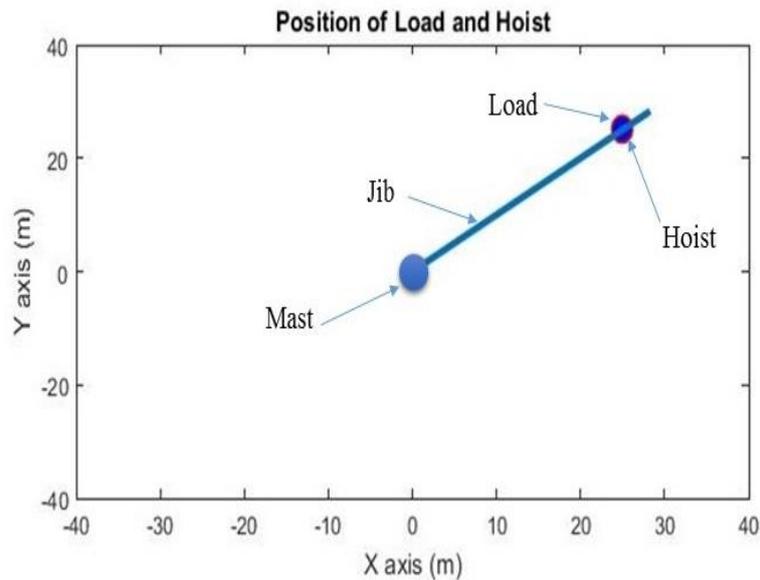


Figure 4-9 Top view of the tower crane system in MATLAB

The pragmatic controller developed in the previous section has been evaluated using MATLAB. Sampling time of the controller $dt = 0.01$ sec is being used. The controller was tested with a fixed cable length of five and ten meters separately. With the used sampling time, it was possible to perform a real-time simulation of the complete control system on a standard computer. Commands were generated to move the hoist from its position to the demanded position of 20 m and rotating of 1 radian from the original position of the crane.

The results of jib and hoist velocity profiles are shown in figure 4-10 and 4-11. However, Figure 4-12 shows the load velocity response while Figure 4-13 shows the behaviour of load and hoist during a travelling to the demanded position with 5-metre cable length. Moreover, Figure 4-14 shows the swing angle of the system with a 5-meter cable length. In addition, Figure 4-15 shows the behaviour of load and hoist during a travelling to the demanded position with 10-metre cable length

Figures 4-10 and 4-11 show the velocity response of the jib and the hoist respectively, as we can see that the acceleration of the drives increased rapidly to pull up the hanging load. In

figure 10, the jib velocity drive reaches around 2 radians per second at the first second then goes down to -0.8 radian per second then changing rapidly up and down to make sure that the load remains under the hoist. When the load has reached the desired velocity, the hoist can travel immediately above the load, only deviating to compensate for disturbances such as wind gusts. As the target position is approached, the desired velocity falls, so the hoist ‘hangs back’ to decelerate the load and finally bring it to a halt when once again it takes up a position immediately overhead. However, figure 4-11 shows that the hoist velocity drive goes up to 3 meters per second at the starting point of movement to pull up the load and then goes down to around zero meters per second which only compensate to make the hoist over the load. Quick changes then occurred to keep the position of the load under the hoist, which kills any swing of the load and the demanded velocity of the hoist will remain around 2.8 meters per second during the movement. Finally, as the target position is approached, the desired velocity falls and increases quickly, at the final stage of stoppage the hoist ‘hangs back’ to decelerate the load and make it stop at the demanded target.

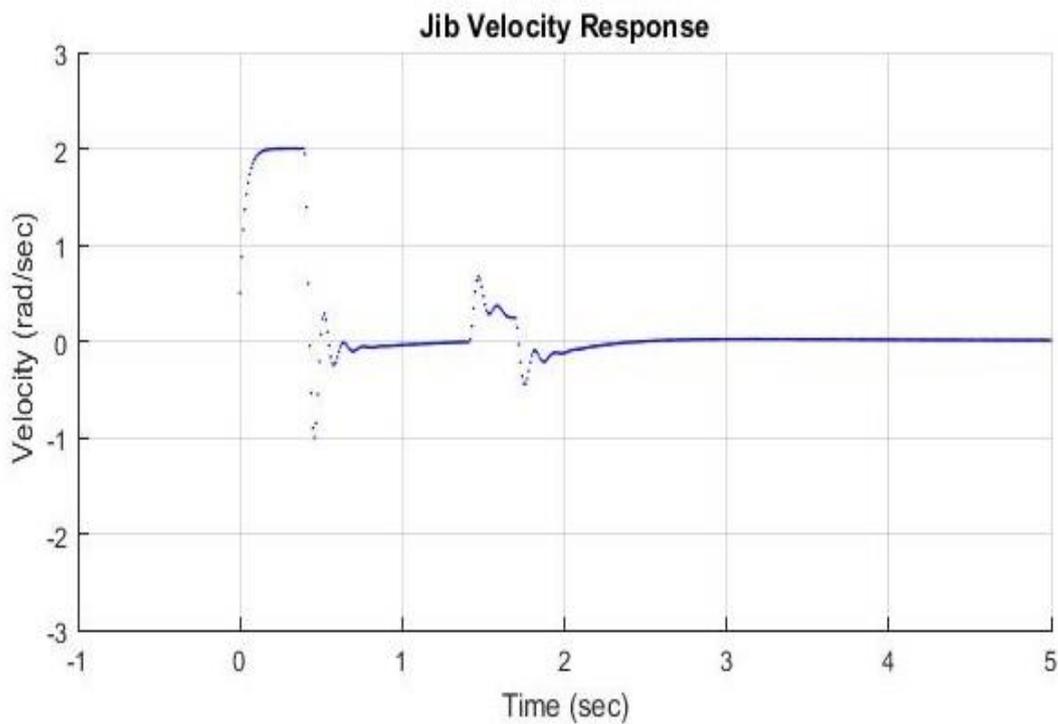


Figure 4-10 Jib velocity response

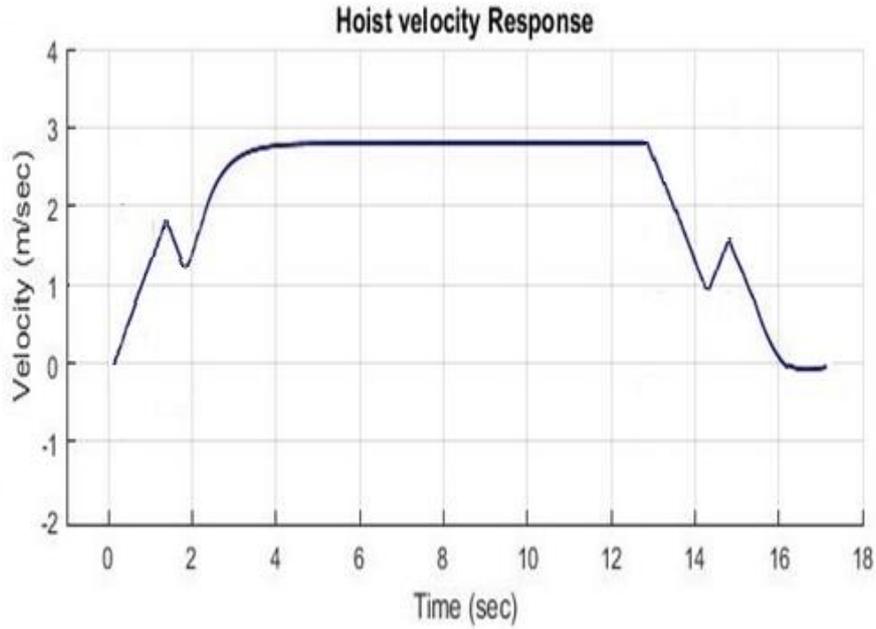


Figure 4-11 Hoist velocity response

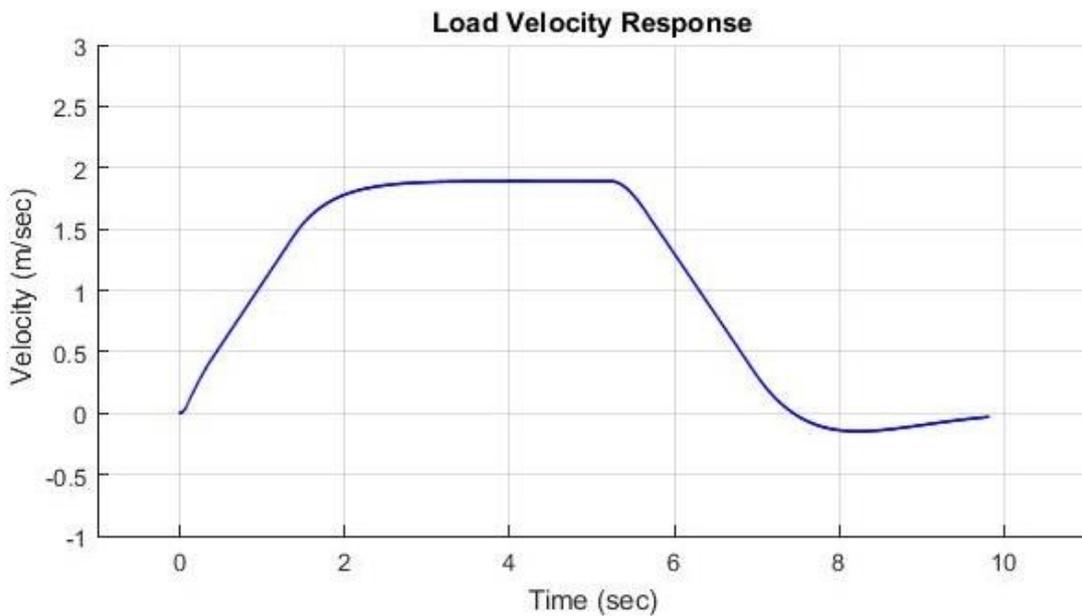


Figure 4-12 Load velocity response

Figure 4-12 shows the load velocity, which increased up to 2 meters per second then decreased gradually until it reaches zero at the final target. While figure 13 displayed the behaviour of load and hoist during travelling with five-meter cable length. The hoist and the load reached the demanded target without overshoot.

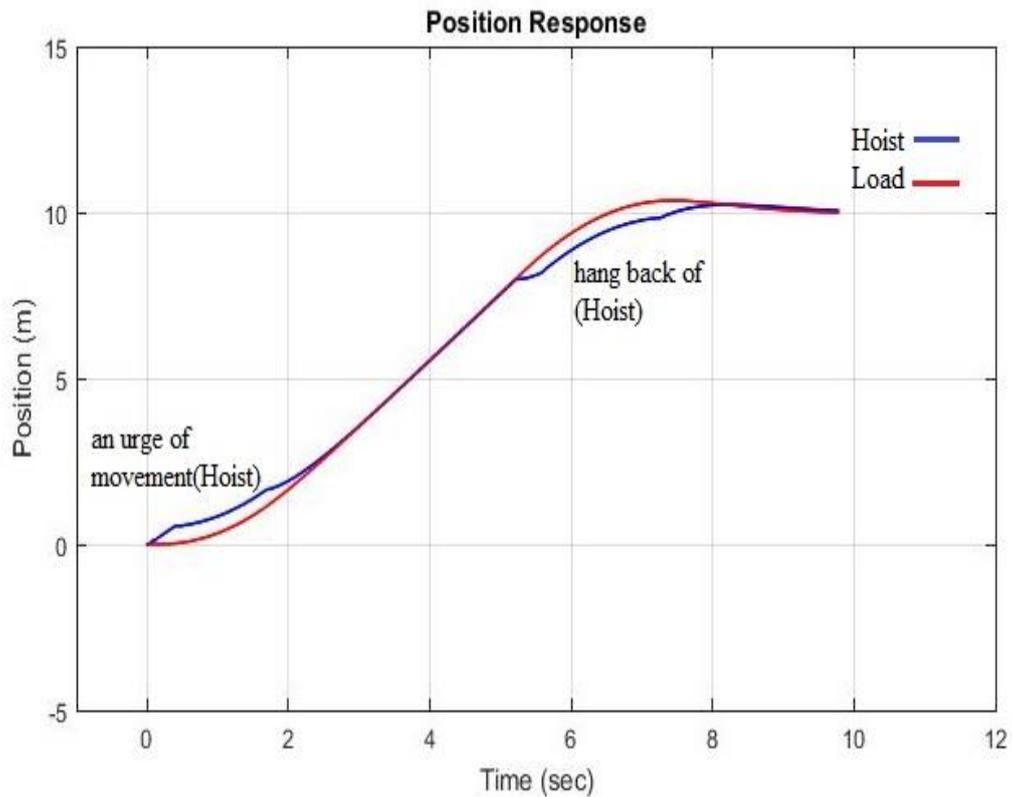


Figure 4-13 Positions of load and hoist during travelling with five-meter cable length

The swing angles of the system, which are Θ and β with five meters cable length have been demonstrated in figure 4-14. Both angles did not exceed 0.1 rad from the starting stage until reaching the final position while remaining at zero with no swing during moving toward the demanded target.

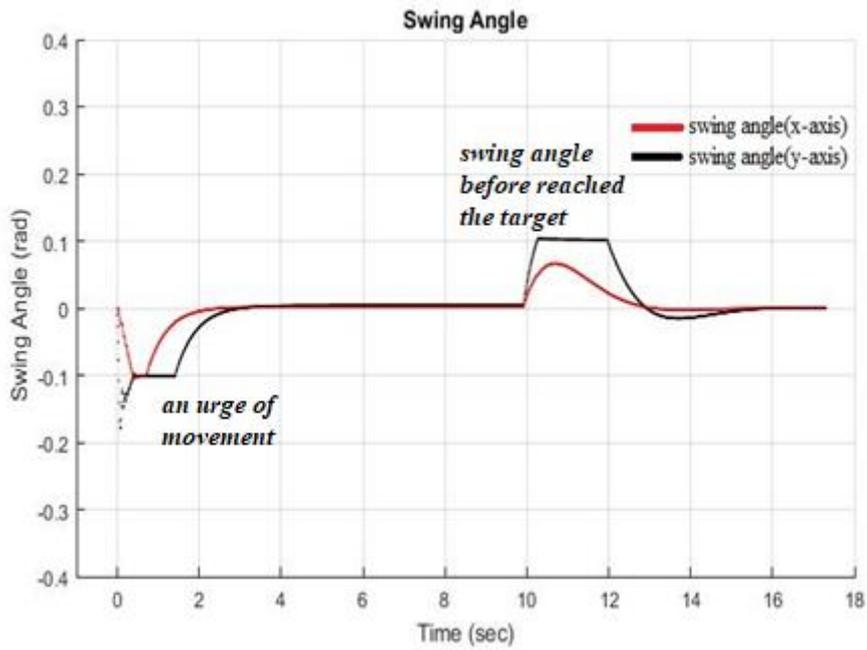


Figure 4-14 Swing angles of the system with five-meter cable length

Further tests have been made to validate the reliability of the proposed technique. Ten-meter cable length has been chosen to test the controller reliability, as shown in figure 4-15 and 4-16.

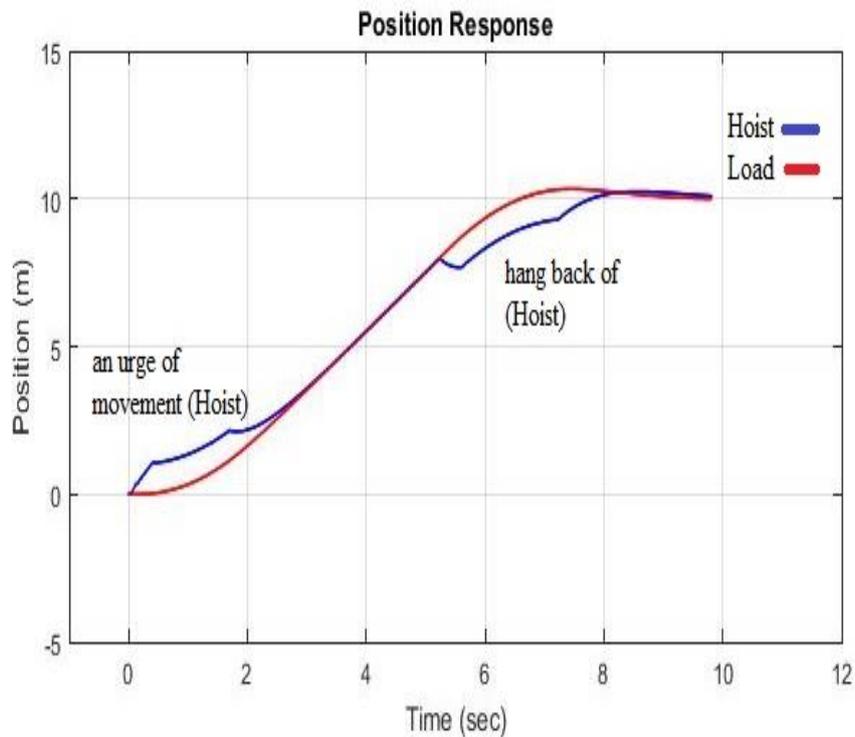


Figure 4-15 Positions of load and hoist during travelling with ten-meter cable length

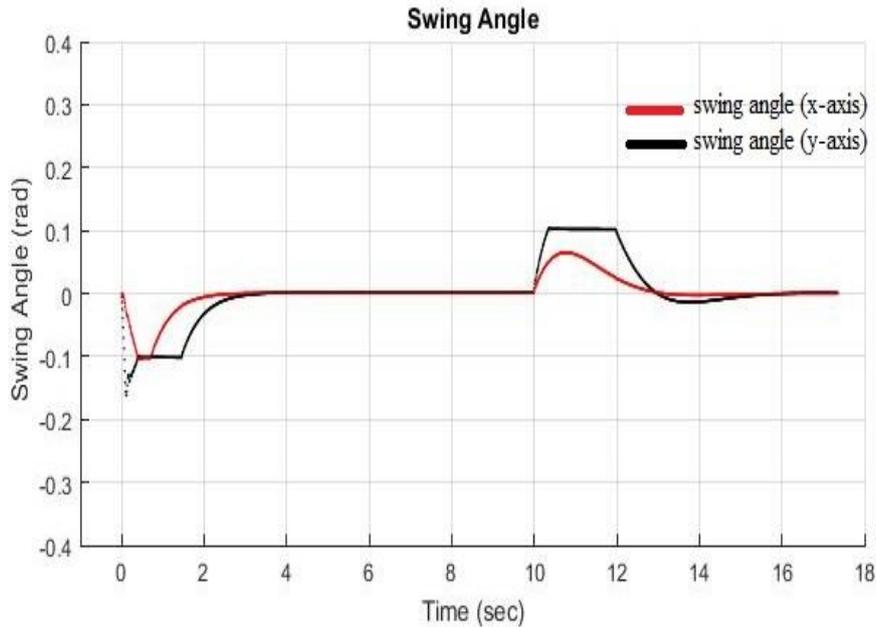


Figure 4-16 Swing angles of the system with ten-meter cable length

It has been noticed from the figure 4-15 and 4-16 that, the performance of the pragmatic controller in the system with ten-metre cable length almost same results except slight overshoot of 0.3 metre which considered too small compared with large workspace. However, the swing angles did not exceed 0.1 radian which means that the controller can successfully reduce the payload swings during and after the crane motion and bring them to near zero when the crane is moving toward the demanded target.

To conclude, the results have shown the simplicity of the proposed strategy comparing with the complexity of the published alternative. Moreover, the results showed the robustness of the proposed strategy with no load swing and it is not sensitive with cable length changing nor with external disturbance.

4.4 Simulation of Tower Crane based on Practical Point of View

In this section, the proposed pragmatic control technique has been evaluated with a test case simulation. The idea of this model is that the whole system is driven by the view from the crane. This model is derived to mimic the real model as a set of state variables have been

defined, choosing names for them that will be compatible with code. The model of the system has been divided into two parts which are the crane axes and world axes. Derive the equation of the target of the load in terms of its coordinates relative to the hoist, in crane axes. So at each step, we have to transform it for the crane movement. Relative to the base of the crane it will rotate this by angle. As the jib rotates, the view rotates in the opposite direction taking the target with it.

Simulation of tower crane has been designed to verify the performance of the proposed technique. The simulation was developed based on the state space modelling of the system, the simulation of the system dynamics is executed with a control input of u_α (angle of the jib) and u_h (hoist movement along the jib) equations. A simulation study has been performed to study the performance of the proposed technique. It is highly desirable to represent the output of the system graphically, preferably in real-time. Therefore, the tower crane model representation was performed using the JavaScript programming language. Figure 4-17 shows a top view of the system.

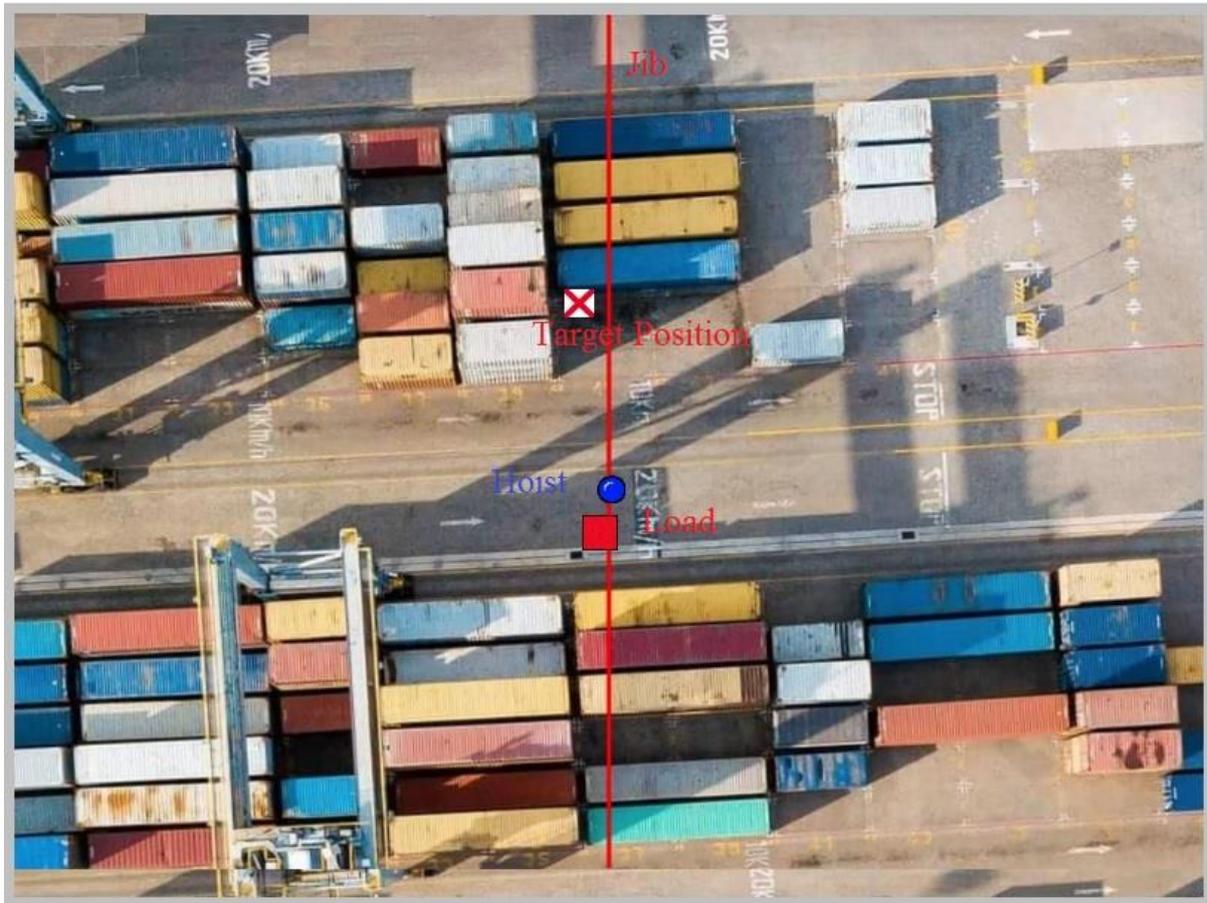


Figure 4-17 Top view of the tower crane system in JavaScript

The red line represents the crane's jib while the blue circle represents the hoist of the crane. In addition, the red square which being underneath the hoist is the representation of the load. The white square with red multiplication sign represents the target position. The pragmatic controller developed in this section has been evaluated using JavaScript. Sampling time of the controller $dt = 0.005$ sec is being used. The controller was tested with a fixed cable length of fifty meters. With the used sampling time, it was possible to perform a real-time simulation of the complete control system on a standard computer.

Figure 4-18 shows that the hoist velocity drive goes up to 10 meters per second at the starting point of movement to pull up the load and then goes down to around -8 meter per second which only compensate to make the hoist over the load. Quick changes then occurred to keep the position of the load under the hoist which kills any swing of the load, and the demanded velocity of the hoist will remain around 5 meters per second during the movement. Finally, as the target position is approached, the desired velocity falls and increases quickly, at the final

stage of stoppage the hoist ‘hangs back’ to decelerate the load and make it stop at the demanded target.

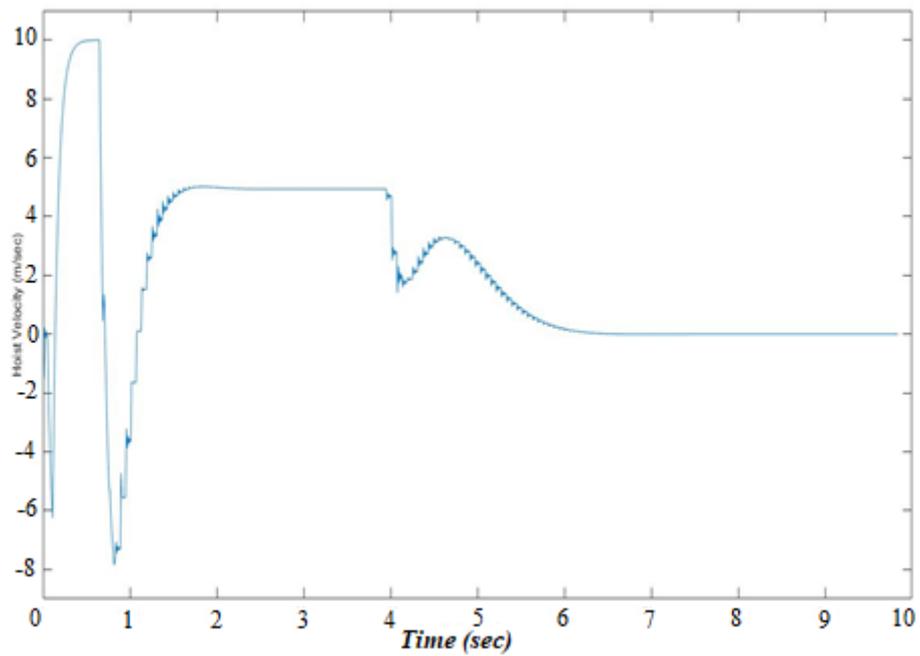


Figure 4-18 Hoist velocity response

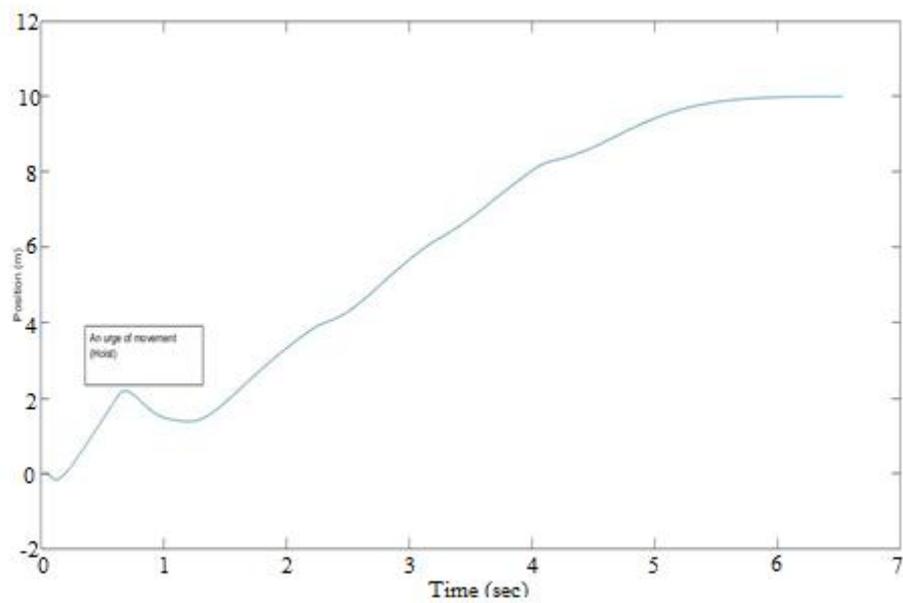


Figure 4-19 Position response of the hoist

As seen in Figure 4-19, which shows the position response of the hoist system during a travelling to the destination position. At the beginning of the movement, an urge of the hoist can be seen which represents the response of the hoist to pull up the load and accelerate the movement while keeping the load under the hoist then move to the demanded position without overshoot.

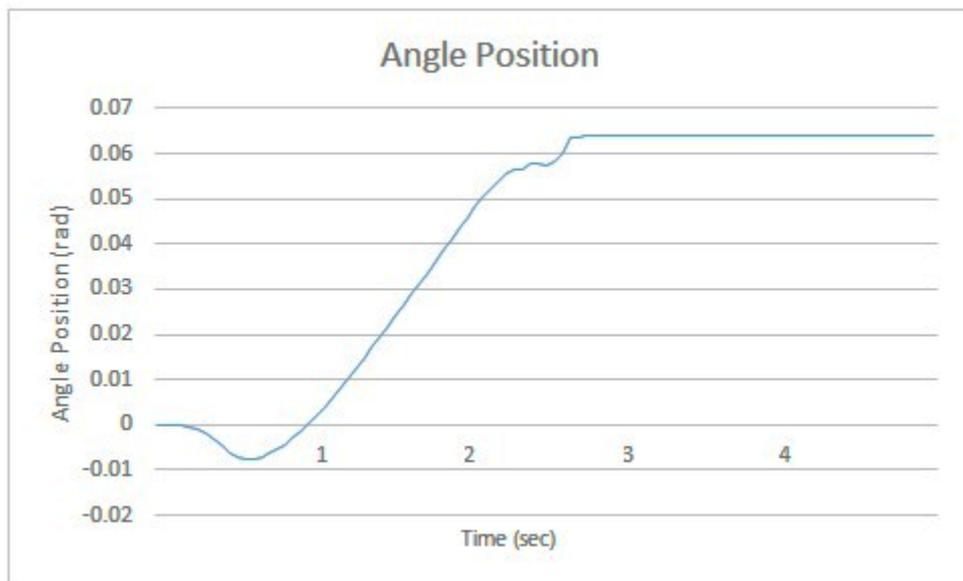


Figure 4-20 Position response of the jib angle

Figure 4-20 shows the position response of the jib angle. The position of the angle goes down to around -0.01 rad to compensate the swing that might occur during the urging stage of movement then goes up gradually to the demanded position which 0.06 rad without overshoot.

Figure 4-21 expresses the load velocity during travelling in both x and y directions. The velocity of the load in the y -direction is going down to -3 m/s as the hoist tries to pull up the load, then goes up to around 5 m/s and decreases when it reached the final position. However, the load velocity in x coordinate goes up to 3 m/s then goes down rapidly to less than zero and goes up to zero.

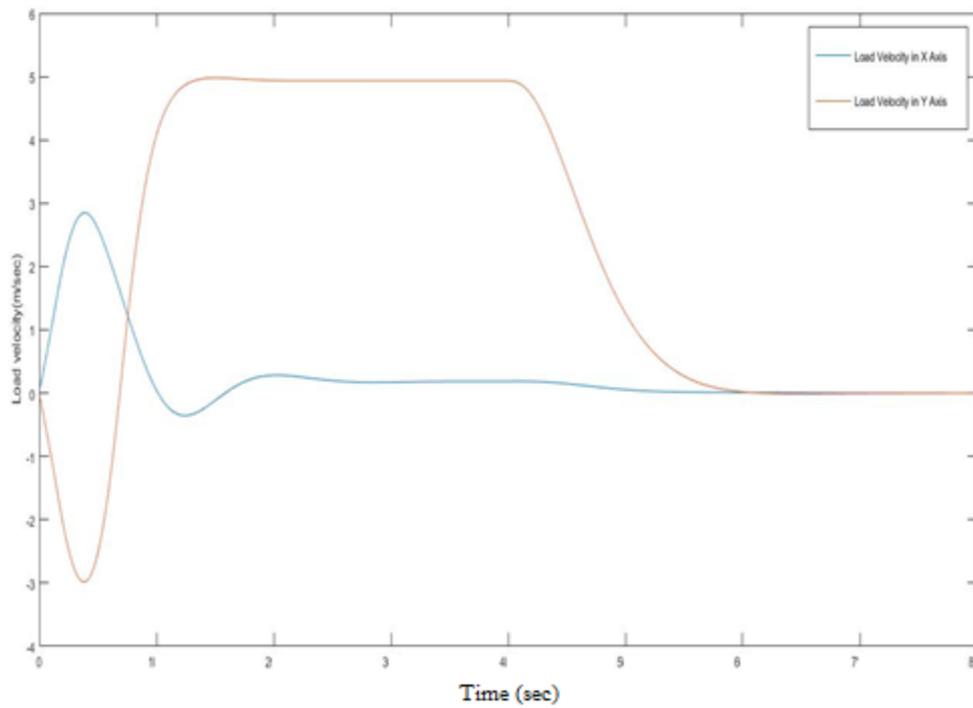


Figure 4-21 Load velocity in x-y coordinates

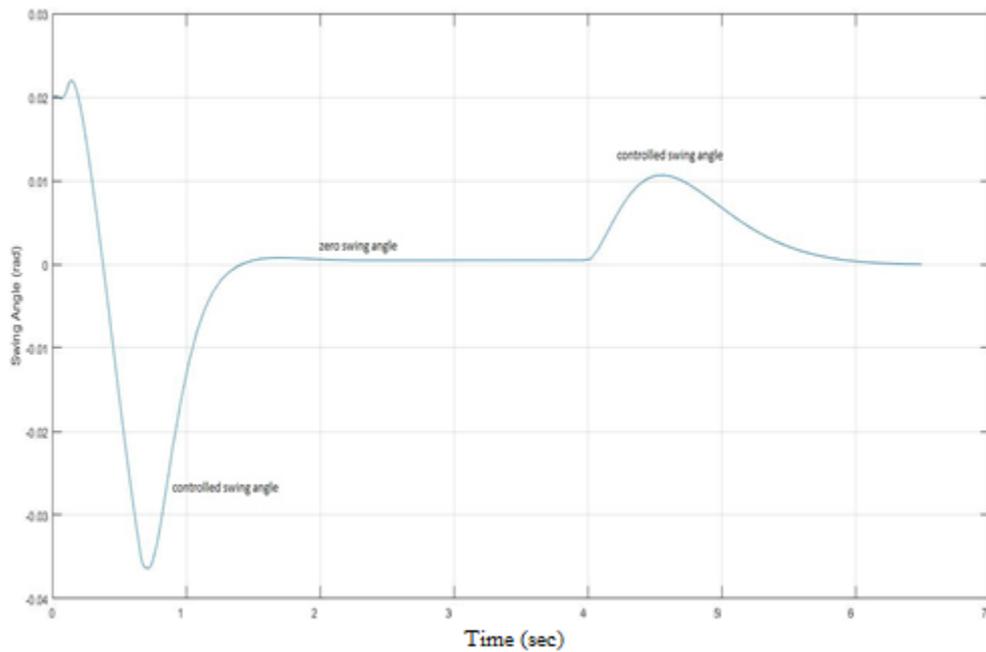


Figure 4-22 Swing angle of the system

The swing angle of the system has been demonstrated in Figure 4-22. The angle is calculated between the hoist and the load the angle is 0.02 radian at starting point of movement while goes

to around 0.035 radian when the hoist tries to pull up the hanging load. The angle then becomes zero while travelling to the final position. When the hoist is reaching the final position, the hoist 'hangs back' to decelerate the load and make it stop at the demanded target which results of 0.01 radian of the hoist while the load remains stable under hoist with no swing.

4.5 Conclusion

In conclusion, it is a novel strategy that the model was derived to mimic the real model. The results have shown the reliability of the proposed strategy. To compare with previous literature which has been described in chapter two, regarding the crane modelling, most researchers have been deriving the model by using the Lagrange approach. Open-loop techniques have been shown to be effective. However, these are extremely sensitive to parameter variations, changing conditions and external disturbances, decreasing the reliability of the system's performance. The previous techniques that have been used to control the crane such as PID, it showed that the PID controllers were developed with the aid of other technique like neural network or use two PID controllers. Also, the fuzzy logic controller considered a complex due to the computational cost like fuzzification, fuzzy operator and defuzzification. In addition, the fuzzy set and the fuzzy rules are difficult to determine.

Furthermore, comparing with published papers in term of performance (which showed the crane's behaviour during the travelling while the stoppage in the final position) that have not been mentioned, we have seen that the stoppage will generate an oscillation of the load. To conclude, these models of cranes show the simplicity of the proposed strategy comparing with the complexity of the published alternative. Thus, the results show the robustness of the proposed strategy with no load swing and it is not sensitive with cable length changing nor with external disturbance.

5 CHAPTER FIVE: - EXPERIMENTAL DESIGN OF GANTRY CRANE SYSTEM

5.1 Mechanical Design Considerations

The gantry crane consists of trolley moving along a girder in translation motion, and the girder moving in a horizontal plane. The trolley consists of a hoist system (rope and hook) to lift and lower the load. The combination of trolley and girder movement allows the load to reach the desired destination. The girder motion is perpendicular to the trolley movement. In this chapter, a one-axis gantry crane is designed and controlled by a pragmatic technique which explained in chapter three. In this section, the design of the mechanical and electrical part of the system will be described.

In the proposed mechanical design, it consists of girder made by aluminium profile, as shown in Figure 5-1. T-slot System 40 aluminium extrusion profile 4040 is used to represent the crane girder

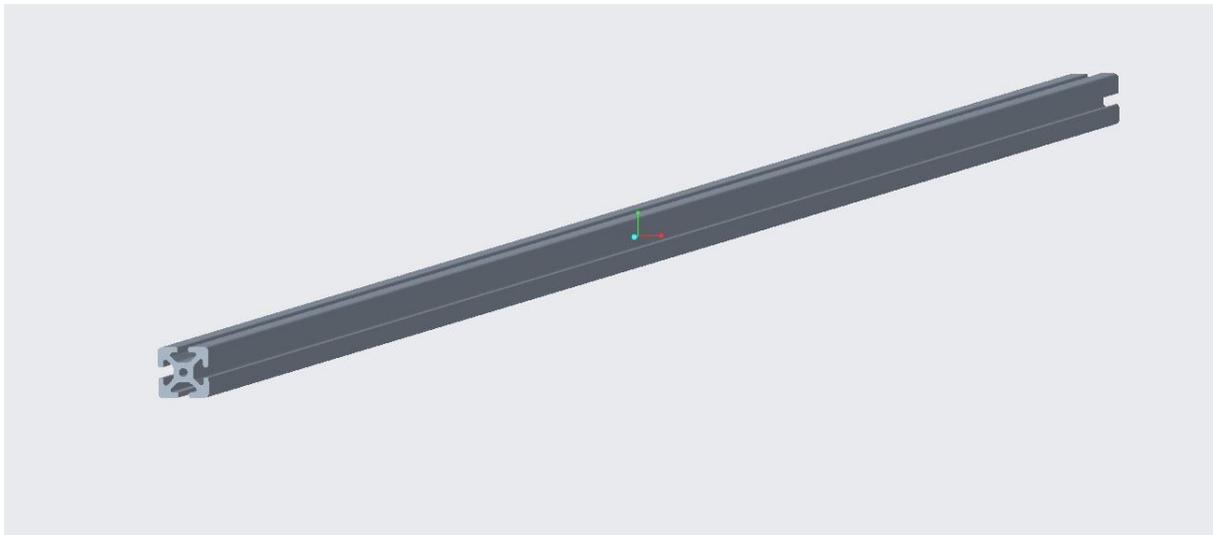


Figure 5-1 T-slot System 40 Aluminum extrusion profile 4040

The motor of the hoist driving and the cable lowering and lifting have been fixed in the rear end of the girder. The girder has a designed hollow bar which allows the belt of the hoist system to move along it.

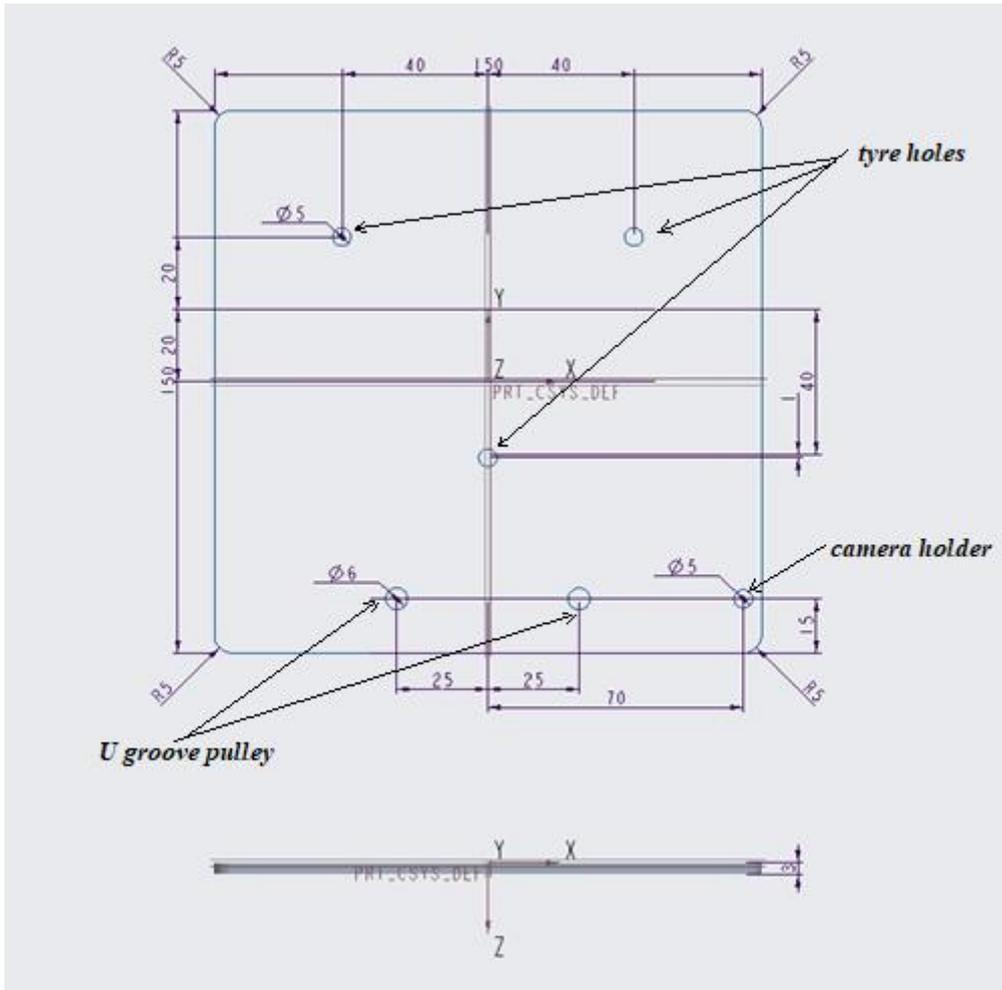


Figure 5-2 Aluminum plate of the hoist

The hoist consists of two aluminium plates which one of them shown in figure (5-2). The dimensions of the plate are 150*150 mm, it has three holes for the tyres, two holes for the U groove pulley and hole for the camera holder. These two plates are fixed by the tyre and U groove chassis which being fixed by screw nut and make it easy for uninstal or maintenance.

Three rubber tyres are used to make the hoist moving along the girder smoothly. Two of the tyres are installed on the top of the girder which moving inside the T slot of the girder, while the other tyre lied underneath the girder inside the T slot as well. The hoist will be moved using a rubber-toothed belt. Besides, two U groove pulleys are installed underneath the hoist system with an idler at the outer end, which allows lowering and lifting the hook as demonstrated in figure 5-3 and 5-4 below:

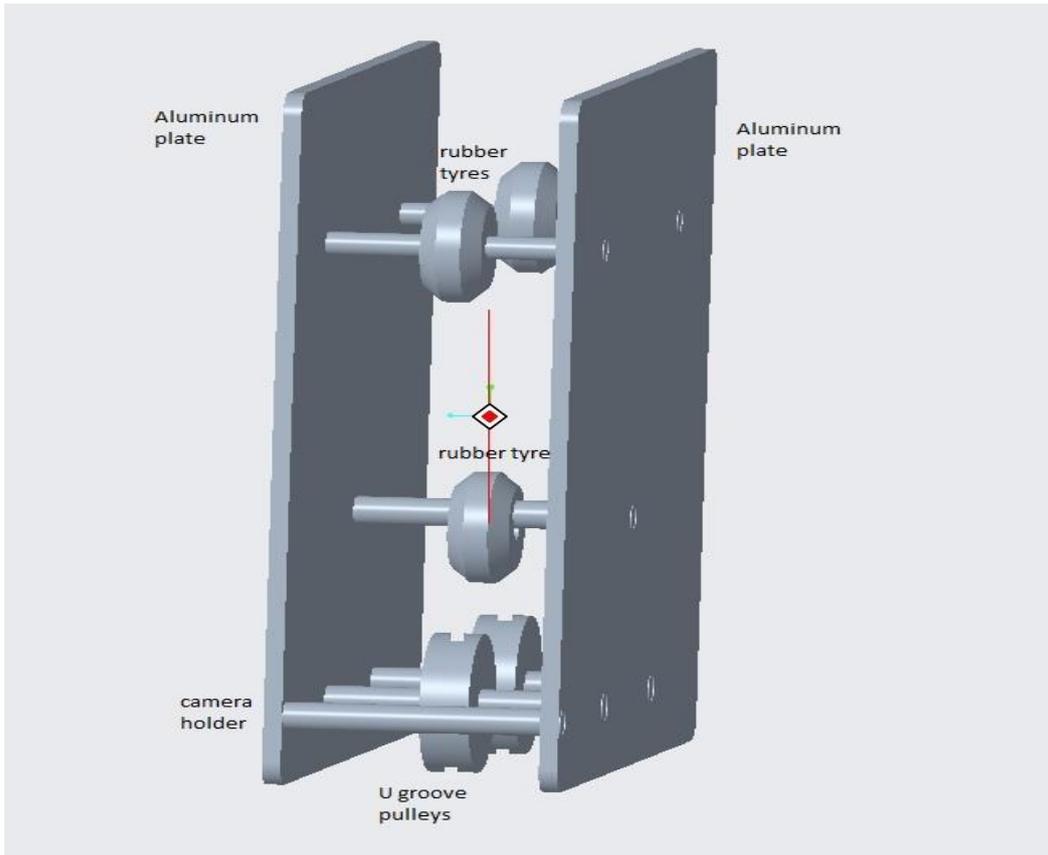


Figure 5-3 The hoist system design



Figure 5-4 The fabricated hoist system

The complete final design shown in Figures 5-5 and 5-6 below, consists of a hoist system and the girder which represented by 40*40 aluminium profile and support columns.

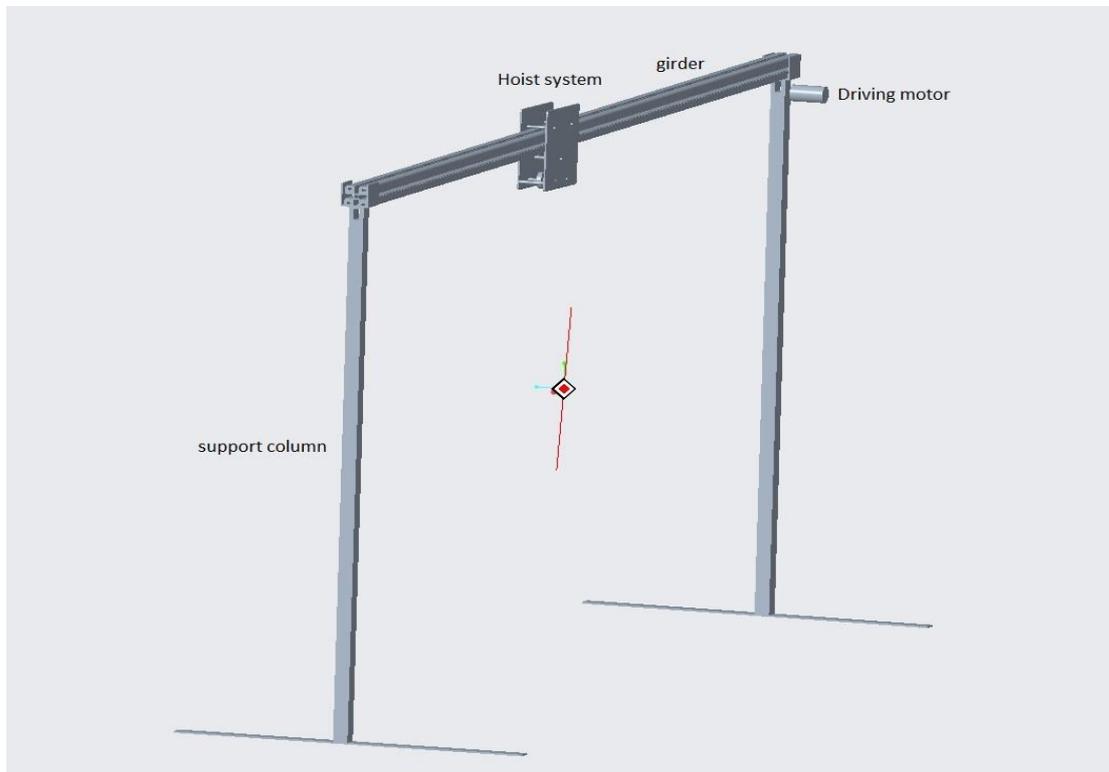


Figure 5-5 Final design of one-axis gantry crane



Figure 5-6 The fabricated one axis gantry crane

5.2 Experimental Components and Setup

5.2.1 Hoist and Cable Motion

The hoist was designed to enable translational movement along the girder. The hoist was powered by DC type electric motor. However, the power to the motor was relayed through a toothed belt. Two-timing belt pulleys were at the same size, relatively small and light and easy to manage. An added advantage to this design was the capability of the hoist to move along the girder without gear conversion. For the lowering and lifting load, another DC geared motor was placed on the girder beside the hoist motor. A U groove pulley was attached to it and worked as robin to collect the cable during the operation. The cable of the hoist will be under this, passing over a pulley to descend to the hook, then up again to pass to an anchor point at the end of the girder, an equal distance above its floor.

Two 12 V DC geared motors were purchased (CHIHAI) model (CHR-3429K-1210-50-ABHL) to drive the hoist and lowering and lifting the cable. These motors, as one of them shown in Figure 5-7 have hardware features of an integrated quadrature encoder with 64 counts per revolution (CPR) of the motor shaft. The motor documentation listed down with several parameters. At maximum load, the motor would draw currents up to 1.75 A. Under no load, the maximum current was rated at 0.3 A. At maximum speed, the stall torque was rated $\geq 20\text{kg.cm}$. The power of the motor was rated at 10.5W (Max 11W).

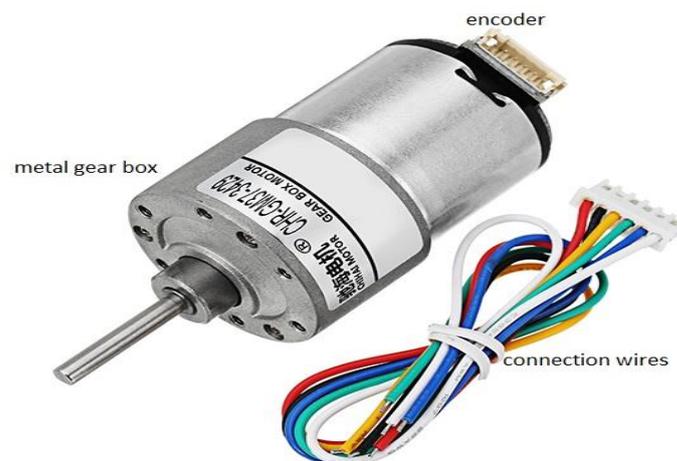


Figure 5-7 Metal Geared motor 200 RPM 31D*30L mm with 64 CPR Encoder

The motor/encoder has six colour-coded wires as follow:

Red: motor (+12V connects motor terminal)

White: motor (-12V connects to other motor terminals)

Blue: encoder (+) 5V /3.3V

Black: encoder GND (-) 5V /3.3V

Green: encoder B phase

Yellow: encoder A phase

To validate the counts per revolution (CPR) of the motor shaft, a real-time calculation has been done to confirm the number of counts per revolution which approximately was 44 count per revolution of the 200 RPM motor. , the distance units are measured by centimetres. To calculate the counts or steps of the encoder's motor. The equation below is used to obtain the steps of the motor's encoder:

$$\frac{(gear\ ratio * CPR)}{(2 * \pi)} \quad (5.1)$$

The hoist motor the steps per centimetre will be obtained as below:

$$hoist\ motor = \frac{(gear\ ratio * CPR)}{(2 * \pi)} / (motor's\ pulley\ radius) \quad (5.2)$$

$$hoist\ motor(kx) = \frac{(50 * 44)}{(2 * \pi)} / 2 = 175\ counts\ per\ centemetre$$

The cable motor the steps per centimetre will be obtained as below:

$$cable\ motor(kc) = \frac{(50 * 44)}{(2 * \pi)} / 0.5 = 700\ counts\ per\ centemetre$$

The type of motor does not affect the precision of the motion control significantly, except stepper motors that rely on a number of steps in order to rotate precisely (Billingsley, 2006). The accuracy in positioning relies on the design of the instrumentation or the system. If the mechanical components are sufficiently robust to resist external disturbances and the position control strategy is derived correctly, the positioning will work precisely.

5.2.2 L298N Dual Motor Controller Module

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time independently. The L298N H-bridge module can be used with DC motors that have a voltage of between 5 and 35V DC, with peak current up to 2A, as shown in Figure 5-8.

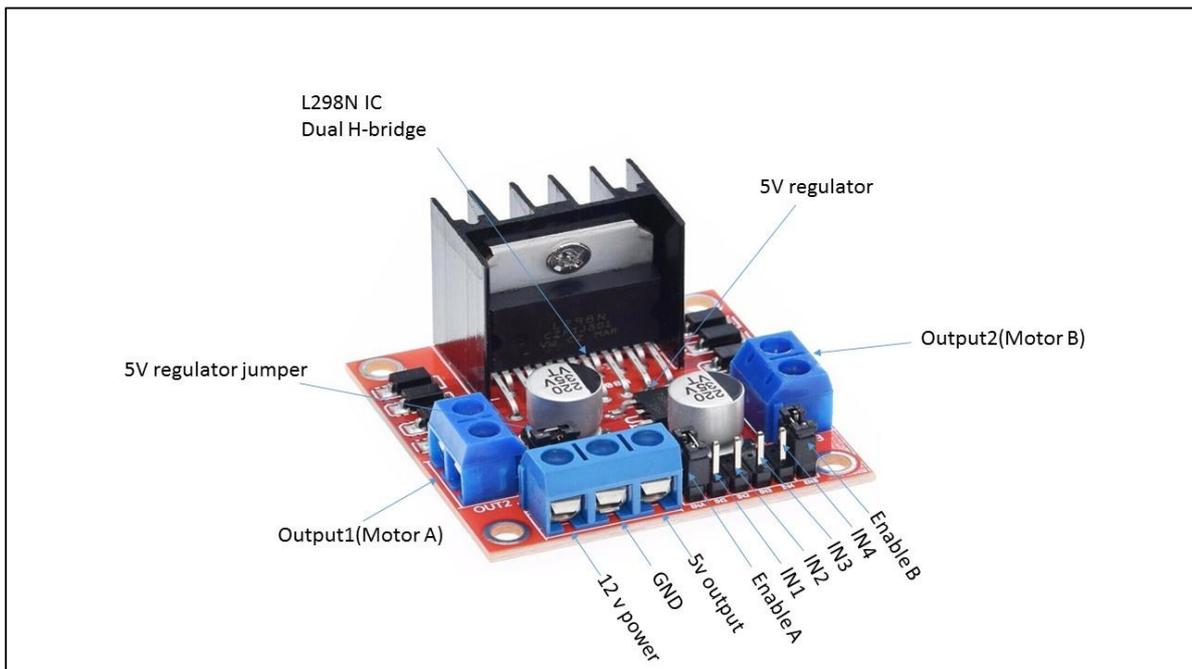


Figure 5-8 L298N Dual Motor Controller Module

5.2.3 Arduino

Arduino is a single-board microcontroller which helps to make the application more attainable. The hardware features operate with an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. It consists of a USB interface, 14 digital I/O pins and 6 Analog input pins that allow the designer to attach various extension boards or motors.

There are many types of Arduino boards in which many of them were third-party compatible versions. The most official versions available are the Arduino Uno R3.

The Arduino Uno board is a microcontroller based on the ATmega328. The board is equipped with sets of digital and analogue input/output (I/O) pins which can be interfaced to

various expansion boards and other circuits. It has 14 digital input/output pins in which six can be used as PWM outputs, an ICSP header, a 16 MHz ceramic resonator, 6 Analog inputs, a USB connection, a power jack and a reset button. This represents all the required features needed for the microcontroller. In order to get started, they are simply connected to a computer by a USB cable or with AC-to-DC adapter or battery. Arduino Uno Board varies from all other boards, and they will not use the FTDI USB-to-serial driver chip in them. It is featured by the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Figure 5-9 below shows the features of Arduino Uno from Sparkfun.

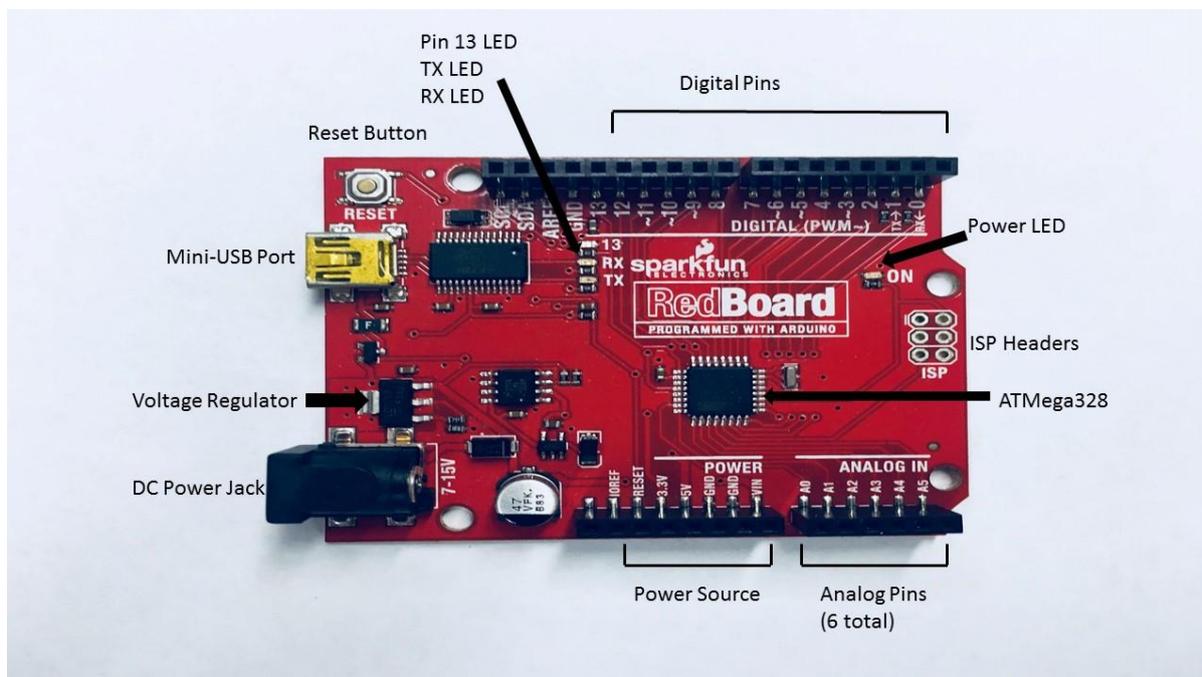


Figure 5-9 Arduino Uno Board (Sparkfun RedBoard)

The Arduino integrated development environment (IDE) is a cross-platform application written in Java and is derived from the IDE for the Processing programming language and the Wiring projects. The Arduino Uno board can be programmed with the Arduino software.

5.2.4 Gantry Crane’s Motor Wiring Connection

First of all, to control one or two DC motors, it needs to be connected to the Arduino board and power supply. Each motor is connected to the output1 and output2 connections on the L298N module. For the cable lowering and lifting motor, connect the encoder GND to the Arduino

GND and connect the encoder Vcc to the Arduino 5V. In addition, the encoder A phase will be connected to Analog pin A2 and the encoder B phase will be connected to Analog pin A3. For the hoist motor, connect the encoder GND to the L298N module GND and connect the encoder Vcc to the Arduino 3.3V. In addition, the encoder A phase will be connected to Analog pin A0, and the encoder B phase will be connected to Analog pin A1.

Next, the Arduino needs to be connected with the L298N module to drive both motors. So EnableA pin in L298N module will be connected to digital pin five which denoted by the tilde (“~”) next to the PIN with a pulse with modulation (PWM). IN1 pin will be connected to digital pin 12 for motor reverse rotation. Moreover, IN2 pin will be connected to digital pin 13 for motor forward.

Now, EnableB pin in L298N module will be connected to digital pin 11 which denoted by the tilde (“~”) next to the PIN with a pulse with modulation (PWM). IN3 pin will be connected to digital pin 9 for motor forward rotation. Moreover, IN4 pin will be connected to digital pin 10 for motor reverse.

Finally, the L298N module will be powered by a 12V power supply through pin 12V and GND on the module.

5.3 Computer Vision Strategy in Gantry Crane

This section will provide a detailed description of using a web camera as a sensor. Computer vision is a field of artificial intelligence which trains computers to understand the visual arts. Using digital images from videos and camera, computers can accurately identify and classify objects. Image processing is a part of computer vision. A computer vision uses the image processing algorithms to test and perform simulation of vision at human standard. A camera is used as a feedback sensor to detect the hanging load of the crane system. Object detection and tracking are challenging and essential tasks in many computer vision applications such as robot navigation and surveillance. To detect an object in a series of images, determination of the location of the object needs a sequence of stages. The tracking method requires to be represented in a way that makes sense to a computer. Object tracking is the process of locating an object or multiple objects over time using a camera by locating its position in every frame

of the video. Object tracking is a challenging task and would be difficult even for the human eye. Therefore, object tracking strategy is usually divided into several stages, which are object representation by shape or colour, object detection and object tracking. This section will contain an explanation of the computer vision strategy which has been used in the crane system. It consists of a camera which has been used as a sensor to detect the load movement as well as a novel method of using the camera to drive the load to the demanded position.

5.3.1 Object Representation

Object representation is an essential part of the tracking process. There are several ways to represent an object in the video stream such as by shape, features or colour. Moreover, each of these techniques would be divided into several sub-techniques, and each method has its advantages and disadvantages. In case of the crane system, methods such as points, geometric shape or edges are difficult to implement as the workplace underneath the crane has similar shapes or geometries especially when dealing with containers in the port. Therefore, represent the load by colour is the best way in the case of the crane.

Usually detection the object by colour effected by spectral power distribution and surface reflectance of the object. Often colour is represented by RGB (red, green, blue) in computer vision.

5.3.2 Segmentation

Image segmentation is the process of dividing or partitioning an image into multiple segments, or parts. This process is useful for object recognition and boundaries in the image or a sequence of images. These segments will cover the whole image. There are different techniques in this process. In the proposed technique, a colour-based segmentation method is used to track the load position during the crane operation by a series of images observed by a camera. A rectangle piece of shiny tape marker is attached over the crane's hook. The use of fluorescent coloured high visibility enhances this piece of tape. This colour is different among the other colours in the workstation area, and it is easy to be detected among them.

5.3.3 Processing Software

In this experiment, processing software has been used to generate and modify a series of image which detect the hanging load of the crane. Processing is an open-source graphical software and integrated development environment (IDE) designed to generate and modify images. Also, it has been created to programming within a visual context and to be used as a production tool, design professionals, and researchers and prototyping. Processing projects are called sketches. Each sketch has its own folder. Processing IDE consists mainly of void setup which set the size and the parameters of the project and void draw where the main program and loops will be defined.

Processing software has been used due utilizing a video which captures by a camera to detect the hanging load of the crane system. Furthermore, it is a faster way to process the data rather than by Arduino itself. The video camera is treated as a sensor which collects the data source from the series of images displayed on the screen.

The video camera has been used rather than other forms of sensors such as potentiometer or accelerometer because it has more accurate data of the position of the load as well as the velocity of the load can be calculated.

5.3.4 Load Position Tracking by The Camera

The crane's load is detected by the camera which is mounted underneath the hoist system that moves along the jib. It provides visual information about the load behaviour during crane operation. This information is processed instantaneously to obtain the load movement that includes the load position and velocity. The location of the load as a displacement from the hoist is found by a simple search for a bright spot in the image. This then gives the angles of the suspension cable, to be multiplied by the cable length for position and velocity calculations.

Originally the code was arranged to have the time-step determined by the availability of each video frame at intervals of 30 milliseconds. When the system failed to perform as expected, the problem was diagnosed as having been caused by variation of the frame interval. In the laboratory environment, the frame rate fell below ten frames per second. To estimate the load

velocity, a simple differencing had to be used. Figure (5-10) shows the initial image of the load tracking. The size of the video is 640 width with 480 height which represent the number of the pixel in the monitor. This size gives a large area around the load to give a tolerance if big load swing occurred.

Window pixel formats define how colour space information is stored in the GPU memory. The RGBA colour space uses the Red Green Blue (RGB) colour model with extra information about the alpha (transparency or opacity) channel. Digital colours are also constructed by mixing three primary colours. the individual colour elements are expressed as ranges from 0 (none of that colour) to 255 (as much as possible), and they are listed in the order red, green, and blue. In addition to the red, green, and blue components of each colour, there is an additional optional fourth component, referred to as the colour's "alpha." Which means opacity of the colour. The alpha values for an image are sometimes referred to collectively as the "alpha channel" of an image. Processing takes the colour numbers and adds a percentage of one to a percentage of another, creating the optical perception of blending. Processing software has a developed video library which needs to be imported from the provided library. Then declare a capture video which allows storing and manipulating video frames from an attached capture device such as a camera.

To perform the detection strategy, the frame rate set to 30 frame per second. The next step if the video available, then read the video as below:

```
if (video.available()) {  
  
    video.read();  
  
    video.loadPixels();  
  
}
```

Where `video.read` is to read the each new frame from the camera. And make its pixels [] array available.

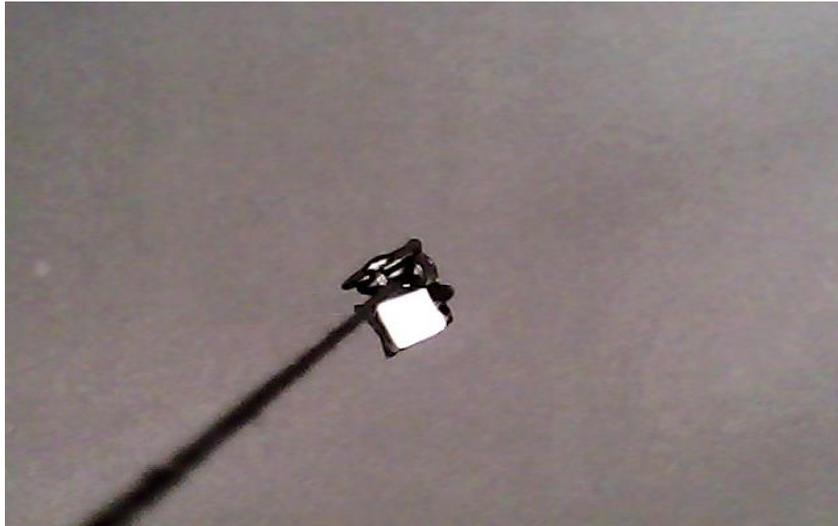


Figure 5-10 Initial image of the load

To detect the desired colour of the marker, the camera needs to read the video for each pixel in the screen using the command below:

```
color currColor = video.pixels[i];
```

Set a loop to detect the marker and use the blue colour as threshold as follow:

```
If ((currColor & 0xff)>255) {  
  
    hit =i;  
  
    Pixels [i] =0xffff0000;
```

Where “0xff” is the hexadecimal number FF which has an integer value of 255.

Moreover, “&0xff” effectively masks the variable, so it leaves only the value in the last 8 bits and ignores all the rest of the bits. Moreover, “0xffff0000” represents the red colour of the marker when it is detected, as shown in the figure 5-11 below:



Figure 5-11 Detected load by the camera

Now after the load has been detected, the position of the load in the camera needs to be calculated.

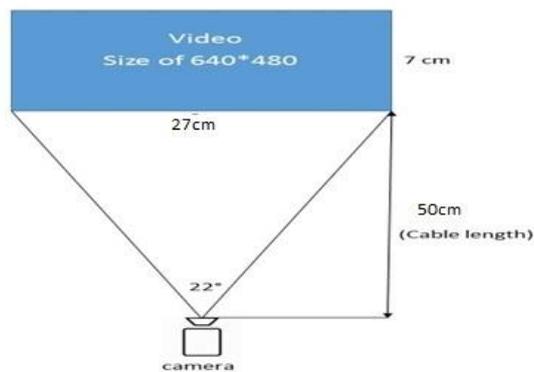


Figure 5-12 Camera calculation set-up

Figure 5-12 shows the camera setting. To calculate the real movement with the camera, we need to know the camera properties. The real measurement of the camera with 640*480-pixel window, the used camera has a span of 22° and coverage of $27*7$ cm. The coverage area of the camera at a certain distance is obtained as in the following equation below:

$$\text{camera factor } kc_{(cm/pixel)} = \frac{\text{camera length}}{\text{cable length} * \text{length in pixel}} \quad (5.3)$$

Using the equation above, the variable kc will be:

$$kc_{(cm/pixel)} = \frac{27}{50 * 640} = 0.000843$$

$$kc = 1185 \text{ pixel per cm}$$

From the image, the coordinates of the load relative to the hoist have been measured, which are $cLoadX$ and $cLoadY$ as if the frame is available, then read the frame and search for the bright spot. Now we have to calculate the actual coordinates of load, by multiplying the coordinates of load seen in camera, kc and cable length as the code below:

To calculate the load seen in camera in pixel, the lines of code below used:

```
cLoadX = -hit % width-width/2; //pixels
cLoadY = height/2-hit/width;
```

Where $cLoadX$ and $cLoadY$ are the load position seen in camera in x,y respectively. As one axis gantry crane used in the experiment, the real load position and velocity in x-axis will be calculated by:

```
LoadXrel=cLoadX*Cable/kc;
vLoadXrel=(LoadXrel-oldLoadXrel)/df;
oldLoadXrel=LoadXrel;
```

Where $LoadXrel$ is the real load position, $cLoadX$ is the load position seen in the camera. The cable is the cable length, kc camera factor, $oldLoadXrel$ is the position of the load in the previous frame, and df is the frame interval.

5.4 Encoder Reading

A two-channel Hall Effect encoder is used to sense the rotation of a magnetic disk on a rear end of the motor shaft. It is an electro-mechanical device that can convert the angular position

(rotation) of a shaft to either an Analog or digital output signals. The quadrature encoder provides a resolution of 64 counts per revolution of the motor shaft when counting both edges of both channels. There are two square wave pulses generated when the motor is rotating.

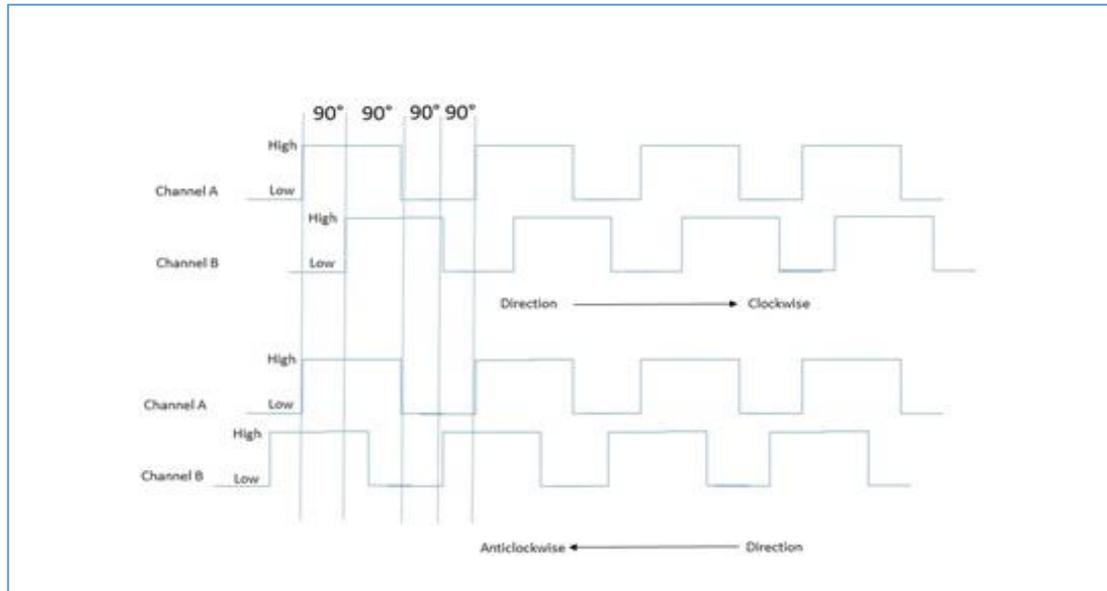


Figure 5-13 Quadrature Encoder A and B Output Signals

The A and B outputs are square waves from 0 V to Vcc. Using two code tracks with sectors positioned 90 degrees out of phase, as shown in figure (5.13), the two output channels of the quadrature encoder indicate both position and direction of rotation. If A leads B, for example, the disk is rotating in a clockwise direction. If B leads A, then the disk is rotating in a counter-clockwise direction. Therefore, by monitoring both the number of pulses and the relative phase of signals A and B, it can track both the position and direction of rotation. These are called single-ended because the A and B signals are both referenced to ground.

5.4.1 Encoder Measurement

The encoder measurement basically depends on the counter. A simple counter sent a value that presents the number of edges in the channel (waveform) that counted. The counter counts the events registered in the source input, and, depending on the state of the up/down line, it either increments the count or decrements it. For example, if the up/down line is “high” the counter increments the count, and if it is “low,” the counter decrements the count. An encoder usually has five wires that need to be connected to the instrument, and, depending on the encoder, these

wires vary in colour. It can use these wires to provide power to the encoder and to read in the A and B signals.

To read the motor encoder, an interrupt routine needs to be used. Interrupts are useful for making things happen automatically in microcontroller programs and can help solve timing problems. It makes the processor responds quickly to critical events. Interrupts in Arduino Board has two types which are external interrupt and pin change interrupt. The external interrupt is interpreted by hardware. These interrupts can be set to trigger on the event of RISING or FALLING or LOW levels.

The external interrupt pins in Arduino Uno is pin 2 and 3. However, Arduinos can have more interrupt pins enabled by using pin change interrupts. In ATmega168/328 based Arduino boards, any pins or all the 20 signal pins can be used as interrupt pins. They can also be triggered using RISING or FALLING edges. In order to use interrupts in Arduino, Interrupt Service Routine (ISR) or an Interrupt handler is an event that has a small set of instructions in it. When an external interrupt occurs, the processor first executes these code that is present in ISR and returns to the state where it left the normal execution. Each time an interrupt occurs, it triggers the associated ISR assuming, and it has turned that interrupt on. Each External Interrupt has its own ISR, and they can be triggered independently by either a rising signal, falling signal, or by both. However, the Pin Change Interrupts share an ISR between all the pins on a port (port B, C, and D).

And anytime a pin changes on that port, it calls the port's ISR which must then decide which pin caused the interrupt. So, Pin Change Interrupts are harder to use, but it is useful of using any Arduino pin. In this work, Arduino pin change interrupts have been used to read the motor encoder.

5.4.2 Reading the Encoder Algorithm

As an Arduino microcontroller has been used in the crane system, a pin change interrupt method has been chosen to read the motor's encoder. There are three stages to perform interrupt using pin changes which are turn on pin change interrupts, select pins to interrupt and write an

ISR for those pins. Firstly, port C has been turned on to perform the interrupt. It has been turned on by setting bit 1 in the PCICR register (PCINT8 – PCINT14) as follow:

```
PCICR |= 0b00000010;    // turn on port C interrupts
```

The second stage of interruption would be select which pin to interrupt. Since the ATMEGA328 has three ports, it also has three masks: PCMSK0, PCMSK1, and PCMSK2. These are set the same way the register for the PCICR was set. So pins in port C has been enabled as follow:

```
cli ();
```

```
PCICR |= 0b00000010;    // turn on port C interrupts
```

```
PCMSK1 |= 0b00001111;   // enables pins PC0 to PC5, pins A0 to A3
```

```
sei ();
```

Where cli (); is a macro that executes an assembler instruction to disable interrupts and turns interrupt off. However, sei () is a macro that executes an assembler instruction to enable interrupts and turns on again.

The final stage is to write the ISR which will be called for each of these interrupts as we can see as follow:

```
ISR (PCINT1_vect) { //interrupt service routine - pin change
```

The bit changes need to be updated on port C

```
    bits = PINC; update position
```

Also, to calculate the movement of the encoder, the changes in bits has to be calculated as:

```
change = ((oldbits & 0b1100) | (bits & 0b11));
```

```
    m1mov = move [change];
```

Where `m1mov` represents the movement of encoder1, however, the movement needs to be updated on time as:

```
If (m1mov) {  
    m1pos += m1mov;  
}
```

The same procedure is occurring for the second encoder except of different pins.

```
change = ((oldbits & 0b110000) | (bits & 0b1100))>>2
```

The change function has been shifted to the right by two bits

```
m2mov = move [change];
```

Where `m2mov` represents the movement of encoder 2, however, the movement need to be updated on the time as:

```
If (m2mov) {  
    m2pos += m2mov;  
}
```

5.5 Positioning Control by Arduino

In general, if the system is linear, the state equations can be expressed in a matrix equation of the form:

$$\dot{x} = Ax + Bu$$

Where: A is the 'system' matrix

B is the 'input' or 'driving' matrix

x the state vector

u the 'input' or 'control' vector

Moreover, the output/s of the system can be expressed in a matrix equation of the form:

$$y = Cx + Du$$

Where: y is the output vector

C is the 'output' matrix

D is the 'feed-forward' matrix

In using state-space techniques to design a controller, it is important to choose state variables that can be measured by sensors. For the third-order servo system, variables that are of interest are position, speed and acceleration. The acceleration of the output shaft in this system is a reflection of the torque generated by the motor. The control of the motors set at a 2-millisecond interval as the following code:

```
if (millis()>lastms){
    lastms= millis()+1;
```

The position error determines the state variable feedback and it has executed with the following code:

```
M1drive= (M1target-M1vel);
```

Where M1drive is the input for motor 1

M1target the demanded target

M1vel is the motor1 velocity

Similarly, applying an input for the motor 2 as below:

```
M2drive= (M2target-M2vel)
```

Where M2 drive is the input for motor 2

M2target the demanded target of motor2

M2vel is the motor2 velocity

The position feedback is applied to the demanded value. Therefore, the demanded position also considered as an input. To observe the velocity variable of the system, it has been assumed that the system has an output of position x , to obtain a damped response, the velocity feedback is needed v . since it cannot be measured directly, we need to estimate the velocity which we called it v_{est} . The estimation velocity can be integrated to obtain a modelled position called x_{slow} as follow:

$$x_{slow} = x_{slow} + v_{est} * dt \quad (5.4)$$

While

$$v_{est} = k * (x - x_{slow}) \quad (5.5)$$

These two lines of codes are able to estimate the velocity signal of the system. The following lines of the code have been written in Arduino part to calculate the estimation velocity of the crane system.

```
m1vel=m1pos-int (m1slow>>3); //counts per 2 ms  
  
m1slow+=m1vel;  
  
m2vel=m2pos-int (m2slow>>3); //counts per 2 ms  
  
m2slow+=m2vel;
```

5.6 Serial Communication Between Arduino and Processing Software

Serial communications provide a secure and flexible way for Arduino board to interact with the computer and other devices. As the camera has been used to detect the crane's load position and velocity by using processing software, the Arduino IDE and the Processing IDE will communicate with each other through serial communication. An Arduino has a serial port. However, serial communication provides connectivity to more than one device. Serial communication is handling with sending and receiving information. To interface between Arduino and processing software, it needs to communicate with each other correctly.

5.6.1 Arduino Section

For the Arduino part, the motors' positions have to be sent to processing. As the position of the motors read by Arduino using built-in encoder has a big number, it needs to be divided by low byte and a high byte. A byte stores an 8-bit unsigned number, from 0 to 255. Low byte and high byte are used when a data type uses more than one byte. The low byte is the byte that holds the least significant part of an integer. The low byte is the eight bits. Therefore, *serial.write()* the command has been used to send the motor position from Arduino to processing software. This command writes binary data to the serial port. This data is sent as a

byte or series of bytes. To obtain a low byte from the motor position reading, the following part of the code has been used:

```
Serial.write (m1pos & 0xff); //low byte
```

However, the following code has been used to obtain the high byte:

```
Serial.write (m1pos>>8); //high byte
```

Where `m1pos` is the position of the first motor, `0xff` is the hexadecimal number FF which has an integer value of 255.

“& 0xff” effectively masks the variable, so it leaves only the value in the last 8 bits, and ignores all the rest of the bits. Since the Java byte type is an 8 bit signed integral type with values in the range -128 to +127. The `0xff` represents +255 which is outside of that range. The value of `m1pos` has been shifted to the right by eight-bit to obtain the highest byte. Similarly, for the second motor, the position is sending by using the following code:

```
Serial.write (m2pos>>8); // high byte  
Serial.write (m2pos & 0xff); // low byte
```

Where `m2pos` is the position of the second motor

To receive the calculated motor target from Processing, firstly, check if there any serial available through the following line of code:

```
if (Serial.available()>3) {
```

Each two bytes will give the target velocities which count per 10 millisecond by using this part of code:

```
m1target=Serial.read();  
if (m1target>127) {m1target-=256;} //this restores the sign bit  
m1target=256*m1target|Serial.read();  
m2target=Serial.read();  
if (m2target>127) {m2target-=256;} //this restores the sign bit  
m2target=256*m2target|Serial.read();
```

Where `m1target` and `m2target` are the demanded velocities of the motors which sent by Processing to Arduino.

5.6.2 Processing Section

The Processing IDE is similar to Arduino in terms of structure. It has set up functions and draw functions which similar to Arduino, which also has a setup and loop function. The setup function in Processing is similar to Arduino, which used to handle one-time initialization. The The Processing IDE can communicate with the Arduino IDE by serial communication. Therefore, it is applicable to send data from the Arduino to the Processing IDE as well as from the Processing IDE to the Arduino. The Processing IDE has a serial library which allows for easy reading and writing data to and from external machines. It allows two computers to send and receive data and gives the flexibility to communicate with custom microcontroller devices, using them as the input or output to Processing program. The serial port is a nine pin I/O port that exists on many PCs and can be emulated through USB. In the crane experiment, the serial library has been imported as the following:

```
import processing.serial.*;
```

Also, an object needs to be created from the serial class using this part of code:

```
Serial myPort;
```

The line `myPort = new Serial (this, "PortName", 115200);` opens the selected port which has been connected to Arduino. Where PortName is COM17 and 115200 is the baud rate.

The draw function in Processing is same as the loop function in Arduino, it is called repeatedly. This fuction checks if any available data on the serial port. The coming data byte will be converted to an integer and read by Processing. Reading the motor positions when it is available by check if there is any coming data from Arduino by using this line of the code:

```
if(myPort.available() >4) {  
  getData();
```

Next, the function `read()` is used to read the coming data. It also Returns a number between 0 and 255 for the next byte that is waiting in the buffer. Using the following lines of the code to read the motor position from Arduino:

```
void getData() {
```

```

m1pos = myPort.read() ;
if (m1pos>127) {m1pos-=256;}
m1pos=(m1pos<<8) |myPort.read();
m2pos = myPort.read() ;
if (m2pos>127) {m2pos-=256;}
m2pos=(m2pos<<8) |myPort.read();

```

After reading the serial port, if statement used which allows the program to make a decision about which code to execute. Therefore, if the position value is greater than 127, the position will be subtracted of 256 which restore the sign bit. Then shift that value 8 bits to the left or read the serial port.

Next, the demanded velocities that has been calculated by Processing needs to be sent to Arduino. The function `write()` used to send those values as following:

```

void sendcommands() {
myPort.write(vMotor1Dem>>8);
myPort.write(vMotor1Dem&255);
myPort.write(vMotor2Dem>>8);
myPort.write(vMotor2Dem&255);
}

```

These values were defined as an integer. The most important part of knowing how many bytes are expected to be transferred between Arduino and Processing. Arduino defines an integer as two bytes, but Processing (Java) defines an integer as four bytes.

5.7 Control algorithm via Processing

After the crane load has been detected based on its brightness, it is easy to calculate its position and velocity. The position of the hoist deduced from the difference between the load position and its target, a velocity demand is calculated, proportional to the distance. This is limited in magnitude to a safe travel velocity as shown in piece of code below:

```

vLoadXdem=(TargetX-LoadX) *kv;
    if (vLoadXdem> vmax) {vLoadXdem= vmax;}
    if (vLoadXdem< -vmax) {vLoadXdem= (-vmax);}

```

Where $v_{LoadXdem}$ is the demanded velocity of the load, $TargetX$ is the load target, and $LoadX$ is the load position, k_v is the feedback gain of 0.5 and v_{max} is set as 10 cm/sec.

The counts of motor's encoder are calculated based on the crane design, which explained previously. It is called $k_x=175$ counts/cm for the hoist motor and $k_y=700$ count/cm for the cable motor. Calculate the hoist position by:

```
HoistX=m1pos/kx + HoistXdatum;
```

Where $HoistX$ is the position of the hoist, $m1pos$ is the motor position, $HoistXdatum$ is hoist position datum.

The velocity of the hoist is calculated by:

```
vHoistX=(HoistX-oldHoistX)/dt;  
oldHoistX=HoistX;
```

where v_{HoistX} is the hoist velocity, $oldHoistX$ is the previous position of the hoist and dt time interval.

Finally giving signals that can be injected into the motor control loops as a demand for motor velocity. The motor velocity demand is constructed by:

```
vMotor1Dem = int(kmotor*(3*LoadXrel + vLoadXdem));  
if(vMotor1Dem> 1000){ vMotor1Dem = 1000;}  
if(vMotor1Dem< -1000){ vMotor1Dem= -1000;}
```

Where $kmotor$ is a factor to convert cm/sec demand to counts for Arduino as we send the motor position in 20 millisecond interval which is = 9

Where $v_{Motor1Dem}$ is the demanded motor velocity.

5.8 Experimental Results and Discussion

In this section, the designed pragmatic strategy for one-axis gantry crane system is implemented on a laboratory-size gantry crane. The proposed pragmatic control technique is developed in the previous section has been evaluated with experimental test.

Several tests have been achieved to evaluate the performance of the proposed strategy. In addition, these tests also carried out to demonstrate their capability in delivering a high

performance of positioning the load as well as suppressing any load swing might have occurred. Three different scenarios have been considered to examine the proposed strategy.

Firstly, operate the crane with no load. Secondly, operate the crane with 1 kg weight and finally add a disturbance during the operation to represent any external disturbance such as wind gust. Besides, by using a camera and the whole system is driven by the view from the crane, the demanded positions were implemented using Processing software as the operator sitting on the crane cabin.

The experimental setup has been mentioned earlier. However, the system was interfaced with Arduino microcontroller board which connected to the computer (Intel® Core™ i7-6700 CPU @ 3.40GHz 3.41 GHz with 16 GB RAM). This setup works with the sampling time $T_s = 0.02$ second.

Figure (5-14) shows the position response of load and hoist during travelling with a 50 cm cable length with excellent performance. The hoist and the load moved to demanded position.

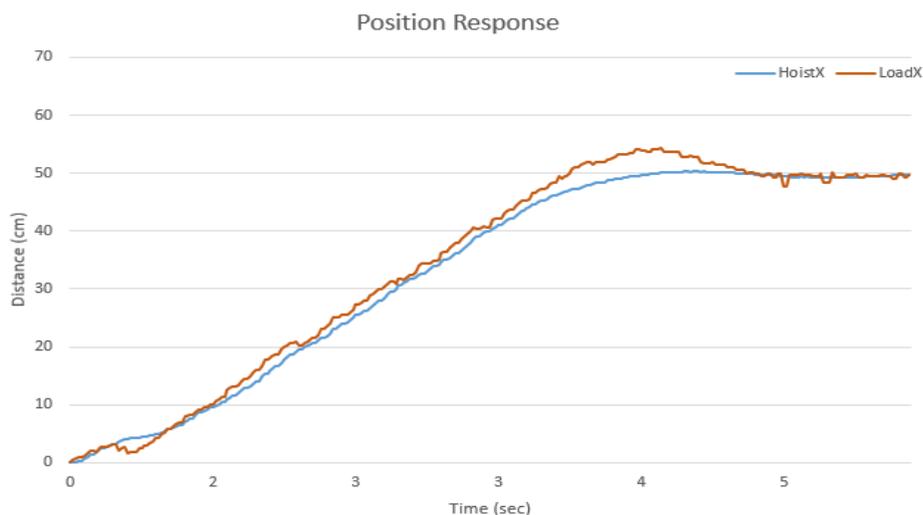


Figure 5-14 Position response of hoist and load with no weight

However, in figure 5-15, a different scenario is used to evaluate the proposed strategy by disturbing the load during the operation. As seen in figure (5-15), the load has been disturbed during the crane operation while the load was trying to compensate for the position deviation. Before the load has reached the desired velocity, the load was disturbed. Therefore, the desired hoist velocity falls to make load returns underneath the hoist, then with small movement back and forth to suppress any load swing until reaching the demanded position. Both scenarios in figures 5-14 and 5-15 is carried out without weight hanging on the crane's hook.

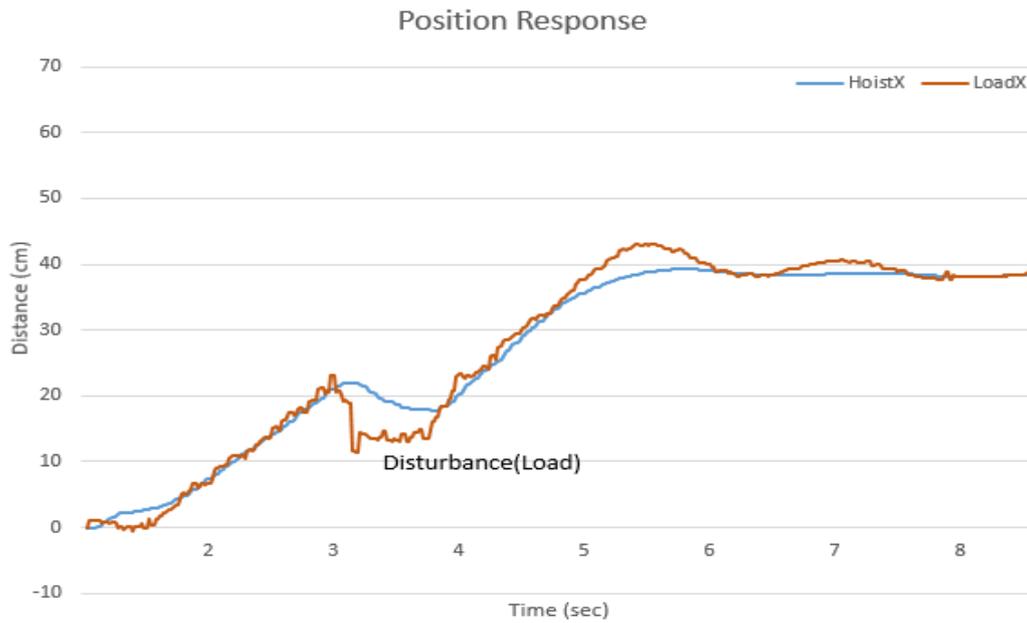


Figure 5-15 Position response of hoist and load with no weight with disturbance

Another test has been made with 1 kg weight, as shown in figure (5-16). It can be seen from the figure discussed earlier that, the proposed strategy could successfully control the position of the crane with high performance in both with weight and without weight.

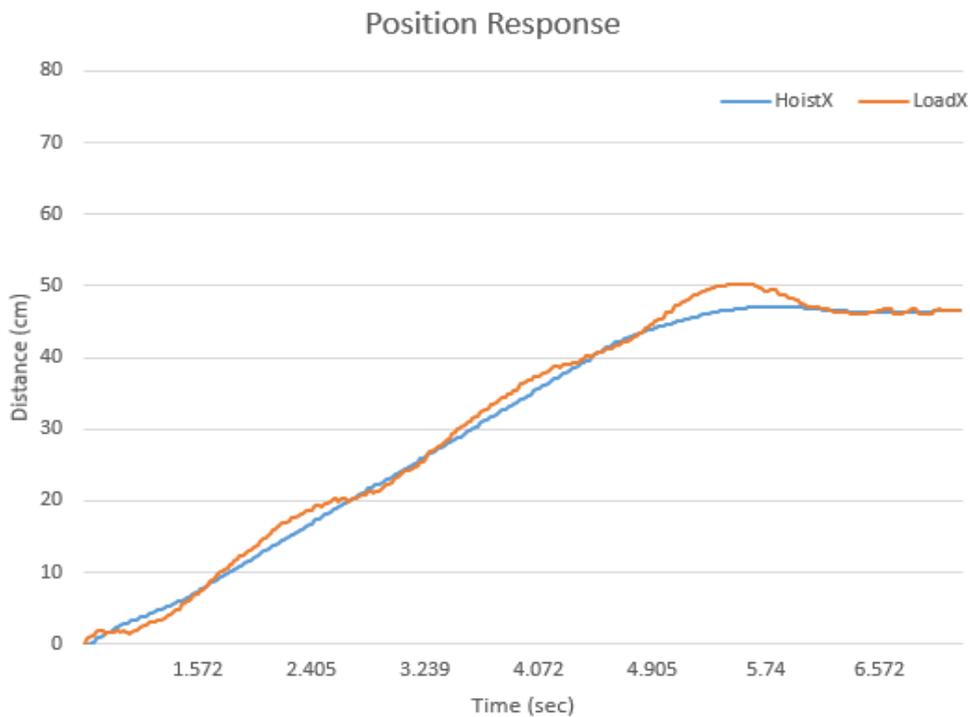


Figure 5-16 Position response of hoist and load with 1 kg weight

Figure 5-17 shows the velocity profile of the hoist during the operation. It is accelerated up to around 20 cm/s and remains stable until the hoist reaches the final destination then goes down under zero and finally goes to zero when it reached the demanded position.

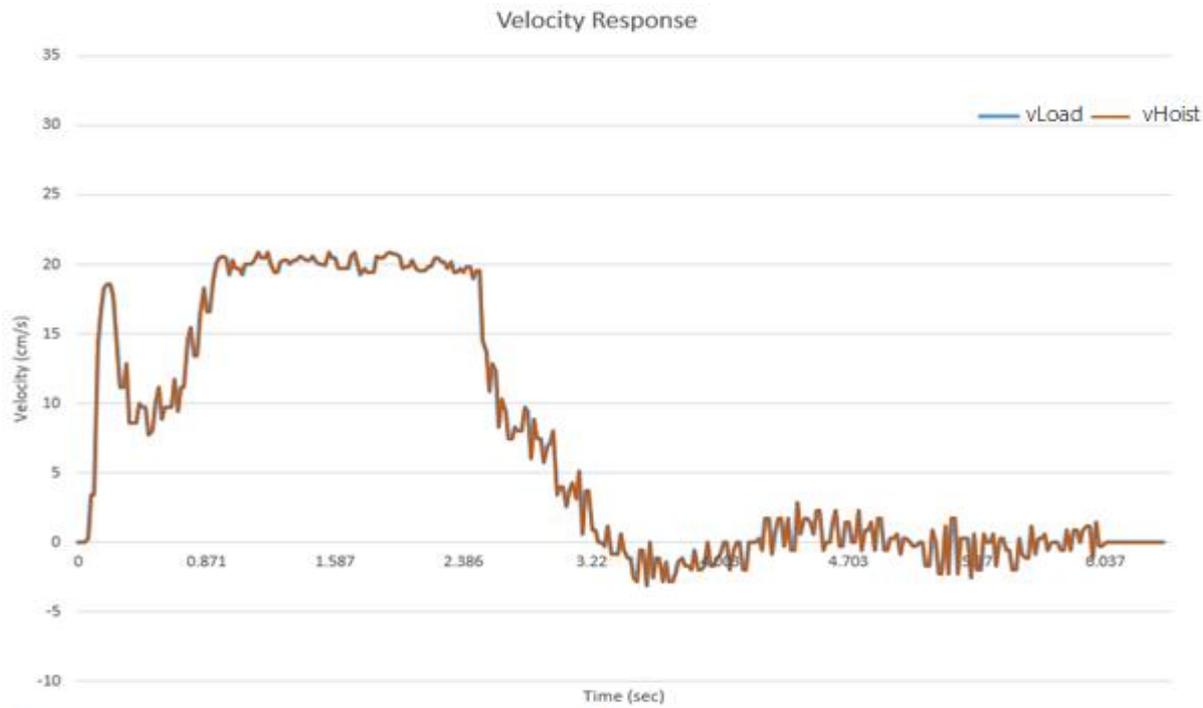


Figure 5-17 Velocity response of hoist and load

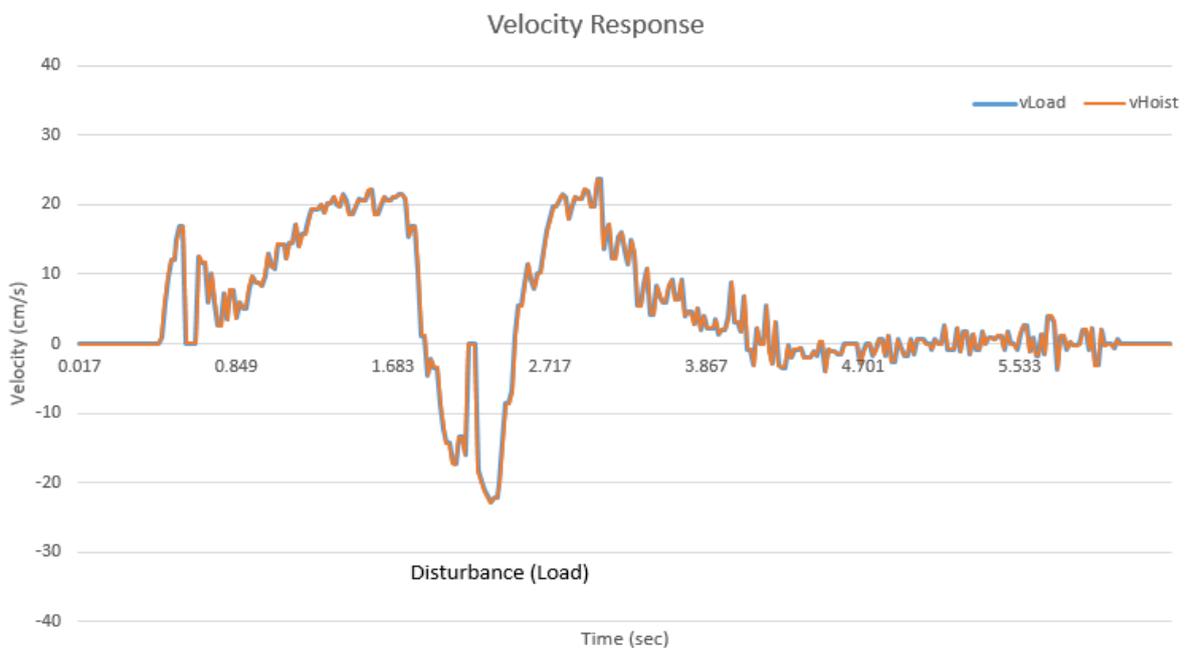


Figure 5-18 Velocity response of hoist and load with disturbance

Another test has been made, as seen in Figure 5-18, which shows the velocity of the hoist and the load profiles when a disturbance interfered on the load during the operation. It can be seen that the hoist reverses the movement to compensate the load position and then goes up to around 20 cm/s till reaching the demanded position. However, Figure 5-19 shows the velocity response of the hoist and the load with 1 kg weight.

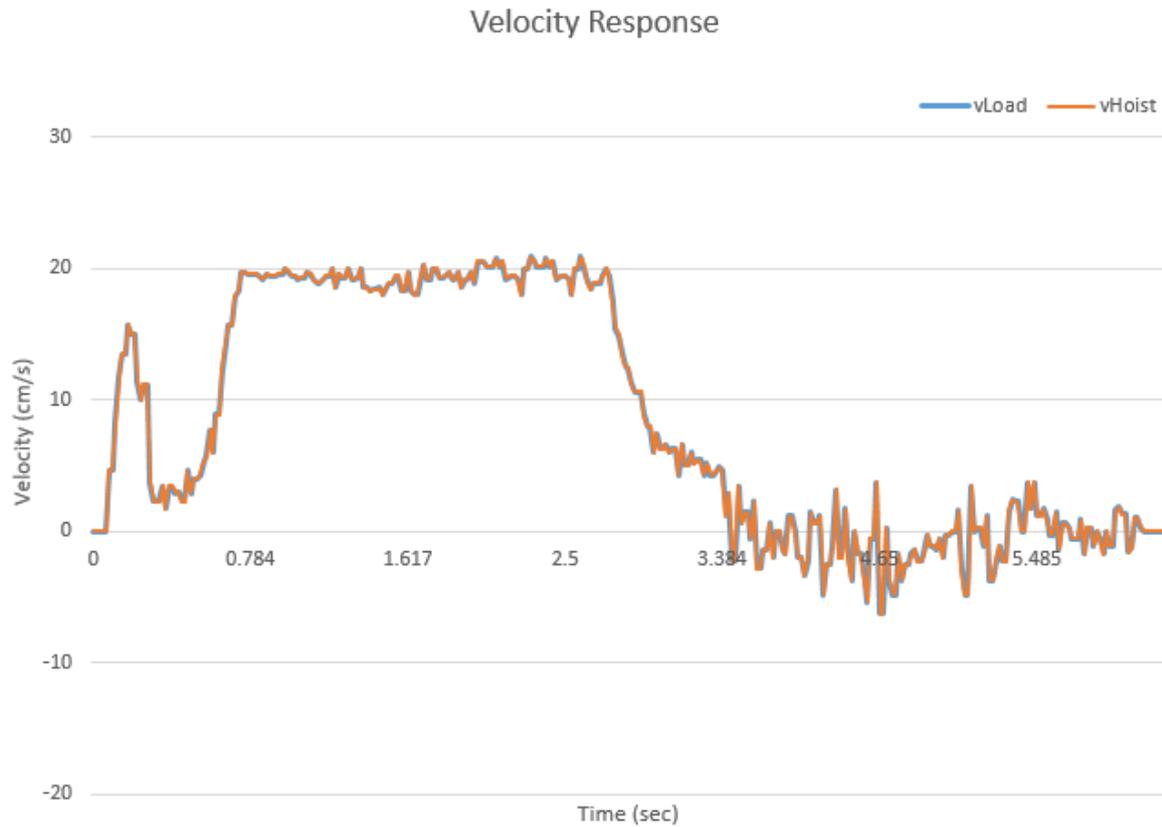


Figure 5-19 Velocity response of hoist and load with 1kg weight

To show the performance of the swing angle during the operation, several tests have been made to evaluate the used strategy. The first test of observing the swing angle is Figure 5-20. In that figure, we can see that the maximum swing occurred at the beginning of a movement which reached to around 0.05 radian that approximately equal to 2.8 degrees. After that is being around 0.01 radian during the operation. The swing angle has been calculated by camera and Processing, which has a high sensitivity to any movement occurred to the hanging load. Figure 5-21 shows the swing angle of the load when it interferes with external disturbance. An external disturbance with around 0.25 radian injected to the load during the operation. However, the

hoist suppressed the generated swing on the load quickly and moved to the final position with no swing.

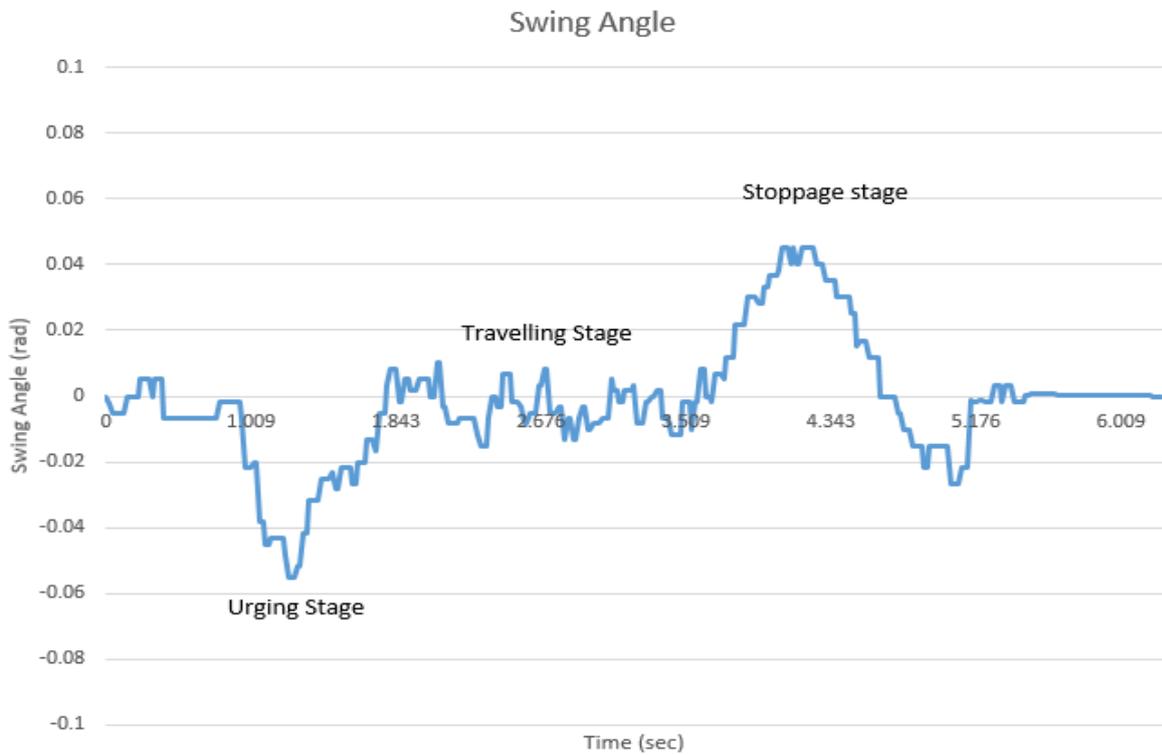


Figure 5-20 Swing angle response with no weight

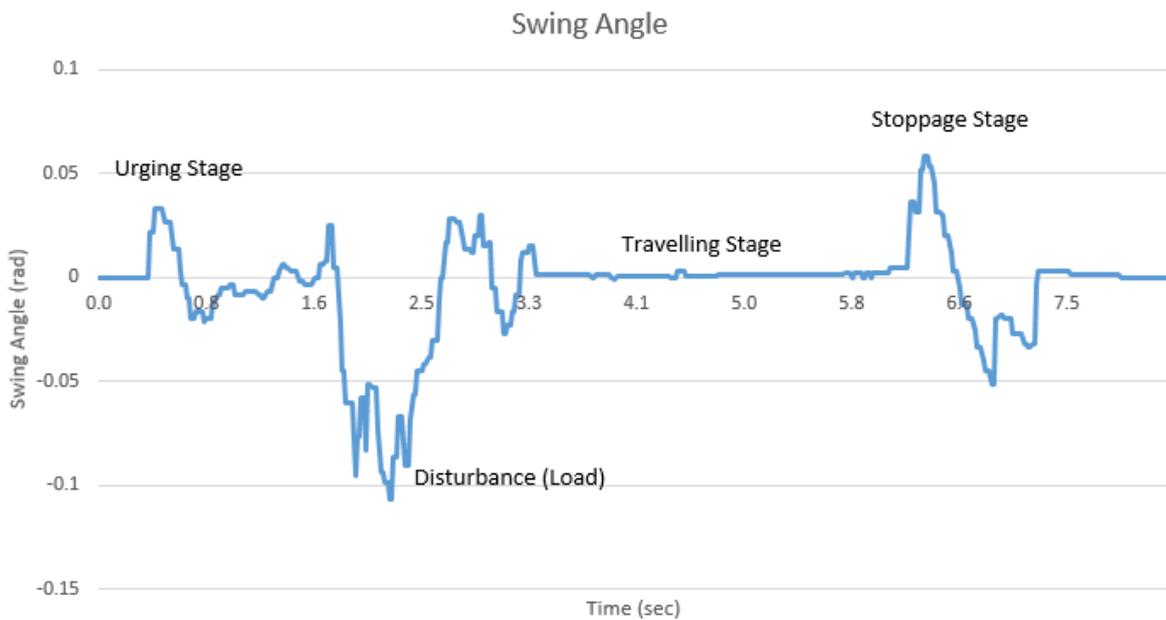


Figure 5-21 Swing angle response with no weight with disturbance

Figures 5-22 and 5-23 showed the swing angle of the load when a 1 kg weight was the hanging load. As we can see that the maximum swing reached over 0.04 radian at the beginning of the operation, then around 0.01 radian during the movement until reach the demanded position. Moreover, in Figure 5-23 an external disturbance interposed to the load which reached over 0.25 radian, then the hoist suppressed it quickly, and they remained without swing until reached the final destination. Thus, these results showed that the proposed strategy could robustly and successfully suppress any load swing which might occur.

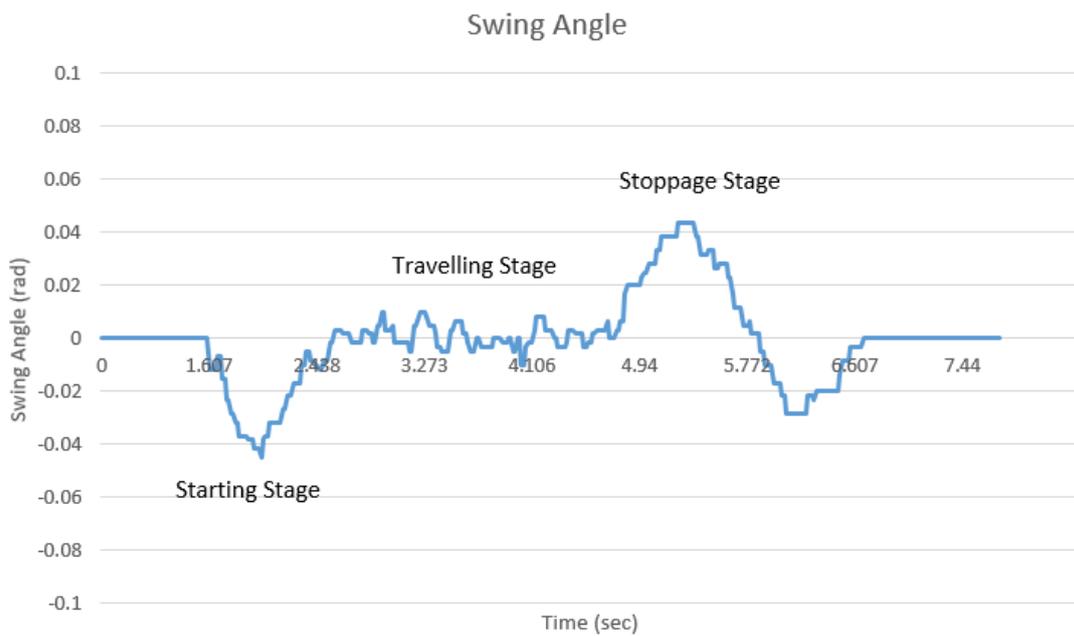


Figure 5-22 Swing angle response with 1 kg weight

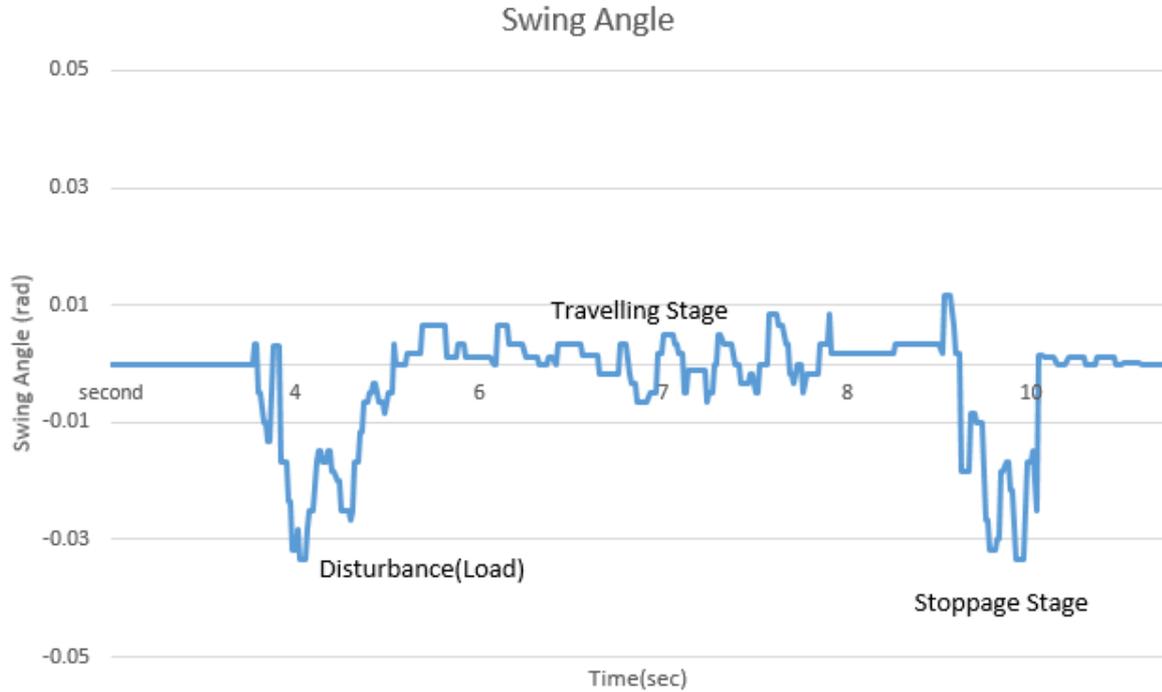


Figure 5-23 swing angle response with 1 kg weight with disturbance

5.9 Conclusion

A single-axis gantry crane has been designed and fabricated in this chapter. The proposed practical control strategy has been elaborated and explained, which is brief. A high-performance gantry crane control operation requires that the load should be transported as fast as possible with high accuracy in load positioning with a minimum load swing as possible. Moreover, the control system design and settings should not be very complex and difficult to understand by the operator. Based on these objectives, the configuration of the proposed pragmatic control system was described. Several tests have been carried out to validate the proposed strategy. As mentioned previously, the three test scenarios under which the gantry crane should be controlled are conducted with the addition of using external disturbance as well as with load masses. The proposed strategy showed a successful and robust performance in term of positioning the transported load and suppress any swing that might occur during the crane operation.

6 CHAPTER SIX: - EXPERIMENTAL DESIGN OF TOWER CRANE SYSTEM

6.1 Mechanical Design Considerations

The tower crane consists of trolley moving along a jib in translation motion, and the jib rotating in a horizontal plane. The trolley consists of a hoist system (rope and hook) to lift and lower the load. The combination of trolley and jib movement allows the load to reach the desired destination.

In general, Tower crane needs to be balanced to compensate for the lifted load. Usually, tower cranes have counterweight in the rear end of the jib. Moreover, the mast of the crane often anchored in the ground, which make the system balanced. The tower crane consists of two major parts which are the mast and the jib. In this section, the design of the mechanical and electrical part of the system will be described.

In the proposed mechanical design, the counterweight has been replaced with the drive motors. Also, the base of the mast has been designed to grab the edge of a table and works as a clamp which secures the system tightly and prevents movement. The jib consists of an aluminium U channel which allows the hoist to move along it smoothly. There will be a five-centimetre gap between them to allow the cable mechanism to hang below the jib. The motors of the hoist and the cable lowering and lifting have been fixed in the rear end of the jib which also works as a counterweight as mentioned earlier. The jib has a designed hollow bar which will be placed into the angle's motor shaft.

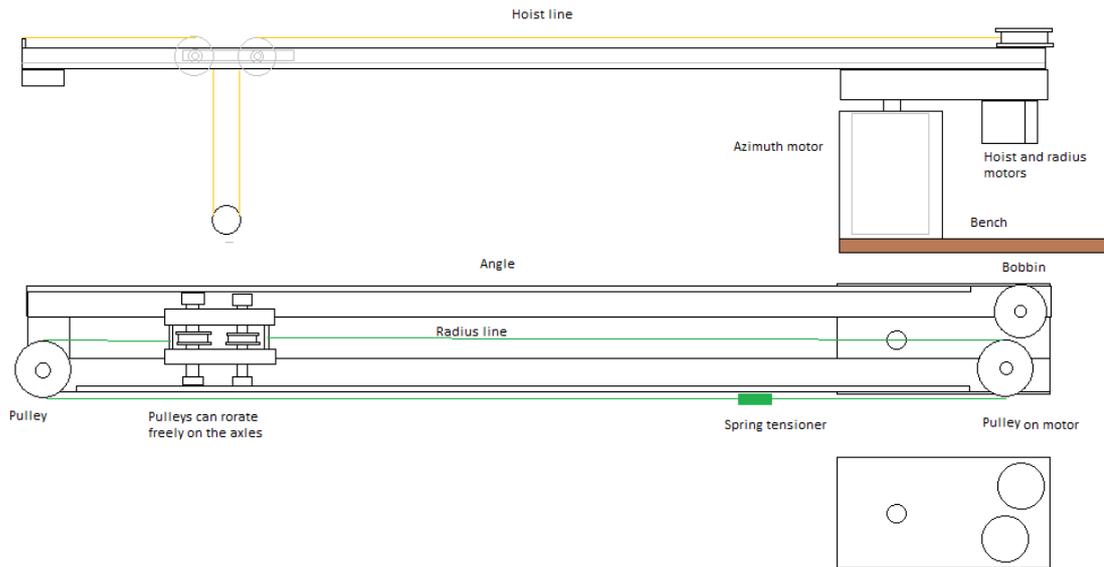


Figure 6-1 Crane jib (side and top view)

The hoist motor will lie just beside the cable motor, with an idler at the outer end. As seen in Figure 6-1, the belt of the hoist will be above this, attached to the hoist and then pass to idler pulley at the end of the jib, an equal distance above its floor. A U groove pulley was attached to it and worked as robin to collect the cable during the operation. The cable of the hoist will be above this, passing over a pulley to descend to the hook, then up again to pass to an anchor point at the end of the boom, an equal distance above its floor.

An essential element to give rigidity between the mast and the jib is a hallowed bar which mounted by a screw into the jib, which is coupled with bearings on the top of the mast and they are assembled by pressure on the mast as shown in Figure (6-2)

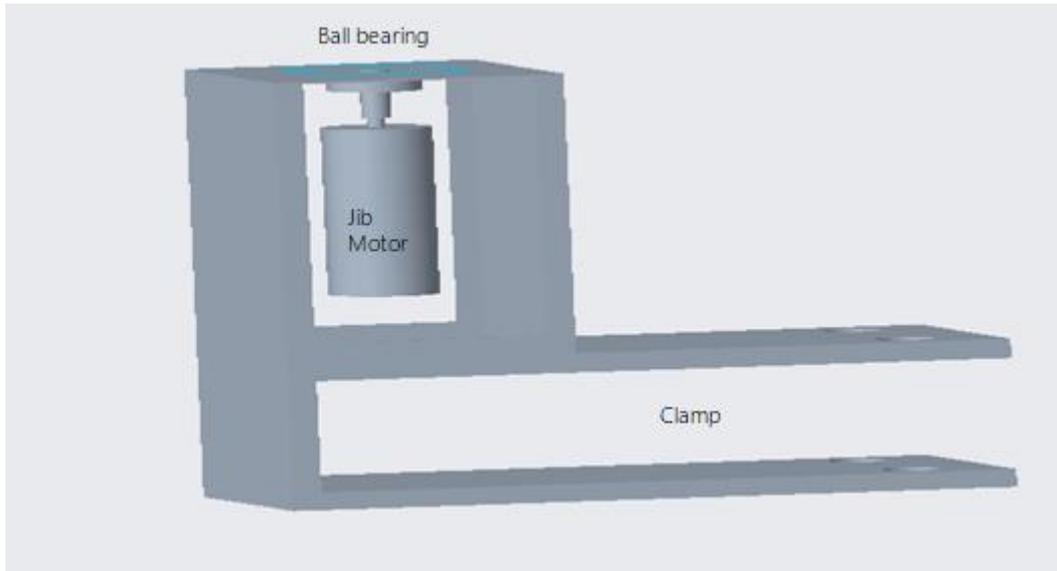


Figure 6-2 Jib angle driver structure

The hoist is shown in Figure 6-3, which consists of a chassis of an aluminium base and four wheels which are for moving into two rails of the jib. Also, it has two U groove pulley placed in the middle of the hoist for the cable mechanism of lowering and lifting the hanging load. The trolley will be moved by utilizing a rubber-toothed belt.

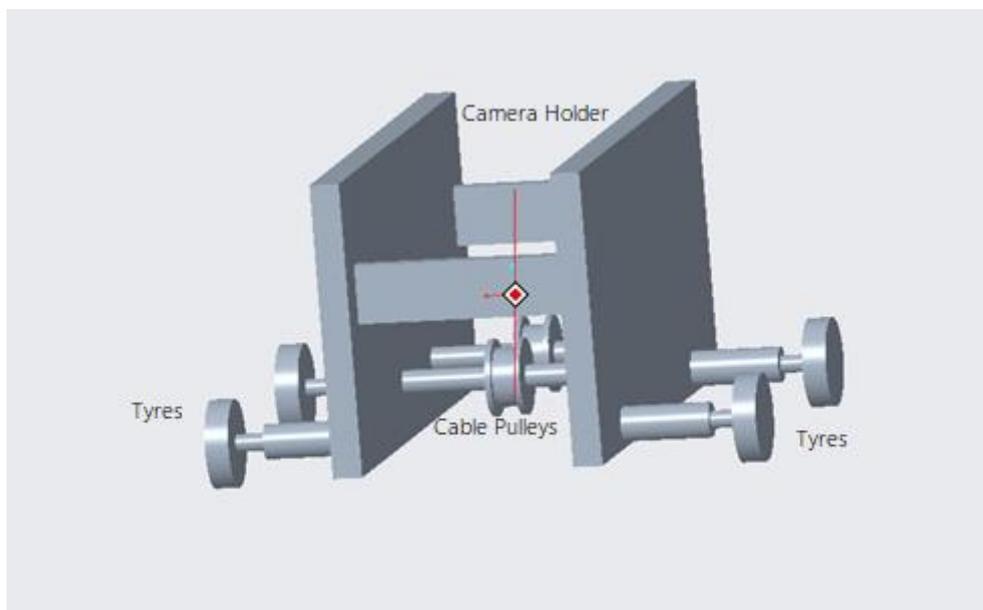


Figure 6-3 Crane hoist system

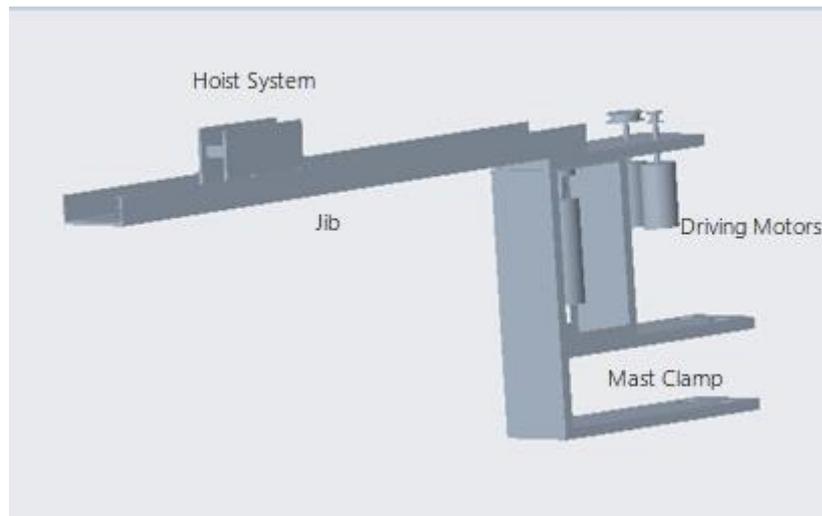


Figure 6-4 The proposed design of the tower crane system

Figure (6-4) above represent the proposed design of tower crane. However, figure (6-5) show of the final fabrication apparatus of the system. It consists of driving motors which drive the hoist along the jib and rotate the jib around 180°.



Figure 6-5 Final design of tower crane

6.2 Experimental Components and Setup

6.2.1 Hoist and Cable Motion

The hoist was designed to enable translational movement along the jib. The hoist was powered by DC type electric motor. However, the power to the motor was relayed through a toothed belt. Two-timing belt pulleys were at the same size, relatively small and light and easy to manage. An added advantage to this design was the capability of the hoist to move along the jib without gear conversion. For the lowering and lifting load, another DC geared motor was placed on the jib beside the hoist motor. A U groove pulley was attached to it and worked as robin to collect the cable during the operation. The cable of the hoist will be above this, passing over a pulley to descend to the hook, then up again to pass to an anchor point at the end of the jib, an equal distance above its floor.

Two 12 V DC geared motors were purchased (Bringsmart) 133RPM to drive the hoist and to lower and lift the cable. These motors have hardware features of an integrated quadrature encoder with 64 counts per revolution (CPR) of the motor shaft. The motor documentation listed down with several parameters. At maximum load, the motor would draw currents up to 1.5 A. Under no load, the maximum current was rated at 0.2 A. The weight of each motor was 0.5 kg. At maximum speed, the nominal torque was rated at 10 kg/cm. The power of the motor was rated at 13W, as shown in Figure 6-6.



Figure 6-6 Metal Geared motor 133 RPM 37D*57L mm with 64 CPR Encoder

The motor/encoder has six colour-coded wires as follow:

Red: motor (+ connects motor terminal)

Black: motor (- connects to other motor terminals)

Blue: Hall sensor Vcc (3.5-20V)

Green: Hall sensor GND

White: Hall sensor B Vout

Yellow: Hall sensor A Vout

To validate the counts per revolution (CPR) of the motor shaft, a real-time calculation has been done to confirm the number of counts per revolution which approximately was 66.6 count per revolution of the 133 RPM motor. , the distance units are measured by centimetres. To calculate the counts or steps of the encoder's motor. The same equation (5.1) in the previous chapter is used to obtain the steps of the motor's encoder:

$$\frac{(gear\ ratio * CPR)}{(2 * \pi)} \quad (5.1)$$

The hoist motor the steps per centimetre will be obtained as below:

$$hoist\ motor = \frac{(gear\ ratio * CPR)}{(2 * \pi)} / (motor's\ pulley\ radius)$$
$$hoist\ motor(kr) = \frac{(30 * 64)}{(2 * \pi)} / 0.5 = 611\ counts\ per\ centemetre$$

6.2.2 Jib Angular Motion

The jib is designed to be mounted over the mast of the crane. The jib is coupled with mast by ball bearing which been assembled by pressure in the internal support of the mast. The jib was powered by DC type electric motor. The motor was coupled by bearing and designing a small hollow shaft to connect with motor's shaft directly. An added advantage to this design was the capability of the jib to rotate without gear conversion.

A 12 V brushed DC geared motor (BRINGSMART) 14 RPM that have hardware features of an integrated quadrature encoder with 64 counts per revolution (CPR) of the motor shaft. This

has been used to drive the jib. The motor documentation listed down with several parameters. At maximum load, the motor would draw currents up to 1.3 A. Under no load, the maximum current was rated at 0.2 A. The weight of each motor was 0.5 kg. At maximum speed, the nominal torque was rated at 60 kg.cm. The power of the motor was rated at 13W.

A real-time calculation has been done to confirm the number of counts per revolution, which approximately was 62.9 count per revolution of the 14 RPM motor, the angle is measured by using radian. To calculate the counts or steps of the encoder's motor, the equation (5.1) in the previous chapter has been used. The jib motor, the steps per radian will be obtained as below:

$$Jib\ motor(ka) = \frac{(270 * 64)}{(2 * \pi)} = 2751\ counts\ per\ radian$$

The type of motor does not affect the precision of the motion control significantly. The accuracy in positioning relies on the design of the instrumentation and the system. If the mechanical components are sufficiently robust to resist external disturbances and the position control strategy is derived correctly, the positioning will work precisely.

6.3 The Electrical Setup of Tower Crane

Basically, to control one or two DC motors, it needs to be connected to the Arduino board and power supply. Also, The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time independently. Each motor is connected to the output1 and output2 connections on the L298N module. For the cable lowering and lifting motor, connect the encoder GND to the Arduino GND and connect the encoder Vcc to the Arduino 5V. In addition, the encoder A phase will be connected to Analog pin A2 and the encoder B phase will be connected to Analog pin A3. For the hoist motor, connect the encoder GND to the L298N module GND and connect the encoder Vcc to the Arduino 3.3V. Further, the encoder A phase will be connected to Analog pin A0, and the encoder B phase will be connected to Analog pin A1.

Next, the Arduino needs to be connected with the L298N module to drive both motors. So EnableA pin in L298N module will be connected to digital pin 5 which denoted by the tilde

("~") next to the pin number with a pulse with modulation (PWM). IN1 pin will be connected to digital pin 12 for motor reverse rotation. Moreover, IN2 pin will be connected to digital pin 13 for motor forward.

Now, EnableB pin in L298N module will be connected to digital pin 11 which denoted by the tilde ("~") next to the pin number with a pulse with modulation (PWM). IN3 pin will be connected to digital pin 9 for motor forward rotation. Moreover, IN4 pin will be connected to digital pin 10 for motor reverse.

Finally, the L298N module will be powered by a 12V power supply through pin 12V and GND on the module.

6.4 Computer Vision Strategy for Tower Crane

In this section, a summary of using a camera as a sensor with its algorithm to detect the crane load. The used strategy in tower crane is similar to the one of gantry crane. The crane's load is detected by the camera which is mounted underneath the hoist system that moves along the jib. It provides visual information about the load behaviour during crane operation. This information is processed instantaneously to obtain the load movement that includes the load position and velocity. The location of the load as a displacement from the hoist is found by a simple search for a bright spot in the image. Then send via serial communication to Arduino which control the speed and position of the driving motors of the system. The position control of the motors is controlled by using the Arduino board. It involves reading the encoders to measure the rotation of the driving motor, which gives us the travelling distance of the trolley as well as the angle of the jib while rotating.

Serial communications provide a secure and flexible way for the Arduino board to interact with the computer and other devices. As the camera has been used to detect the crane's load position and velocity by using processing software, the Arduino IDE and the Processing IDE will communicate with each other through serial communication. An Arduino has a serial port. However, serial communication provides connectivity to more than one device. Serial communication is handling with sending and receiving information. To interface between Arduino and processing software, it needs to communicate with each other correctly.

6.5 The Control Algorithm of the Tower Crane

In place of estimating speed as a simple difference in the count, the low-pass filter described below was applied. In the process of the calculation, the result is multiplied by a factor of eight, by shifting it three binary places to the left. In consequence, the quantization of the control demand is reduced to one per cent of full speed.

Rather than applying mark-space pseudo-linear control, bang-bang control is applied to the motor drive, only being zero if the speed error is zero. The mark-space is now asynchronously determined by the motor velocity itself. The motor speed accurately responds to unit increments in demand, this 'sliding motion' eliminating errors due to friction or striction.

First, the encoder reading routine is described and explained below, then that of the low-pass filter.

6.5.1 Reading the Encoders by Arduino

The two-phase encoders cycle through 00, 01, 11, 10 for rotation in one direction and 00, 10, 11, 01 for the reverse. The increment or decrement of position can thus be determined by comparing the present inputs with the previous value. A four-bit composite is constructed by combining the present value with the old value shifted two places, thus 0000, 0101, 1111 and 1010 indicate that no move has occurred, 0001, 0111, 1110 and 1000 indicate a positive move while 0010, 1011, 1101 and 0100 indicate a negative one. A table is defined with values of 0, 1 or -1 corresponding to these combinations:

```
int move[16] = {0, 1, -1, 0, -1, 0, 0, 1, 1, 0, 0, -1, 0, -1, 1, 0};
```

If both bits have changed, however, there is an error that means that a transition has been missed. At 6000 rpm, 4800 transitions per second can occur on each of the three encoders, but if the interval between samples is substantially less than 150 microseconds, the risk will be small. That sets a limit on the duration of the interrupt routine.

Although the hobbyist software of the Arduino presents individual bits to the user, byte-wide port operations are still performed. Three pairs of adjacent bits of PORTC are connected to the sensors. The setup code

```
cli();  
  
PCICR|=0b00000010;
```

```

//turn on port C interrupts

PCMSK1 |= 0b00111111;

// enable pins PC0 to PC5, pins A0 to A5

sei();

```

Sets up an interrupt on the change of any input, while the routine itself is

```

ISR (PCINT1_vect){
    bits = PINC;
//for motor 1
    change = ((oldbits & 0b1100) | (bits & 0b11));
    m1pos += move[change];
//for motor 2
    change = ((oldbits & 0b110000) | (bits & 0b1100))>>2;
    m2pos += move[change];
//for motor 3
    change = ((oldbits & 0b11000000) | (bits & 0b110000))>>4;
    m3pos += move[change];
    oldbits=bits<<2;
}

```

6.5.2 Estimating Velocity by Arduino

The simple process of subtracting the latest value from the previous one produces a rough quantized signal. In terms of the z transform this estimated velocity is

$$ve(z) = (1 - z^{-1}) x(z) / dt \quad (6.1)$$

Where dt is the sample interval, x is the input and ve is the velocity to be estimated.

At lower speeds, this will give a ‘steppy’ value. At low speed, some intervals will contain a step and others none. The following code will generate a smoothed velocity.

First a lagged signal $xslow$ is constructed:

```
xslow += (x - xslow)*k*dt;
```

And the estimated velocity $\mathcal{V}\mathcal{E}$ is then given in the loop by

$$ve = k * (x - xslow);$$

In z-transform terms,

$$xslow = k * dt * x / (z - 1 + k * dt) \quad (6.2)$$

So

$$ve(z) = k * \left\{ (z - 1) / (z - 1 + k * dt) \right\} x \quad (6.3)$$

Which can be rewritten instead as

$$ve(z) = k * \left\{ (1 - z^{-1}) / (1 - [1 - k * dt]z^{-1}) \right\} x \quad (6.4)$$

Clearly if $k = 1/dt$ this reduces to the original simple difference.

For implementation within an Arduino, the execution time can be shortened by avoiding floating-point operations. The following code achieves this:

```
m1vel=m1pos-int(m1slow>>3); //counts per 2 ms
m1slow+=m1vel;
```

Where this part of the loop is executed at 2 *ms* intervals, and all variables are integers, except for *m1slow*, which is long. This gives smoothing with an equivalent time-constant of 16*ms*.

With the factor imposed by the shift, it was found that full speed on the motor corresponded to a velocity demand of 220. With a top speed of 27 cm/second, that means that the speed quantization is just over one millimetre per second.

6.5.3 Frame Rate and Timing

The location of the load is found as a displacement from the hoist by a simple search for a bright spot in the image. This then gives the angles of the suspension cable, to be multiplied by the cable length for position and velocity calculations as described below:

$$LoadXrel = (cLoadX * Cable) / kc \quad (6.5)$$

$$LoadYrel = (cLoadY * Cable) / kc \quad (6.6)$$

Where *LoadXrel* and *LoadYrel* are the positions of the load axes, *Cable* is the length of the cable and *kc* is camera factor= 850 pixel/rad

Then, the load velocities have been calculated by subtracting the load position of each frame as below:

$$vLoadXrel = (LoadXrel - oldLoadXrel) / df \quad (6.7)$$

$$vLoadYrel = (LoadYrel - oldLoadYrel) / df \quad (6.8)$$

Where $vLoadXrel$ and $vLoadYrel$ are the load velocities, $LoadXrel$ and $LoadYrel$ are the positions of the load axes, $oldLoadXrel$ and $oldLoadYrel$ are the position of the load axes in the previous frame, and df is the frame interval.

Originally the code was arranged to synchronise all calculations to each video frame at intervals of 30 ms. When the system failed to perform as expected, the problem was diagnosed as having been caused by the frame interval and its variability. In the laboratory environment, the frame rate fell below 10 fps. The code was therefore restructured so that other calculations were made at fixed time intervals, while the load offset was calculated whenever a frame became available.

Since the frame interval can vary with the brightness, we have to use a variable dt for measuring velocities. Rates then will be ‘per second’ rather than ‘per frame’ as described in the code below:

```
if (video.available()) {
  dt=(millis()-framestartms)/1000;
  print(millis()-framestartms); //time data arrived
  print("\t");
  framestartms=millis();
  frames++;
  if(frames>100){frames=0;}
  myPort.clear(); //clear the old reply
  sendcommands(); //Send again to ask for latest counts
```

Several schemes were tried for the serial communication between the desktop machine and the Arduino. In the latest version, the Arduino reports the encoder counts at fixed 20 ms intervals and the desktop replies with a new speed demand.

6.5.4 Control Algorithm via Processing for Tower Crane

After the crane load has been detected based on its brightness, it is easy to calculate its position and velocity. The position of the hoist deduced from the difference between the load position and its target, a velocity demand is calculated, proportional to the distance. This is limited in magnitude to a safe travel velocity as shown below:

$$\begin{aligned}
vLoadXdem &= (TargetX - LoadX) * kv; \\
&\text{if } (vLoadXdem > vmax) \{vLoadXdem = vmax;\} \\
&\text{if } (vLoadXdem < -vmax) \{vLoadXdem = (-vmax);\} \\
vLoadYdem &= (TargetY - LoadY) / ktv; \\
&\text{if } (vLoadYdem > vmax) \{vLoadYdem = vmax;\} \\
&\text{if } (vLoadYdem < -vmax) \{vLoadYdem = (-vmax);\}
\end{aligned} \tag{6.9}$$

Where $vLoadXdem$ and $vLoadYdem$ is the demanded velocity of the load, $TargetX$ and $TargetY$ is the load target, and $LoadX$ and $LoadY$ is the load position, kv is a feedback gain of 0.5 and $vmax$ is set as 10 cm/sec.

The counts of motor's encoder are calculated based on the crane design, which explained previously. It is called $kx=620$ counts/cm for the hoist motor and $ky=2700$ count/cm for the jib motor. Calculate the position by:

$$\begin{aligned}
HoistX &= m1pos / kx + Hoistdatum; \\
Angle &= m2pos / ky + Angdatum;
\end{aligned} \tag{6.10}$$

Where $HoistX$ is the position of the hoist, $m1pos$ is the motor position, $Hoistdatum$ is hoist position datum.

An $Angle$ is the position of the jib, $m2pos$ is the jib motor position, $Angdatum$ is hoist position datum.

The velocity of the hoist calculated simply by:

$$\begin{aligned}
vHoistX &= (HoistX - oldHoistX) / dt; \\
oldHoistX &= HoistX; \\
vAng &= (Ang - oldAng) / dt;
\end{aligned} \tag{6.11}$$

where $vHoistX$ is the hoist velocity, $oldHoistX$ is the previous position of the hoist and dt time interval. And $vAng$ is the angular velocity of the jib, Ang is jib angle while $oldAng$ is the previous jib angle in prior frame

Finally giving signals that can be injected into the motor control loops as a demand for motor velocity. The motor velocity demand is constructed by:

$$\begin{aligned}
vMotor1Dem &= \text{int} (kmotor1 * (3 * LoadXrel + vLoadXdem)); \\
vMotor2Dem &= \text{int} (kmotor2 * (3 * LoadYrel + vLoadYdem));
\end{aligned} \tag{6.12}$$

$$\begin{aligned}
& \text{if } (vMotor1Dem > 1000) \{ vMotor1Dem = 1000; \} \\
& \text{if } (vMotor1Dem < -1000) \{ vMotor1Dem = -1000; \} \\
& \text{if } (vMotor2Dem > 1000) \{ vMotor2Dem = 1000; \} \\
& \text{if } (vMotor2Dem < -1000) \{ vMotor2Dem = -1000; \}
\end{aligned} \tag{6.13}$$

Where $kmotor$ is a factor to convert cm/sec demand to counts for Arduino as we send the motor position in 20 millisecond interval, $vMotor1Dem$ and $vMotor2Dem$ are the demanded motors velocities

Now we transform the target position for change in crane axes

$$\begin{aligned}
c &= \cos(Angle); \\
s &= \sin(Angle);
\end{aligned} \tag{6.14}$$

where c and s sine and cosine of the angle moved

$$temp = (TargetX + LoadX) * c + TargetY * s - LoadX \tag{6.15}$$

move target with yard

$$TargetY = -(TargetX + LoadX) * s + TargetY * c \tag{6.16}$$

$$TargetX = temp \tag{6.17}$$

Finally, give the load target position by clicking on the demanded position on the screen which sends new targets to the motors.

6.6 Experimental Results and Discussion

In this section, the designed pragmatic strategy for tower crane system is implemented on a laboratory-size gantry crane. The proposed pragmatic control technique is developed in the previous section has been evaluated with experimental test. Several tests have been achieved to evaluate the performance of the proposed strategy. In addition, these tests also carried out to demonstrate their capability in delivering a high performance of positioning the load as well as suppressing any load swing might have occurred. Three different scenarios have been considered to examine the proposed strategy. Firstly, operate the crane with no load. Secondly, operate the crane with 1 kg weight and finally add a disturbance during the operation to represent any external disturbance such as wind gust. Also, by using a camera and the whole system is driven by the view from the crane, the demanded positions were implemented using Processing software as the operator sitting on the crane cabin. The experimental setup has been

mentioned earlier. However, the system was interfaced with the Arduino microcontroller board which connected to the computer (Intel® Core™ i7-6700 CPU @ 3.40GHz 3.41 GHz with 16 GB RAM). This setup works with the sampling time $T_s = 0.02$ second. Commands were generated to move the hoist from its position to the demanded position of 10 cm, 13 cm and rotating of 3,2 radians from the original position of the crane.

Figure (6-7) shows the position response of hoist during travelling with a 60 cm cable length without weight. The hoist moved to a demanded position which is 10 cm and hang back before reached the demanded position to eliminate any swing of the load might occur.

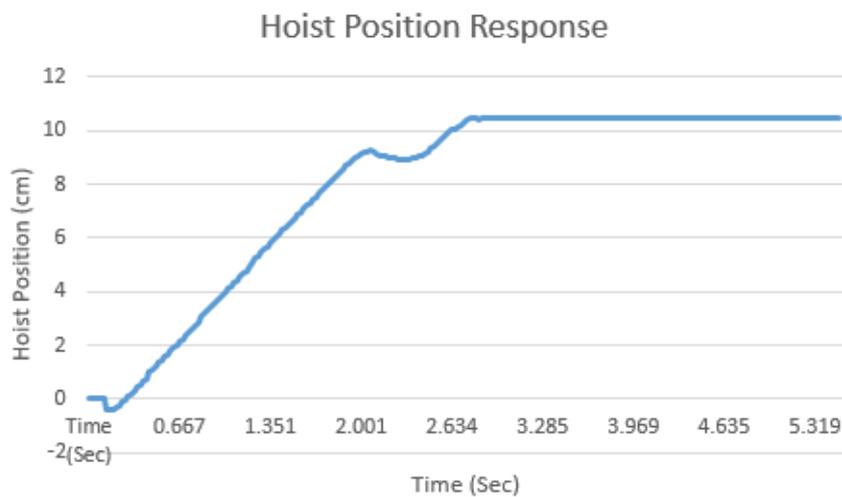


Figure 6-7 Position response of tower hoist with no weight

Figure 6-8 shows the position response of jib during travelling with a 60 cm cable length without weight. The jib moved to the demanded position, which is 3 rad. Likewise, there is a hang back as soon as reached the demanded position to stop the load without swing.



Figure 6-8 Position response of tower jib with no weight

Figure 6-9 shows the velocity profile of the hoist during the operation. It is accelerated up to around 6 cm/s and remains stable until the hoist reaches the final destination then goes down under zero and finally goes to zero when it reached the demanded position.

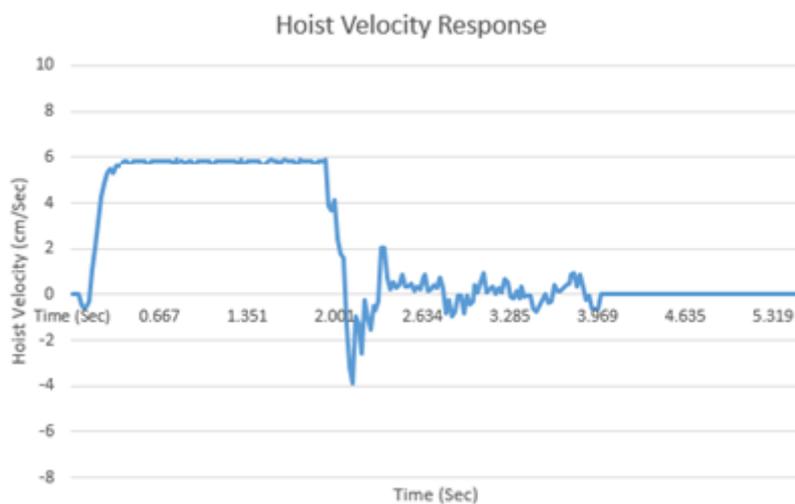


Figure 6-9 Velocity response of tower hoist

As we can see below, Figure 6-10 shows the velocity profile of the jib during the operation. It is accelerated up to around 1.2 rad/s and remains stable until the hoist reaches the final destination then goes down to less than 1 rad/s and finally goes to zero when it reached the demanded position.

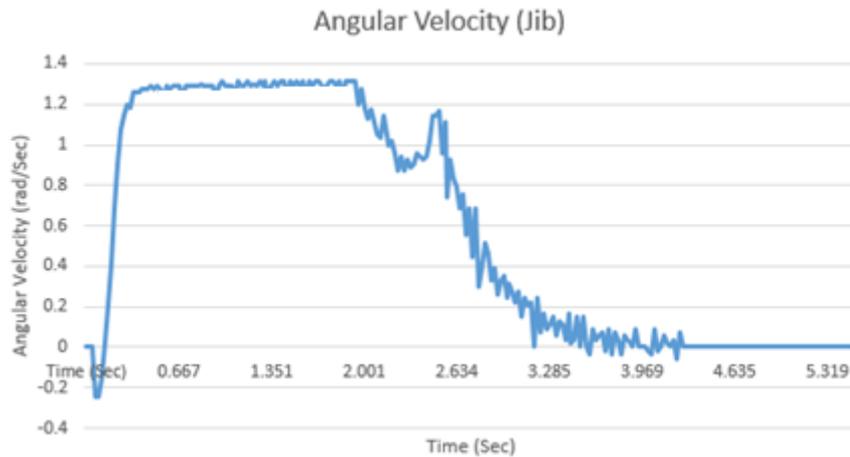


Figure 6-10 Velocity response of tower hoist

To show the performance of suppressing the swing angles during the operation, several tests have been made to evaluate the used strategy. The observing swing angle Beta which related to the jib movement shown in Figure 6-11. It is 0.0004 rad which is unnoticeable.

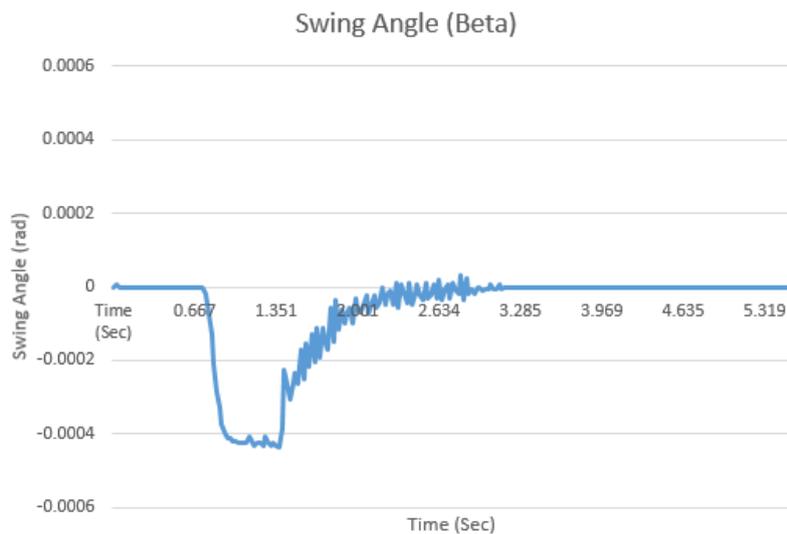


Figure 6-11 The swing angle “Beta” response with no weight

However, in Figure 6-12, it is shown that the swing angle “theta” that is related to the hoist movement. The maximum swing occurred at the beginning of a movement which reached to around 0.06 radian. After that is being around zero radians during the operation then goes up to 0.02 rad in the stoppage stage to suppress load swing during the deaccelerating. The swing angle has been calculated by camera and Processing, which has a high sensitivity to any movement occurred to the hanging load.

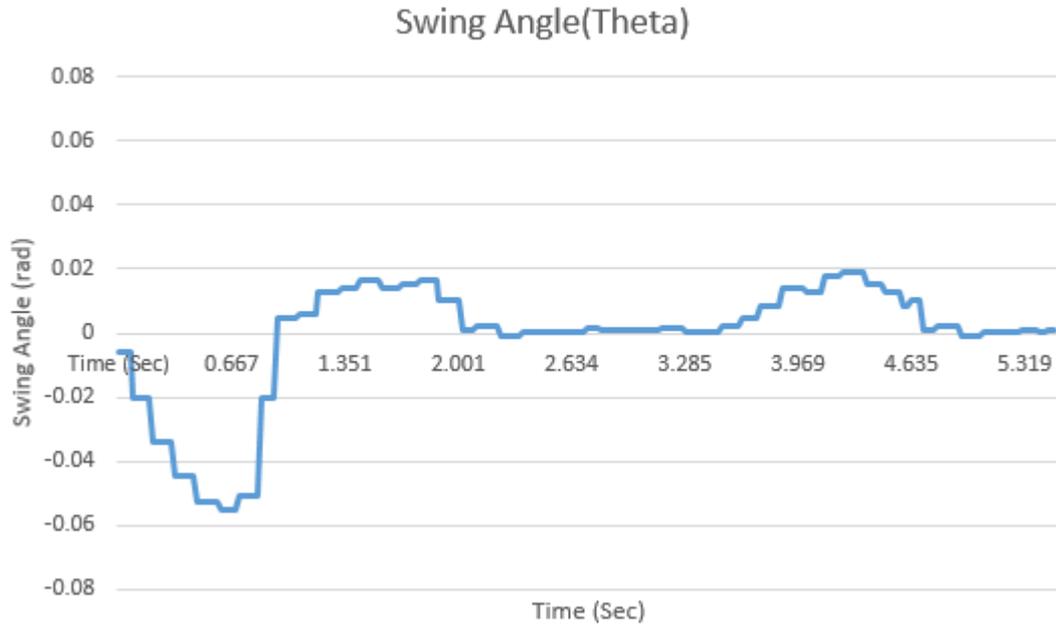


Figure 6-12 The swing angle “theta” response with no weight

The second scenario has been executed to evaluate the proposed technique with 1 kg Load weight. As shown in Figures 6-13 and 6-14, it can be seen that the proposed strategy can successfully control the position of the crane with high performance in both cases with weight and without weight.

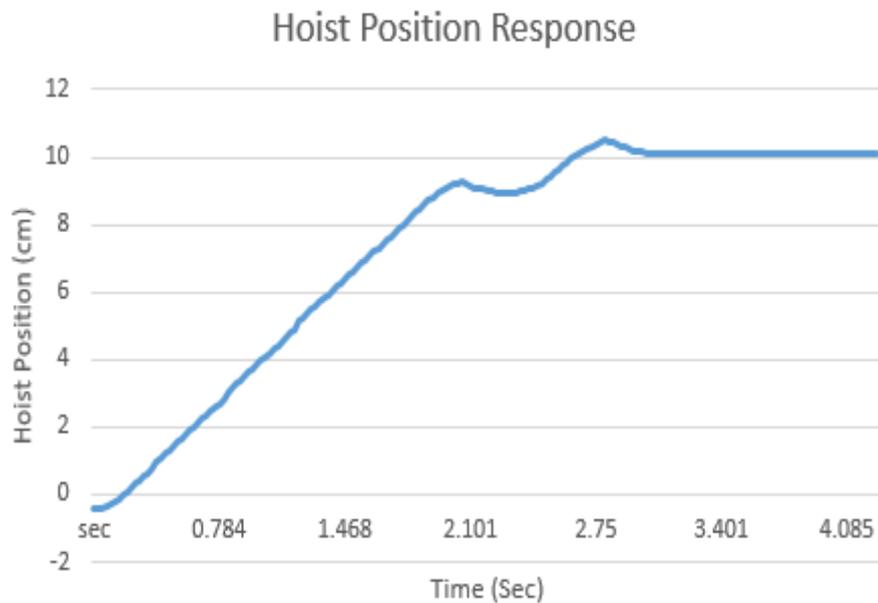


Figure 6-13 Position response of tower hoist with 1 kg weight

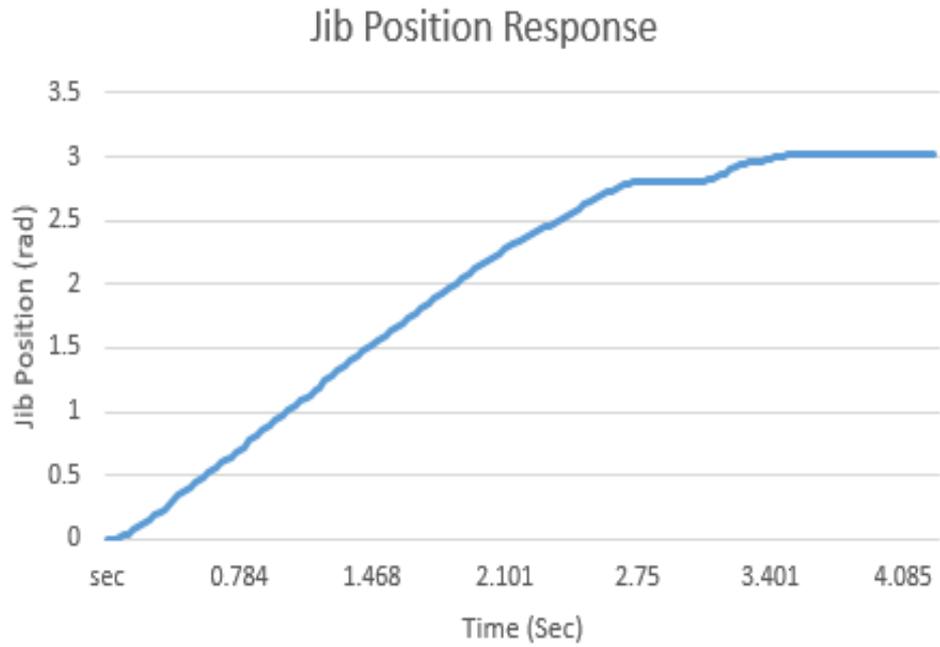


Figure 6-14 Position response of tower jib with 1 kg weight

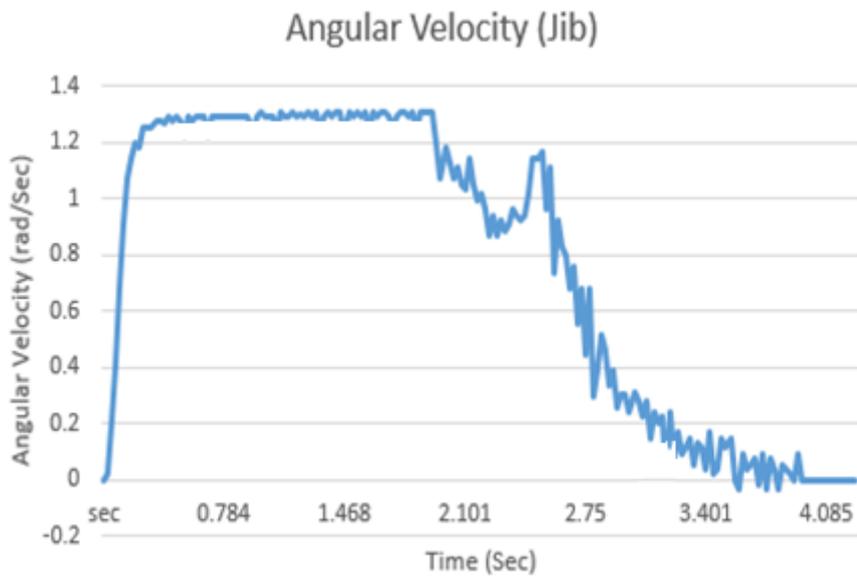


Figure 6-15 Velocity response of tower jib

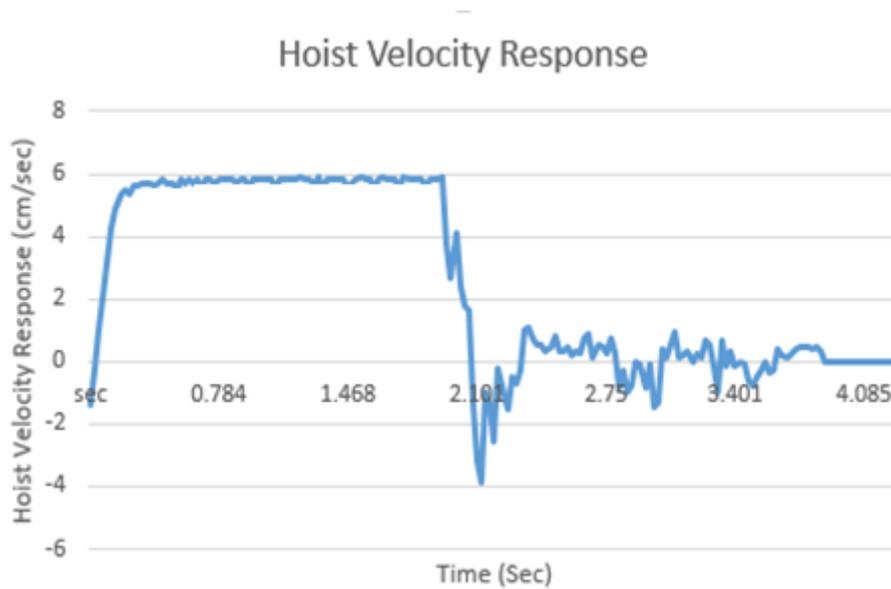


Figure 6-16 Velocity response of tower hoist

Figure 6-15 shows the velocity profile of the jib during the operation. It is accelerated up to around 1.2 rad/s and remains stable until the hoist reaches the final destination then goes down to 0.8 rad/s and finally goes to zero when it reached the demanded position. However, the velocity response of the hoist shown in Figure 6-16, it is accelerated up to around 6 cm/s and remain stable until the hoist reaches the final destination then goes down -4 cm/s and finally goes to zero when it reached the demanded position.

The third scenario has been carried out to evaluate the proposed technique. The load of the crane has been injected with a hit to mimic the disturbance of the wind during the crane operation.

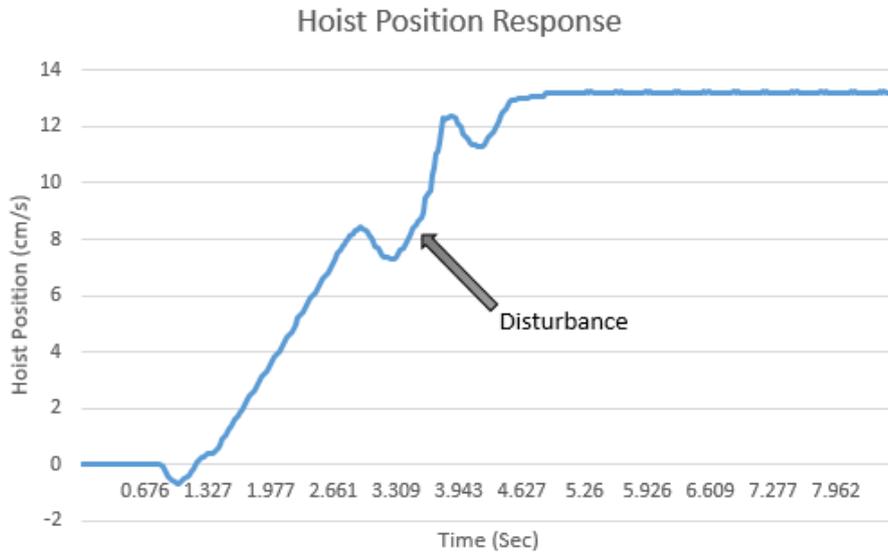


Figure 6-17 Position response of tower hoist with disturbance

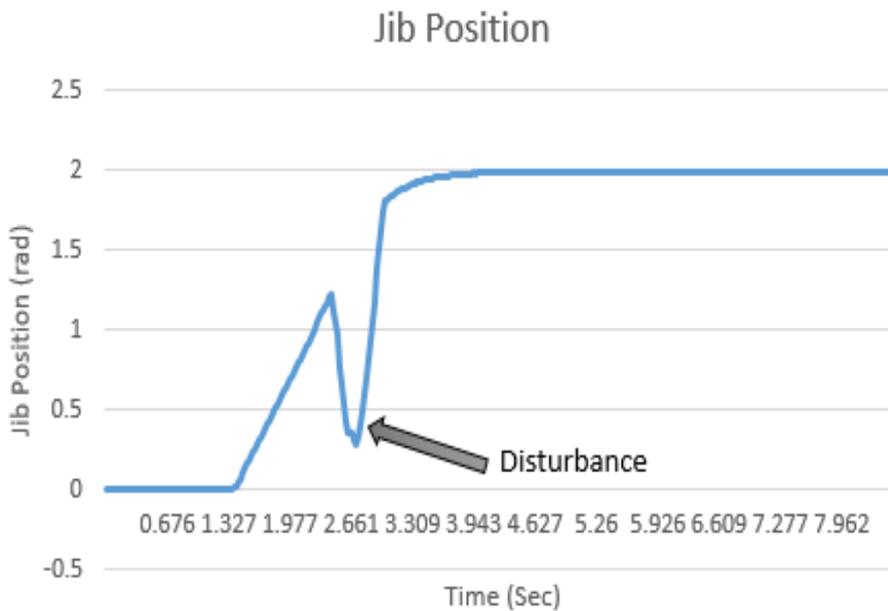


Figure 6-18 Position response of tower jib with disturbance

Figures 6-17 and 6-18 show the swing angle of the load when it interferes with external disturbance. An external disturbance injected to the load during the operation. However, the hoist and the jib suppressed the generated swing on the load quickly and moved to the final position with no swing.

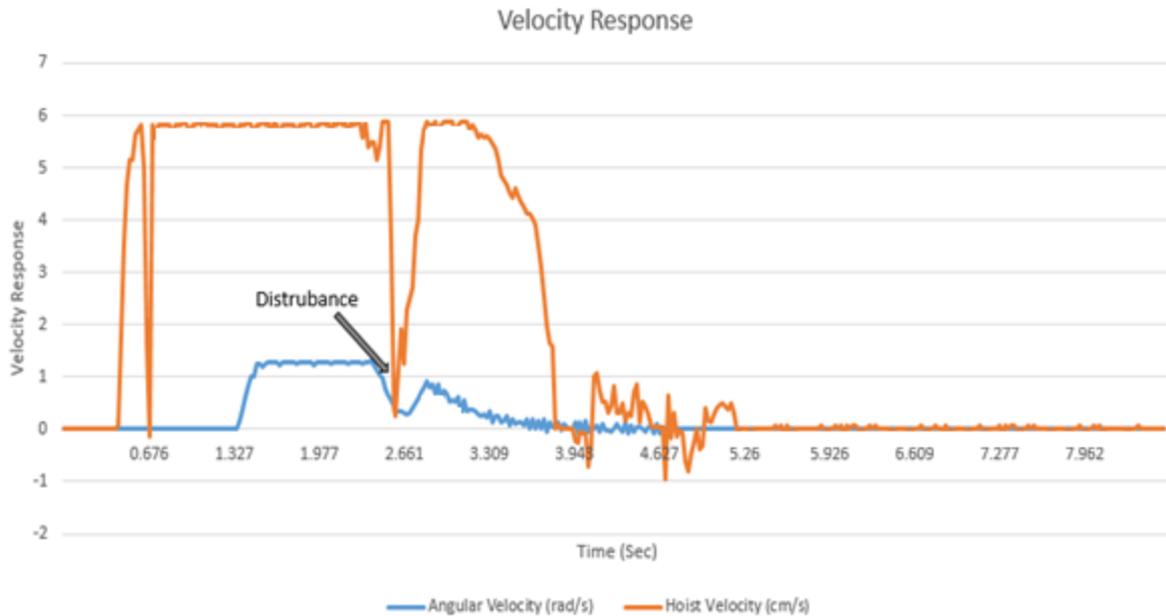


Figure 6-19 The velocity response of hoist and jib with disturbance

In addition, Figure 6-19 shows the velocity response of hoist and jib when the load injected with an external disturbance. The velocities of the hoist and jib have been decreased to compensate for the movement and eliminate the generated swing of the load.

6.7 Conclusion

A tower crane has been designed and fabricated in this chapter. The proposed practical control strategy has been elaborated and explained. A high-performance tower crane control operation requires that the load should be transported as fast as possible with high accuracy in load positioning with a minimum load swing as possible. Moreover, the control system design and settings should not be very complex and difficult to understand by the operator. Based on these objectives, the configuration of the proposed pragmatic control system was described. Several tests have been carried out to validate the proposed strategy. As mentioned previously, the three test scenarios under which the tower crane should be controlled are conducted with the addition of using external disturbance as well as with load masses. The proposed strategy showed a successful and robust performance in term of positioning the transported load and suppress any swing that might occur during the crane operation.

7 CHAPTER SEVEN: - CONCLUSION AND FUTURE WORKS

7.1 Introduction

The main objective of this work is to design robust, fast, and practical controllers for crane systems. The controllers are designed to transfer the load from point to point as fast as possible and, at the same time, the load swing is kept small during the transfer process until it reaches the demanded destination.

The work presented in this thesis describes the investigation of a pragmatic controller that can be implemented in real-time to control the crane system. A novel practical control strategy is proposed and analysed. Gantry and tower crane systems have been designed. The proposed strategy for a crane system has been successfully implemented in gantry and tower cranes.

To predict the behaviour of the systems, it is required to simulate the controller's performance before implementing the controller. To accomplish this, a mathematical model that realistically describes the behaviour of the crane system is needed. Both cranes have been modelled using the state-space model and simulated using JavaScript language and MATLAB software. The used methodology shows the simplicity in term on designing and implementing the control, and the results show the excellent performances can be achieved through the design.

An embedded microcontroller Arduino has been used in both systems due to its simplicity and is considered an excellent tool for developing interactive objects, taking inputs from a variety of switches or sensors and controlling motors and other outputs. Arduino systems can be stand-alone, or they can be connected to a computer using USB. The pragmatic strategy was devised for the crane's control, in which position sensing of the load was performed by a camera that observed a corner reflector mounted on the hook. The Experimental results demonstrated that the pragmatic technique could work in practice effectively.

7.2 System Models

The proposed strategy borrowed techniques from early autopilots in which loops were nested one inside the other. At each stage, demand is calculated for the next inward loop, this being limited in magnitude. This highly nonlinear multivariable controller is nevertheless easy to comprehend and tune. Gantry and tower crane were models mathematically using the state-space model. However, most previous researchers have been used Lagrange method to model those systems. The ‘pragmatic’ paradigm has its roots set in design practices of autopilots for automatic landing half a century ago. For the original strategy, these concepts were transferred to the problem of the crane as follows.

- From the difference between the load position and its target, a velocity demand is calculated, proportional to the distance. This is limited in magnitude to a safe travel velocity.
- From the difference between the velocity demand and the actual load velocity, a demanded load acceleration is calculated. This is also limited in magnitude representing a limit on the demanded cable angles.
- To achieve this demanded acceleration, an offset is demanded between the lateral hoist position and that of the load, finally giving signals that can be injected into the motor control loops as demand for motor velocity.

For the tower-crane simulation, the strategy was made easier to implement by working in coordinates that rotated with the jib angle and were centred on the hoist. As the jib rotated, the target coordinates were rotated and moved in the opposite sense. In this way, all calculations were aligned with the coordinates of the downward-looking camera. In the simulation, it was then necessary to apply transformations to the load velocity to keep true to the kinetics, but of course, this is not needed in the physical model.

The computer simulations show that the designed controllers in both gantry and tower cranes are successfully transferring the load to its final destination without residual swing. The controllers transfer the load smoothly without inducing an overshoot in the trolley position. As the target position is approached, the desired velocity falls, so the hoist ‘hangs back’ to decelerate the load and finally brings it to a halt; where once again it takes up a position that is immediately overhead. To validate the designed controllers, an experimental setup was built.

Although the designed controllers work correctly in the computer simulations, we have faced many problems which will be explained in the next section.

7.3 Hardware Model

The gantry crane consists of hoist moving along a girder in translation motion, and the girder moving in a horizontal plane. The trolley consists of a hoist system (rope and hook) to lift and lower the load. The combination of hoist and girder movement allows the load to reach the desired destination. The hoist was designed to enable translational movement along the girder. The hoist was powered by DC type electric motor. However, the power to the motor was relayed through a toothed belt. Two-timing belt pulleys were at the same size, relatively small and light and easy to manage. An added advantage to this design was the capability of the hoist to move along the girder without gear conversion. For the lowering and lifting load, another DC geared motor was placed on the girder beside the hoist motor. A U groove pulley was attached to it and works as robin to collect the cable during the operation. The cable of the hoist will be under this, passing over a pulley to descend to the hook, then up again to pass to an anchor point at the end of the girder, an equal distance above its floor.

For the tower crane apparatus, hardware construction started with the selection of three geared motors, each with a two-phase incremental encoder. One rotated the angle of the jib, and another controlled the radius of the hoist from which the load was suspended, and the third controlled the length of the suspension cable. Mounted on the hoist was a camera looking vertically downwards, locating the angular deviations from the vertical of a corner reflector mounted on the hook. Thus, the essential state variables could be measured of the positions of hoist and load, while their velocities could be estimated.

An Arduino was used to apply control to the motors and to monitor their encoders. This communicated via a USB serial link to a desktop machine which handled the vision processing and the overall strategy. The desktop machine also served to gather real-time data for plotting the responses.

In the process of making the model work, many problems and tasks had to be overcome. With rotary encoders giving 48 transitions per revolution of motors that could turn at over 6,000

rpm, a novel interrupt-based algorithm had to be devised to track motor position and estimate velocity. The communication structure and control topology had to overcome the latency of the serial communication. The camera reported frames at intervals that varied with light level so that the discrete-time control caused violent oscillation when the original strategy was applied. Attention then turned to the control of a simplified single-axis model. Recognising the dominance of the discrete-time aspects, a second simulation was constructed. Asynchronous parallel streams represented the events of frame arrival, Arduino message transfer and simulation of the dynamics. Only then was some resemblance seen between the performance of the simulation and that of the physical model. A simple strategy was found for damping the swing. The control structure centred on assigning motor velocity control solely to the Arduino. The velocity demand sent to it was the sum of two components. To stabilise and cancel the swing, the first component was proportional to the displacement of the load, as seen in the camera, then added the velocity demand computed from the error between the load position and its target, limited to a safe travel velocity.

The designed controllers showed that an efficient positioning control and a suppression for the swing angle of the crane while travelling to almost zero swing. In addition, the technique is demonstrated with the aid of a MATLAB and JavaScript state-space model. The simulated results, as well as the validation of the experiment, have shown an accurate position for the trolley along the jib to the demanded position with no swing of the load. The used strategy showed robust and efficient results in term of simplicity and reduced the load swing comparing with the previously published work.

7.4 Limitations of The Experimental Set-Up

7.4.1 The Problem of Quantization

In addition to the calculation of the parameters of the system, calibrations were made of the experimental equipment. There were 175 steps per centimetre movement of the hoist. At top motor speed, the hoist moved at 27 centimetres per second. The maximum hoist acceleration was measured from a sequence of counts that were reported at 20 ms intervals. The change in velocity was seen to be about one count per millisecond per 20 ms interval.

From this, the possibility of stable bang-bang control was inferred. At top speed, the encoder counts at some 5,000 counts per second. The software to read all three encoders has no more than 200 microseconds to avoid missing counts. That is not the only problem. At top speed, there will be just ten counts in each motor drive update period of 2 milliseconds. In the range up to the ‘travel speed’ of 10 cm/sec this is reduced to three or four levels at which velocity could be commanded. Increasing the interval between drive updates would increase the number of levels, but the delay would risk compromising stability. Any proportional control with mark-space drive of the motors would be ‘soft’, requiring a substantial speed error before the maximum drive is applied. In place of estimating speed as a simple difference in the count, the low-pass filter described below was applied. In the process of the calculation, the result is multiplied by a factor of eight, by shifting it three binary places to the left. In consequence, the quantization of the control demand is reduced to one per cent of full speed. Rather than applying mark-space pseudo-linear control, bang-bang control is applied to the motor drive, only being zero if the speed error is zero. The mark-space is now asynchronously determined by the motor velocity itself. The motor speed accurately responds to unit increments in demand, this ‘sliding motion’ eliminating errors due to friction or stiction. First, the encoder reading routine is described and explained below, then that of the low-pass filter.

7.4.2 Reading the Encoders

The two-phase encoders cycle through 00, 01, 11, 10 for a rotation in one direction and 00, 10, 11, 01 for the reverse. The increment or decrement of position can thus be determined by comparing the present inputs with the previous value. A four-bit composite is constructed by combining the present value with the old value shifted two places, thus 0000, 0101, 1111 and 1010 indicate that no move has occurred, 0001, 0111, 1110 and 1000 indicate a positive move while 0010, 1011, 1101 and 0100 indicate a negative one.

A table is defined with values of 0,1 or -1 corresponding to these combinations:

```
int move[16] = {0, 1, -1, 0, -1, 0, 0, 1, 1, 0, 0, -1, 0, -1, 1, 0};
```

If both bits have changed, however, there is an error that means that a transition has been missed. At 6000 rpm, 4800 transitions per second can occur on each of the three encoders, but if the interval between samples is substantially less than 150 microseconds the risk will be small. That sets a limit on the duration of the interrupt routine.

Although the hobbyist software of the Arduino presents individual bits to the user, byte-wide port operations are still performed. Three pairs of adjacent bits of PORTC are connected to the sensors.

7.4.3 Estimating Velocity

The simple process of subtracting the latest value from the previous one produces a rough quantized signal. In terms of the z transform this estimated velocity is

$$ve = (1 - z^{-1}) x / dt \quad (7.1)$$

where dt is the sample interval, x is the input and ve is the velocity to be estimated. At lower speeds, this will give a ‘steppy’ value. At a low speed, some intervals will contain a step and others none. The following code will generate a smoothed velocity. First, a lagged signal ‘ $xslow$ ’ is constructed:

$$xslow += (x - xslow) * k * dt \quad (7.2)$$

and the estimated velocity ve is then given in the loop by

$$ve = k * (x - xslow) \quad (7.3)$$

In z-transform terms,

$$xslow = k * dt * x / (z - 1 + k * dt) \quad (7.4)$$

Therefore

$$ve = k * \{(z - 1) / (z - 1 + k * dt)\} \quad (7.5)$$

which can be rewritten instead as

$$ve = k * \left\{ (1 - z^{-1}) / (1 - [1 - k * dt]z^{-1}) \right\} x \quad (7.6)$$

Clearly if $k = 1/dt$ this reduces to the original simple difference. For implementation within an Arduino the execution time can be shortened by avoiding floating point operations. The following code achieves this:

```
m1vel=m1pos-int(m1slow>>3); //counts per 2 ms
m1slow+=m1vel;
```

where this part of the loop is executed at 2 ms intervals and all variables are integers, except for `m1slow` which is long. This gives smoothing with an equivalent time-constant of 16 ms. With the factor imposed by the shift, it was found that full speed on the motor corresponded to a velocity demand of 220. With a top speed of 27 cm/second that means that the speed quantization is just over one millimetre per second.

7.4.4 Frame Rate and Timing

The location of the load is found as a displacement from the hoist by a simple search for a bright spot in the image. This then gives the angles of the suspension cable, to be multiplied by the cable length for position and velocity calculations. Originally the code was arranged to synchronise all calculations to each video frame at intervals of 30 ms. When the system failed to perform as expected, the problem was diagnosed as having been caused by the frame interval and its variability. In the laboratory environment, the frame rate fell below 10 fps. The code was therefore restructured so that other calculations were made at fixed time intervals, while the load offset was calculated whenever a frame became available. Several schemes were tried for the serial communication between the desktop machine and the Arduino. In the latest version, the Arduino reports the encoder counts at fixed 20 ms intervals and the desktop replies with a new speed demand.

7.5 Recommendations and Future Work

With the success of our experimental study, this study considers further work directions, including improvements to the proposed technique. In this section, I emphasise the possible extensions to the proposed technique, as discussed below.

- We designed our control algorithms to work with any fixed cable length. Most likely, they will perform adequately if load hoisting is included in the transfer manoeuvres. However, further simulations and experimental tests are needed to investigate the effect of the rate of change of the cable length on the controller performance.
- The camera was used to measure the swing motion of the payload. It was not accurate enough to get high resolution. A more accurate camera should be considered for better results.
- Mechanical improvement of the crane is suggested for better results. Due to the slip in the active joints, accurate results could not be obtained. The controller might have produced better results with a more mechanically sound crane.
- Collaboration with the manufacturer of such cranes might be a much easier way to demonstrate the effectiveness of the original strategy with practical time-constants and camera frame-rates.

REFERENCES

- ABE, A. 2011. Anti-sway control for overhead cranes using neural networks. *International Journal of Innovative Computing, Information and Control*, 7, 4251-4262.
- ADEY, A., COALES, J. & STILES, J. 1963. Predictive control of an on-off system with two control variables. *IFAC Proceedings Volumes*, 1, 41-46.
- AHMAD, M., MISRAN, F., RAMLI, M. & ISMAIL, R. R. Experimental investigations of low pass filter techniques for sway control of a Gantry crane system. 2010 2nd International Conference on Electronic Computer Technology, 2010. IEEE, 1-4.
- AHMAD, M. A., ISMAIL, R. M. T. R., RAMLI, M. S., NASIR, A. N. K. & HAMBALI, N. Feed-forward techniques for sway suppression in a double-pendulum-type overhead crane. 2009 International Conference on Computer Technology and Development, 2009. IEEE, 173-178.
- AL-MOUSA, A. A. 2000. *Control of rotary cranes using fuzzy logic and time-delayed position feedback control*. Virginia Tech.
- AL-MOUSA, A. A., NAYFEH, A. H. & KACHROO, P. 2003. Control of rotary cranes using fuzzy logic. *Shock and Vibration*, 10, 81-95.
- ALHASSAN, A., MOHAMED, Z., ABDULLAHI, A. M., BATURE, A. A., HARUNA, A. & TAHIR, N. M. 2018. Input shaping techniques for sway control of a rotary crane system. *Jurnal Teknologi*, 80.
- ALMUTAIRI, N. B. & ZRIBI, M. 2016. Fuzzy controllers for a gantry crane system with experimental verifications. *Mathematical Problems in Engineering*, 2016.
- ANNUAR, K. A. M., HADI, N. A. A., HARUN, M. H., HALIM, M. F. M. A., MIRIN, S. N. S. & AZAHAR, A. H. 2018. Dynamic modelling and analysis of 3D overhead gantry crane system. *International Journal of Engineering & Technology*, 7, 1257-1262.
- ARVIN, R., SOLIHIN, M., HELTHA, F., TAN, R. H. & AMMAR, A. Modeling and control design for rotary crane system using MATLAB Simscape Toolbox. 2014 IEEE 5th Control and System Graduate Research Colloquium, 2014. IEEE, 170-175.
- AVILA, J. R., ALCÁNTARA-RAMÍREZ, R., JAIMES-PONCE, J. & SILLER-ALCALÁ, I. Design of a self-balancing tower crane. Proceedings of the 2nd WSEAS International Conference on Circuits, Systems, Signal and Telecommunications, 2008. World Scientific and Engineering Academy and Society (WSEAS), 109-115.
- BARIŠA, T., BARTULOVIĆ, M., ŽUŽIĆ, G., ILEŠ, Š., MATUŠKO, J. & KOLONIĆ, F. Nonlinear predictive control of a tower crane using reference shaping approach. 2014 16th International Power Electronics and Motion Control Conference and Exposition, 2014. IEEE, 872-876.
- BILLINGSLEY, J. 2006. *Essentials of mechatronics*, Wiley Online Library.

- BILLINGSLEY, J. & GHUDE, S. 2015. Significant advance in logical predictive control. *Electronics Letters*, 51, 1240-1241.
- BÖCK, M. & KUGI, A. 2013. Real-time nonlinear model predictive path-following control of a laboratory tower crane. *IEEE Transactions on Control Systems Technology*, 22, 1461-1473.
- BREUNING, P. 2015. *Linear model predictive control of a 3D tower crane for educational use*.
- BRUINS, S. 2010. Comparison of different control algorithms for a gantry crane system. *Intelligent Control and Automation*, 1, 68.
- CAKAN, A. & ONEN, U. 2016. Position Regulation And Sway Control Of A Nonlinear Gantry Crane System. *International Journal of Scientific & Technology Research*, 5, 121-124.
- CHESTNUT, H. & WETMORE, V. 1959. Predictive control applied to a simple position control. *General Electric Engineering Laboratory, Report*.
- COALES, J. & NOTON, A. 1956. An on-off servo mechanism with predicted change-over. *Proceedings of the IEE-Part B: Radio and Electronic Engineering*, 103, 449-460.
- DIEP, D. & KHOA, V. 2014. PID-controllers tuning optimization with pso algorithm for nonlinear gantry crane system. *International Journal of Engineering and Computer Science*, 3.
- DODDS, S. 1984. A predicted signed switching time high precision satellite attitude control law. *International Journal of Control*, 39, 1051-1061.
- DUONG, S. C., UEZATO, E., KINJO, H. & YAMAMOTO, T. 2012. A hybrid evolutionary algorithm for recurrent neural network control of a three-dimensional tower crane. *Automation in Construction*, 23, 55-63.
- FALAHIAN, R., DASTJERDI, M. M. & GHARIBZADEH, S. Pragmatic modeling of chaotic dynamical systems through artificial neural network. 2014 21th Iranian Conference on Biomedical Engineering (ICBME), 2014. IEEE, 103-108.
- FATEHI, M. H., EGHTEHAD, M. & AMJADIFARD, R. 2014. Modelling and control of an overhead crane system with a flexible cable and large swing angle. *Journal of Low Frequency Noise, Vibration and Active Control*, 33, 395-409.
- GOLAFSHANI, A. R. 1999. Modeling and optimal control of tower crane motions.
- GRAICHEN, K., EGRETZBERGER, M. & KUGI, A. Suboptimal model predictive control of a laboratory crane. 8th IFAC Symposium on Nonlinear Control Systems, 2010. Citeseer, 397-402.
- GÜRLEYÜK, S. S., BAHADIR, Ö., TÜRKKAN, Y. & ÜŞENTİ, H. 2008. Improved three-step input shaping control of crane system. *constraints*, 6, 10.

- HE, W. & GE, S. S. 2016. Cooperative control of a nonuniform gantry crane with constrained tension. *Automatica*, 66, 146-154.
- HILHORST, G., PIPELEERS, G. & SWEVERS, J. Reduced-order multi-objective \mathcal{H}_∞ control of an overhead crane test setup. 52nd IEEE Conference on Decision and Control, 2013. IEEE, 770-775.
- HO, D. T., NGUYEN, H. & NGUYEN, Q. C. Input shaping control of an overhead crane. 7th Vietnam Conf on Mechatronics (VCM 2014), 2014. 303-311.
- HUA, Y. J. & SHING, Y. K. Adaptive coupling control for overhead crane systems. 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005., 2005. IEEE, 6 pp.
- HUANG, B. 2003. A pragmatic approach towards assessment of control loop performance. *International Journal of Adaptive Control and Signal Processing*, 17, 589-608.
- HUANG, J., LIANG, Z. & ZANG, Q. 2015. Dynamics and swing control of double-pendulum bridge cranes with distributed-mass beams. *Mechanical Systems and Signal Processing*, 54, 357-366.
- HUANG, J., XIE, X. & LIANG, Z. 2014. Control of bridge cranes with distributed-mass payload dynamics. *IEEE/ASME Transactions on Mechatronics*, 20, 481-486.
- HUH, C.-D. & HONG, K.-S. 2002. Input shaping control of container crane systems: limiting the transient sway angle. *IFAC Proceedings Volumes*, 35, 445-450.
- HUSSIEN, S., GHAZALI, R., JAAFAR, H. & SOON, C. Robustness analysis for PID controller optimized using PFPSO for underactuated gantry crane system. 2015 IEEE International Conference on Control System, Computing and Engineering (ICCSC), 2015. IEEE, 520-525.
- JAAFAR, H. I., MOHAMED, Z., JAMIAN, J. J., ABIDIN, Z., FAIZ, A., ANUAR, M. K. & AB GHANI, Z. 2013. Dynamic behaviour of a nonlinear gantry crane system. *Procedia Technology*, 11, 419-425.
- JALANI, J. Robust fuzzy logic controller for an intelligent gantry crane system. First International Conference on Industrial and Information Systems, 2006. IEEE, 497-502.
- JIAN, T. Y. & MOHAMED, Z. 2017. Modelling and sway control of a double-pendulum overhead crane system. *Applications of Modelling and Simulation*, 1, 15-21.
- JU, F., CHOO, Y. & CUI, F. 2006. Dynamic response of tower crane induced by the pendulum motion of the payload. *International Journal of Solids and Structures*, 43, 376-389.
- KARKOUB, M. A. & ZRIBI, M. 2001. Robust control schemes for an overhead crane. *Journal of Vibration and Control*, 7, 395-416.

- KAUR, A., PRIYAHANSHA, S. K. & SINGH, T. 2014. Position control of overhead cranes using fuzzy controller. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 3, 9341-9350.
- KIM, D., PARK, Y., PARK, Y.-S., KWON, S. & KIM, E. Dual stage trolley control system for anti-swing control of mobile harbor crane. 2011 11th International Conference on Control, Automation and Systems, 2011. IEEE, 420-423.
- KIM, D. & SINGHOSE, W. Reduction of double-pendulum bridge crane oscillations. The 8th International Conference on Motion and Vibration Control, 2006. 300-305.
- KIM, D. & SINGHOSE, W. 2010. Performance studies of human operators driving double-pendulum bridge cranes. *Control Engineering Practice*, 18, 567-576.
- LEE, S.-G. 2013. Sliding mode controls of double-pendulum crane systems. *Journal of Mechanical Science and Technology*, 27, 1863-1873.
- LEW, J. Y. & HALDER, B. Experimental study of anti-swing crane control for a varying load. Proceedings of the 2003 American Control Conference, 2003., 2003. IEEE, 1434-1439.
- LEW, J. Y. & KHALIL, A. Anti-swing control of a suspended load with a robotic crane. Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334), 2000. IEEE, 1042-1046.
- LIU, D.-T., GUO, W.-P. & YI, J.-Q. Composite sliding mode fuzzy control for double-pendulum-type overhead crane. 2005 International Conference on Machine Learning and Cybernetics, 2005. IEEE, 767-772.
- LIU, D.-T., GUO, W.-P., YI, J.-Q. & ZHAO, D.-B. Double-pendulum-type overhead crane dynamics and its adaptive sliding mode fuzzy control. Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), 2004. IEEE, 423-428.
- LUMKES JR, J. H. 2001. *Control strategies for dynamic systems: design and implementation*, CRC Press.
- MAGHSOUDI, M. J., MOHAMED, Z., HUSAIN, A. R. & TOKHI, M. O. 2016. An optimal performance control scheme for a 3D crane. *Mechanical Systems and Signal Processing*, 66, 756-768.
- MAJID, M. A., IBRAHIM, W. S. W., MOHAMAD, S. & BAKAR, Z. A. A comparison of PID and PD controller with input shaping technique for 3D gantry crane. 2013 IEEE Conference on Systems, Process & Control (ICSPC), 2013. IEEE, 144-148.
- MASOUD, Z., ALHAZZA, K., ABU-NADA, E. & MAJEED, M. 2014. A hybrid command-shaper for double-pendulum overhead cranes. *Journal of Vibration and Control*, 20, 24-37.

- MASOUD, Z. N. & DAQAQ, M. F. 2006. A graphical approach to input-shaping control design for container cranes with hoist. *IEEE Transactions on Control Systems Technology*, 14, 1070-1077.
- MATUŠKO, J., ILEŠ, Š., KOLONIĆ, F. & LEŠIĆ, V. 2015. Control of 3D tower crane based on tensor product model transformation with neural friction compensation. *Asian Journal of Control*, 17, 443-458.
- NGUYEN, Q. C. & KIM, W.-H. Nonlinear adaptive control of a 3d overhead crane. 2015 15th International Conference on Control, Automation and Systems (ICCAS), 2015. IEEE, 41-47.
- NGUYEN, T. H., KHANH, T. G., THANH, N. T., DUONG, B. T. & MINH, P. X. 2017. Anti-sway tracking control of overhead crane system based on pid and fuzzy sliding mode control. *Vietnam Journal of Science and Technology*, 55, 116.
- OMAR, H. M. 2003. *Control of gantry and tower cranes*. Virginia Tech.
- OMAR, H. M. & NAYFEH, A. H. A simple adaptive feedback controller for tower cranes. ASME 2001 Design Engineering Technical Conference and Computers and Information in Engineering Conference, Pittsburgh, PA, September, 2001. 9-12.
- OMAR, H. M. & NAYFEH, A. H. 2003. Gain scheduling feedback control for tower cranes. *Journal of Vibration and Control*, 9, 399-418.
- PARK, B.-J., HONG, K.-S. & HUH, C.-D. Time-efficient input shaping control of container crane systems. Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No. 00CH37162), 2000. IEEE, 80-85.
- PARK, H., CHWA, D. & HONG, K.-S. 2007. A feedback linearization control of container cranes: Varying rope length. *International Journal of Control, Automation, and Systems*, 5, 379-387.
- PARK, M.-S., CHWA, D. & HONG, S.-K. 2008. Antisway tracking control of overhead cranes with system uncertainty and actuator nonlinearity using an adaptive fuzzy sliding-mode control. *IEEE Transactions on Industrial Electronics*, 55, 3972-3984.
- PAULUK, M., KORYTOWSKI, A., TURNAU, A. & SZYMKAT, M. Time optimal control of 3d crane. Proceedings of the 7th Inter. Conference on Methods and Models in Automation and Robotics, 2001. 927-936.
- PERIG, A. V., STADNIK, A. N. & DERIGLAZOV, A. I. 2014. Spherical pendulum small oscillations for slewing crane motion. *The Scientific World Journal*, 2014.
- QIAN, D., TONG, S. & LEE, S. 2016. Fuzzy-logic-based control of payloads subjected to double-pendulum motion in overhead cranes. *Automation in Construction*, 65, 133-143.

- RAMLI, L., MOHAMED, Z., ABDULLAHI, A. M., JAAFAR, H. & LAZIM, I. M. 2017. Control strategies for crane systems: A comprehensive review. *Mechanical Systems and Signal Processing*, 95, 1-23.
- RAUH, A., PRABEL, R. & ASCHEMANN, H. Oscillation attenuation for crane payloads by controlling the rope length using extended linearization techniques. 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), 2017. IEEE, 307-312.
- RUBIO-AVILA, J., ALCANTARA-RAMIREZ, R., JAIME-PONCE, J. & SILLER-ALCALÁ, I. 2007. Design, construction, and control of a novel tower crane. *International journal of mathematics and computers in simulation*, 1.
- SADATI, N. & HOOSHMAND, A. Design of a gain-scheduling anti-swing controller for tower cranes using fuzzy clustering techniques. 2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), 2006. IEEE, 172-172.
- SAMIN, R. E., MOHAMED, Z., JALANI, J. & GHAZALI, R. A hybrid controller for control of a 3-DOF rotary crane system. 2013 1st International Conference on Artificial Intelligence, Modelling and Simulation, 2013a. IEEE, 190-195.
- SAMIN, R. E., MOHAMED, Z., JALANI, J. & GHAZALI, R. Input shaping techniques for anti-sway control of a 3-DOF rotary crane system. 2013 1st International Conference on Artificial Intelligence, Modelling and Simulation, 2013b. IEEE, 184-189.
- SANTHI, L. & LB, M. 2014. Position control and anti-swing control of overhead crane using LQR. *International Journal of Scientific Engineering and Research*, 3, 26-30.
- SANTHI, L. R., CAMPUS, T. & VISHU, V. 2018. Control of position and swing angle of overhead crane using different controllers.
- SHEBEL, A., MAAZOUZ, S., MOHAMMED ABU, Z., MOHAMMAD, A. & AYMAN AL, R. 2011. Design of fuzzy PD-controlled overhead crane system with anti-swing compensation. *Engineering*, 2011.
- SINGHOSE, W. & KIM, D. Manipulation with tower cranes exhibiting double-pendulum oscillations. Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007. IEEE, 4550-4555.
- SINGHOSE, W., PORTER, L., KENISON, M. & KRIKKU, E. 2000. Effects of hoisting on the input shaping control of gantry cranes. *Control engineering practice*, 8, 1159-1165.
- SMOCZEK, J. & SZPYTKO, J. 2008. A mechatronics approach in intelligent control systems of the overhead traveling cranes prototyping. *Information Technology and Control*, 37.

- SMOCZEK, J. & SZPYTKO, J. 2014. Evolutionary algorithm-based design of a fuzzy TBF predictive model and TSK fuzzy anti-sway crane control system. *Engineering Applications of Artificial Intelligence*, 28, 190-200.
- SOLIHIN, M. I., KAMAL, M. & LEGOWO, A. Optimal PID controller tuning of automatic gantry crane using PSO algorithm. 2008 5th International Symposium on Mechatronics and Its Applications, 2008. IEEE, 1-5.
- SOLIHIN, M. I., LEGOWO, A. & AKMELIAWATI, R. Robust PID anti-swing control of automatic gantry crane based on Kharitonov's stability. 2009 4th IEEE Conference on Industrial Electronics and Applications, 2009. IEEE, 275-280.
- SUN, N., FANG, Y. & CHEN, H. 2014a. A new antishwing control method for underactuated cranes with unmodeled uncertainties: Theoretical design and hardware experiments. *IEEE Transactions on Industrial Electronics*, 62, 453-465.
- SUN, N., FANG, Y., CHEN, H. & HE, B. 2014b. Adaptive nonlinear crane control with load hoisting/lowering and unknown parameters: Design and experiments. *IEEE/ASME Transactions on Mechatronics*, 20, 2107-2119.
- SUN, N., FANG, Y., CHEN, H. & LU, B. 2016a. Amplitude-saturated nonlinear output feedback antishwing control for underactuated cranes with double-pendulum cargo dynamics. *IEEE Transactions on Industrial Electronics*, 64, 2135-2146.
- SUN, N., FANG, Y., CHEN, H., LU, B. & FU, Y. 2016b. Slew/translation positioning and swing suppression for 4-DOF tower cranes with parametric uncertainties: Design and hardware experimentation. *IEEE Transactions on Industrial Electronics*, 63, 6407-6418.
- SUN, N., FANG, Y., ZHANG, Y. & MA, B. 2011. A novel kinematic coupling-based trajectory planning method for overhead cranes. *IEEE/ASME Transactions on Mechatronics*, 17, 166-173.
- TANG, R. & HUANG, J. 2016. Control of bridge cranes with distributed-mass payloads under windy conditions. *Mechanical Systems and Signal Processing*, 72, 409-419.
- TINKIR, M., ÖNEN, Ü., KALYONCU, M. & ŞAH, Y. Modeling and control of scaled a tower crane system. 2011 3rd International Conference on Computer Research and Development, 2011. IEEE, 93-98.
- TUMARI, M. M., SAEALAL, M. S., GHAZALI, M. R. & WAHAB, Y. A. H_∞ controller with graphical LMI region profile for Gantry Crane System. The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems, 2012. IEEE, 1397-1402.
- TUMARI, M. M., SHABUDIN, L., DAUD, M. & ZAWAWI, M. 2013. Experimental Investigation on Active Sway Control of a Gantry Crane System using PID Controller.

- VAUGHAN, J., KIM, D. & SINGHOSE, W. 2010. Control of tower cranes with double-pendulum payload dynamics. *IEEE Transactions on Control Systems Technology*, 18, 1345-1358.
- VAUGHAN, J. E. 2008. *Dynamics and control of mobile cranes*. Georgia Institute of Technology.
- VUKOSAVLJEV, M. & BROUCKE, M. E. Control of a gantry crane: a reach control approach. 53rd IEEE Conference on Decision and Control, 2014. IEEE, 3609-3614.
- VUKOV, M., VAN LOOCK, W., HOUSKA, B., FERREAU, H. J., SWEVERS, J. & DIEHL, M. Experimental validation of nonlinear MPC on an overhead crane using automatic code generation. 2012 American Control Conference (ACC), 2012. IEEE, 6264-6269.
- WAHYUDI, M. I. S. 2007. Sensorless anti-swing control for automatic gantry crane system: Model-based approach. *International Journal of Applied Engineering Research*, 2, 147-161.
- WEIPING, G., DIANTONG, L., JIANQIANG, Y. & DONGBIN, Z. Passivity-based-control for double-pendulum-type overhead cranes. 2004 IEEE Region 10 Conference TENCN 2004., 2004. IEEE, 546-549.
- WIN, T. M. 2016. *Robotic Tower Crane Modelling Control (RTCMC)*. The University of New South Wales.
- WU, T.-S., KARKOUB, M., YU, W.-S., CHEN, C.-T., HER, M.-G. & WU, K.-W. 2016. Anti-sway tracking control of tower cranes with delayed uncertainty using a robust adaptive fuzzy control. *Fuzzy Sets and Systems*, 290, 118-137.
- WU, Z., XIA, X. & ZHU, B. 2015. Model predictive control for improving operational efficiency of overhead cranes. *Nonlinear Dynamics*, 79, 2639-2657.
- YANG, J. H. & YANG, K. S. 2007. Adaptive coupling control for overhead crane systems. *Mechatronics*, 17, 143-152.
- YANYANG, L., WEI, X. & LI, W. Anti-swing control of the crane system based on input shaping technique. 2011 Chinese Control and Decision Conference (CCDC), 2011. IEEE, 2788-2791.
- YU, X., LIN, X. & LAN, W. 2014. Composite nonlinear feedback controller design for an overhead crane servo system. *Transactions of the Institute of Measurement and Control*, 36, 662-672.
- ZHANG, M., MA, X., RONG, X., TIAN, X. & LI, Y. 2016. Adaptive tracking control for double-pendulum overhead cranes subject to tracking error limitation, parametric uncertainties and external disturbances. *Mechanical Systems and Signal Processing*, 76, 15-32.

ZHANG, M., MA, X., RONG, X., TIAN, X. & LI, Y. 2017. Error tracking control for underactuated overhead cranes against arbitrary initial payload swing angles. *Mechanical Systems and Signal Processing*, 84, 268-285.