CONCEPTUAL DESIGN AND SIMULATION OF AN INVERTED PENDULUM WHEELCHAIR

MUHAMMAD FAJAR

A Thesis submitted in partial fulfilment of the
requirement of Liverpool John Moores University
for the degree of Master of Philosophy

December 2013

Abstract

The need to develop advanced wheelchairs especially for improving better mobility and comfort to help disabled people has led to an investigation into spherical inverted pendulum wheelchair concept. The wheelchair's concept is based around the dynamic and control of spherical inverted pendulum.

The investigation starts with the stabilisation of the inverted pendulum using some various control strategies and command tracking capabilities also evaluated. Several different type control strategies are evaluated. These include (1) pole placement (2) PID and (3) Linear Quadratic Regulator (LQR). The stabilisation and tracking command performance of each control strategy is examined through simulation. The result shows all these three control strategies are capable to control the inverted pendulum system. But as the spherical inverted pendulum system is a MIMO system which has eight states and two inputs the LQR control strategy is more convenient to use for controlling the spherical inverted pendulum wheelchair.

The dynamic equation for the spherical inverted pendulum wheelchair is presented and the modelling in SimMechanics also developed. The model is controlled by feedback control using LQR. The simulation shows that the body which represents the chair and the occupant is balanced on a spherical ball through four rollers (two driven and two idler) successfully. Thus the new wheelchair concept based upon the dynamic and control of spherical inverted pendulum has the potential to offer improved mobility compared with existing wheelchairs in the market place.

Acknowledgements

I would like to thank my supervisor, Dr Steve Douglas, for his patience, guidance and support. I would also like to thank Dr James Barry Gomm for his support and help. Without their help the completion of this thesis would not have been possible. Besides the excellent supervision that they provided during the thesis project, they very encouraged and supported me throughout the long and onerous period of the project.

I would like to thank all those members of staff at the Liverpool John Moores University, who help provide software package and facilities to enable this project.

Finally, I would also like to thank my mom, wife and my two wonderful kids for their encouragement over recent years to complete the project.

Table of Contents

List of Figures

Symbols

b	Damper constant of cart friction ($N.s/m$)
E	motor voltage ($volts$)
e	Tracking error
F	Force (N)
F_h	Force on horizontal axis (N)
F_n	Normal Force (N)
F_x	Force matrix
F_v	Force on vertical axis (N)
g	Gravity (m/s^2)
I	Moment of Inertia ($kg.m^2$)
I_{Bx}	Inertia Moment of the body about x-axis
I_{Bz}	Inertia Moment of the body about z-axis
I_b	Inertia Moment of the ball ($kg.m^2$)
I_{Mx}	Inertia Moment of the roller/motor in x-axis ($kg.m^2$)
I_{Mz}	Inertia Moment of the roller/motor in z-axis ($kg.m^2$)
I_x	Moment of Inertia about x axis ($kg.m^2$)
I_y	Moment of Inertia about y axis ($kg.m^2$)
I_z	Moment of Inertia about z axis ($kg.m^2$)
J	Cost function
K	Feedback gain matrix
K_b	motor back emf constant ($volts/(rad.s)$)
K_d	Derivative gain

K_i	Integral gain
K_p	Proportional gain
K_t	motor torque constant (*Kg.m/amp*)
k	Spring constant of cart friction (*N/m*)
L	Lagrangian
l	High of pendulum CG (*m*)
M	Mass of cart (*Kg*)
M_B	Mass of the body (*Kg*)
M_b	Mass of the ball (*Kg*)
M_c	Controllability matrix
M_x	Mass matrix
M_v	Mass of the top cart (*Kg*)
m	Mass of pendulum (*Kg*)
N	Number of the state
n	Gear ratio
p_i	The i^{th} pole of the system
Q	State variable weighting matrix
q	Angle matrix of the SIPW
$\dot{q}$	Angular velocity matrix of the SIPW
$\ddot{q}$	Angular acceleration matrix of the SIPW
R	Input weighting matrix
R_b	Radius of the ball (*m*)
R_m	motor resistance (*ohms*)

R_r	Radius of the roller (*m*)
R_x	Control input weighting matrix/Rest (corriolis and gravity) matrix
r	Input signal reference
T_{lkB}	Linear kinetic energy of the body (*Joule*)
T_{lkb}	Linear kinetic energy of the ball (*Joule*)
T_{rkB}	Rotational kinetic energy of the body (*Joule*)
T_{rkb}	Rotational kinetic energy of the ball (*Joule*)
T_{rkm}	Rotational kinetic energy of the roller/motor (*Joule*)
u	Control input
u_x	Control input in *x*-*y* plane
u_z	Control input in *z*-*y* plane
V_B	Potential energy of the body (*Joule*)
V_b	Potential energy of the ball (*Joule*)
v_b	Linear velocity of the ball (m/s^2)
x	Displacement in *x* axis (*m*)
$\dot{x}$	Velocity in *x* direction (*m/s*)
$\ddot{x}$	Acceleration in *x* direction (m/s^2)
x_b	Ball displacement in *x* axis (*m*)
$\dot{x}_b$	Linear velocity of ball in *x* direction (*m/s*)
x_B	Body displacement in *x* direction (*m*)
$\dot{x}_B$	Linear velocity of body in *x* direction (*m/s*)
y_B	Body displacement in *y* direction (*m*)
$\dot{y}_B$	Linear velocity of body in *y* direction (*m/s*)

z	Displacement in z axis (m)
$\dot{z}$	Velocity in z direction (m/s)
$\ddot{z}$	Acceleration in z direction (m/s^2)
θ	Angle between pendulum/body and vertical axis (rad)
$\dot{\theta}$	Angular velocity of pendulum/body (rad/s)
$\ddot{\theta}$	Angular acceleration of pendulum/body (rad/s^2)
θ_x	Angle of pendulum/body in x-y plane (rad)
$\dot{\theta}_x$	Angular velocity of pendulum/body about x-y plane (rad/s)
$\ddot{\theta}_x$	Angular acceleration of pendulum/body in x-y plane (rad/s^2)
θ_z	Angle of pendulum/body in z-y plane (rad)
$\dot{\theta}_z$	Angular velocity of pendulum/body about z-y plane (rad/s)
$\ddot{\theta}_z$	Angular acceleration of pendulum/body in z-y plane (rad/s^2)
μ	Coefficient of friction
μ_{Bb}	Friction coefficient between body and ball
μ_{bg}	Friction coefficient between ball and ground
ξ	Integral of the tracking error
φ_x	Angle of roller/motor respect to pendulum/body position in x-y plane (rad)
$\dot{\varphi}_x$	Angular velocity of roller/motor about x-y plane (rad/s)
$\ddot{\varphi}_x$	Angular acceleration of roller/motor in x-y plane (rad/s^2)
φ_z	Angle of roller/motor respect to pendulum/body position in z-y plane (rad)

$\dot{\varphi}_z$	Angular velocity of roller/motor about *z-y* plane (*rad/s*)
$\ddot{\varphi}_z$	Angular acceleration of roller/motor in *z-y* plane (rad/s^2)
τ_{Bbx}	Viscous friction body and ball in *x-y* plane
τ_{bgx}	Viscous friction between ball and ground in *x-y* plane
τ_{Bbz}	Viscous friction body and ball in *z-y* plane
τ_{bgz}	Viscous friction between ball and ground in *z-y* plane
τ_{mx}	Motor torque in *x-y* plane (*Nm*)
τ_{mz}	Motor torque in *z-y* plane (*Nm*)
ω_b	Angular velocity of the ball (rad/s^2)

Abbreviations

CAD	Computer Aided Design
CG	Centre of Gravity
CS	Coordinate System
CSSL	Continuous System Simulation Language
DAEs	Differential Algebraic Equations
DoF	Degree of Freedom
EPW	Electric Powered Wheelchair
LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
MATLAB	Matrix Laboratory
MIMO	Multi Input Multi Output
ODE	Ordinary Differential Equations
PADO	Pitch Angle Disturbance Observer
PID	Proportional Integral Derivative
SIPW	Spherical Inverted Pendulum Wheelchair
SISO	Single Input Single Output
SS	State Space
SST	Sitting Standing Transporter
XML	Extensive Markup Language
3D	3 Dimensions

CHAPTER 1

INTRODUCTION

1.1. Background

Robotics is a branch of engineering science that deals with design, modelling, control and utilisation of robots. Robots are widely used for assisting human life for example to automate mass production and to a lesser extent to provide assistance to help human in everyday life activities, etc.

Robotics technologies have the potential to improve lifestyle of people suffering from some forms of incapacity or individuals with disabilities by restoring their functional abilities that have been reduced or even lost. Mobility is one such ability that can be restored by utilising a wheelchair (Simpson, 2005). A wheelchair is a mechanical device that uses wheels and mechanical support to overcome or to provide mobility for the elderly, disabled and patients. Wheelchair technologies have been developed as mobile robots to create modern electric powered wheelchairs by researchers in recent decades (Pei et al., 2007). These endeavours are aimed to create smart wheelchairs with advanced abilities to accomplish more complex tasks like sense information and respond in useful ways based upon traditional wheelchair mechanical configurations.

However, there are still many challenges that may be overcome by applying novel technologies to develop advanced wheelchairs especially for

improving adequately mobility and comfort for many certain users. Improvements in technologies on advanced or smart wheelchairs are needed to accommodate some people with disabilities and they can expand their ability to drive independently and to enhance driving safety (Ju et al., 2009).

In this research the conceptual design and simulation of a new wheelchair concept with novel motion capabilities is studied. The crux of this concept is to help disabled people by providing a wheelchair type transport to provide significantly better mobility. The wheelchair's concept is based around the dynamic and control of spherical inverted pendulum. This design arrangement consists of a ball acting as spherical wheel for the wheelchair, this arrangement enables the wheel chair to translate in any direction and rotate. The wheelchair is balanced by driving the ball, using four rollers, such that the wheelchair does not fall over but remain stables and moves via spherical inverted pendulum control concept to the author’s knowledge this arrangement has not been investigated before.

1.2. Aims and Objectives of the Research

In the introduction background it was explained that need to be investigated to create a new wheelchair with improvement capabilities in motion to help the disable people to improve their mobility and maneuverability in everyday life environment.

Thus the aim of this research is to create a conceptual design and to

simulate a new wheelchair that can provide improvement in mobility and maneuverability. The wheelchair concept is based upon the control of spherical inverted pendulum control. This concept has the potential to offer improved mobility and maneuverability compared with existing wheelchairs because in this concept the need of caster wheels to balance existing wheelchair can be eliminated. Without casters, wheelchair is able to moves in any direction thus the improvement in mobility can be achieved.

The technical challenge of the work is such that in this research is focused solely on the simulation techniques of the new wheelchair based on spherical inverted pendulum control. Under this aim, the following objectives are proposed to be achieved in this project:

- To investigate various controls for the inverted pendulum system can be applied such as pole placement, PID and linear quadratic regulator (LQR).
- To develop an alternative inverted pendulum design that consists of two carts (bottom cart (vehicle), top cart with inverted pendulum) as it design gives possibility to control the bottom cart (vehicle) to get better characteristic on set point tracking while another cart moving on top to stabilise the inverted pendulum.
- To create design and simulation of a spherical inverted pendulum wheelchair that is designed and simulated in SimMechanics with state feedback control using LQR control strategy.

The development of this a new wheelchair concept based upon spherical inverted pendulum will make contributions to the inverted pendulum application

knowledge base including:

- Developing simulation in stabilisation of the inverted pendulum system using various control strategies in both Matlab Simulink and SimMechanics.
- Proposing the new alternative inverted pendulum design construction with two carts (bottom cart (vehicle), top cart with inverted pendulum) for improving characteristic on the command tracking.
- Modelling and controlling a new concept wheelchair, the spherical inverted pendulum wheelchair (SIPW), for improving mobility to help wheelchair user.

1.3. Thesis Organisation

The thesis proposes conceptual design and simulation of a spherical inverted pendulum wheelchair (SIPW). It consists of nine chapters and is organised as follows. After an introduction in this Chapter the wheelchair history, inverted pendulum applications and the simulation techniques are presented in Chapter 2. The history of the wheelchair from the beginning until now days is described briefly, then the application of the inverted pendulum especially in the robotic fields in term of improving the motion capability of the wheelchair also described and the simulation techniques that can be used for dynamics system simulation also discussed briefly. Chapter 3 provides the mathematical modelling of the two DoF inverted pendulum. The dynamics of this inverted pendulum can

be presented in the form of transfer function equation and in the form of state space model. This mathematic model then is used to design the control strategy for stabilising the inverted pendulum. Chapter 4 presents the simulation of the two DoF inverted pendulum using SimMechanics and Simulink modelling with some various control strategies which are based on the mathematical modelling developed in the previous chapter. This work is to investigate and to demonstrate some various control methods of two DoF inverted pendulum system can be applied. Chapter 5 proposes a new an alternative inverted pendulum design. The alternative inverted pendulum design that consists of two carts which gives possibility to get better characteristic on set point tracking is simulated and discussed. Chapter 6 presents how to simulate the spherical inverted pendulum, a dynamics of multibody system, in slide mode which is designed with SimMechanics. Moreover this chapter also shows the advantages of the use SimMechanics for multibody dynamics simulation, especially for the non-experts. Chapter 7 describes the mathematical modelling development, the control design and the SimMechaincs modelling for the new proposed wheelchair concept (SIPW) based upon spherical inverted pendulum control. This chapter also shows and demonstrates that the SIPW which is a dynamics of multibody system is successfully designed within SimMechanics modelling that is stabilised using the LQR feedback method with 8 state variables. Chapter 8 discusses the contributions and the research results in sight of previous chapters. In Chapter 9 the conclusions drawn from the project work are presented and also number of potential projects for future work is listed.

CHAPTER 2

LITERATURE REVIEW

2.1. Wheelchair History

Wheelchairs are widely used as assistive devices to help provide mobility for the elderly, disabled and patients. There are three basic types of wheelchairs: manual wheelchairs, scooters and electric powered wheelchairs (EPW) (Cooper et al., 2006). Manual wheelchairs are not electric powered which are mainly used by those who still have upper body strength to provide a force. EPW are powered with electric motors and provide mobility for those who do not have the capability such as due to poor health, weak upper bodies and some other incapacity.

The creation of a wheeled chair for helping elderly and disabilities people mobility started in Europe at sixteenth century (Kamenetz, 1969). In this period the wheelchair was a manual device and the technology advancement was very slow. The first electric powered wheelchair was created in 1940 with manual on-off switch and a single speed. The last twenty years since the early of 1990 has seen the fastest development in wheelchair technologies when researchers began to use technologies originally developed for mobile robots to create modern electric powered wheelchairs or smart wheelchairs (Simpson, 2005).

Commonly there are many elderly and mildly disabled users may have been satisfied for helping their mobility by using traditional manual or electric

powered wheelchairs. However there are some disabled people who find it difficult or even impossible to use wheelchairs independently. They have some severe disabilities such as low vision, cognitive deficit, suffering from physical dexterity and coordination, visual field reduction, cerebral spasticity, tremors, paraplegia that causes them lack independent mobility and need carer to assist them (Simpson, 2005). Therefore it becomes a challenge for researchers to develop wheelchair technologies to provide a certain degree of safe autonomy for them and the idea of a smart wheelchair have started to gain support (Fehr, Langbien, and Skaar, 2000).

A smart wheelchair is an electric powered wheelchair that is equipped with a set of controllers and sensors to provide an independent mobility device for people with severe disabilities in a safe and interactive way and has additional functionality (Matsumoto, Ino and Ogasawara, 2001). People with severe disabilities usually have little or even no abilities to express their wishes, make choices and therefore cannot drive a wheelchair independently. Therefore, the current development of advanced or smart wheelchairs attempts to deal with this problem. Smart wheelchairs have used some alternative input methods instead of a joystick. Voice recognition for example has been used in NavChair (Levine et al., 1999), SENARIO (Katevas et al., 1997), TetraNauta (Cagigas and Abascal, 2004). However this work still has many difficulties, some voice control applications prove very difficult to implement in term of protection from misrecognized voice commands (McGuire, 1999), (Simpson and Levine, 2002). Other more advanced alternative inputs that have been applied are detection of

sight path of user in a Wheeleseley Robotic Wheelchair (Cooper et al., 2002) and detection of position and orientation of a user's head in Osaka University Wheelchairs (Kuno, Shimada and Shirai, 2003), Watson (Matsumoto, Ino and Ogasawara, 2001). Sensors are used in smart wheelchairs to perceive their surrounding for avoiding obstacles. For examples VAHM (Bouris et al., 1993), OMNI (Borgolte et al., 1998), Rolland (Lankenau and Rofer, 2001) has applied infrared and sonar (Simpson, 2005). While cameras are used for landmark detection like in Rolland, MAid (Katevas et.al., 1996) and as a mean of head-eye tracking for wheelchair control like in Watson, Mr. HURRY (Moon et al., 2003), Siamo (Mazo, 2001). Some wheelchairs are operated autonomously where the user only gives a final destination and then the wheelchair will plan and go to the target location, like TetraNauta. This wheelchair category is proper for people who has severe disabilities and very less ability to plan and execute a path to destination and spends the same environment of their everyday life activities.

However the current state of development of advanced wheelchairs does not meet adequately mobility and comfort for many certain users (Ding and Cooper, 2005). There are still many challenges that novel technological development can potentially overcome. Improvements in technologies on advanced or smart wheelchairs are needed to accommodate some people with severe disabilities and they can expand their ability to drive independently and to enhance driving safety.

Therefore in this research the design and simulation a new wheelchair concept with novel motion capabilities to help disabled young people will be

investigated. The young people are selected as generally a lot lighter than older people. The wheelchairs motion will be based upon the dynamic and control of a driven cart mounted inverted pendulum concept. This concept has been selected as it has the potential to offer much improved mobility compared with exiting wheelchairs. To the author knowledge the type of wheelchair based on spherical inverted pendulum configuration is a novel idea.

2.2. The Inverted Pendulum Applications

Wheelchair users often face access difficulties in daily life when they have to pass through over obstacles such as curbs, steps, and irregular terrains and getting in and out of cars. An improved wheelchair design is needed to overcome these problems and to increase both mobility and manoeuvrability of the wheelchairs. An inverse pendulum mounted upon moveable cart model applied to a wheelchair will give the user greater manoeuvrability (Nawawi, Ahmad and Osman, 2006) and increase mobility (Adam and Robert, 2004). These improvements are accomplished because stabilising small front wheels needed in conventional wheelchair can be eliminated. Except the inverted pendulum technology can be applied to keep larger rear wheels in balance and have greater ability to roll over some obstacles (Akihiro and Toshiyuki, 2009).

A cart mounted inverted pendulum is relatively simple mechanical system which is inherently unstable and defined by highly nonlinear dynamic equations (Hauser, Alessandro, and Ruggero, 2005). The inverted pendulum model has been

widely used as a teaching aid and in research experiments around the world. As a teaching aid, the inverted pendulum used because it's an imaginable unstable nonlinear dynamics problem commonly discussed in control engineering (Hauser et al., 2005), and various control algorithms, ranging from conventional through to intelligent control algorithms, has been applied and evaluated (Jung and Kim, 2008). There are a number of associated control problems that can be derived from the inverted pendulum models such as rocket control, the dynamic balance of skiing, bicycle/motorcycle dynamics (Hauser et al., 2005). Recently, a mobile inverted pendulum model with two wheels has been applied to various robotic problems such as designing walking or legged humanoid robot, robotic wheelchairs and personal transport systems (Kim et al., 2006).

The application of the wheeled inverted pendulum in robotic engineering has the goal of improving mobility such as running, steering and trajectory following (Seong and Takayuki, 2007). Several investigations using the inverted pendulum concept have been applied to wheelchairs to improve its adaptability have been conducted. SST (Sitting-Standing Transporter) is a self-balancing wheelchair that has capability of supporting a person or wheelchair occupant in the sitting and standing position with only two parallel wheels (Adam and Robert, 2004). The SST is based on the inverted pendulum with counter balance. Another application of inverted pendulum in wheelchair that can be used to help a wheelchair bound person to climb over steps without caregiver assistance has been proposed at Kanagawa Institute of Technology (Yoshihiko et al., 2008). In this proposed robotic wheelchair, the inverted pendulum control has been

conducted successfully and the user can move forward to the step although the rider felt a small shock. Another application is PADO (Pitch Angle Disturbance Observer) (Akihiro & Toshiyuki, 2009). PADO has proven that it is possible to stabilize the wheelchair motion and remove the casters. Without casters, the wheelchair is able to achieve sophisticated mobility like a short turn motion, go over steps and so on.

2.3. Simulation Techniques for Dynamic Systems

Simulation is the imitation by using a mathematical model of the operation of a real-world process or system over time (Banks, 1999). Thus simulation is to represent the behaviour of a system through a certain model performing an experiment on the system is attainable. The advantages of simulation method are it can shorten development time, reduce cost and explore a range of alternatives designs. However, the use of simulation method has some disadvantages such as level of model accuracy, the assumptions and constraints have to be defined carefully in order to get valid simulation result.

There are two dynamic system models: continuous time and discrete time models. Continuous time models have variable values changing continuously over time whereas discrete time models only change their variable values at certain instances in time. In general, such models can be modelled using differential algebraic equations (DAEs) (Ascher and Petzold, 1998). Many modelling and simulation languages have been developed, and in earliest simulation languages

was based on CSSL (Continuous System Simulation Language) (Strauss et.al., 1967). They were procedural and provided a low-level description of a system in terms of ordinary differential equations. From these languages emerged two important developments: equation-based or graph based modelling, and object-oriented modelling.

Graph based modelling have been used or developed in three paradigms: bond graphs, linear graphs and block diagrams. Bond graph modelling is based on junctions that transform elements through bonds (Rossenberg and Karnopp, 1983). The bonds represent power flow between modelling elements. Bond graphs are domain independent but they are not convenient for 3D mechanics and continuous hybrid systems. Linear graph modelling is similar to bond graphs, these linear graphs represent the energy flow through the system, expressed by through and across variables (also called terminal variables) (Durfee et.al., 1991). Linear graphs are domain independent and, unlike bond graphs, they can be easily extended to model 3D mechanics and hybrid system. The third modelling technique is based on block diagrams, as in Simulink. Block diagrams modelling are specified by connecting inputs and outputs of primitive models such as integrators, multipliers, or adders.

Object-oriented based on modelling is the latest development. Object-oriented modelling has benefits of simplified model creation and maintenance. An important principle of object-oriented modelling is that of information hiding: an object can only be accessed through its public interface, which is independent of the underlying implementation (Calderon, Paredish and Khosla, 2000). The same

principle can be applied to modelling by making a clear distinction between the physical interactions of an object with its environment (*interface*) and its internal behaviour (*implementation*). Object oriented model design results in a hierarchical organization of models and simplifies the tasks of reusing, maintaining, and extending families of simulation models. The most comprehensive support for object oriented principles is Modelica and SimMechanics.

2.4. Summary

This chapter describes the background of the research. As explained above, this project the conceptual design and simulation of a new wheelchair concept is studied as the need of improvement of the wheelchair mobility. The crux of this concept is to provide better mobility of wheelchair compared to the existing wheelchair available in the market. The wheelchair's concept is based around the dynamic and control of spherical inverted pendulum and simulated using SimMechanics.

CHAPTER 3

MATHEMATICAL MODELLING OF THE TWO DOF INVERTED PENDULUM

3.1. Introduction

In the design of the spherical inverted pendulum wheelchair (SIPW), to understand fundamental building system is started by undertaking the simulation of common or classic inverted pendulum system as shown in Figure 3.1. The SIPW has four DoF while this model system is simpler as it only has two DoF: one DoF rotation moving of the pendulum and one DoF horizontally moving of the cart. A fundamental of this works is to investigate and demonstrate the use of SimMechanics/Matlab to support simulation and understanding of the various dynamics systems and control strategies. Thus in this chapter will derived mathematical modelling of the inverted pendulum with two DoF as the starts of the works in the design of the Spherical Inverted Pendulum Wheelchair (SIPW)

3.2. Inverted Pendulum

The inverted pendulum system is a classic control problem that has been widely used as a teaching aid and in research experiments around the world

(Hauser et al., 2005). It is a suitable process to test prototype controllers due to its high non-linearities and lack of stability (Hauser et al., 2005).

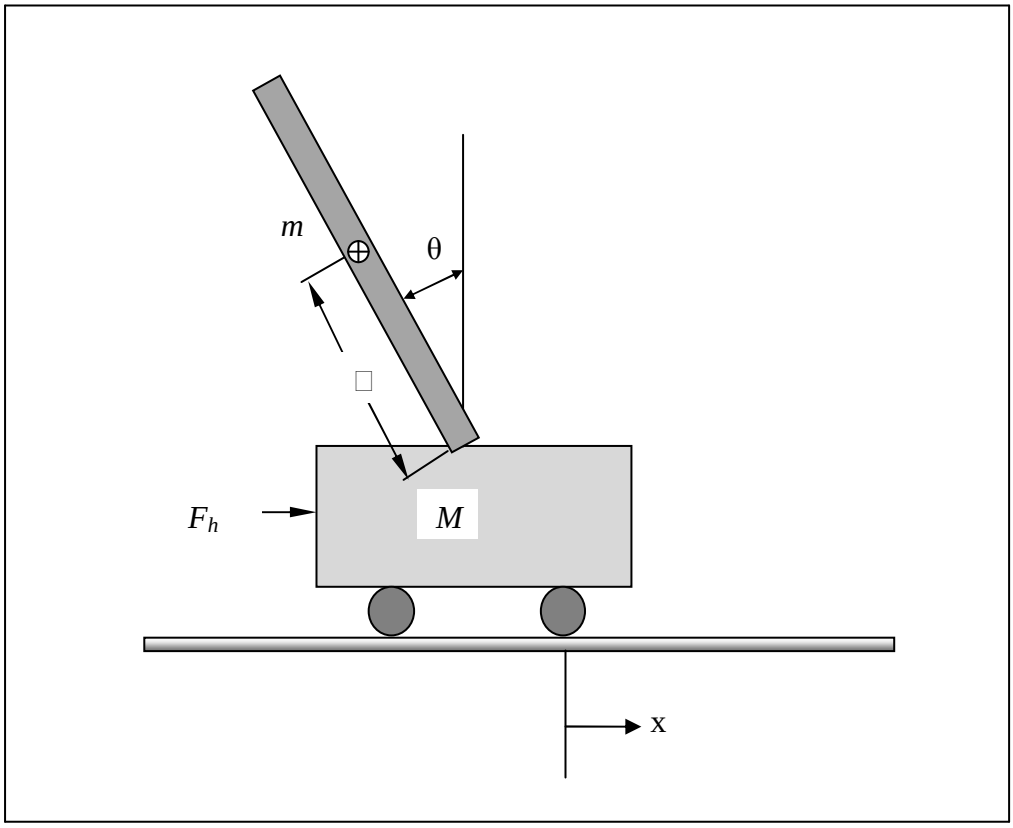


Figure 3.1 Free body diagram of the inverted pendulum system

The system as shown in Figure 3.1 consists of a pendulum with mass *m*, hinged by an angle θ from vertical axis on a cart with mass *M*, which is free to move in the *x* direction. A force F_h, is required to push the cart horizontally and dynamic equations relationship between the cart and inverted pendulum are required so that it is possible to keep the pendulum upright stable while the cart moves by following a set reference of velocity point or desired path.

3.3. Mathematical Modelling

To derive the dynamic equations or mathematical model for an inverted pendulum system, the free body diagram shown in Figure 3.2 is considered:

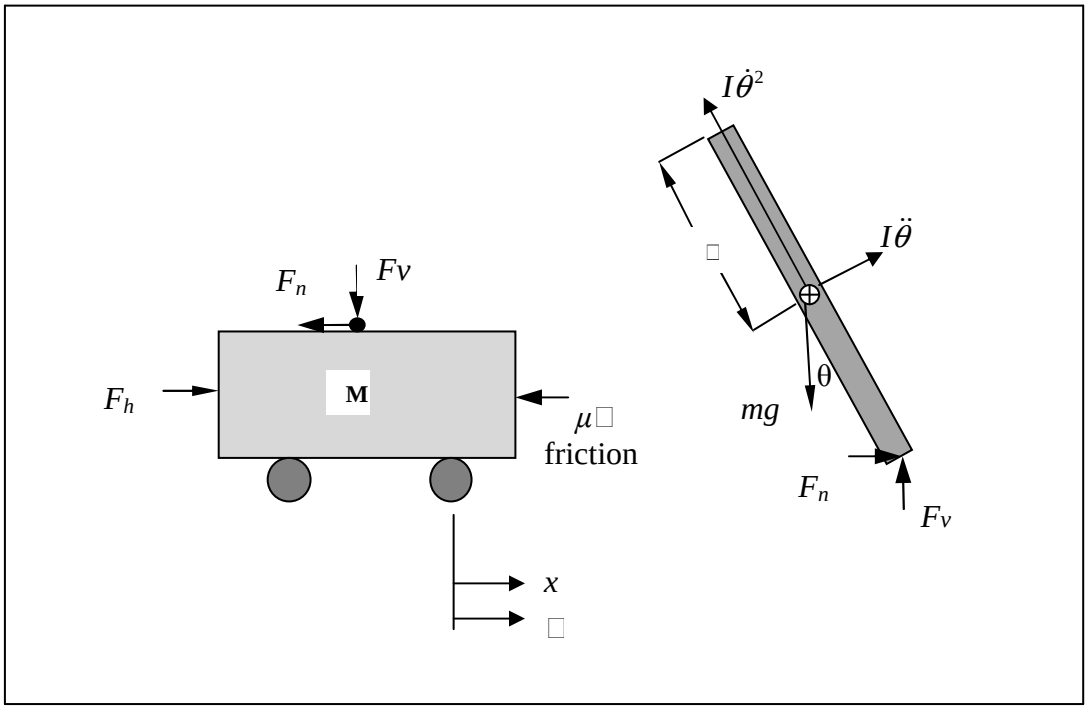


Figure 3.2 Free body diagram of the system

Summing forces of the cart in horizontal direction based on the Newton law gets:

$$M\ddot{x} + \mu\dot{x} + F_n = F_h \tag{3.1}$$

Summing forces of the pendulum in horizontal direction gives:

$$m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F_n \tag{3.2}$$

Substituting F_n in equation (3.2) into equation (3.1), the first dynamic equation for the system becomes:

$$(M+m)\ddot{x}+\mu\dot{x}+ml\ddot{\theta}\cos\theta-ml\dot{\theta}^2\sin\theta=F_h \tag{3.3}$$

Summing forces perpendicular to the pendulum gets:

$$F_v\sin\theta+F_n\cos\theta-mg\sin\theta=ml\ddot{\theta}+m\ddot{x}\cos\theta \tag{3.4}$$

Summing moments around CG of the pendulum gives:

$$-F_v l\sin\theta-F_n l\cos\theta=I\theta \tag{3.5}$$

Combining equations (3.4) and (3.5), the second dynamic equation for the system becomes:

$$\left(I+ml^2\right)\ddot{\theta}+mgl\sin\theta=-ml\ddot{x}\cos\theta \tag{3.6}$$

Since the inverted pendulum must be kept on vertical position, therefore it is assumed that $\theta(t)$ and $\dot{\theta}(t)$ are very small quantities such that $\sin\theta\approx\theta$, $\cos\theta=-1$, and $\theta\dot{\theta}=0$. Thus dynamic equations above will become (where *u* represent input):

$$\left(I+ml^2\right)\ddot{\theta}-mgl\theta=ml\ddot{x} \tag{3.7}$$

$$(M+m)\ddot{x}+\mu\dot{x}-ml\ddot{\theta}=u \tag{3.8}$$

3.3.1. Transfer Function Modelling

To get transfer function model form of the two DoF inverted pendulum system, the dynamic equations (3.7) and (3.8) need to be represented in the form of Laplace transform equations as below:

$$\left(I + ml^2\right)\theta(s)s^2 - mgl\,\theta(s) = mlx(s)s^2 \tag{3.9}$$

$$(M + m)x(s)s^2 + \mu x(s)s - ml\,\theta(s)s^2 = u(s) \tag{3.10}$$

Solve the first equation, gets:

$$x(s) = \left[\frac{\left(I + ml^2\right)}{ml} - \frac{g}{s^2}\right]\theta(s) \tag{3.11}$$

Substituting equation (3.11) into (3.10):

$$(M + m)\left[\frac{\left(I + ml^2\right)}{ml} + \frac{g}{s}\right]\theta(s)s^2 + \mu\left[\frac{\left(I + ml^2\right)}{ml} - \frac{g}{s}\right] - ml\,\theta(s)s^2 = u(s) \tag{3.12}$$

Rearranging, gives:

$$\frac{\theta(s)}{u(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{\mu\left(I + ml^2\right)}{q}s^3 - \frac{(M + m)mgl}{q}s^2 - \frac{\mu mgl}{q}s} \tag{3.13}$$

Where, $q = \left[(M + m)\left(I + ml^2\right) - (ml)^2\right]$

Transfer functions with displacement of cart as output *x(s)* can be derived in similar technique gives the following:

$$\frac{x(s)}{u(s)} = \frac{\frac{\left(I + ml^2\right)s^2 - gml}{q}}{s^4 + \frac{\mu\left(I + ml^2\right)}{q}s^3 - \frac{(M + m)mgl}{q}s^2 - \frac{\mu mgl}{q}s} \tag{3.14}$$

3.3.2. State Space Modelling

To build state space modelling of the system, the task is to derive the elements of the matrices, and to write the system model in the standard form:

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

$$y = Cx + Du$$

The matrices *A* and *B* are properties of the system and are determined by the system structure and elements. The output equation matrices *C* and *D* are determined by the particular choice of output variables.

To get state space model form of the inverted pendulum system, it required to eliminate $\ddot{\theta}$ from dynamic equations (3.7) and (3.8) to get equation for $\ddot{x}$ and vice versa as shown below:

Eliminating $\ddot{\theta}$ from equations (3.7) and (3.8) gives:

$$\ddot{x} = \frac{-\left(I + ml^2\right)\mu}{I(M+m) + Ml^2}\dot{x} + \frac{m^2 gl^2}{I(M+m) + Ml^2}\theta + \frac{I + ml^2}{I(M+m) + Ml^2}u \qquad (3.15)$$

Eliminating $\ddot{x}$ from equations (3.7) and (3.8) gives:

$$\ddot{\theta} = \frac{-ml\mu}{I(M+m) + Ml^2}\dot{x} + \frac{mgl(M+m)}{I(M+m) + Ml^2}\theta + \frac{ml}{I(M+m) + Ml^2}u \qquad (3.16)$$

Define the state variables as:

$$x_1 = x$$

$$x_2 = \dot{x}$$

$$x_3 = \theta$$

$$x_4 = \dot{\theta}$$

The state space equation then can be obtained from equations (3.15) and (3.16) as follows:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{-\left(I + ml^2\right)\mu}{I\left(M + m\right) + Mml^2} x_2 + \frac{m^2 gl^2}{I\left(M + m\right) + Mml^2} x_3 + \frac{I + ml^2}{I\left(M + m\right) + Mml^2} u$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{-ml\mu}{I\left(M + m\right) + Mml^2} x_2 + \frac{mgl\left(M + m\right)}{I\left(M + m\right) + Mml^2} x_3 + \frac{ml}{I\left(M + m\right) + Mml^2} u$$

Then the state space equations can be presented in the vector matrices form as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)\mu}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-ml\mu}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \qquad (3.17)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \qquad (3.18)$$

Thus,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)\mu}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-ml\mu}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3.4. Summary

This chapter discusses and shows how the mathematic modelling of two DoF inverted pendulum can be developed. The dynamics of this inverted pendulum can be presented in the form of transfer function equation and in the form of state space model. This mathematic model then is used to design the control strategy for stabilising the inverted pendulum. Thus in the next chapter 4 simulation of the two DoF inverted pendulum is presented using SimMechanics and Simulink with some various control strategies, which are based on the mathematic modelling developed in this chapter.

CHAPTER 4

CONTROL OF THE TWO DOF INVERTED PENDULUM

4.1. Introduction

The conventional or classical control theory is based on the input-output relationship, or transfer function. Therefore, a primary disadvantage of the classical control strategy is that there is only a single free feedback controller, *K*, which can be adjusted. Modern control theory is based on the description of system equations in terms of '*n*' first-order differential equations, which may be combined into a first-order vector-matrix differential equation, or state space. It should allow more freedom in adjusting *N* control variables for an N^{th} order system.

For the two DoF inverted pendulum system, the pendulum angle and the cart velocity or position need to be controlled and thus requires a multi-output system which is relatively simple to solve using the state space method such as discussed in the previous chapter. This chapter 4 discusses a state space with pole placement, linear quadratic regulator (LQR), and PID control techniques for controlling the cart's velocity to follow on the desired velocity while maintaining the angle of the pendulum as 0° (upright pendulum position).

4.2. Pole Placement Control

Control of the pendulum system can be designed by using the controller design of the pole placement or the pole assignment control strategy. The pole placement control is designed to put the poles of the closed loop system on the desired location by the state feedback through the appropriate state feedback gain matrix if the given system is perfectly controllable. When the system model is given as shown in the formula:

$$\dot{x} = Ax + Bu \tag{4.1}$$

The state feedback controller is as:

$$u = -Kx \tag{4.2}$$

This form is called state feedback. The state feedback gain matrix K has dimension $1 \times n$ where n is the number of the state. Substituting equation (4.1) into equation (4.2) gives result as:

$$\dot{x}(t) = (A - BK)x(t) \tag{4.3}$$

The solution of this equation is given by:

$$x(t) = e^{(A-BK)t}x(0) \tag{4.4}$$

Where, $x(0)$ is the initial state caused by external disturbances. The stability and transient response characteristics are determined by the eigenvalues of matrix A - BK**.** If matrix K is chosen properly, the matrix A - BK can be made an asymptotically stable matrix, and for all $x(0) \neq 0$, it is possible to make $x(t)$ approach 0 as t approaches infinity. A schematic block diagram of this type of control system is shown in Figure 4.1.

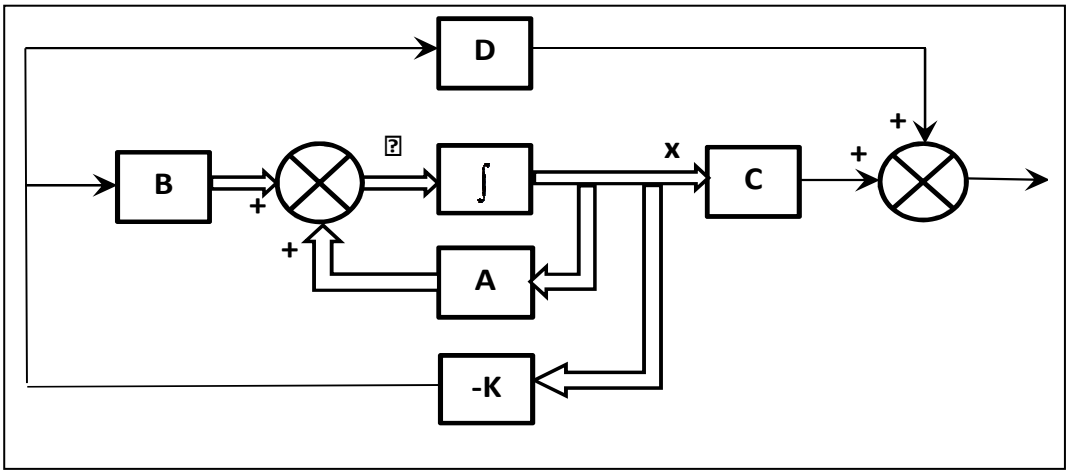


Figure 4.1 Schematic block diagram for closed loop control system in state space form (Ogata,2002)

In this work to simulate the two DoF inverted pendulum system, let assuming the two DoF inverted pendulum properties values are the same as physical properties provided by the inverted pendulum Solidworks model, as shown in Figure 4.7 developed as:

M	mass of the cart	13.26	(kg)
m	mass of the pendulum	2.88	(kg)
μ	friction of the cart	0.0	
l	length to pendulum CG	0.21	(m)
I	Moment Inertia of the pendulum	0.04	($kg*m\hat{}2$)

In this simulation the friction between the cart and the ground is assumed to be zero to give sliding moves on surface for the cart.

Substituting all the inverted pendulum parameter values given above into state space equation derived in chapter 3: equations (3.17) and (3.18) and considering the cart position as outputs to the inverted pendulum state space model becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1.5387 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 41.0617 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0717 \\ 0 \\ 0.2596 \end{bmatrix} u \tag{4.9}$$

$$[y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{4.10}$$

Using Matlab command (appendix 1) shows the poles of this model are:

p =

0

0

6.4079

-6.4079

As can be seen there is a one pole laying in the right hand plane at 6.4079 thus the system is open loop unstable. To stabilise the dynamics of the inverted pendulum plant obviously requires some of feedback controllers to be designed. This problem can be solved by finding a suitable *K* matrix using a full state feedback type 1 servo system. The schematic block diagram of this control is as shown in Figure 4.2 below.

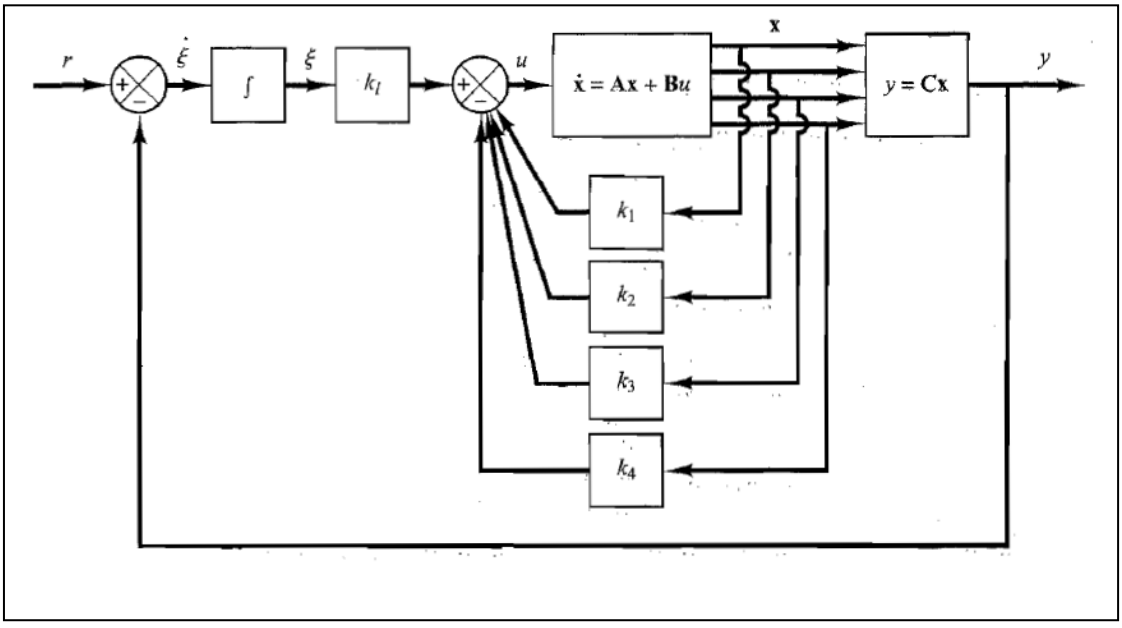


Figure 4.2 Full state feedback type 1 servo system (Ogata,2002)

Where *r* represents the reference input signal to the cart. The four states (x_1, x_2, x_3, x_4) represent position of the cart, velocity of the cart, pendulum's angle and angular velocity of the pendulum and *y* represents the output signal. A controller have to be designed so that when reference input is given to the system, the pendulum should be displaced, but eventually return to zero (upright) and the cart should move with its position or velocity as commanded in the reference input.

In the pole placement control, pole locations can be arbitrarily placed if and only if the inverted pendulum plant is controllable. Therefore the controllability matrix of the plant is determined first before calculating the feedback gain matrix *K*. The controllability matrix is given as:

$$M_c = \begin{bmatrix} B & AB & A^2B & .... & A^{n-1}B \end{bmatrix} \tag{4.11}$$

If M_c is nonzero the plant is controllable

From the equation (4.9) the state space model we get:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1.5387 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 41.0617 & 0 \end{bmatrix}$$

And,

$$B = \begin{bmatrix} 0 \\ 0.0717 \\ 0 \\ 0.2596 \end{bmatrix}$$

Since the controllability matrix M_c is given by:

$$M_c = \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = \begin{bmatrix} 0 & 0.0717 & 0 & 0.3994 \\ 0.0717 & 0 & 0.3994 & 0 \\ 0 & 0.2596 & 0 & 10.6597 \\ 0.2596 & 0 & 10.6597 & 0 \end{bmatrix}$$

And the determinant solution gives that $|M_c| = 0.4362$, thus the inverted pendulum plant is controllable and arbitrary pole placement is possible.

Referring to Figure 4.2 the state feedback control law for the inverted pendulum with set point or tracking is:

$$u = -Kx + k_I \xi \tag{4.12}$$

$$\dot{\xi} = r - y \tag{4.13}$$

In which r is the input signal reference to be tracked by y, thus ξ represents integral of the tracking error. For type servo system 1 state error equations is given as:

$$\dot{e} = \hat{A}e + \hat{B}u_e \tag{4.14}$$

Where,

$$\hat{A} = \begin{bmatrix} A & 0 \\ -X & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1.5387 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 41.0617 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0717 \\ 0 \\ 0.2596 \\ 0 \end{bmatrix}$$

The control signal is given by equation:

$$u_e = -\hat{K}e \tag{4.15}$$

Where,

$$\hat{K} = \begin{bmatrix} K & -K_i \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & -K_i \end{bmatrix}$$

In order to get a reasonable speed and damping in the response of the designed inverted pendulum plant system, the desired closed-loop poles were chosen to be at:

$$s = p_i (i=1,2,3,4.5),$$

where:

$$p_1 = -7 + 7i, \quad p_2 = -7 - 7i, \quad p_3 = -7, \quad p_4 = -7, \quad p_5 = -7$$

And the state-feedback gain matrix is calculated by using the Ackerman command of MATLAB and the MATLAB program to calculate the state feedback gain matrix *K* is as attached in appendix 2. The program gives result as:

K =

1.0e+003 *

-4.7142 -1.5334 3.3472 0.5583 6.5957

Thus, we get the feedback gains to control the designed inverted pendulum system are:

$$K = [k_1 \quad k_2 \quad k_3 \quad k_4] = [-4714.2 \quad -1533.4 \quad 3347.2 \quad 558.3]$$

And

$$K_i = 6595.7$$

These feedback gains will be used in the simulation of designed two DoF inverted pendulum using Simulink model and SimMechanics model as discussed below:

4.2.1. Simulink Model of the Two DoF Inverted Pendulum with Pole Placement Control

Simulink is a tool for simulation that is based on block diagram modelling. Its primary interface is a graphical block diagramming tool and a customisable set of block libraries. Simulink is widely used in control theory and design. It can be

used to simulate and analyse dynamic systems as well such as the inverted pendulum. The inverted pendulum in state space using Simulink modelling is as shown in Figure 4.3.

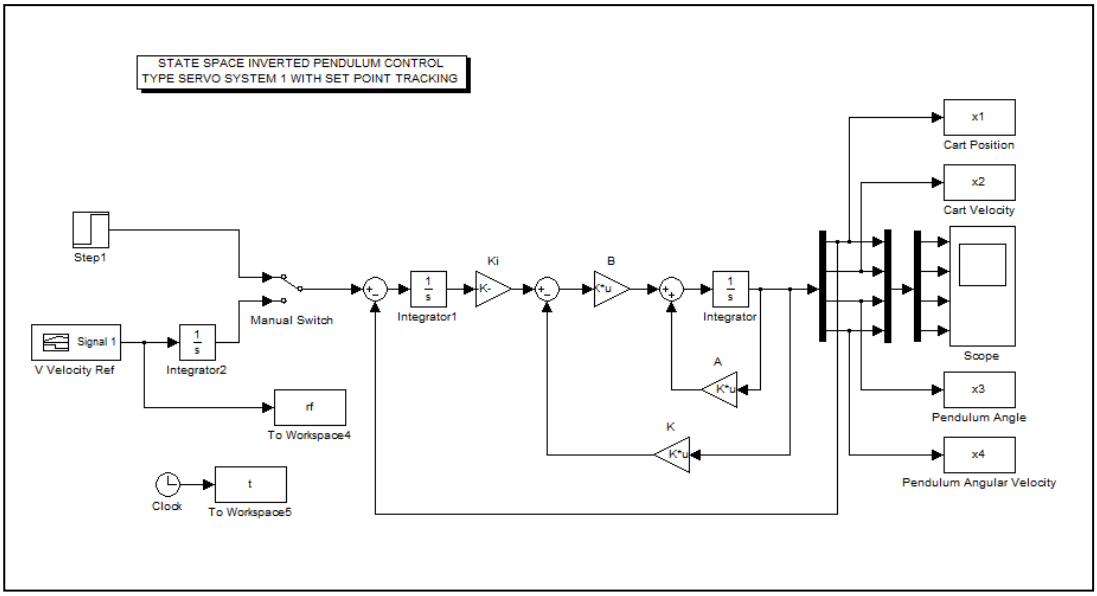
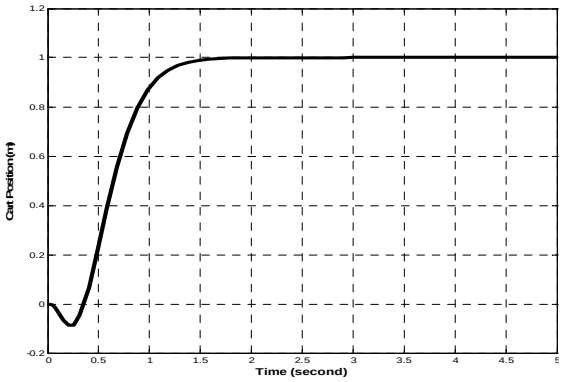


Figure 4.3 The two DoF inverted pendulum in state space using simulink

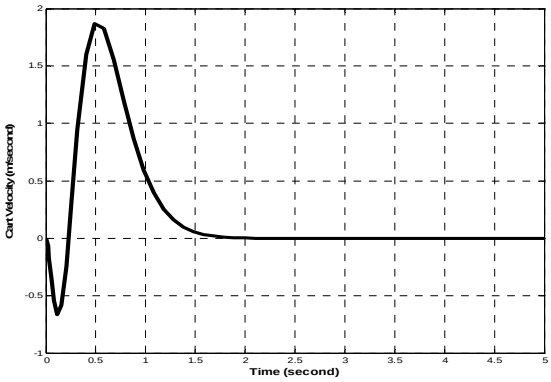
Choosing the state variables as $\left(x_1 = x, x_2 = \dot{x}, x_3 = \theta, x_4 = \dot{\theta}\right)$ and will require four state variables in the feedback controller. The system parameters are variables defined in the MATLAB workspace. Applying the state space inverted pendulum model, discussed in previous chapter 4.2 i.e. equation 4.9, and the feedback controller gains which were calculated using pole placement method and the Matlab M-file script to produce the gains can be seen in appendix 2. The gains calculation result is:

$$\hat{K} = \left[K, K_i\right] = \left[k_1, k_2, k_3, k_4, K_i\right] = \left[-4714.2 \quad -1533.4 \quad 3347.2 \quad 558.3 \quad 6595.7\right]$$

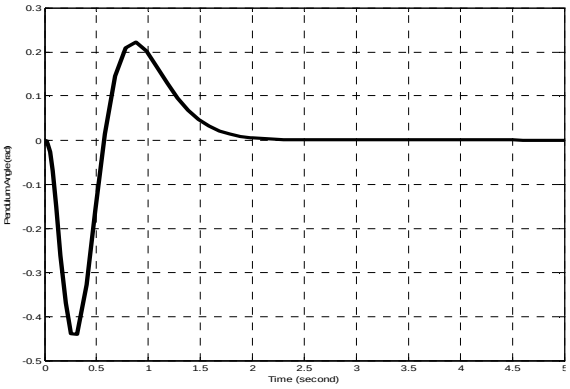
The simulation result for step response is shown in Figure 4.4. The graph shows that the pendulum rod can be stabilised at zero point at about 2 seconds while the cart can be stabilised successfully as well and reaches the new position as the reference input at 1 m.



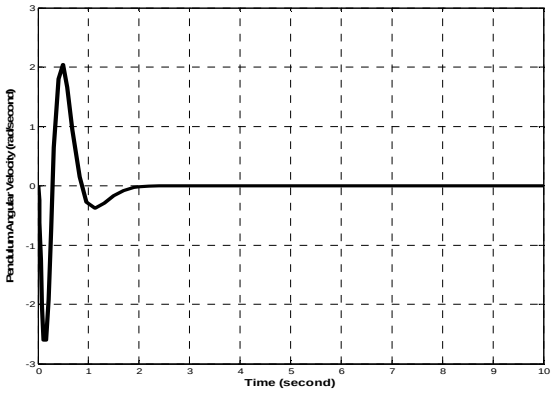
(a) Cart Position x_1



(b) Cart Velocity x_2



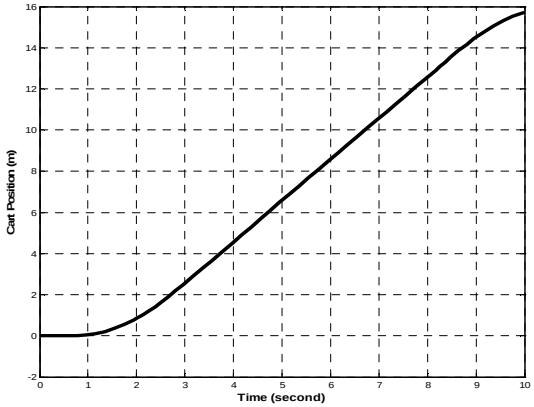
(c) Pendulum Angle x_3



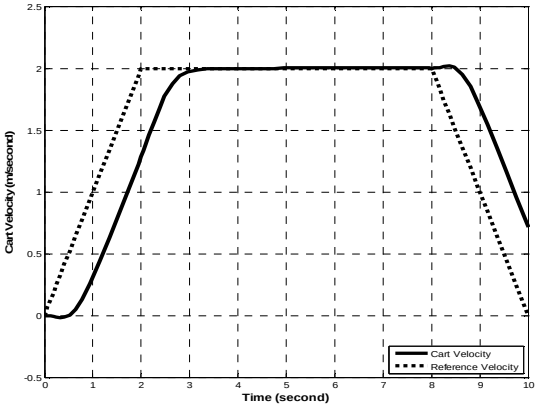
(d) Pendulum Angular Velocity x_4

Figure 4.4 Simulink state variables step response

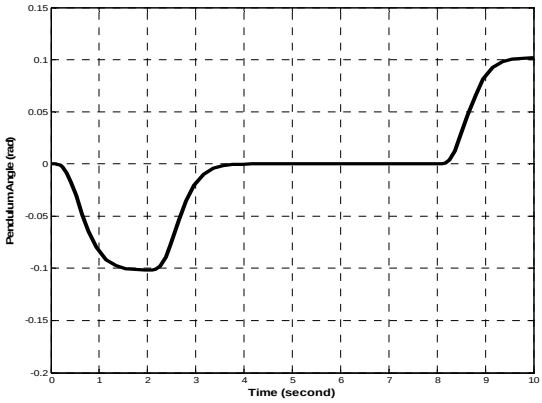
When the cart is given a prescribed velocity, the state variables response result is as shown in Figure 4.5.



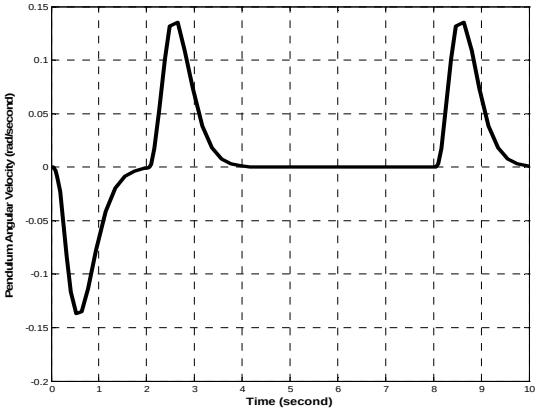
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.5 Simulink state variables result for cart velocity reference

This graph shows that the cart velocity lags about 0.5 seconds from the cart velocity reference. However the cart velocity can be controlled successfully to track the reference given while at same time stabilised the angle pendulum rod.

4.2.2. SimMechanics Model of the Two DoF Inverted Pendulum with Pole Placement Control

SimMechanics is a toolbox in Simulink-Matlab which can be used to simulate the dynamics of multibody systems which is a problem commonly found in mechanical engineering science. SimMechanics is based on an object oriented modelling and the plants or mechanical systems are modelled by connected block diagrams. SimMechanics blocks represent physical components and geometric and kinematic relationships directly. However, SimMechanics model can be interfaced seamlessly with ordinary Simulink block diagrams. Thus, it becomes possible to design a SimMechanics model operating under Simulink control system in one common environment. The SimMechanics modelling for the inverted pendulum is shown as Figure 4.6 below.

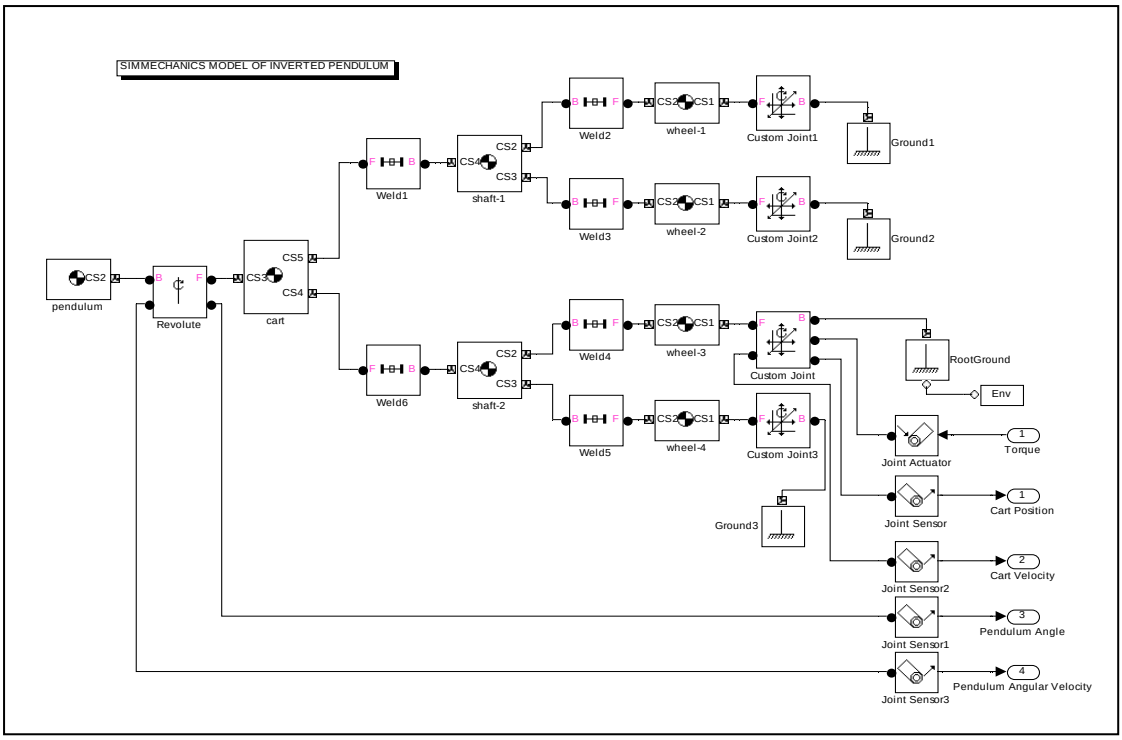


Figure 4.6 An Inverted pendulum model in SimMechanics

The model in Figure 4.6 consists of block libraries for bodies, joints, sensors and actuators elements. Each body represent a part or component of inverted pendulum assembly. Modelling such a component in traditional ways can become quite difficult. Standard Simulink blocks have distinct input and output ports. The connections between those blocks are called signal lines, and represent inputs to and outputs from the mathematical functions. SimMechanics offers four ways to visualise and animate machines:

- use machine default body geometry
- convex hull from body CS location
- equivalent ellipsoid from mass properties
- external graphics file.

External graphics files can be used to visualise the inverted pendulum SimMechanics model since this provides more realistic visualisation than the others. The two DoF inverted pendulum assembly was built in Solidwork and then CAD assembly in Solidworks was imported to SimMechanics. The visualisation of the two DoF inverted pendulum physical model in SimMechanics model is shown in Figure 4.7.

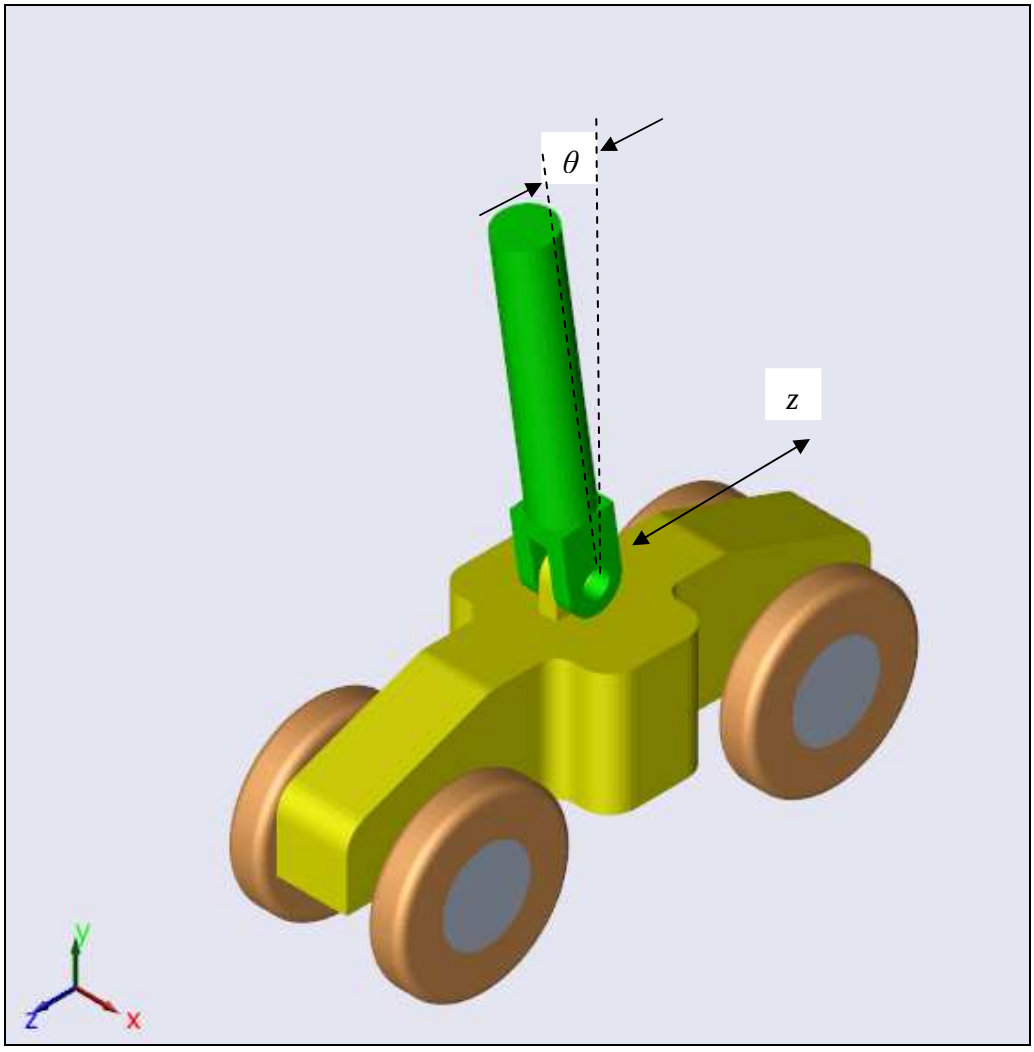


Figure 4.7 The two DoF Inverted pendulum SimMechanics model visualisation

As already mentioned, the SimMechanics blocks of two DoF inverted pendulum model can be interfaced with Simulink block diagrams. Thus, it possible to design a SimMechanics model system whereas control system is in Simulink blocks diagram as shown in Figure 4.8 below.

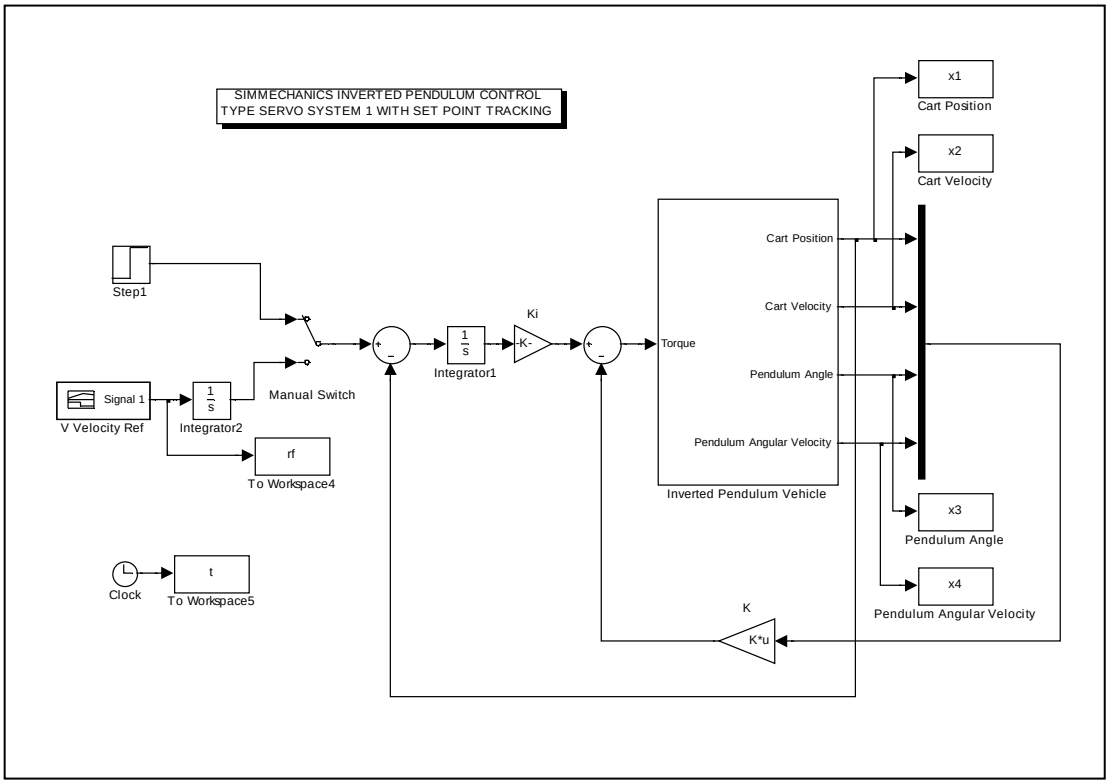
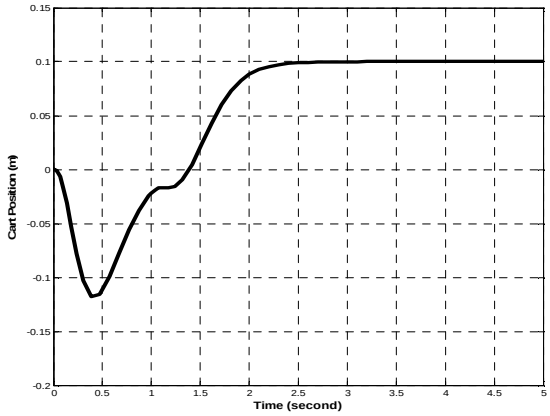


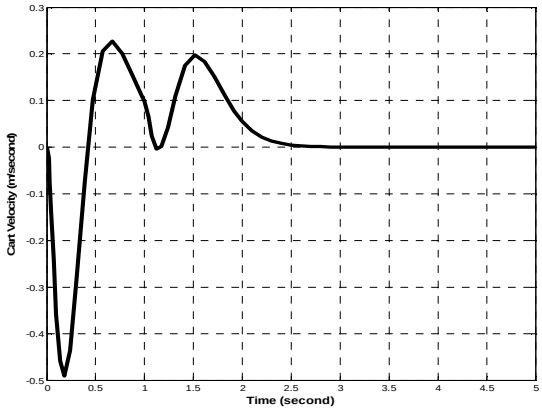
Figure 4.8 The Inverted pendulum control in SimMechanics-Simulink

As seen from the Figure 4.8 the inverted pendulum vehicle is modelled using SimMechanics and controlled by feedback controller. The feedback controller gains matrix was calculated using pole placement control strategy:

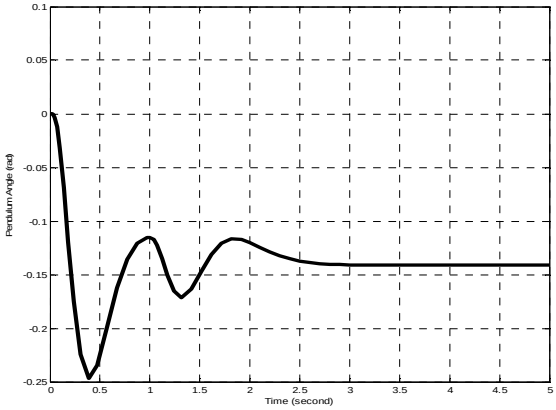
$$\hat{K} = [K, K_i] = [k_1, k_2, k_3, k_4, K_i] = [-4714.2 \quad -1533.4 \quad 3347.2 \quad 558.3 \quad 6595.7]$$



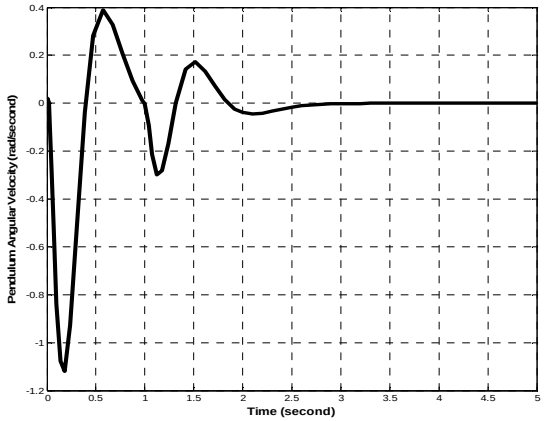
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



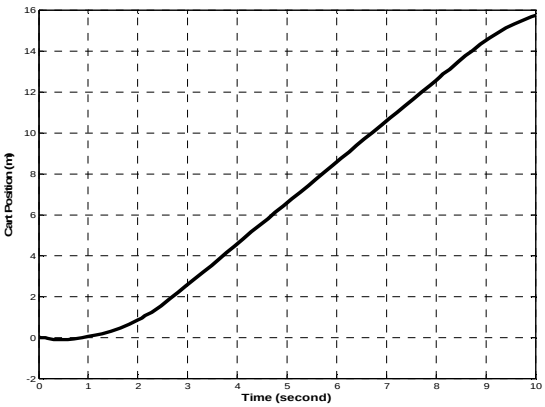
(d) Pendulum Angular Velocity x_4

Figure 4.9 SimMechanics state variables result for step response

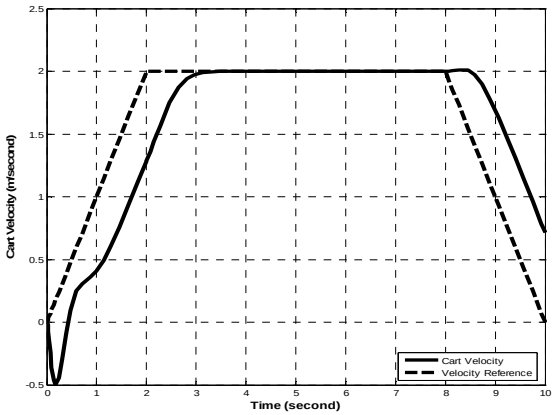
The state variables result to step response can be seen in Figure 4.9. The graph is shown that the pendulum rod can be stabilised at zero point at about 2.5 seconds while the cart can be stabilised successfully as well and reaches the new position as the reference input at 1 m. Compared with Simulink model result, SimMechanics model need a longer time of about 0.5 second to stabilise the

system. It can be seen from the Figure 4.9 that the pendulum is stable at 0.14 radian after running 2.5 seconds whereas the Simulink model the pendulum stable at 0 radian after 2 seconds. However the result demonstrates that response between SimMechanics model and state space model in Simulink are very close.

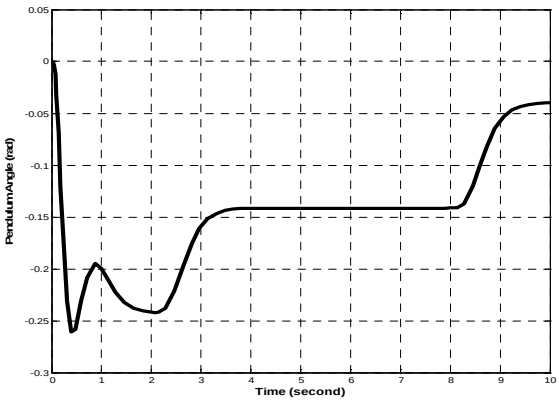
When the cart is given a prescribed velocity, the state variables response result is as shown in Figure 4.10.



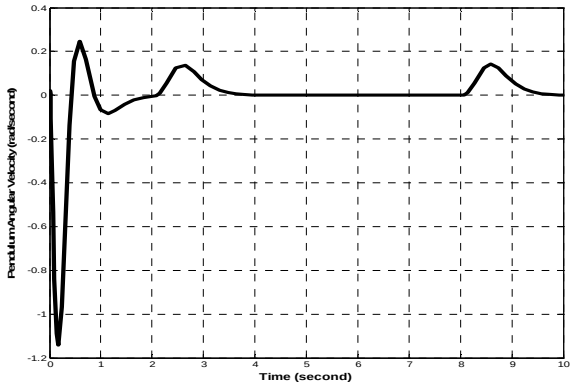
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.10 SimMechanics state variables result for cart velocity reference

The graph shows that the cart velocity lags about 0.5 seconds from the cart velocity input reference. However the cart velocity can be controlled successfully to track the reference input given while at same time stabilized the angle pendulum rod.

4.3. Linear Quadratic Regulator (LQR) Control

Another way to choose the feedback gains to control the two DoF inverted pendulum system is the quadratic optimal control method to minimize a quadratic cost function. One of the most common optimal controllers is the LQR (Linear Quadratic Regulator) controller. The quadratic criterion to be minimized is:

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{4.16}$$

Cost function J can be interpreted as energy function, x is the state variables of the systems which is weighted by Q and u is control input of the systems which is weighted by R. In LQR, the gain matrix K for the state feedback controller law as equation (4.2) is found by minimising this cost function. Minimisation the cost function J, results in moving x to zero via little control energy and in turn guarantees that the systems will be stable. The two weighting matrices $Q > 0$ and $R > 0$, are symmetric and positive definite gain matrices and should be selected. The gain K (matrix) is determined by first solving the algebraic Riccati equation:

$$PA + A^T P - PBQ^{-1}B^T P + Q = 0 \qquad (4.17)$$

Then K can be calculated as:

$$K = Q^{-1}B^T P \qquad (4.18)$$

This calculation can be difficult by hand. However the MATLAB lqr command can be used. The Matlab lqr command solves for the gain vector K given A, B, Q, and R. This approach is used in all LQR designs and simplest way is to assume as:

$$Q = \begin{bmatrix} x_w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & y_w & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$R = 1,$

The element in the (1,1) position will be used to weight the cart's position and the element in the (3,3) position will be used to weight the pendulum's angle and denoted as x_w and y_w weighting variables. Then K matrix will be produced an optimal controller could be found by using K = lqr (A,B,Q,R), the m-file code in Matlab can be seen in appendix 3.

The weighting x_w and y_w variables could be changed to see the various responses. If x_w and y_w are increased even higher, improvement to the response should be found. But, in order to satisfy the design requirements of keeping x_w and y_w as small as possible since in this problem the values of x and y have been used to describe the relative weight of the tracking error in the cart's position and pendulum's angle versus the control effort. The higher the values of x_w and y_w, the more control effort will be required, but the smaller the tracking error. Using all parameter values of the two DoF inverted pendulum and by trial and error choosing the weighting values as x_w = 10000 and y_w = 1000, the following values for controller gains K matrix are determined:

K =

-100.0000 -88.8858 545.3054 87.6982

And the reference gain is

Kr =

-100.0000

These feedback gains will be used in the simulation of the designed two DoF inverted pendulum using state space Simulink model and SimMechanics model and will be discussed in 4.3.1 and 4.3.2.

4.3.1. Simulink Model of the Two DoF Inverted Pendulum with LQR Control

The inverted pendulum in state space with feedback controller using LQR in Simulink modelling is as shown in Figure 4.11.

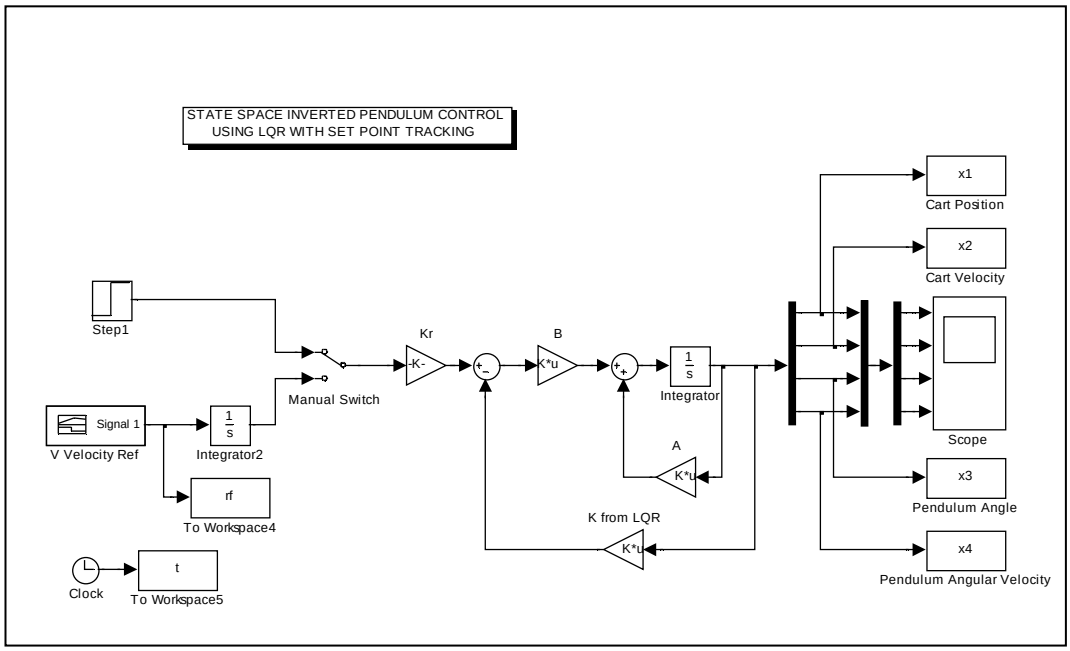
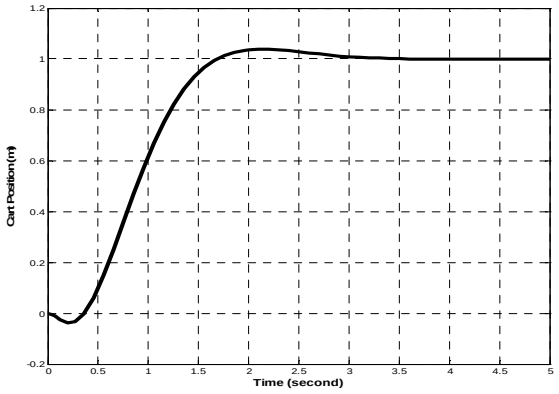


Figure 4.11 The inverted pendulum LQR feedback control in state space using simulink

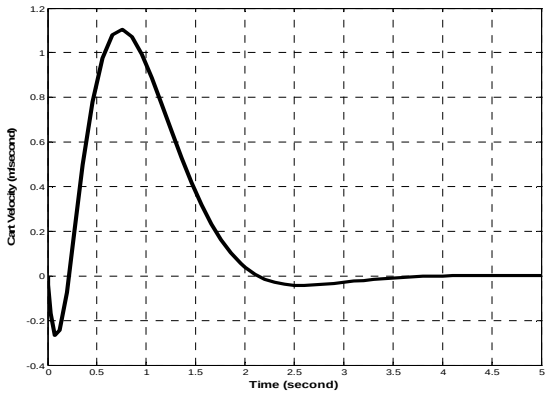
The state space model and the LQR feedback control is shown in Figure 4.11. Applying the state space inverted pendulum model, discussed in previous chapter i.e. equation 4.9, and the feedback controller gains which were calculated using LQR method as in appendix 3 that is:

$$[K, k_r] = [k_1, k_2, k_3, k_4, k_r] = [-100.0 \quad -88.9 \quad 545.3 \quad 87.7 \quad -100.0]$$

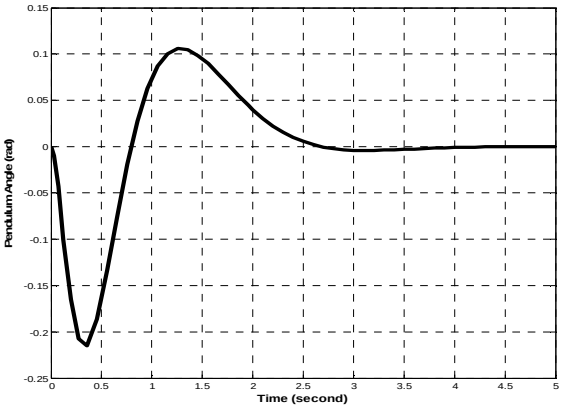
produced results as shown in Figure 4.12. The graph shows that the pendulum rod can be stabilised at zero point at about 3 seconds while the cart can be stabilised successfully as well and reaches the new position as the reference input at 1 m. There is small overshoot only about 5 %.



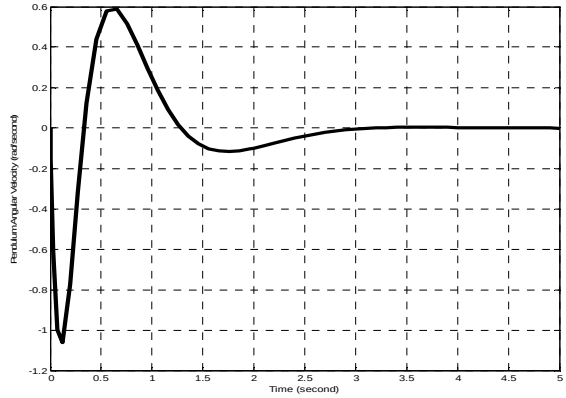
(a) Cart Position x_1



(b) Cart Velocity x_2



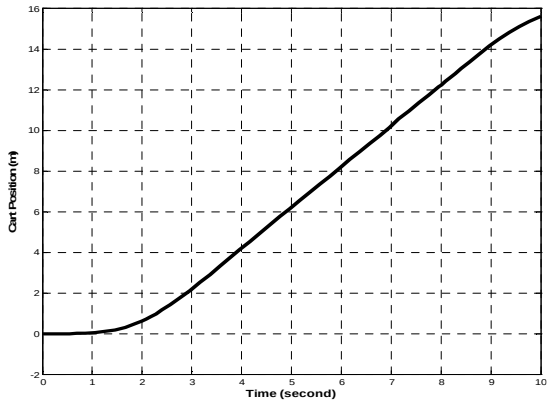
(c) Pendulum Angle x_3



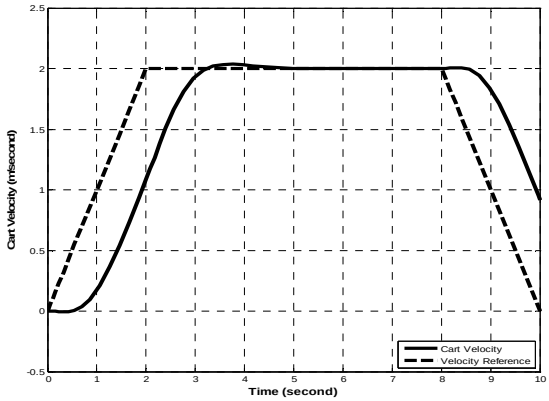
(d) Pendulum Angular Velocity x_4

Figure 4.12 Simulink LQR control state variables result for step response

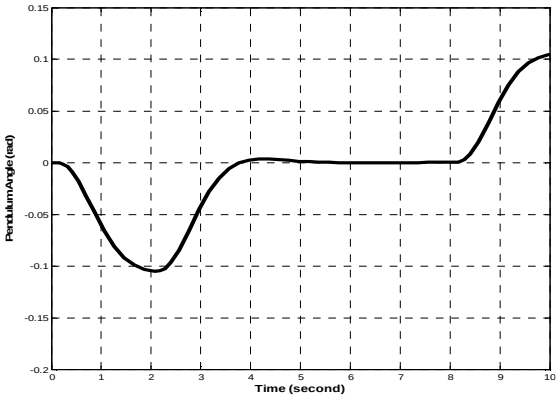
When the cart is given a prescribed velocity, the state variables response result is as shown in Figure 4.13.



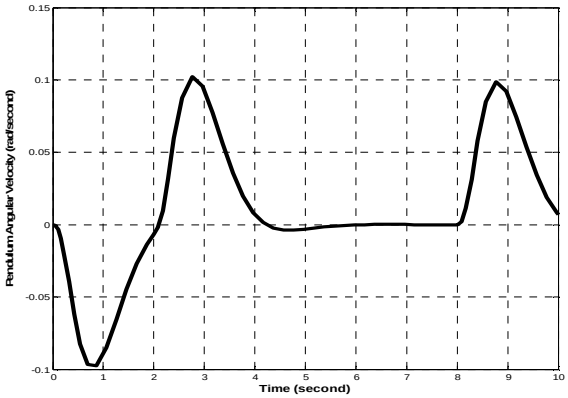
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.13 Simulink LQR control state variables result for cart velocity reference

The graph above is shown that the cart velocity lags about 1 second from the cart velocity reference. However the cart velocity can be controlled successfully to track the reference input given while at same time stabilised the angle pendulum rod.

4.3.2. SimMechanics Model of the Two DoF Inverted Pendulum with LQR Control

The SimMechanics modelling for the two DoF inverted pendulum using LQR control is shown as Figure 4.14 below.

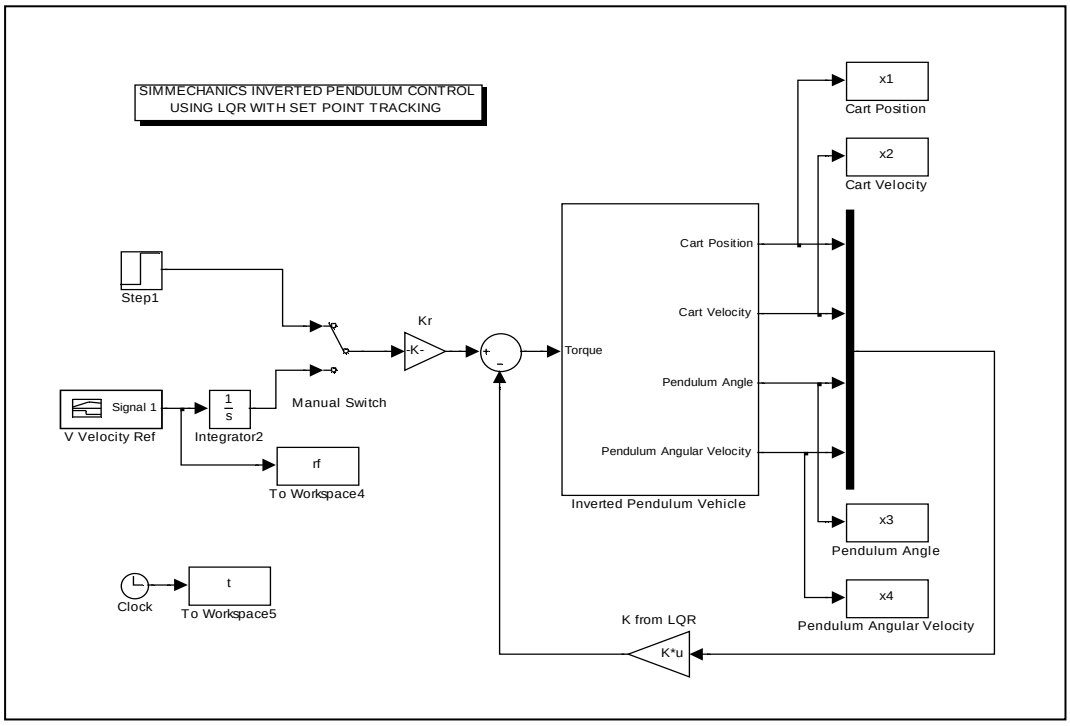
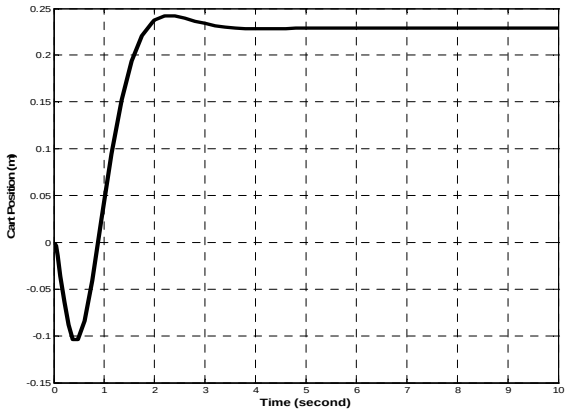


Figure 4.14 The inverted pendulum LQR feedback control in SimMechanics

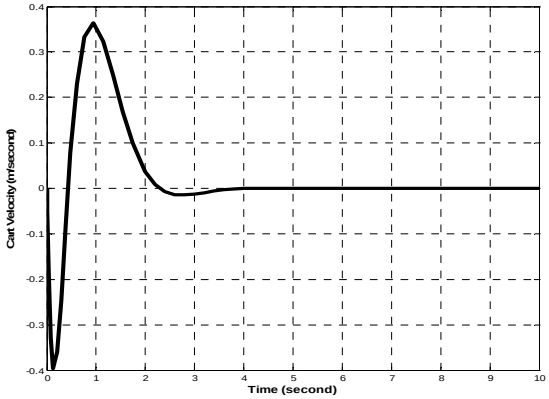
As can be seen from Figure 4.14 the inverted pendulum vehicle discussed previously is modelled using SimMechanics and controlled by using LQR feedback controller. The feedback controller gains matrix was calculated using linear quadratic regulator (LQR) method that is:

$$[K, k_r] = [k_1, k_2, k_3, k_4, k_r] = [-100.0 \quad -88.9 \quad 545.3 \quad 87.7 \quad -100.0]$$

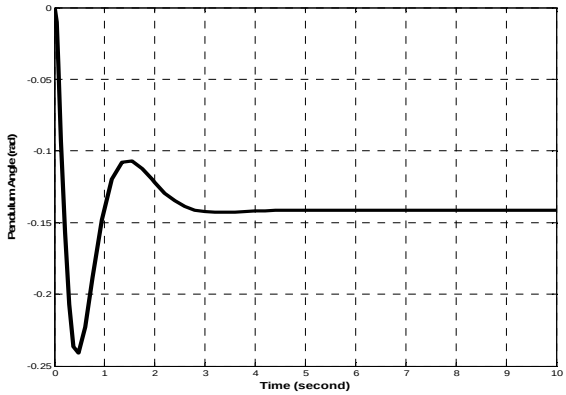
The state variables result to step response input is as Figure 4.15.



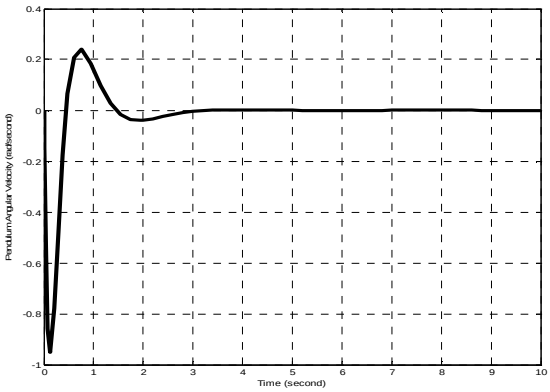
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



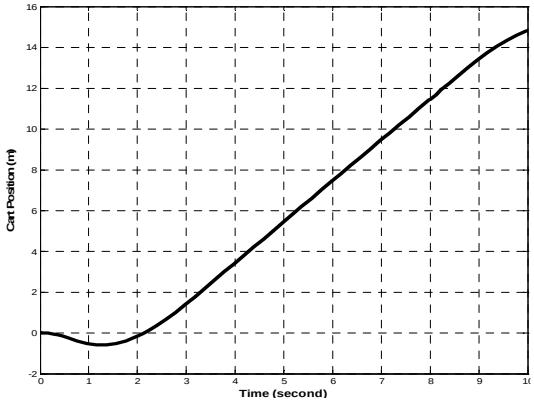
(d) Pendulum Angular Velocity x_4

Figure 4.15 Simmechanics with LQR control state variables result for step response

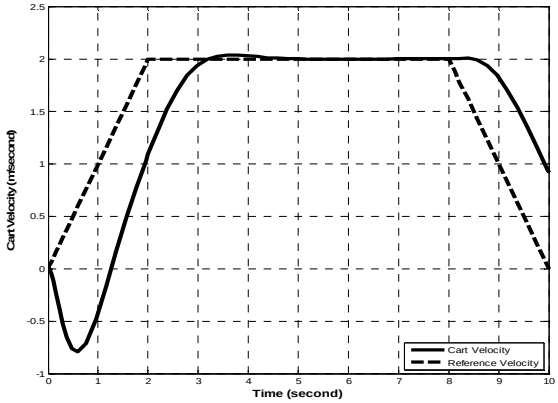
The graph is shown that the pendulum rod can be stabilized at zero point at about 3.5 seconds while the cart can be stabilised successfully as well and reaches the new position as the reference input at 1 m. Again compared with Simulink

model result, SimMechanics model needs 0.5 seconds longer to stabilise the system. It can be seen from the Figure 4.15 that pendulum stable at 0.14 radian after running 3.5 seconds whereas the Simulink model stable at 0 point after 3 seconds. However the result demonstrates that there is little substantial difference between SimMechanics model and state space Simulink model.

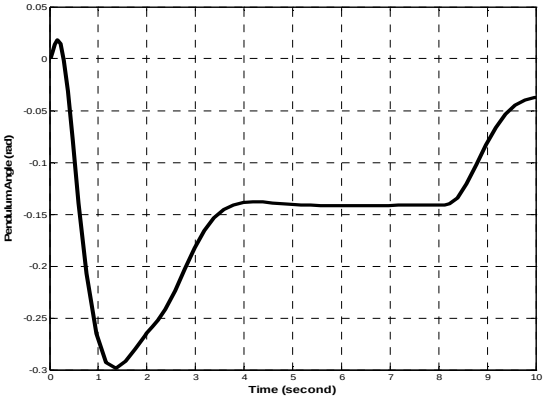
When the cart is given a prescribed velocity input, the state variables result is as Figure 4.16.



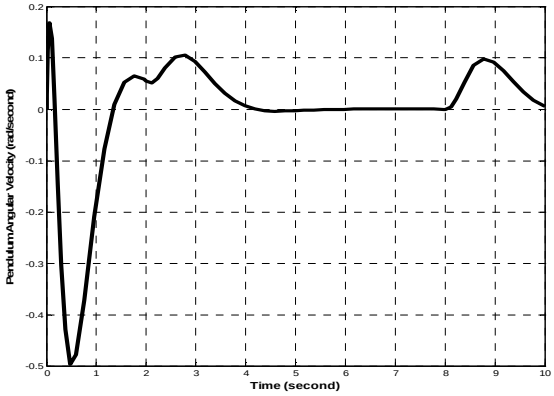
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.16 SimMechanics with LQR control result for cart velocity reference

The result as can be seen in Figure 4.16 shows that the cart velocity lags about 1 second from the cart velocity reference. However the cart velocity can be controlled successfully to track the reference input given while at same time stabilized the angle pendulum rod.

4.4. PID Control

PID (Proportional-Integral-Derivative) control is the widest type of automatic control used in industry. PID control has survived many changes in technology, from mechanics and pneumatics to microprocessors. The microprocessor has had a dramatic influence on the PID controller. Practically all PID controllers made today are based on microprocessors. This has given opportunities to provide additional features like automatic tuning, gain scheduling and continuous adaptation.

PID controller will correct the error between the output and the desired input or set point. PID controller has the general form as:

$$u = K_p e + K_i \int e dt + K_d \frac{de}{dt} \qquad (5.1)$$

Where K_p is proportional gain, K_i is the integral gain, and K_d is the derivative gain. The variable e represents tracking error, the difference between desired input (set point) and actual output. This error signal e will be sent to the PID controller and the controller will do calculation (algorithm) involves three separate parameters;

the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent error and the Derivative determines the reaction to the rate at which the error has been changing. Then the new signal *u* will be sent to the plant or inverted pendulum system in my case and new output will be sent again to find new error signal *e*. The controller takes this new error signal and computes again. This process goes on and on during the simulation.

Effects of Proportional, Integral and Derivative action on the plant or inverted pendulum system are:

1. Proportional controller (K_p) will have the effect of reducing the rise time and steady state error, but never eliminate it.
2. Integral controller (K_i) will have the effect of eliminating the steady-state error, but it may make the transient response worse.
3. Derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

Effects of each controllers K_p, K_i, and K_d on a closed-loop system are summarized in the Figure 5.1 shown below.

Controller	**Rise Time**	**Overshoot**	**Settling Time**	**SS Error**
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small change	Decrease	Decrease	Small change

Figure 4.17 The effects of K_p, K_i, *and* K_d to the dynamic system

There are several methods to obtain the values or gains of K_p, K_i, and K_d, one of those methods is that proposed by Ziegler and Nichols in 1940's (Astrom et al, 1995). The method first applied to open loop plant a step input and then observes the open loop response. This traditional tuning tended to provide a dynamic system with poor performance. Thus automatic tuning strategy was selected as it can give fast and useable gains for PID control blocks. A key method for auto tuning is to use relay feedback (Astrom, 1988). By using this technique can tune to obtain gain parameters with good performance and ideal response time. Therefore the author used PID auto tuner from the Simulink block to get the best performance of inverted pendulum dynamic system. The procedure to obtain PID gains controller using auto tuning in Simulink PID block is as follows:

- Launch the PID tuner to compute a linear plant.
- Tune the controller in PID tuner by adjusting the response time and the PID tuner calculates the gains that can stabilise the system.
- Export the gains values to the PID control block and simulate the performance in Simulink.

4.4.1. Simulink Model of the Two DoF Inverted Pendulum with PID Control

As the objective is to control the pendulum's position, which should remain to the vertical position or at 0 point radian, therefore the reference signal for the pendulum's angle position should be zero. The force applied to the cart is added as a disturbance to the inverted pendulum system that causes the pendulum unstable. Then the PID controller will be used to stabilise the pendulum's angle at upright position or zero while the cart is controlled by another PID to move to the new desired position. Thus the author proposes to control the two DoF inverted pendulum system using two PID controllers for controlling both cart and the pendulum respectively.

The block diagram of the two DoF inverted pendulum plant in Simulink state space model with two different PID controllers is shown as in Figure 4.18. It can be seen from the Figure there are 2 different PID controllers. The 'PID Cart' is used to control state variable x_1 that is cart position and 'PID Pendulum' is used to control state variable x_3 that is pendulum angle.

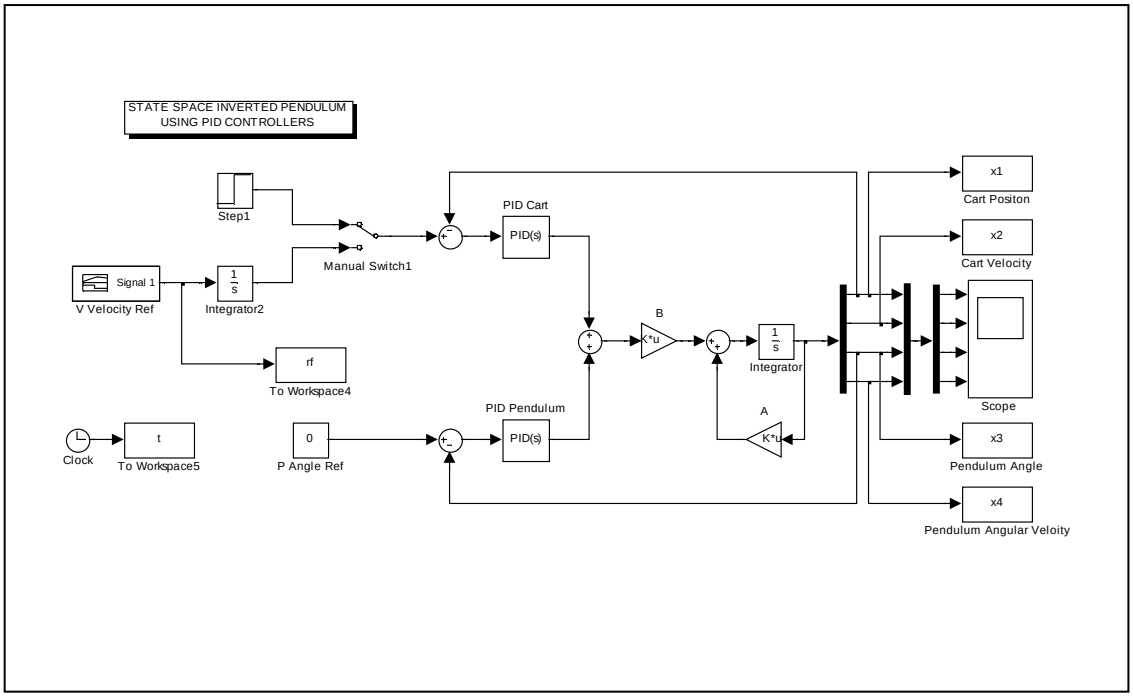


Figure 4.18 Simulink inverted pendulum state space model with two PID controllers

Applying inverted pendulum state space model as equation (4.9) and by using PID tuner the controller gains used inside for 'PID Cart' which controls cart as shown in Figure 4.19 are as

K_p = -414.8, K_i = -5.1, and K_d = -467.4

The gains inside 'PID Pendulum' which controls pendulum as shown in Figure 4.20 are as:

K_p = 4157.3, K_i = 3539.5, and K_d = 443.3

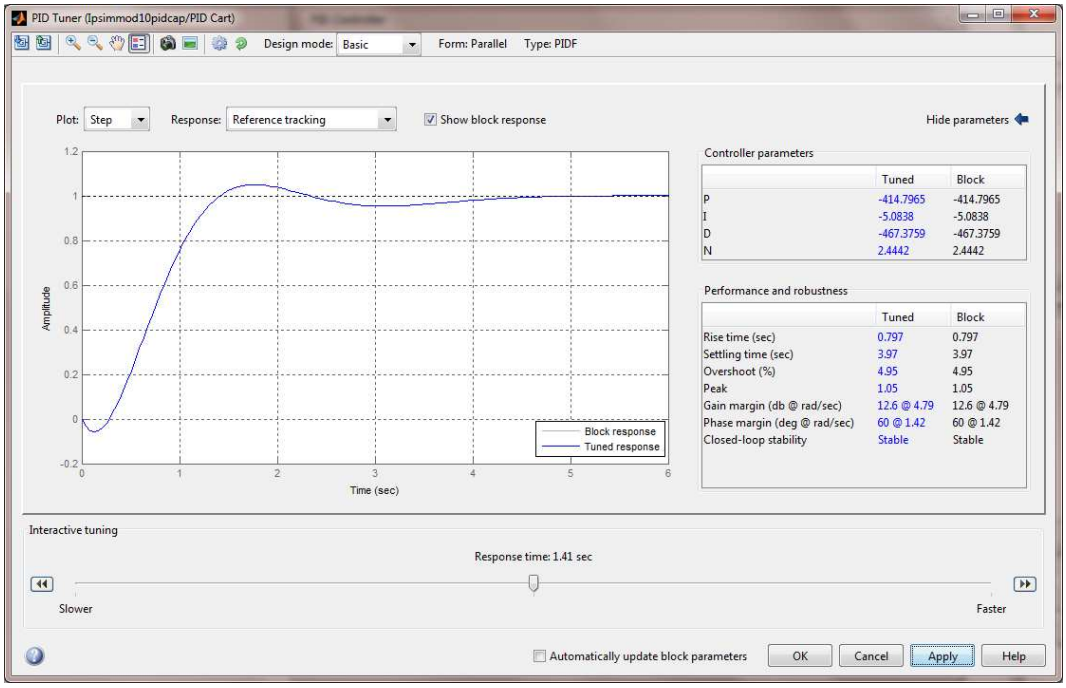


Figure 4.19 PID Cart to control cart position

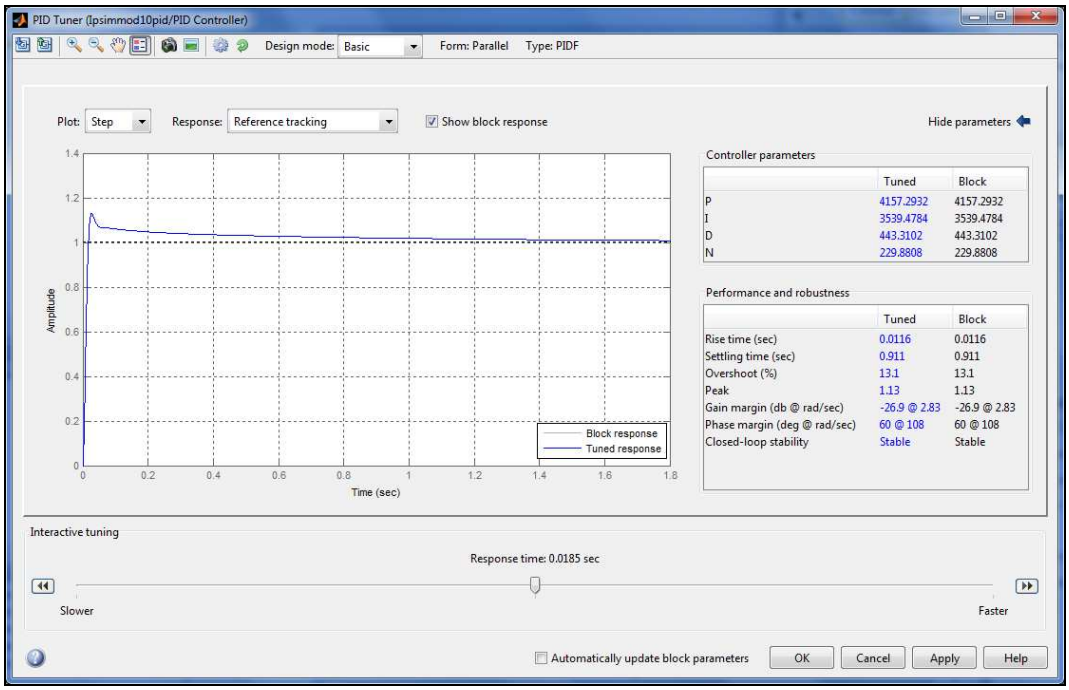
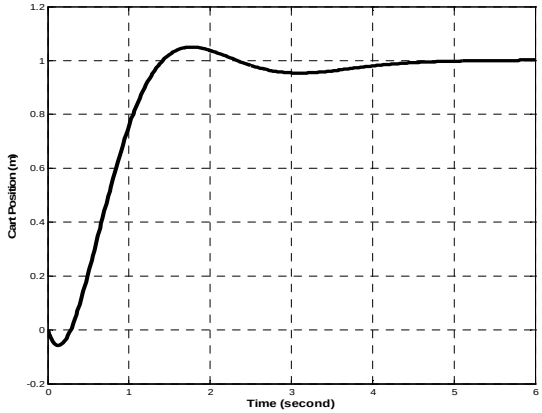
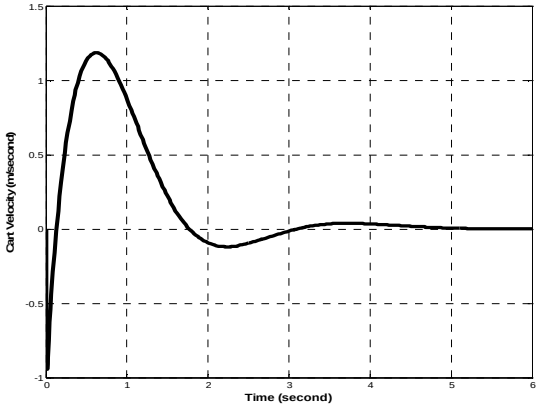


Figure 4.20 PID Pendulum to control pendulum angle

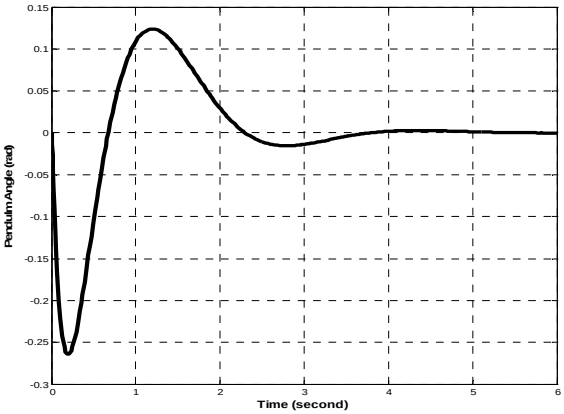
The simulation result of step responses for two PID controllers with consideration to stabilise the pendulum angle and cart position is as shown in Figure 4.21.



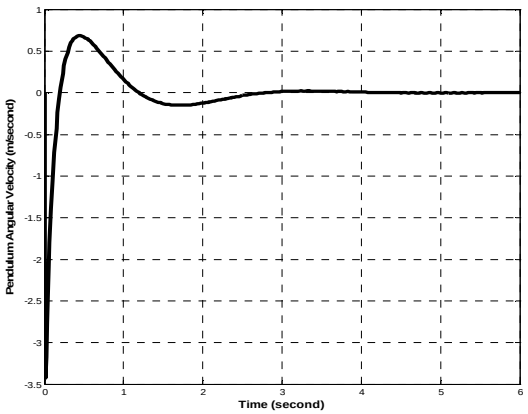
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3

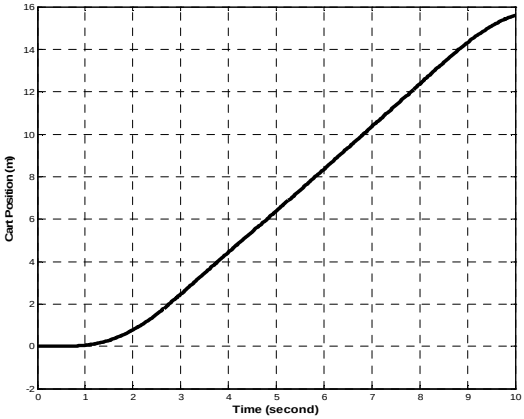


(d) Pendulum Angular Velocity x_4

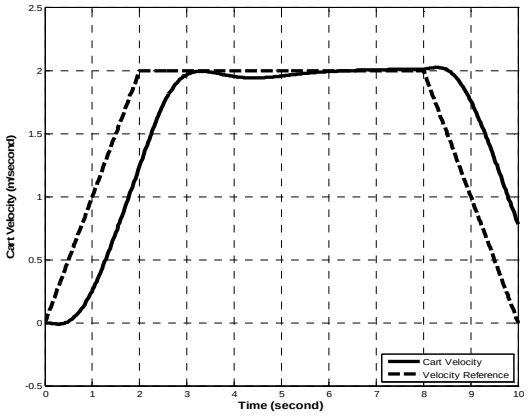
Figure 4.21 Step response of the system in Simulink using two PID controllers

From Figure 4.21, it can be seen that by applying two PID controllers to the two DoF inverted pendulum system, all the state variables of the inverted pendulum system can be stabilised successfully with rise time about 1 second, settling time about 4 seconds and overshoot about 5 %. ‘PID Cart’ and ‘PID Pendulum’ were used to control the position of the cart and the angle of the pendulum respectively. As a conclusion, the model of two DoF inverted pendulum system can be stabilized successfully by applying two PID controllers simultaneously if they are tuned properly.

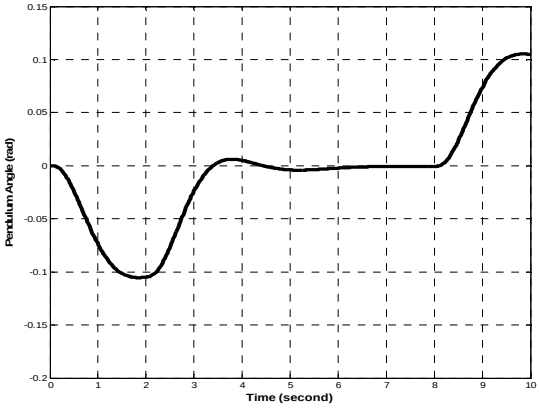
Moreover using the same PID values as Figure 4.19 for ‘PID Cart’ and Figure 4.20 for ‘PID Pendulum’, when the cart is given a prescribed velocity, the state variables result is shown in Figure 5.22. The graph shows that the cart velocity clearly follows the prescribed velocity although there is a lags about 0.5 seconds from the cart velocity input reference. However it can be concluded that cart velocity can be controlled successfully to track the reference input given while at same time stabilised the angle pendulum rod remain on vertical position by using two PID controllers.



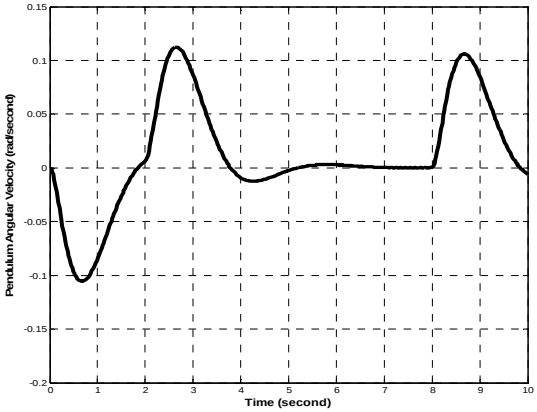
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.22 Result Simulink model controlled with two PID for velocity reference

4.4.2. SimMechanics Model of the Two DoF Inverted Pendulum with PID Control

A SimMechanics model with the same inverted pendulum parameter values is shown in Figure 4.23.

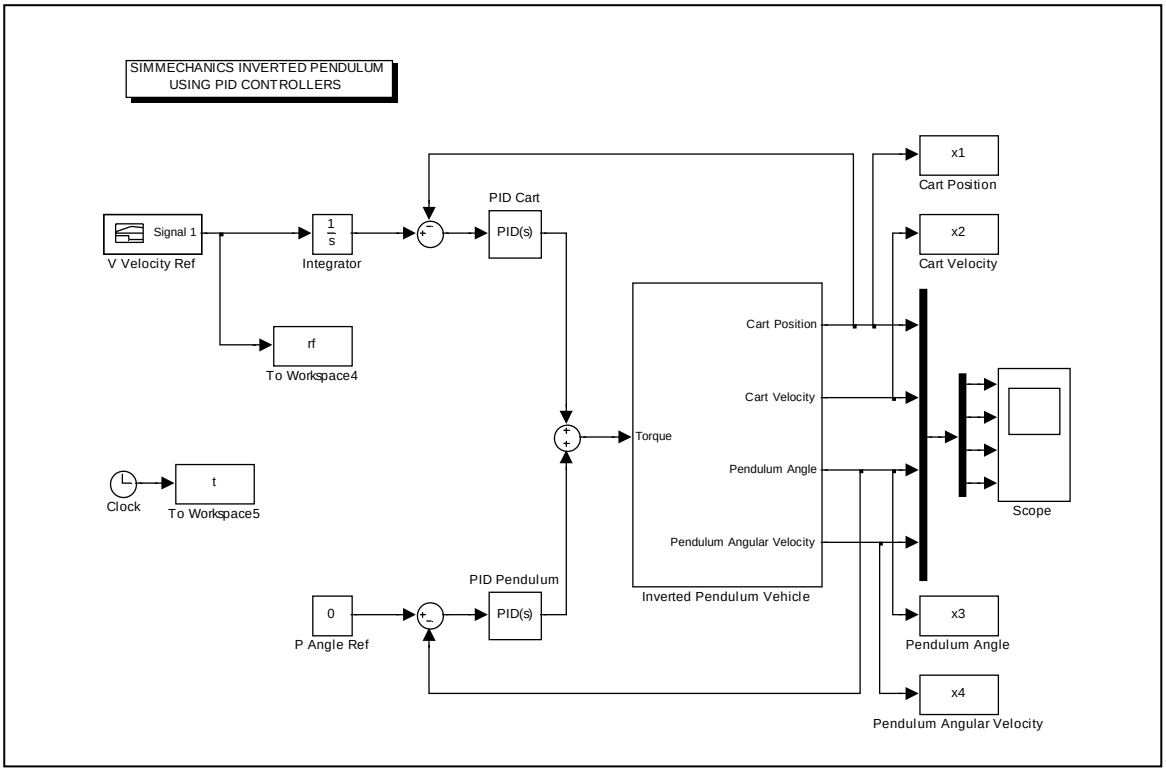


Figure 4.23 Simmechanics two DoF inverted pendulum model with two PID controllers

To determine the controller gains the Matlab PID tuner was used. The controller gains used inside 'PID Cart' which controls cart as shown in Figure 4.24 are as:

K_p = -2084.1, K_i = -64.4, and K_d = -823.8

The gains inside ‘PID Pendulum’ which controls pendulum as shown in Figure 4.25 are as:

K_p = 4136.6, K_i = 3534.8, and K_d = 439.5

These gains are successfully controlling the inverted pendulum system using SimMechanics as a diagram shown in Figure 5.13.

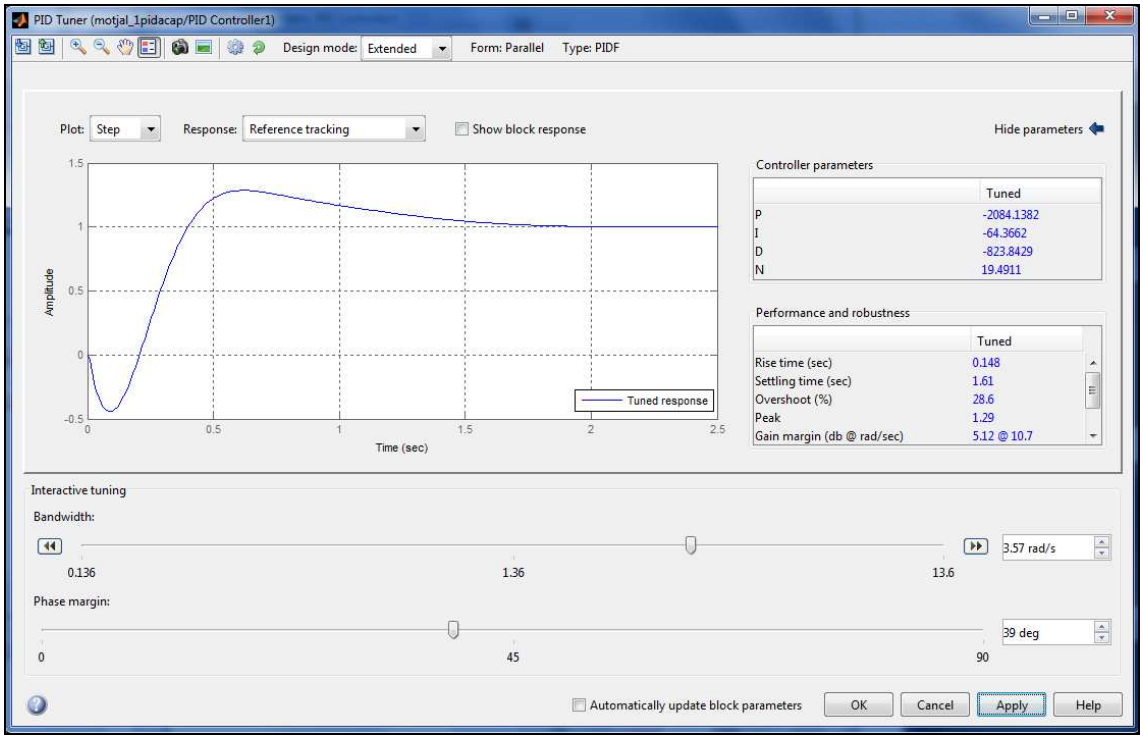


Figure 4.24 PID tuner for cart position

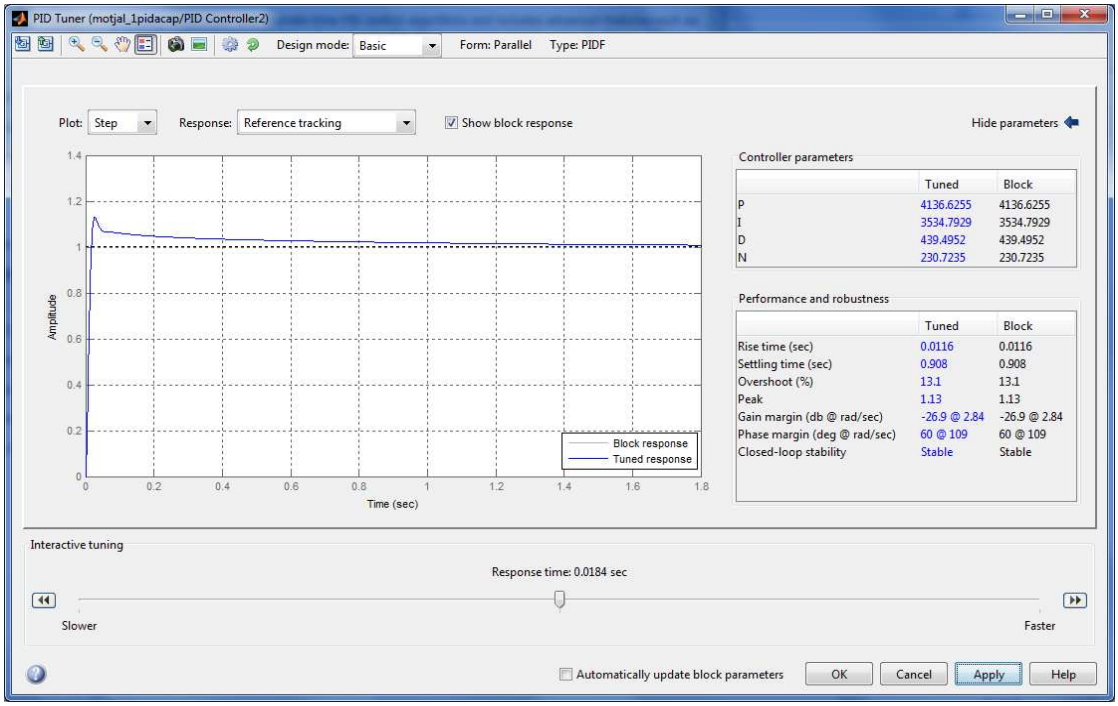
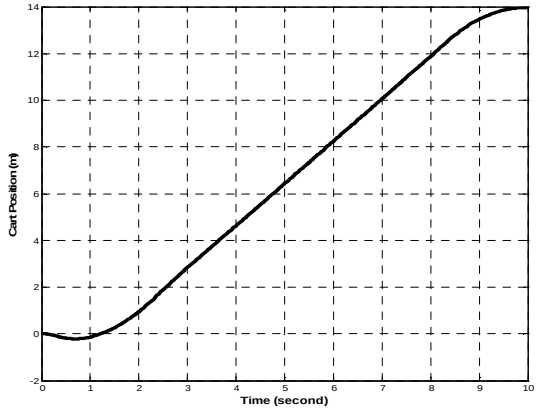
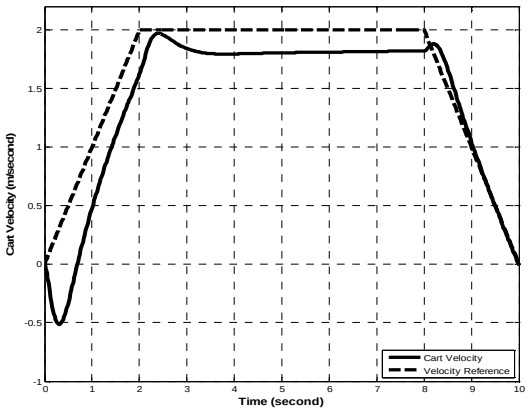


Figure 4.25 PID tuner for pendulum's angle control

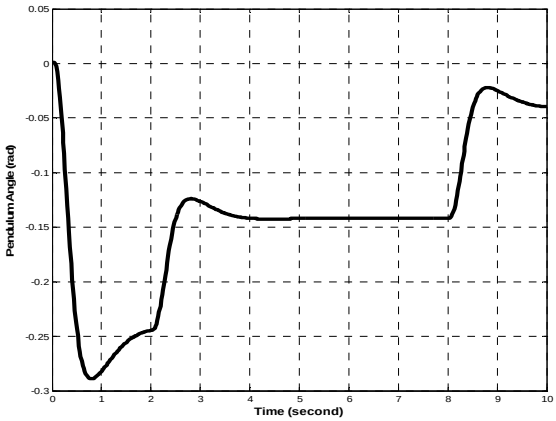
The simulation result of the cart performing a prescribed velocity reference input for two PID controllers is shown in Figure 4.26. The graph shows that the cart velocity is very close to the reference input given. There is a lagging but only less than 0.5 seconds from the cart velocity input reference. However the cart velocity can be controlled successfully to track the reference input given while at same time stabilised the angle pendulum rod.



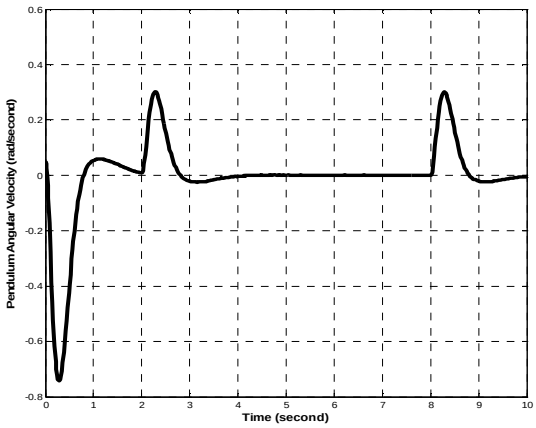
(a) Cart Position x_1



(b) Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4

Figure 4.26 Result SimMechanics model controlled with two PID for velocity reference

4.5. Summary

Progress to date of the design and simulation of spherical inverted pendulum wheelchair (SIPW) work is reported, which has demonstrated some various control methods of two DoF inverted pendulum system can be applied. Three control methods, pole placement, LQR and PID are capable of controlling the two DoF inverted pendulum's angle and the cart's position or velocity of the both Simulink model and SimMechanics model.

In this work, the two DoF inverted pendulum is modelled and simulated in both ways Simulink model and SimMechanics model. A SimMechanics model differs significantly from a Simulink model in how it represents the two DoF inverted pendulum system. A Simulink model represents mathematics of the two DoF inverted pendulum motion, i.e., the differential equations that predict the inverted pendulum system future state from its present state. The mathematical model enables Simulink to simulate the system. In contrast, a SimMechanics model represents a physical structure of the inverted pendulum, the geometric and kinematic relationships of its component bodies. Thus mathematical model no longer needed to be developed as SimMechanics converts this structural representation to an internal, equivalent mathematical model. The inverted pendulum is represented by connected block diagrams. The Physical modelling environment SimMechanics makes the task easier than the Simulink one where the dynamic system equation should be developed first before building the block diagram of the system.

Based on simulation results, it can be said that the pole placement method can be useful to design controllers for the two DoF inverted pendulums with satisfactory performance. Another alternative technique, LQR also provides satisfactory result in controlling the two DoF inverted pendulum. Compared with the pole placement method, the LQR result gives time to stabilise the system about one second slower but LQR use smaller gains controller of *K* matrix. It means that the system uses lesser effort or energy to stabilise the system. This is the advantage to use LQR method the poles are placed in such way through the cost function to get optimal gains for not only in stabilising the system but also in controlling effort. The conventional controller such as PID is implemented to control the two DoF inverted pendulum in this work. The PID controller widely known is a good controller to control the single-input-single-output (SISO) system. Thus with only one PID controller cannot be used to control the cart and the pendulum simultaneously as the two DoF inverted pendulum is a multi-input-multi-output (MIMO) system. Therefore in this work is implemented to use two PID controllers to control the two DoF inverted pendulum as can be seen in Figure 4.18. 'PID Cart' is used to control the cart whereas 'PID Pendulum' is used to control the pendulum. And the result shows that although there exists a lagging for less than one second but the cart is able to follow the cart prescribed velocity or on another word that the model of two DoF inverted pendulum system can be stabilized successfully by applying two PID controllers simultaneously if they are tuned properly.

CHAPTER 5

DOUBLE CARTS INVERTED PENDULUM CLOSED LOOP PID CONTROL

5.1. Introduction

In the previous chapter, it was shown that all three of control methods of the conventional controllers (pole placement, LQR and PID) are capable of controlling the two DoF inverted pendulum's angle and the cart's position or velocity of the both Simulink model and SimMechanics model. The PID control approach that was proposed to use two PID controllers solves the problem to control inverted pendulum system which is MIMO system successfully. But a problem still exists when the two DoF inverted pendulum system is given a prescribed velocity this leads to the cart’s velocity lagging the input by about 0.5 second. To overcome this problem, the author proposed an alternative inverted pendulum design that consists of two carts as seen in Figure 5.1. This means an inverted pendulum on cart system moving on top of another cart. This design gives possibility to control the bottom cart (vehicle) to get better characteristic on set point tracking while another cart moving on top to stabilise the inverted pendulum. Thus the author expects to get better characteristic for the system to track the set point such cart velocity reference input.

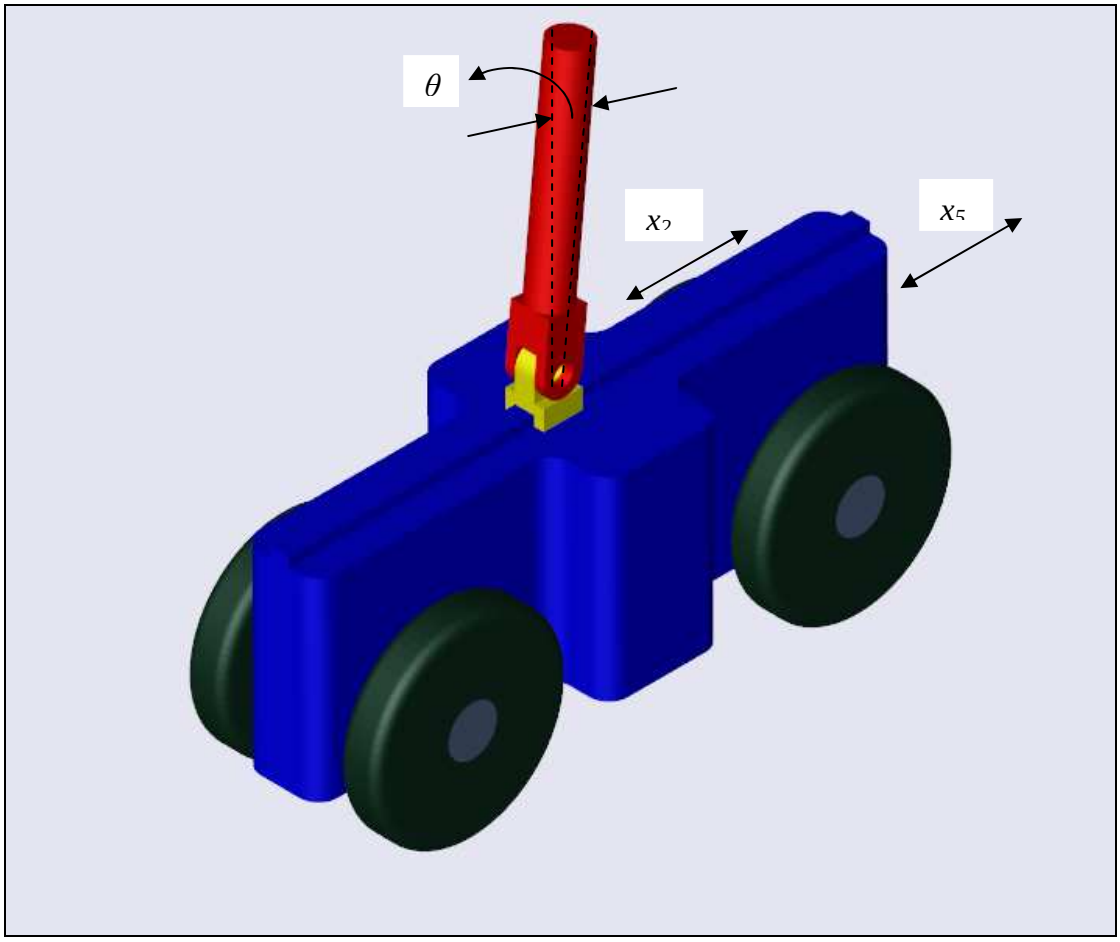


Figure 5.1 Double carts inverted pendulum model 3D visualisation using Solidworks in SimMechanics

5.2. SimMechanics Modelling of Double Carts Inverted Pendulum

To simulate this design, a SimMechanics model was developed. The Simmechanics block diagram for the top cart of double carts inverted pendulum is as shown in the Figure 5.2 below.

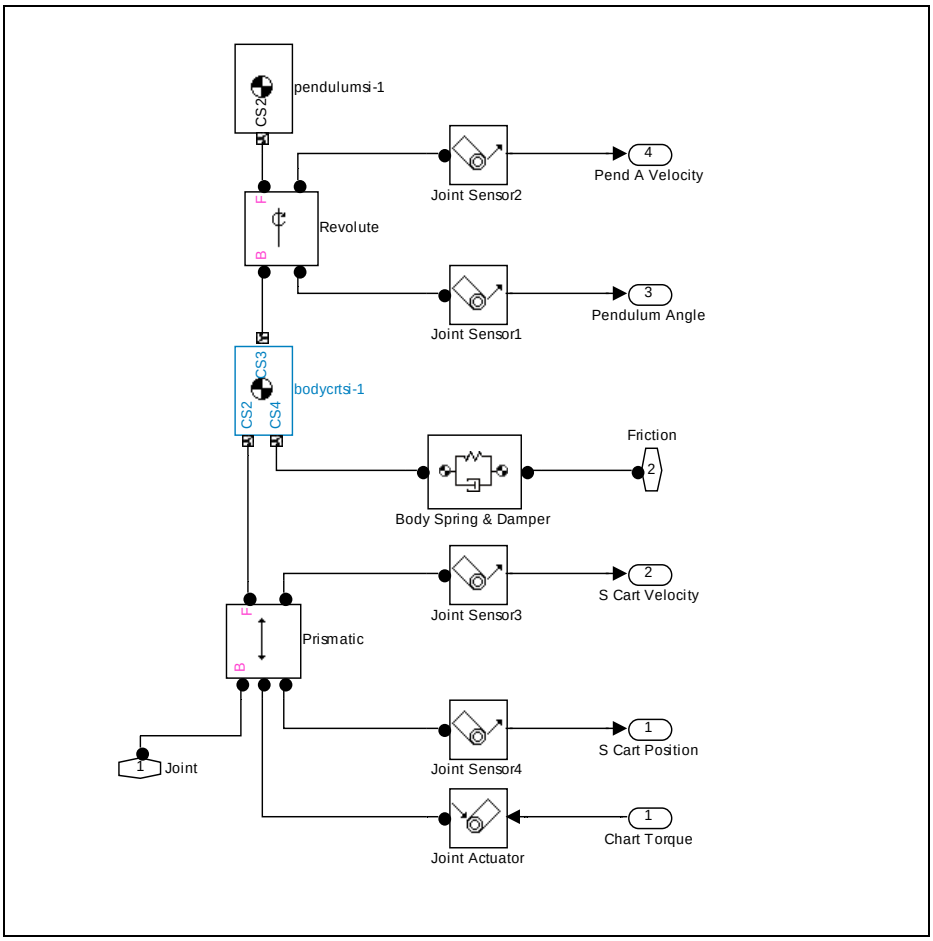


Figure 5.2 The SimMechanics block diagram for top cart of double carts inverted pendulum

The bottom cart or vehicle SimMechanics block diagram can be seen in Figure 5.3 below that structurally consists of a cart or vehicle body, two shafts and four wheels that totally have mass as 8.51 kg.

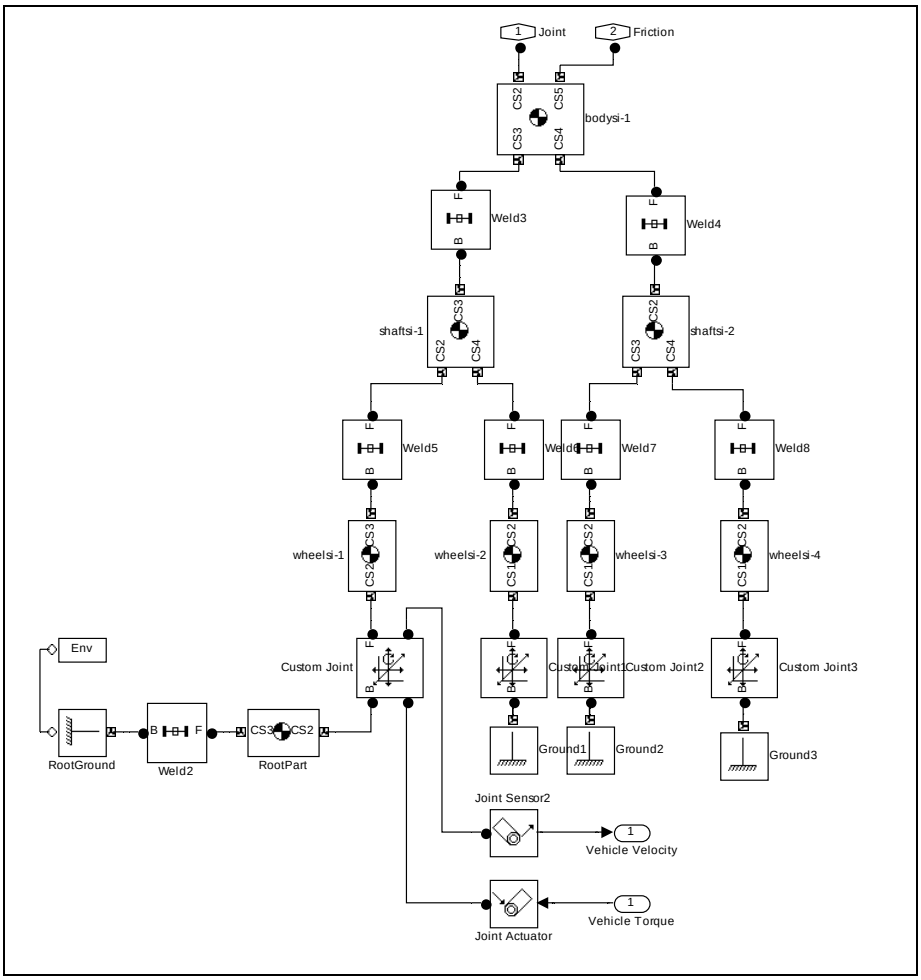


Figure 5.3 The SimMechanics block diagram for bottom cart or vehicle

5.3. Physical Properties of Double Carts Inverted Pendulum

As discussed in previous chapter SimMechanics offers four ways to visualise and animate machines:

- Use machine default body geometry
- Convex hull from body CS location
- Equivalent ellipsoid from mass properties
- External graphics file

The author used external graphics files to visualise this design the double carts inverted pendulum SimMechanics model because this one is more realistic visualisation than the others. Furthermore by using Solidworks can be got physical properties needed to simulate the systems.

The double carts inverted pendulum assembly was built in Solidwork first and then CAD assembly in Solidworks was imported to SimMechanics trough SimMechanics Link that converts Solidworks assembly document into XML document. The 3D visualisation of the double carts inverted pendulum physical model in SimMechanics model is as shown in Figure 5.1.

The Solidworks model provides the physical properties of the double carts inverted pendulum model as follows:

M	Mass of the top cart	0.37	(*kg*)
m	Mass of the pendulum	0.98	(*kg*)
k	Spring constant of cart friction	10	(*N/m*)
b	Damper constant of cart friction	1	(*N*sec/m*)
l	Length to pendulum CG	0.08	(*m*)
I	Moment Inertia of the pendulum	0.01	(*kg*m^2*)
M_v	Total mass of the vehicle (bottom cart)	8.51	(*kg*)

5.4. Control Design

To stabilise the double carts inverted pendulum system in SimMechanics model and to make possible the control of both carts and the pendulum, 3 PID controllers are proposed to control the system.

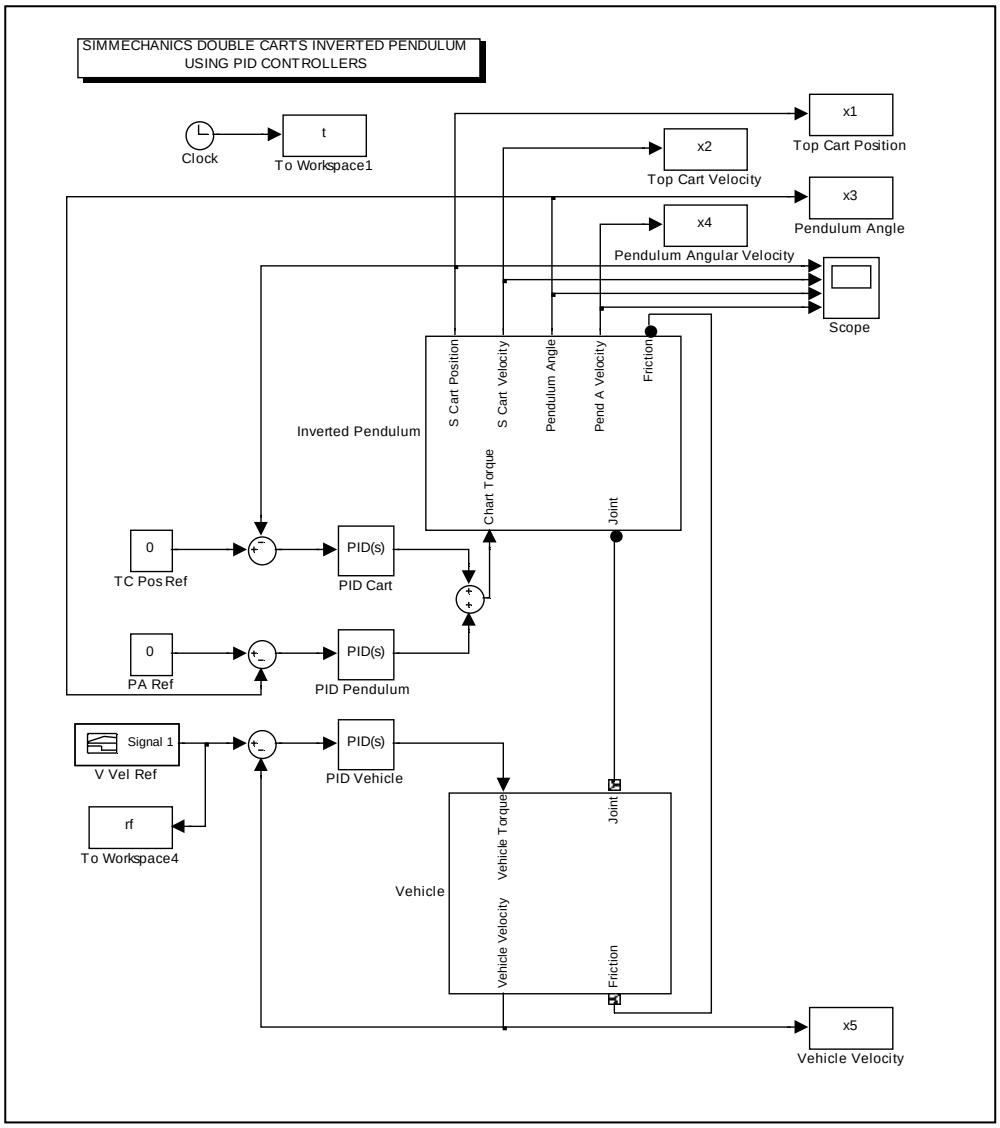


Figure 5.4 The double carts inverted pendulum with three PID controllers in SimMechanics

The Simmulink-SimMechanics block diagram with three different PID controllers strategy is shown as in Figure 5.4. The three PIDs are the ‘PID Cart’ which is used to control state variable x_1 that is top cart position, the ‘PID Pendulum’ which is used to control state variable x_3 that is pendulum angle and the ‘PID Vehicle’ which is used to control state variable x_5 that is bottom cart or vehicle velocity.

Again to determine the controller gains the Matlab PID tuner was applied. The controller gains used inside ‘PID Vehicle’ which controls bottom cart or vehicle velocity is as Figure 5.5 below. From step respond of PID tuning in the Figure 5.5 can be got parameter values inside the ‘PID Vehicle’ are as:

K_p = 617.9, K_i = 0.6, and K_d = 0.

From the Figure 5.5, it can be seen that the vehicle velocity of the double carts inverted pendulum system can be controlled and satisfy. The settling time is about 0.06 s and the rise time is about 0.02 s.

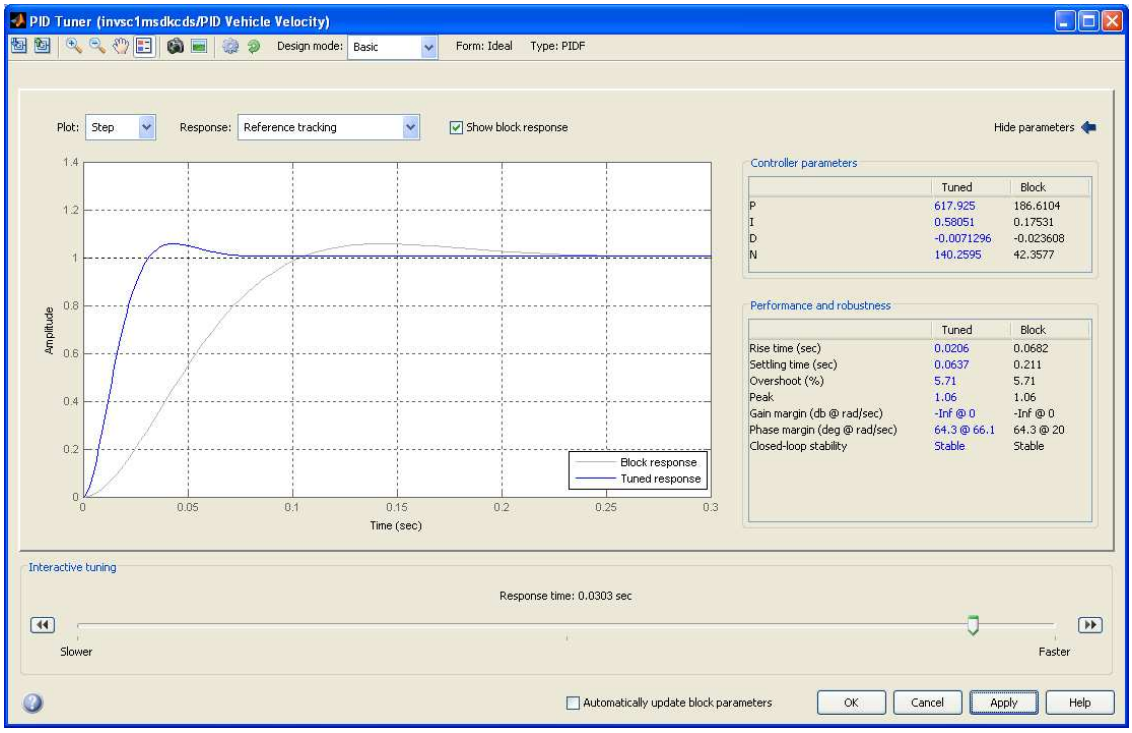


Figure 5.5 PID tuners for bottom cart or vehicle control

The controller gain used inside 'PID Pendulum' which controls pendulum's angle is as Figure 5.6 below. From step respond of PID tuning in the Figure 5.6 can be got parameter values inside the 'PID Pendulum' are as:

K_p = -346.7, K_i = 1.4, and K_d = 0.07

From the Figure 5.6, it can be seen that the pendulum angle of the double carts inverted pendulum system can be controlled and satisfy. The settling time is about 1.07 s and the rise time is about 0.01 s.

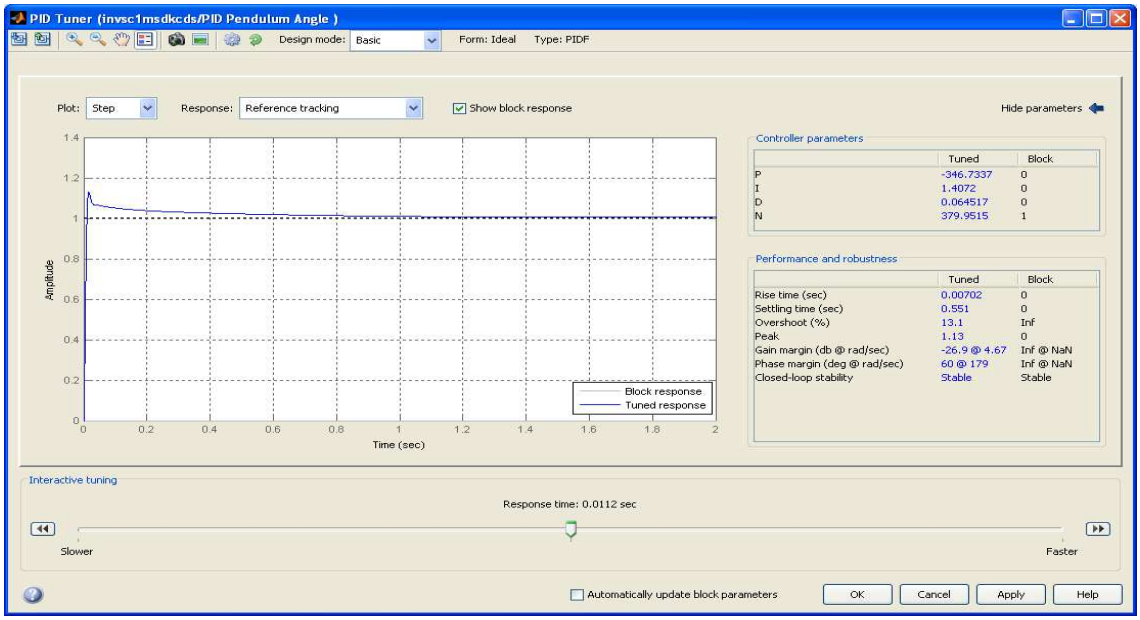


Figure 5.6 PID tuners for pendulum's angle control

The controller gain used inside 'PID Cart' which controls top cart's position is as Figure 5.7 below. From step respond of PID tuning in the Figure 5.7 can be got parameter values inside the 'PID Cart' are as:

K_p = -82.7, K_i = 12.2, and K_d = 0

From the Figure 5.7, it can be seen that the cart position of the double carts inverted pendulum system can be controlled and satisfy. The settling time is about 1.04 s and the rise time is about 0.03 s with the position constrains of the top cart movement $0.3 \leq x_1 \leq 0.3$.

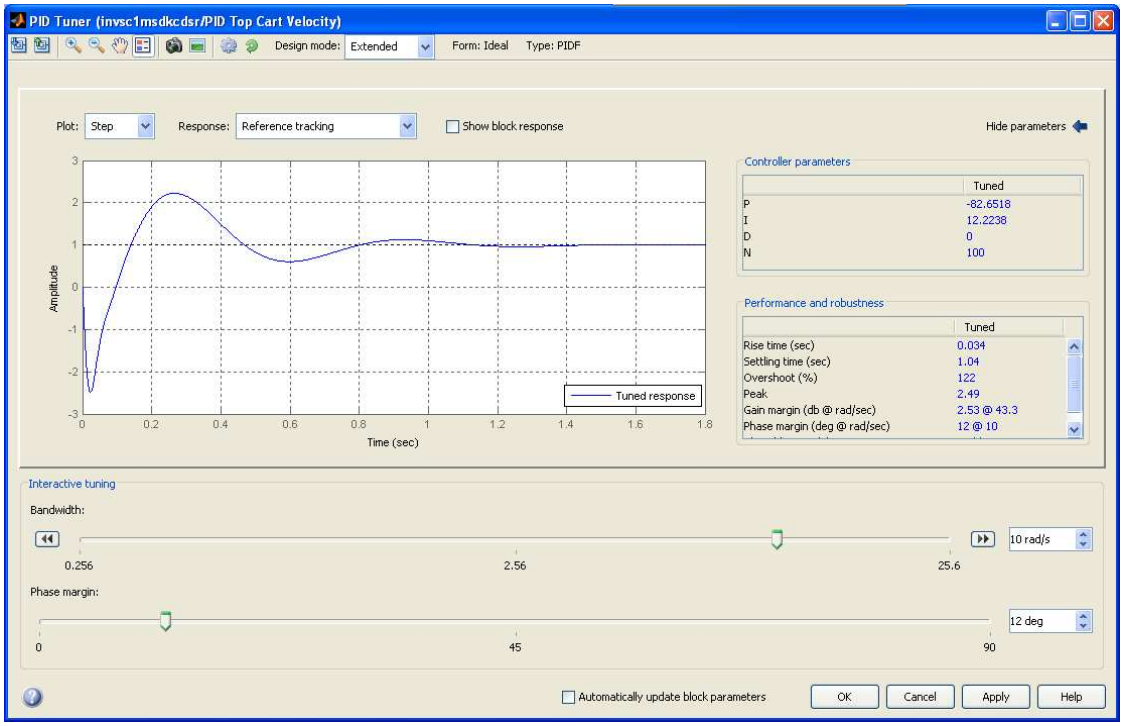
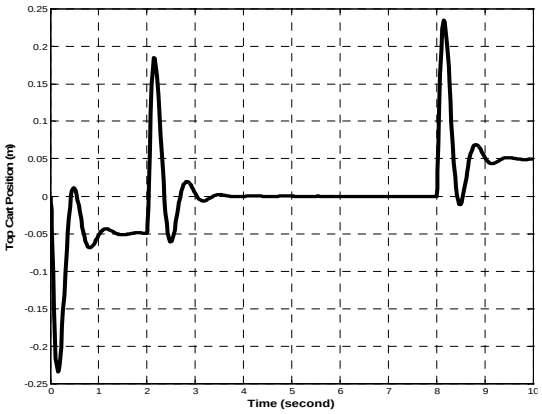


Figure 5.7 PID tuners for top cart position or velocity control

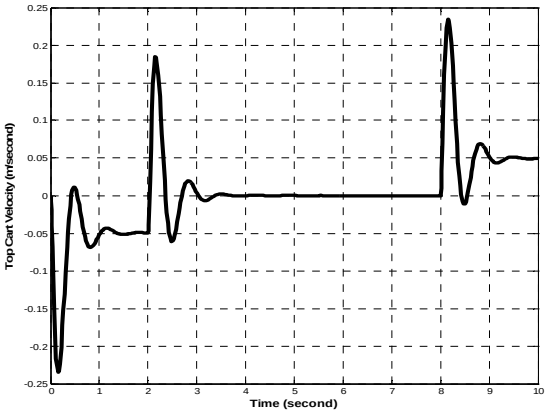
5.5. Simulation Result

The control strategy of using three PID controllers with the determined gain values hailed to successfully controlling the double carts inverted pendulum system using SimMechanics model as can be seen a diagram in Figure 5.8.

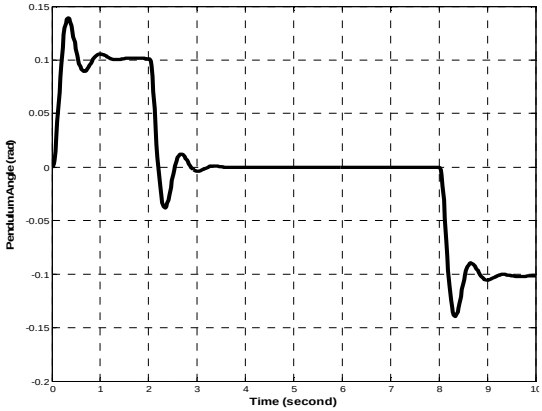
The 10 seconds simulation with vehicle velocity reference input as shown in Figure 5.8 shows that the design strategy by creating two carts inverted pendulum system gives very satisfying result. As we can see from the result that the bottom cart or vehicle velocity is perfectly tracking the reference input given while the top cart is moving on top of the vehicle and successfully stabilising the pendulum rod remain at about 0 radian.



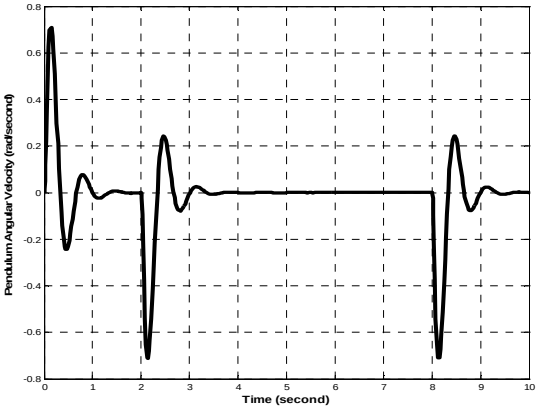
(a) Top Cart Position x_1



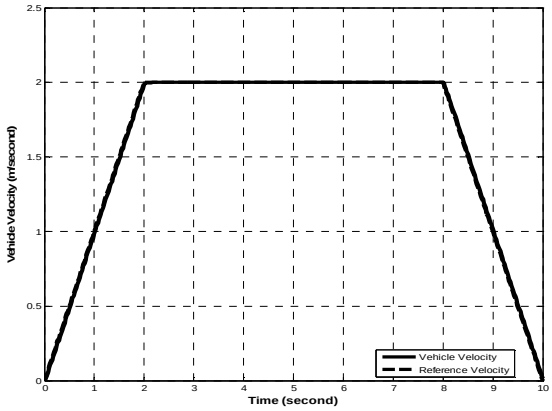
(b) Top Cart Velocity x_2



(c) Pendulum Angle x_3



(d) Pendulum Angular Velocity x_4



(e) Vehicle Velocity x_5 and Reference Velocity

Figure 5.8 SimMechanics double carts inverted pendulum with three PID controllers and bottom cart or vehicle velocity reference input

5.6. Summary

An alternative inverted pendulum design that consists of two carts (bottom cart (vehicle), top cart) and pendulum which is controlled using PID is presented in this chapter 5. This design gives possibility to control the bottom cart (vehicle) to get better characteristic on set point tracking while another cart moving on top to stabilise the pendulum stand upright. Three PID controllers are used to control both carts and the pendulum as can be seen in figure 5.4. 'PID Vehicle' is used to control bottom cart or vehicle velocity, 'PID Pendulum' is used to control pendulum angle and the 'PID Cart' is used to control top cart position or velocity. The simulation result shows that the system can be stabilised successfully while the vehicle follows the prescribed velocity. The vehicle is able to follow the reference very satisfactory with no lagging. From the result also shows that the sudden change of acceleration can lead to instability of the system.

The advantage of the two carts design is the system can follow the prescribed velocity given very satisfactory, with no lagging. This is because the velocity of the system is no longer associated with the pendulum. However, there is disadvantage in this design. As the pendulum is stabilised by the moving second or top cart on the vehicle or bottom cart, thus the movement of the top cart is constrained by the long dimension of the bottom cart. As a result the capability of the system to stabilise the pendulum is highly depend on the long dimension of the bottom cart. If the top cart need to move exceed the long dimension constrain when stabilising the pendulum, the design will fail to stabilise the system. Thus

this design has limitation in stabilising the systems. The next chapter will investigate and discuss the more complex design i.e. the spherical inverted pendulum design in sliding mode as the cart moves sliding not rolling which has four DoF i.e. two DoF rotation moving of the pendulum in *x*-axis and *z*-axis and two DoF horizontally moving of the cart in *x*-axis direction and *z*-axis direction.

CHAPTER 6

SPHERICAL INVERTED PENDULUM SLIDE MODE WITH SIMMECHANICS SIMULATION

6.1. Introduction

The SIPW is designed based on spherical inverted pendulum concept which has four DoF i.e. two DoF rotation moving of the pendulum in *x*-axis and *z*-axis and two DoF horizontally moving of the cart in *x*-axis direction and *z*-axis direction. In this chapter discusses how to develop and simulate the spherical inverted pendulum in slide mode i.e. the cart move sliding not rolling while stabilising the pendulum using SimMechanics. The schematics diagram of the spherical inverted pendulum in slide mode is shown in Figure 6.1. This spherical inverted pendulum slide mode is discussed first to get fundamental understanding of the spherical pendulum concept knowledge and control using SimMechanics before continuing to the design of the SIPW that based on rolling mode which is discussed in chapter 7.

A cart mounted inverted pendulum is relatively simple mechanical system which is inherently unstable and defined by highly nonlinear dynamic equations. The inverted pendulum model has been widely used as a teaching aid and in research experiments around the world because it’s an imaginable unstable

nonlinear dynamics problem commonly discussed in control engineering (Hauser et al., 2005), It is well established benchmark problem that provides many challenging problems to control design. Most of the researches focused on two types of inverted pendulum modelling and control: linear and rotational. For example, the swing-up and balancing problems were studied in (Chan, 2004) and (Liao, 2005) and the inverted-pendulum control problem of the multi-joint was studied in (Tsai, 2006) and etc.

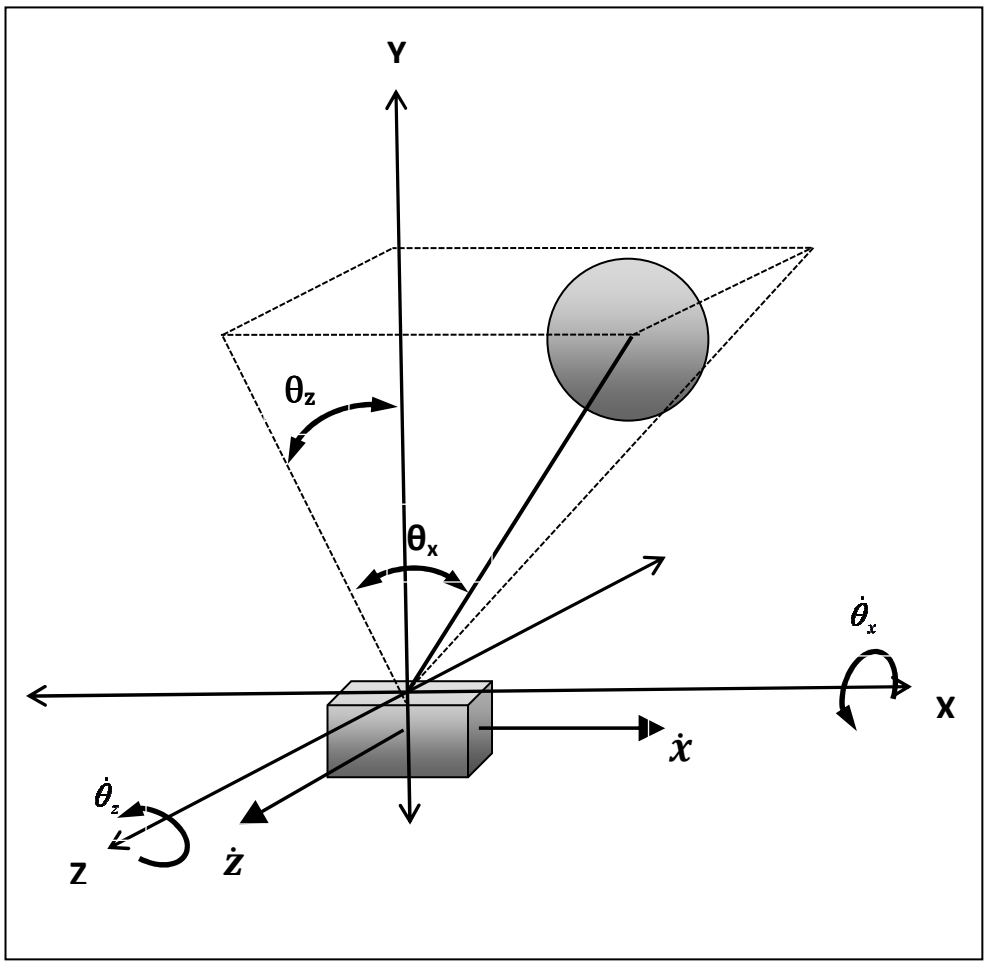


Figure 6.1 Schematic diagram of the spherical inverted pendulum in slide mode system with state variables geometry

Some researchers recently have considerable attention in investigating two-dimensional or spherical inverted pendulum problems. Generally speaking, aside from balancing control problem, the trajectory tracking control of such pendulums is also included due to an *x-y* mobile base. A mathematical modelling was proposed and then constructed a stable controller for a spherical inverted pendulum in (Liu *et. al.* 2007). In addition, there were some approaches developed for stabilising control of spherical inverted-pendulum systems, examples were based on PID controllers by (Wang, 2011), based on LQR by (Vicente, 2007), the cursive least-squares estimation (RLS) by (Yang et. al. 2000), decoupled neural network reference compensation technique by (Jung and Cho, 2004) and etc. Although a lot of control algorithms are researched in the systems control design, PID and LQR are still the common control approaches to overcome the problem and most widely used controller structure in the realization of a control system. As the spherical inverted pendulum system as a MIMO and LQR is easier to design than PID for MIMO system thus in this design the author has used a Linear Quadratic Regulator (LQR) Control strategy to stabilise the spherical inverted pendulum slide mode system.

6.2. Modelling of Spherical Inverted Pendulum Slide Mode

The spherical inverted pendulum model is developed by using Matlab-SimMechanics software. This software is permits to build models easily using drag and drop, and then visualise, animate and simulate them.

6.2.1. SimMechanics Modelling of Spherical Inverted Pendulum Slide Mode

As it has been explained above in SimMechanics, the dynamics system is made based on physic system. Physic system for the spherical inverted pendulum model that considers the dynamic of the pendulum and the cart motion consists of:

- 2 prismatic actuators (motor *x* and motor *z*), which represent motion of the cart in *x*-direction and *z*-direction.
- 1 custom joint that consist of 2 prismatic in *x*-direction and *z*-direction, which represent two DOFs of the cart motion.
- 1 custom joint that consist of 2 revolute around *x*-axis and *z*-axis, which represent two DOFs of the pendulum rotation.
- 2 bodies, which represent the cart and the pendulum
- 1 join initial condition, which is to set the initial angular position of both degrees of freedom of the pendulum
- 8 sensors which are to measure cart linear position and velocity in *x*-direction and *z*-direction, angular position and velocity of both degrees of freedom of the pendulum

- 2 input ports (U_x and U_z), to connect input signals to system
- 8 output ports (x1-TX, x2-TZ, x3-TXAV, x4-TZAV, x5-CPX, x6-CPZ, x7-CVX and x8-CVZ), to read output signals from system
- Reference systems that consist of machine environment, root ground, weld and root part.

The completed SimMechanics block diagrams of the spherical inverted pendulum system is as shown in Figure 6.2.

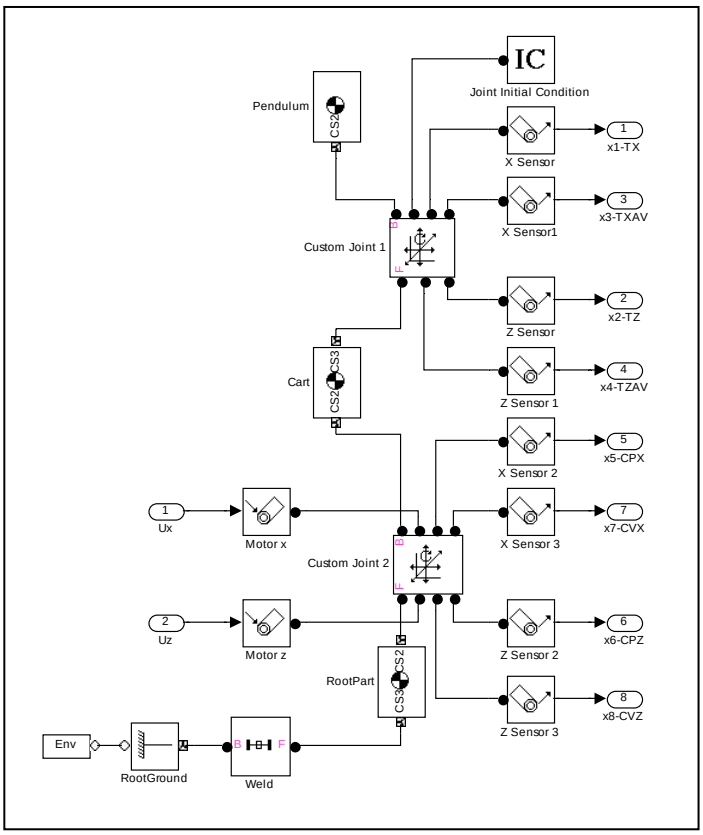


Figure 6.2 Simmechanics model of spherical inverted pendulum slide mode

As SimMechanics offers visualisation tool thus the spherical inverted pendulum system can be visualised in a more realistic model by using external graphics files. The spherical inverted pendulum assembly was built in Solidwork and then CAD assembly in Solidworks was imported to SimMechanics. The SimMechanics visualisation window display of the spherical inverted pendulum slide mode physically can be seen in Figure 6.3.

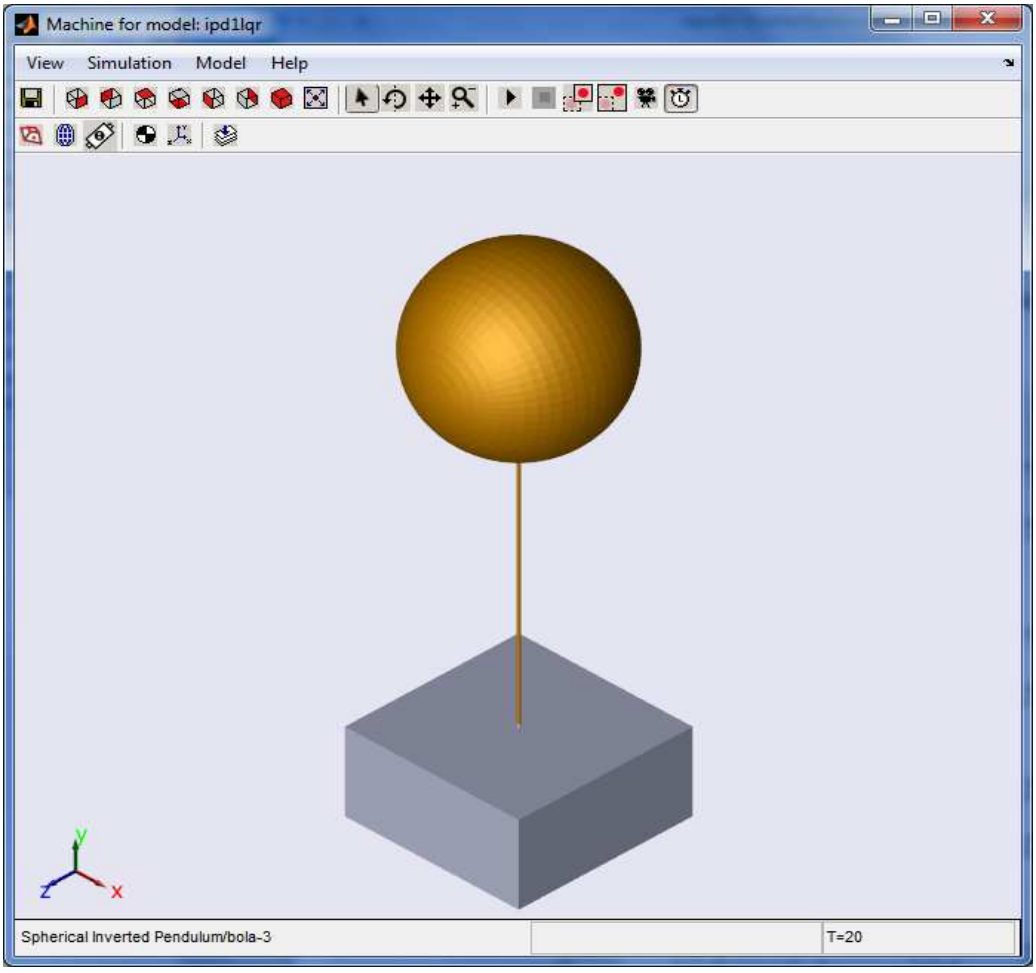


Figure 6.3 SimMechanics visualisation window display of the spherical inverted pendulum in slide mode system

6.2.2. Physical Geometry and Parameters of the Spherical Inverted Pendulum Slide Mode

The physical geometry description of the four DoF spherical inverted pendulum slide mode can be seen in the schematic diagram of the systems with all state variables geometry as in Figure 6.1. This model has 8 state variables. The physical geometry describing all 8 state variables of the systems model are defined as follows:

θ_x	=	Pendulum angle that rotates around *z*-axis	(*rad*)
$\dot{\theta}_x$	=	Pendulum angular velocity around *z*-axis	(*rad/s*)
θ_z	=	Pendulum angle that rotates around *x*-axis	(*rad*)
$\dot{\theta}_z$	=	Pendulum angular velocity around *x*-axis	(*rad/s*)
x	=	Cart position on *x*-axis direction	(*m*)
$\dot{x}$	=	Cart velocity on *x*-axis direction	(*m/s*)
z	=	Cart position on *z*-axis direction	(*m*)
$\dot{z}$	=	Cart velocity on *z*-axis direction	(*m/s*)

The four DoF inverted pendulum slide mode parameter values used for the simulation are the same as physical properties provided by the SolidWorks model, as shown in Figure 6.3 developed as:

Pendulum : $m = 65.5$ (Kg)

$Ix = Iy = Iz = 1.7$ ($Kg.\text{m}^2$)

Cart : $M = 20\ (Kg)$

$Ix = Iz = 0.51\ (Kg.m^2)$

$Iy = 0.83\ (Kg.m^2)$

Gravity : $g = -9.8\ (m/s^2)$

Distance between Pendulum's CG to Cart's CG : $l = 1\ (m)$

Dynamic behaviour of the system then can be seen by means of SimMechanics Simulation. But the system would be unstable as there is still no controller. For working with the stable system, it needs to calculate a controller which will need a mathematical model of it. Besides, that model must be linear. Thus at the next step in 6.2.3 and 6.2.4 will get a linear mathematical model.

6.2.3. Linearisation of Dynamic Model

By using Simulink-Matlab control design tool can be got a representation of system from the SimMechanics model directly into LTI state space model in the form:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

where x is the model's state vector, y is outputs and u is inputs.

The system's state vector in SimMechanics model as shown in Figure 6.2 was ordered as in Figure 6.4. :

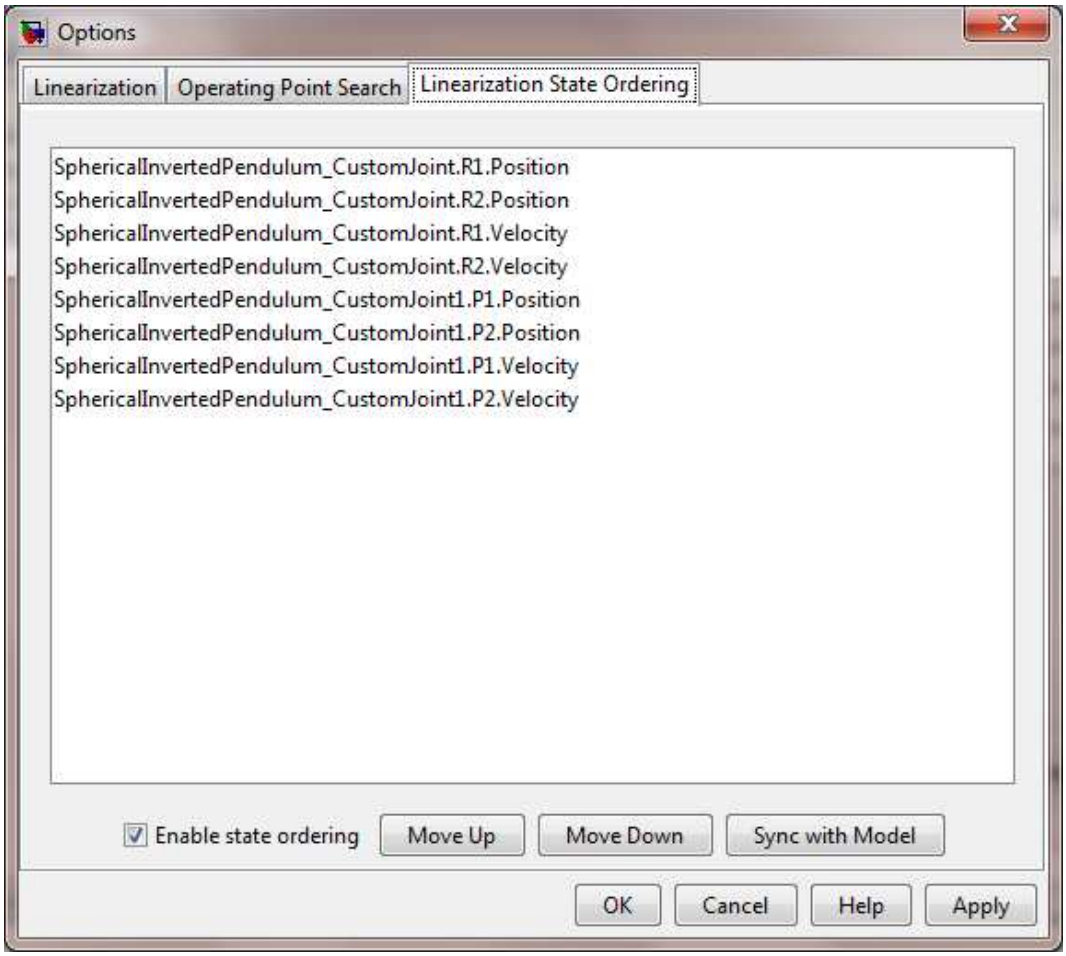


Figure 6.4 State variables of the spherical inverted pendulum slide mode ordering

This ordering implies that $x = \begin{bmatrix} \theta_x & \theta_z & \dot{\theta}_x & \dot{\theta}_z & x & z & \dot{x} & \dot{z} \end{bmatrix}^T$ and $u = \begin{bmatrix} u_x & u_z \end{bmatrix}^T$, where θ_x is the angle between pendulum and projected line of pendulum into *x-y* plane, θ_z is the angle between pendulum and projected line of pendulum into *z-y* plane, *x* is the cart position in the *x* direction and *z* is the cart position in the *z* direction (as can be seen in Figure 6.3). Since the pendulum must

be kept on vertical position, therefore the system model will be linearised near zero or $\theta_x \approx 0$, $\theta_z \approx 0$ thus pendulum initially angles position set to zero. After executing Simulink control design tool will get all necessary information to make a state space model of the system as presented in 6.2.3.

6.2.4. Mathematical Model

After executing Simulink control design tool the result of the system linearisation around $\theta_x = 0$ and $\theta_z = 0$ gave properties matrices of the state space model, *A* and *B*, as:

A =

	x1	*x2*	*x3*	*x4*	*x5*	*x6*	*x7*	*x8*
x1	0	0	1	0	0	0	0	0
x2	0	0	0	1	0	0	0	0
x3	37.89	0	0	0	0	0	0	0
x4	0	37.89	0	0	0	0	0	0
x5	0	0	0	0	0	0	1	0
x6	0	0	0	0	0	0	0	1
x7	29.02	0	0	0	0	0	0	0
x8	0	-29.02	0	0	0	0	0	0

B =

	u1	*u2*
x1	0	0
x2	0	0
x3	0.04517	0
x4	0	-0.04517
x5	0	0
x6	0	0
x7	0.04629	0
x8	0	0.04629

The spherical inverted pendulum sliding mode also can be presented in state space using Simulink modelling as shown in Figure 6.5 and the matrices *A* and *B* put into the gain parameters *A* and *B* in the model.

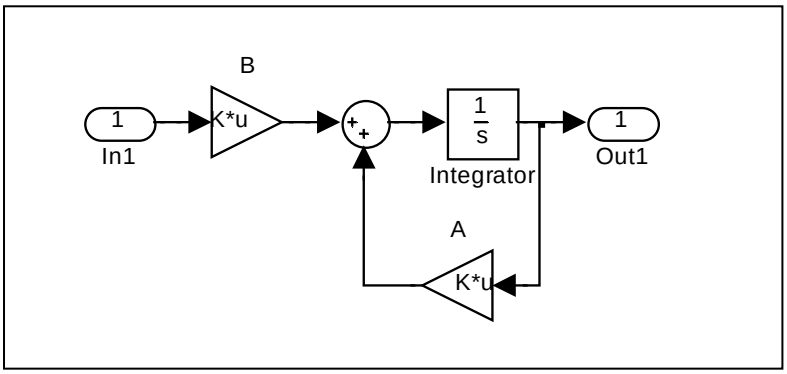


Figure 6.5 The state space model of the spherical inverted pendulum

6.3. Control Design

One of the most common optimal controllers is the LQR (Linear Quadratic Regulator) controller. The quadratic criterion or cost function to be minimized is as equation (4.16) i.e :

$$J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$$

And the gain matrix *K* to minimise this cost function need to be found. Minimisation the cost function *J*, results in moving *x* to zero via little control energy and state distance from origin as possible. The matrices $Q > 0$ and $R > 0$,

are symmetric and positive definite gain matrices. The gain K (matrix) is determined by first solving the algebraic Riccati equation (4.17) i.e.:

$$PA + A^T P - PBQ^{-1}B^T P + Q = 0$$

Then K can be calculated as equation (4.18) i.e.:

$$K = Q^{-1}B^T P$$

This calculation can be difficult by hand. However the MATLAB lqr command can be used. The Matlab lqr command solves for the gain vector K given A, B, Q, and R. This approach is used in our LQR design and simplest way is to assume as:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.00001 & 0 \\ 0 & 0.00001 \end{bmatrix}$$

Then K matrix will be produced an optimal controller could be found by using code K = lqr (A, B, Q, R), m-file code in Matlab. Using all parameter values of the spherical inverted pendulum and choosing Q and R as above, the following controller gains K matrix was determined as:

$$K = \begin{bmatrix} 4266.1 & 0 & 1196.7 & 0 & -316.2 & 0 & -566.5 & 0 \\ 0 & -4266.1 & 0 & -1196.7 & 0 & -316.2 & 0 & -566.5 \end{bmatrix}$$

This feedback controller gain was used in the simulation of the spherical inverted pendulum SimMechanics model that will be discussed in next section. The Simulink block diagram of model with feedback control using LQR method is as Figure 6.6.

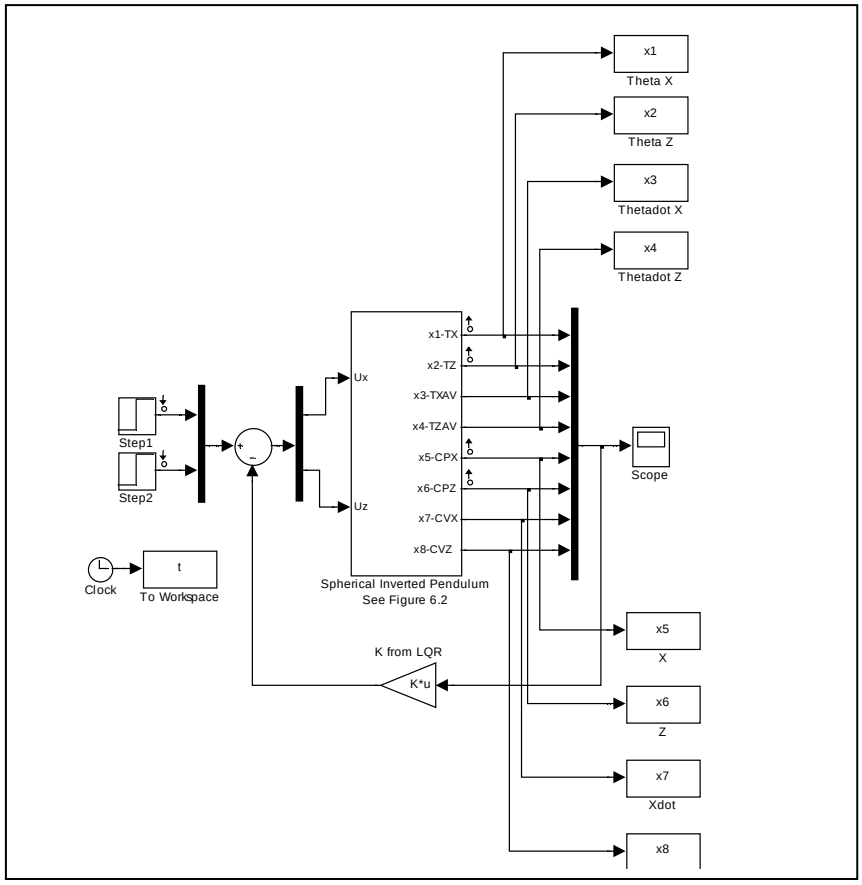
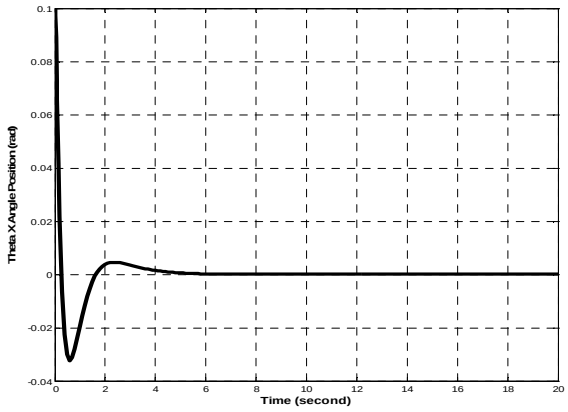


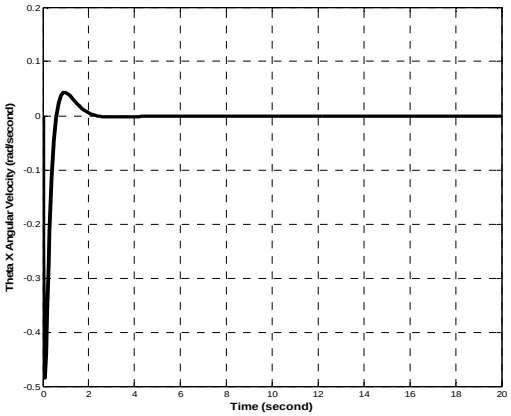
Figure 6.6 The spherical inverted pendulum with LQR feedback control

6.4. Simulation Result

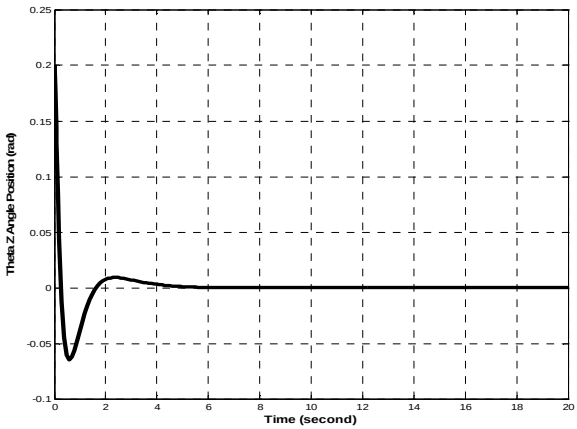
As discussed in modelling section the spherical inverted pendulum SimMechanics model produces 8 outputs i.e. x1-TX, x2-TZ, x3-TXAV, x4-TZAV, x5-CPX, x6-CPZ, x7-CVX and x8-CVZ which represent system's state variables $x = \begin{bmatrix} \theta_x & \theta_z & \dot{\theta}_x & \dot{\theta}_z & x & z & \dot{x} & \dot{z} \end{bmatrix}^T$.



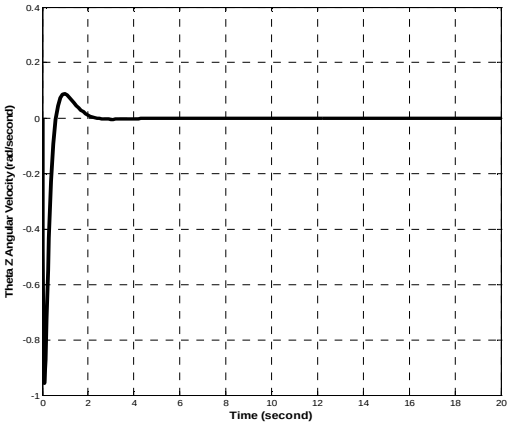
(a) Angle Position of θ_x



(b) Angular Velocity of θ_x



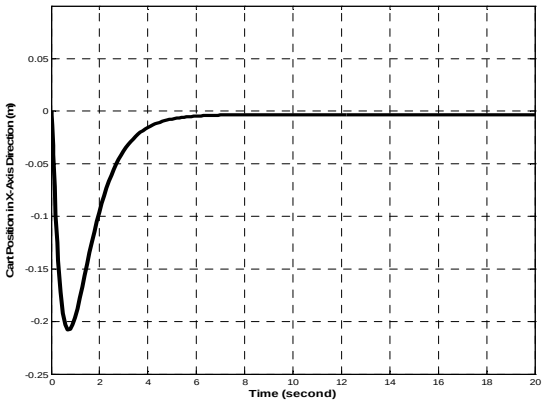
(c) Angle Position of θ_z



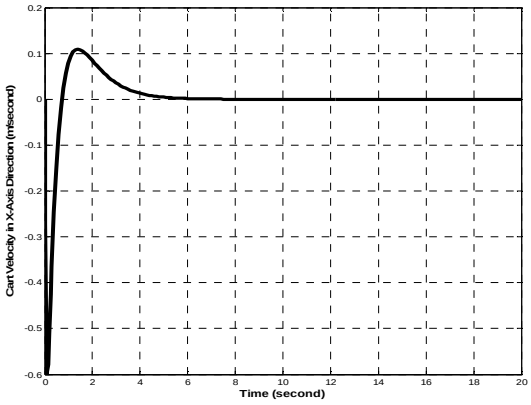
(d) Angular Velocity of θ_z

Figure 6.7 Pendulum of the System Responses

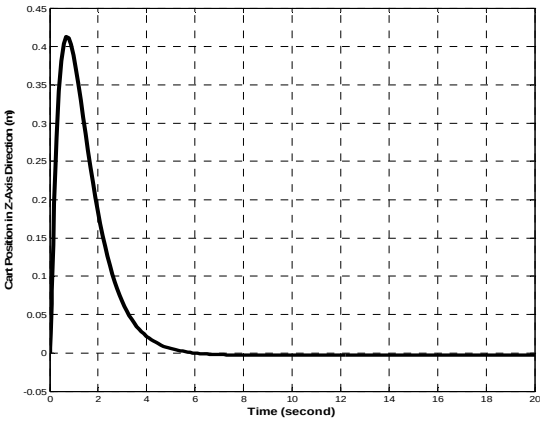
For simulation the system is set an initial condition of the pendulum as θ_x at 0.1 rad and θ_z at 0.2 rad. It means that the cart needs to move at the same time to eliminate the pendulum angles θ_x and θ_z to zero radians so that the pendulum is on vertically straight position and make the stabilisation of the system. The pendulum systems responses can be seen in Figure 6.7 and the cart responses can be seen in Figure 6.8.



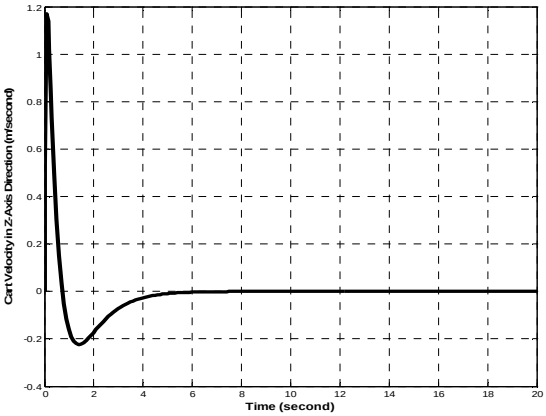
(a) Cart Position in X



(b) Cart Velocity in X



(c) Cart Position in Z



(d) Cart Velocity in Z

Figure 6.8 Cart of the System Responses

It can be seen that LQR controller can realise stabilisation of spherical inverted pendulum with good robustness. Figures show the performance of the spherical inverted pendulum with LQR controller has settling time about 4 seconds and the percentage overshoot about 1 %. The displacements of the cart when the initial condition of the pendulum applied as θ_x = 0.1 rad and θ_z = 0.2 rad to the system are -0.2 meter in *x*-axis direction and 0.4 meter in *z*-axis direction. Thus the LQR controller is able to stabilise the spherical inverted pendulum low percentage overshoot. In term of steady state error the controller had shown very outstanding performance by giving zero error at time 5 seconds and more. The spherical inverted pendulum slide model which is developed using SimMechanics and using LQR control gives very satisfactory simulation result.

6.5. Summary

This chapter has described how to simulate the spherical inverted pendulum, a dynamics of multibody system, in sliding mode which is successfully designed with SimMechanics. The control strategy used is based on the LQR feedback control for the stabilisation of the spherical inverted pendulum system. The simulation result study has been done in Simulink environment shows that LQR control strategy is capable to control multi input and multi output of spherical inverted pendulum system successfully and gives satisfy of controlling the spherical inverted pendulum system.

Moreover this chapter also shows that the use of SimMechanics is an interesting and important add-on to the simulation environment Simulink. It allows to include the SimMechanics spherical inverted pendulum model subsystems into Simulink models. Furthermore especially for non-experts will benefit from the visualisation tools provided in SimMechanics that it can facilitate the modelling and the interpretation of results. Using SimMechanics was impressive to realise the spherical inverted pendulum modelling. It can be say in another words that SimMechanics proved to be a very suitable tool for multibody dynamics simulation. Thus in the next chapter 7 the author discusses the development of the SIPW design and simulation using SimMechanics with LQR feedback control strategy.

CHAPTER 7

SPHERICAL INVERTED PENDULUM WHEELCHAIR

7.1. Introduction

In this chapter discusses the development of the Spherical Inverted Pendulum Wheelchair (SIPW) design and simulation using SimMechanics with LQR feedback control method. The wheelchair concept is based upon the control of spherical inverted pendulum concept is a robot that is balanced on a ball as shown in Figure 7.1. The upper cylindrical tube represents simplify wheelchair occupant. The two plates supported by four columns represent physical properties of the wheelchair. The ball is the method of creating traction between the SIPW and the ground.

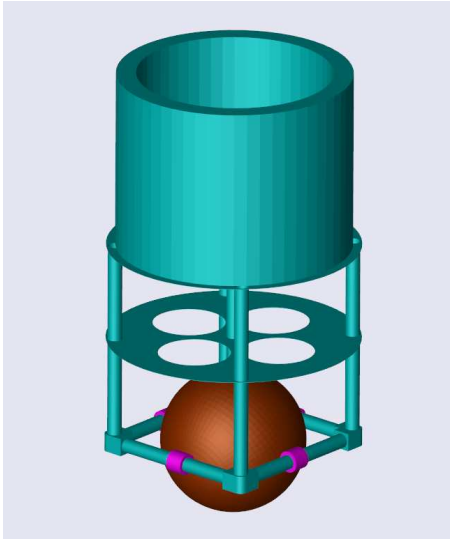
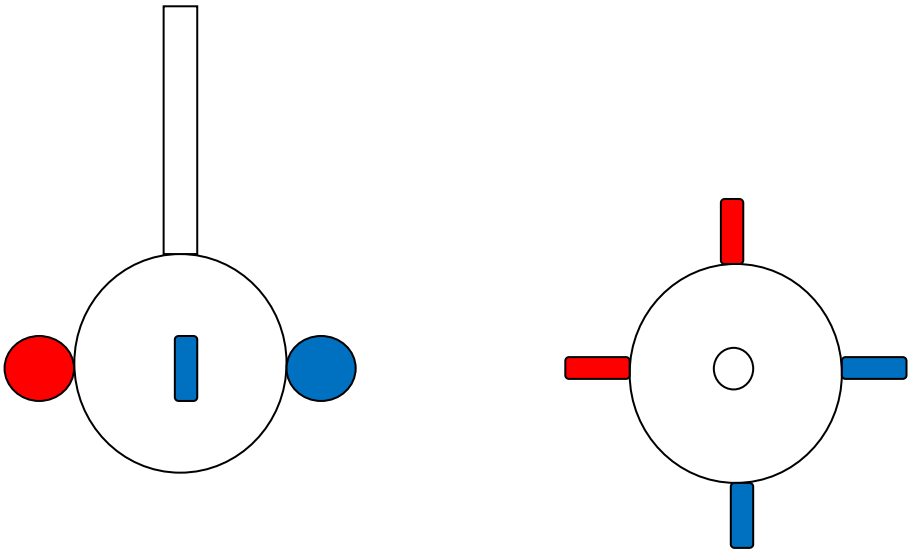


Figure 7.1 The SIPW concept

This design arrangement with ball acts as spherical wheel for the wheelchair, allowing the wheelchair moves in any direction. The wheelchair is balanced by driving the ball such that the wheelchair does not fall over but moves. This can be done by using mouse ball drive mechanism concept. This driving mechanism is via actuated rollers which drive the ball to produce motion. The mouse ball drive mechanism construction is considered as can be seen in Figure 7.2. The arrangements of ball and rollers are as orthogonal fixed four rollers at ball centre axis. The rollers consist of two driven rollers (coloured in red) and two idler rollers (coloured in blue).



Side View

Top View

Figure 7.2 Mouse ball driving mechanism concepts

7.2. Modelling of Spherical Inverted Pendulum Wheelchair

This section describes the derivation of dynamic modelling of spherical inverted pendulum wheelchair (SIPW) required as basis for stabilising controller design development. This starts with making some assumptions to create simplified model of SIPW which consists of seven rigid bodies i.e. the ground, the ball, the body, and four rollers. The Lagrangian method is used to derive SIPW model equation of motion.

7.2.1. Assumptions and Parameters

The dynamics of the actual SIPW are very complex and difficult to develop, thus here is a need to create a simplified dynamic model. The following assumptions are used to simplify the SIPW model made in one planar system model which dynamic modelling can be derived using Lagrangian approach:

- No deformations happen on all parts of SIPW: the ball, the four rollers, the body and the ground are assumed as rigid bodies.
- No slip happens between the ball and the rollers and between the ball and the ground.
- The friction between the ball and the ground and between the rollers and the ball are modelled as viscous damping.
- The control inputs are as torques that applied to the body and the ball.
- The motion is decoupled in two separate planar models (in *x-y* plane and *z-*

y plane) and that equations of motion in these two planes are assumed identical.

- The body angle and roller angle (= motor angle) can be measured directly.
- The ball moves only horizontally and does not move in the vertical direction thus this model is designed to move on flat floor without steep inclination.

The sketch of the simplified SIPW in one planar model in *x-y* plane showing the coordinates and measured parameters i.e. the body angle and the roller angle is shown in Figure 7.3.

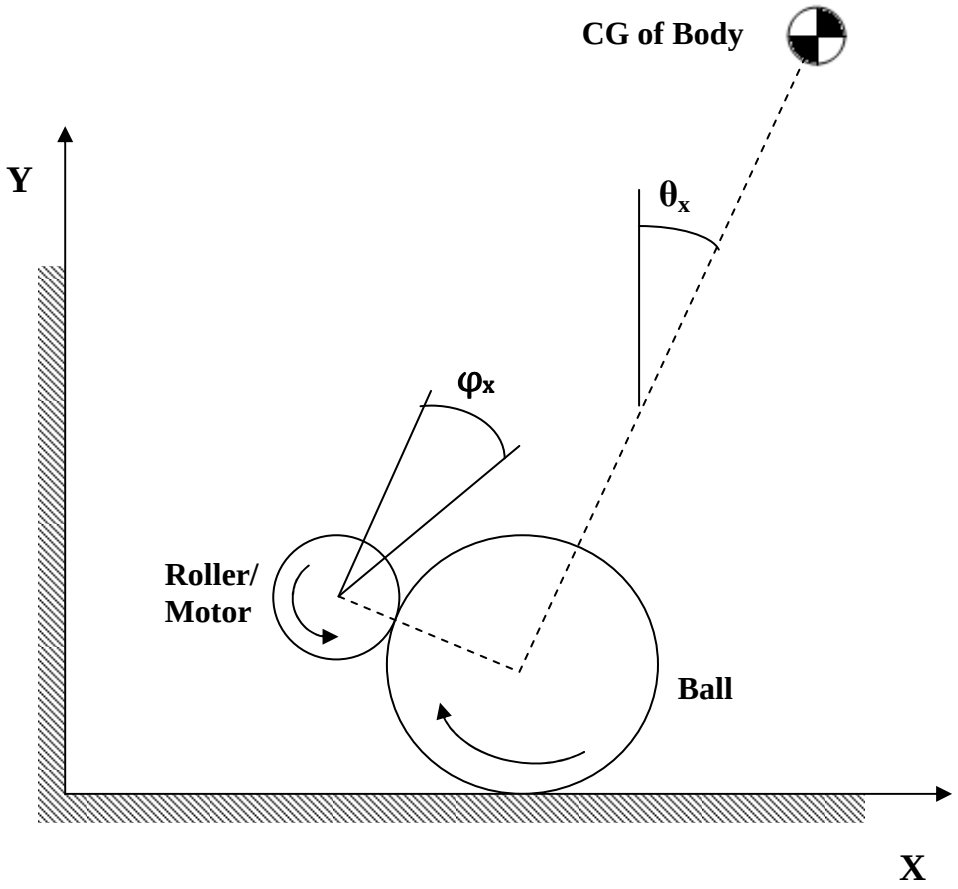


Figure 7.3 Sketch of the simplified SIPW in the *x-y* plane model

The physical parameters describing the simplified SIPW model which are used to derive the mathematical dynamics modelling for one planar i.e. in *x-y* plane are defined as follows:

M_B	=	Mass of the body	= 37.342 (Kg)
M_b	=	Mass of the ball	= 2.353 (Kg)
l	=	Distance between ball CG and body CG	= 0.737 (m)
R_b	=	Radius of the ball	= 0.200 (m)
R_r	=	Radius of the roller (motor)	= 0.030 (m)
n	=	Gear ratio = R_r/R_b	
I_b	=	Inertia Moment of the ball	= 0.061 (Kgm^2)
I_{Bx}	=	Inertia Moment of the body about x-axis	= 3.800 (Kgm^2)
I_{Bz}	=	Inertia Moment of the body about z-axis	= 3.800 (Kgm^2)
I_{Mx}	=	Inertia Moment of the roller/motor in x-axis	= $0.4x10^{-4}$(Kgm^2)
I_{Mz}	=	Inertia Moment of the roller/motor in z-axis	= $0.4x10^{-4}$(Kgm^2)
μ_{Bb}	=	Friction coefficient between body and ball	= 0.2
μ_{bg}	=	Friction coefficient between ball and ground	= 0.2

7.2.2. Derivation of Mathematical Model

The dynamics modelling of the simplified SIPW as has been explained above that only developed in one planar model i.e. in *x-y* plane using Lagrangian approach. The code that is used can be seen as in appendices 5-8. Another one planar model i.e. in *z-y* plane is assumed to be identical dynamics model.

The development of the mathematical model of the simplified SIPW using Lagrangian approach starts with determining the energy of the dynamics systems i.e. the energy of the body, the ball, and the motor (roller). Each these parts have kinetic energy and potential energy as follows:

The ball energy:

The ball angle position =

$$\theta_x + n\varphi_x \tag{7.1}$$

The ball position in *x* direction =

$$x_b = R_b\left(\theta_x + n\varphi_x\right) \tag{7.2}$$

Angular velocity of the ball =

$$\dot{\theta}_x + n\dot{\varphi}_x \tag{7.3}$$

Linear velocity of the ball in *x* direction =

$$\dot{x}_b = R_b\left(\dot{\theta}_x + n\dot{\varphi}_x\right) \tag{7.4}$$

Thus,

Linear kinetic energy of the ball is:

$$T_{lkb} = \frac{M_b R_b^2\left(\dot{\theta}_x + n\dot{\varphi}_x\right)^2}{2} \tag{7.5}$$

Rotational kinetic energy of the ball is:

$$T_{rkb} = \frac{I_b\left(\dot{\theta}_x + n\dot{\varphi}_x\right)^2}{2} \tag{7.6}$$

As assumed that there is no vertical movement of the SIPW thus the potential energy of the ball should be,

$$V_b = 0 \tag{7.7}$$

The body energy:

The body angle position =

$$\theta_x \tag{7.8}$$

The body position in *x* direction =

$$x_B = x_b + l\sin(\theta_x) \tag{7.9}$$

The body position in *y* direction =

$$y_B = l\cos(\theta_x) \tag{7.10}$$

Angular velocity of the body =

$$\dot{\theta}_x \tag{7.11}$$

Linear velocity of the body in *x* direction =

$$\dot{x}_B = \dot{x}_b + l\cos(\theta_x)\dot{\theta}_x \tag{7.12}$$

Linear velocity of the body in *y* direction =

$$\dot{y}_B = -l\sin(\theta_x)\dot{\theta}_x \tag{7.13}$$

Thus,

Linear kinetic energy of the body is:

$$T_{lkB} = \frac{M_B\left(\dot{x}_b + l\cos(\theta_x)\dot{\theta}_x\right)^2}{2} \tag{7.14}$$

Substitute $\dot{x}_b$ with equation (7.4) and solve the equation will get:

$$T_{lkB} = \frac{M_B\left(l^2\dot{\theta}_x^2 + 2\cos(\theta_x)nlR_b\dot{\varphi}_x\dot{\theta}_x + 2\cos(\theta_x)lR_b\dot{\theta}_x^2 + n^2R_b^2\dot{\varphi}_x^2 + 2nR_b^2\dot{\varphi}_x\dot{\theta}_x + R_b^2\dot{\theta}_x^2\right)}{2} \tag{7.15}$$

Rotational kinetic energy of the body is,

$$T_{rkB} = \frac{I_{Bx}\dot{\theta}_x^2}{2} \tag{7.16}$$

The potential energy of the body is,

$$V_B = M_B g l \cos(\theta_x) \tag{7.17}$$

The motor energy:

Motor position =

$$\theta_x + \varphi_x \tag{7.18}$$

Motor angular velocity =

$$\dot{\theta}_x + \dot{\varphi}_x \tag{7.19}$$

Thus,

Angular kinetic energy of the motor is,

$$T_{rkm} = \frac{I_{Mx}\left(\dot{\theta}_x + \dot{\varphi}_x\right)^2}{2} \tag{7.20}$$

Then,

The Lagrangian L is the total kinetic energy minus the total potential energy:

$$L = T - V$$

$$L = T_{lkb} + T_{rkb} + T_{lkB} + T_{rkB} + V_b + V_B \tag{7.21}$$

And the Lagrangian equation dictates as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Fi \tag{7.22}$$

Where,

$$q = \begin{bmatrix} \theta_x \\ \varphi_x \end{bmatrix}$$

And the force F matrix is determined by:

- The effects of the viscous friction between ball and ground :

$$\tau_{bgx} = -\mu_{bg}\dot{\theta}_x$$

- The torque applied to the ball by the roller (motor):

  The model of voltage controlled DC motor torque with negligible inductance is as (Yamamoto, 2008):

$$\tau_{mx} = \frac{K_t\left(E_x + K_b\dot{\varphi}_x\right)}{R_m}$$

  Where E is motor voltage, K_t is the motor torque constant, K_b is the motor back emf constant and R_m is the motor resistance.

- The effects of the viscous friction between ball and body:

$$\tau_{Bbx} = -\mu_{Bb}\dot{\varphi}_x$$

Thus the force matrix is determined as:

$$F = \begin{bmatrix} \tau_{bgx} \\ \tau_{mx} + \tau_{Bbx} \end{bmatrix}$$

Evaluating equation (7.22) using Matlab symbolic package and rearranging gets the equation of motion in the following form:

$$M_x(q,\dot{q})\ddot{q} + R_x(q,\dot{q}) = F_x(q,\dot{q},v_x) \tag{7.23}$$

Where,

M_x is the 2x2 mass matrix with columns, $Mx(:,1)$ and $Mx(:,2)$ as:

$$M_x(:,1) = \begin{bmatrix} I_{Bx} + I_{Mx} + I_b + l^2 M_B + R_b^2 M_B + R_b^2 M_b + 2lR_b M_B \cos(\theta_x) \\ I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + nlR_b M_B \cos(\theta_x) \end{bmatrix}$$

$$M_x(:,2) = \begin{bmatrix} I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + lR_b M_B \cos(\theta_x) \\ I_{Mx} + I_b n^2 + n^2 R_b^2 M_B + n^2 R_b^2 M_b \end{bmatrix}$$

R_x is the 2x1 remainder matrix as:

$$R_x = \begin{bmatrix} -lR_b M_B \cos(\theta_x)\dot{\theta}_x^2 - glM_B \sin(\theta_x) \\ -nlR_b M_B \dot{\theta}_x^2 \sin(\theta_x) \end{bmatrix}$$

Fx is the 2x1 force matrix as:

$$F_x = \begin{bmatrix} \tau_{bgx} \\ \tau_{mx} + \tau_{Bbx} \end{bmatrix}$$

The equation (7.23) can be rearranged to be:

$$\ddot{q} = M_x^{-1} + (F_x - R_x) \tag{7.24}$$

And this equation can be written in nonlinear state space standard form as:

$$\dot{x} = f(x) \tag{7.25}$$

Where,

$$x = \begin{bmatrix} \theta_x & \varphi_x & \dot{\theta}_x & \dot{\varphi}_x \end{bmatrix}^T$$

7.2.3. Linearisation of Dynamic Model

As this SIPW design uses LQR control method thus it requires that the equation model of the SIPW system be linear. The nonlinear state space equation (7.25) is linearised using a first order approximation as follows:

$$\dot{\hat{x}} = f(\bar{x}) + J(\bar{x})\hat{x} \tag{7.26}$$

Where,

$\hat{x}$ is the linearised state = $[q \quad \dot{q}]^T$

$\bar{x}$ is the point which linearise about = $[0 \quad \varphi_x \quad 0 \quad 0]^T$

$J(\bar{x})$ is the Jacobian at the point which linearise about

Taking $\hat{x}$ to be the state vector *x*, so that the equation (7.26) can be rewritten in standard linear state space form as follows:

$$\dot{x} = Ax + Bu \tag{7.27}$$

Where by using Matlab symbolic package for the *x*-*y* plane gets the following matrices:

$$A_x = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_x(3,1) & 0 & A_x(3,3) & A_x(3,4) \\ A_x(4,1) & 0 & A_x(4,3) & A_x(4,4) \end{bmatrix}$$

$$B_x = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_x(3,1) & 0 \\ B_x(4,1) & 0 \end{bmatrix}$$

Where,

$$A_x(3,1) = \frac{glM_B\left(I_{Mx} + I_b n^2 + n^2 R_b^2 M_B + n^2 R_b^2 M_b\right)}{D}$$

$$A_x(3,3) = -\frac{\mu_{bg}\left(I_{Mx} + I_b n^2 + n^2 R_b^2 M_B + n^2 R_b^2 M_b\right)}{D}$$

$$A_x(3,4) = \frac{\left(\mu_{Bb} + \frac{K_t K_b}{R_m}\right)\left(I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + nlR_b M_B\right)}{D}$$

$$A_x(4,1) = -\frac{glM_B\left(I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + nlR_b M_B\right)}{D}$$

$$A_x(4,3) = \frac{\mu_{bg}\left(I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + nlR_b M_B\right)}{D}$$

$$A_x(4,4) = -\frac{\left(\mu_{Bb} + \frac{K_t K_b}{R_m}\right)\left(I_{Bx} + I_{Mx} + I_b + l^2 M_B + R_b^2 M_B + R_b^2 M_b + 2lR_b M_B\right)}{D}$$

And,

$$B_x(3,1) = -\frac{K_t\left(I_{Mx} + I_b n + nR_b^2 M_B + nR_b^2 M_b + nlR_b M_B\right)}{R_m D}$$

$$B_x(4,1) = \frac{K_t\left(I_{Bx} + I_{Mx} + I_b + l^2 M_B + R_b^2 M_B + R_b^2 M_b + 2lR_b M_B\right)}{R_m D}$$

Where,

$$\begin{aligned} D = {} & I_{Bx} I_{Mx} + I_b I_{Mx} - 2nI_b I_{Mx} + n^2 I_{Bx} I_b + n^2 I_b I_{Mx} + l^2 I_{Mx} M_B + R_b^2 I_{Mx} M_B + R_b^2 I_{Mx} M_b \\ & + n^2 l^2 I_b M_B + n^2 R_b^2 I_{Bx} M_B + n^2 R_b^2 I_{Mx} M_B + n^2 R_b^2 I_{Bx} M_b + n^2 R_b^2 I_{Mx} M_b \\ & + 2lR_b I_{Mx} M_B - 2nR_b^2 I_{Mx} M_B - 2nR_b^2 I_{Mx} M_b + n^2 l^2 R_b^2 M_B M_b - 2nlR_b I_{Mx} M_B \end{aligned}$$

7.2.4. Complete Mathematical Model

The dynamics modelling of the simplified SIPW as has been explained previously in chapter 7.2.1 that both planar models in *x-y* plane and in *z-y* plane are assumed to be identical. Therefore the derivation is only repeated for the dynamic planar model in *z-y* plane. Then the completed equations of the SIPW is combination between the *x-y* planar model and *z-y* planar model into one set of linear state space matrices with state vector as the following:

$$x = \begin{bmatrix} \theta_x & \varphi_x & \dot{\theta}_x & \dot{\varphi}_x & \theta_z & \varphi_z & \dot{\theta}_z & \dot{\varphi}_z \end{bmatrix}^T$$

And the matrices are as:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ A_x(3,1) & 0 & A_x(3,3) & A_x(3,4) & 0 & 0 & 0 & 0 \\ A_x(4,1) & 0 & A_x(4,3) & A_x(4,4) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & A_z(3,1) & 0 & A_z(3,3) & A_z(3,4) \\ 0 & 0 & 0 & 0 & A_z(4,1) & 0 & A_z(4,3) & A_z(4,4) \end{bmatrix}$$

Where A_z values are the same as the A_x values as defined previously in 7.2.3 with the same physical parameters for the z-y plane.

And,

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ B_x(3,1) & 0 \\ B_x(4,1) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & B_z(3,1) \\ 0 & B_z(4,1) \end{bmatrix}$$

Where B_z values are the same as the B_x values as defined previously in 7.2.3 with the same physical parameters for the *z-y* plane.

These the complete mathematical model for the SIPW in linear state space form that is used as basis to find the gain matrix controller.

7.3. Control Design

There are a number of controller options that may be able to stabilise the SIPW system, but in this project as mentioned previously in chapter 6 gains controller calculated using LQR control strategy is selected to control the system. The LQR control strategy has been discussed in previous chapters. It is based upon state feedback law which is the control signal for the system based on the desired state reference x_{ref} and actual state of the system x such as:

$$u = K\left(x_{ref} - x\right) \tag{7.28}$$

In the SIPW system, the control signal input vector $u = \left[\tau_{bgx} \quad \tau_{mx} + \tau_{Bbx} \quad \tau_{bgz} \quad \tau_{mz} + \tau_{Bbz}\right]^T$ is torque that applied to the ball and the state vector of the system $x = \left[\theta_x \quad \varphi_x \quad \dot{\theta}_x \quad \dot{\varphi}_x \quad \theta_z \quad \varphi_z \quad \dot{\theta}_z \quad \dot{\varphi}_z\right]^T$. The LQR find gain controller matrix K which optimises the control signal input by minimalising the cost function $J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$ near linearised point. Where Q is 8x8 matrix of weight for the state and R is the weight for the applied torque. The diagonal terms in the Q matrix correspond to the body angle relative to vertical in the *x-y* plane θ_x, the roller angle relative to the body in the *x-y* plane φ_x, the body angle relative to vertical in the *z-y* plane θ_z, the roller angle relative to the body in the *z-y* plane φ_z, the angular velocity of the body in the *x-y* plane $\dot{\theta}_x$, the angular velocity of the roller in the *x-y* plane $\dot{\varphi}_x$, the angular velocity of the body in the *z-y* plane $\dot{\theta}_z$, the angular velocity of the roller in the *z-y*

plane $\dot{\varphi}_z$. The matrices Q and R used to obtain the control gain matrix K for stabilising the SIPW system are:

$$Q = \begin{bmatrix} 60000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

And,

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

These weighting values were selected and assumed by trial and error only for reasonable balancing the systems and not to achieve a certain goal. These choosing are also intended to give high importance to the body and the ball position and low importance to the torque used. Then K matrix will be found for the SIPW system by using code K = lqr (A, B, Q, R), m-file code in Matlab. Using all parameter values of the SIPW system and choosing Q and R as discussed above, the following controller gains K matrix was determined as:

$$K = \begin{bmatrix} -2715 & -5 & -478 & -20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2715 & -5 & -478 & -20 \end{bmatrix}$$

This feedback gain matrix then is used in the simulation of the SIPW in SimMechanics modelling. It will be discussed in the next section 7.4 and 7.5.

7.4. SimMechanics Block Diagram of SIPW Modelling

The modelling of SIPW in SimMechanics as a block diagram modelling is shown in Figure 7.4. The model consists of:

- Seven rigid bodies, which represent the ground, the ball, the body (represent chair and occupant), and four rollers.
- A custom joint connects the ground and the ball. The joint consist of 2 prismatic in *x*-direction and *z*-direction, 2 revolute in *x*-axis and *z*-axis which represent ball motion.
- A custom joint connects the ball and the body that consist of 2 revolute around *x*-axis and *z*-axis, which represent body motion.
- Four revolute joints connect the body to four rollers (x_1 roller, x_2 roller, z_1 roller, and z_2 roller).
- A joint initial condition, which is to set the initial angular position of body.
- The ball rolling constraint block is to give constraint for relative motion of the ball respect to the ground.
- The roller constraint block is to give constraint for relative motion of the rollers and the ball
- The sensors block is to sense position and velocity of the ball and the body, and to give input signals to system.
- The controller block is to control the SIPW system.
- The actuator block is to apply torques to the rollers.

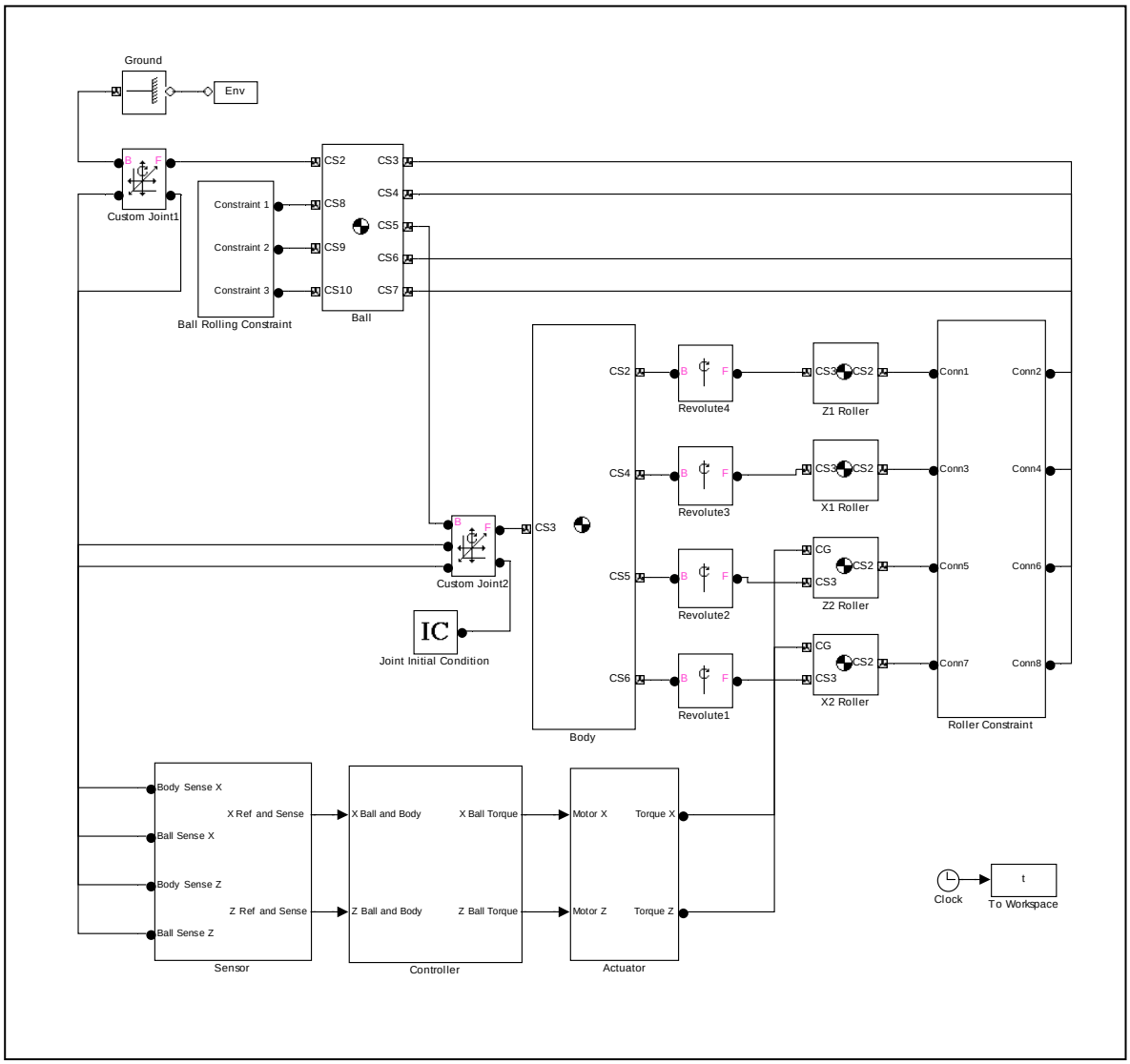


Figure 7.4 SIPW in SimMechanic block diagram modelling

The ball rolling constraint block gives limitation of the ball motion. As it mentioned previously in section 7.2.1 that in this SIPW design the ball moves only horizontally and does not move in the vertical direction thus the constraint equation for the ball motion is defined as:

$$\begin{bmatrix} v_{bx} \\ v_{by} \\ v_{bz} \end{bmatrix} = \begin{bmatrix} R_b \omega_{bz} \\ 0 \\ R_b \omega_{bx} \end{bmatrix} \tag{7.29}$$

Where v_b is the linear velocity of the ball, ω_b is the angular velocity of the ball, R_b is the ball radius, *x* and *z* are axes that are parallel to the ground, and *y* is axis that is vertical to the ground. The constraint makes the ball moves by rolling on the ground without slipping and cannot move in the vertical direction. Figure 7.5 shows the contents inside the ball rolling constraint block.

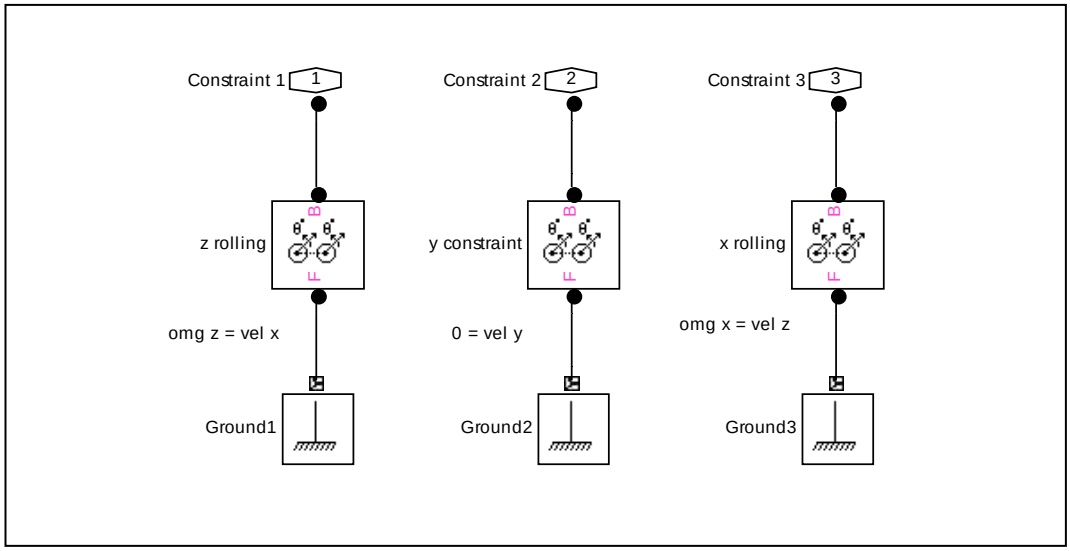


Figure 7.5 The ball rolling constraint

The rollers constraint block gives constraint for relative motion of the rollers and the ball. It makes sure that the rollers rotate without slipping on the

ball to meet with assumption mentioned previously in section 7.2.1 that in the SIPW design assumed to be no slip happens between the ball and the rollers. The constraint equation for the ball motion relative to the ball is defined as:

$$\begin{bmatrix} R_{x1r}\omega_{x1r} \\ R_{z1r}\omega_{z1r} \end{bmatrix} = \begin{bmatrix} -R_b\omega_x \\ -R_b\omega_z \end{bmatrix} = \begin{bmatrix} R_{x2r}\omega_{x1r} \\ R_{z2r}\omega_{z1r} \end{bmatrix} \tag{7.30}$$

Where *x*1*r* and *x*2*r* refer to the rollers that rotate about *z*-axis, *z*1*r* and *z*2*r* refer to the rollers that rotate about *x*-axis. The constraint also makes the rollers and the ball rotate in correct direction. Figure 7.6 shows inside the roller constraint block.

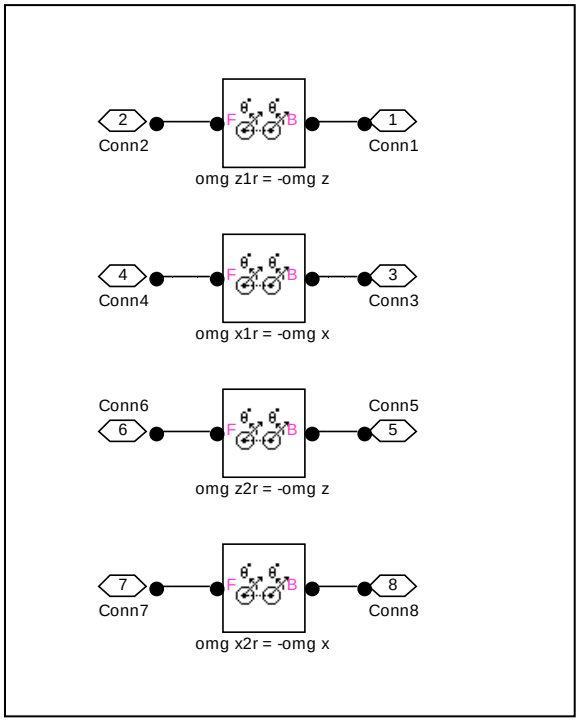


Figure 7.6 The roller constraint

Figure 7.7 shows inside the sensor block diagram in Figure 7.4. The block contains input signals block and state variables sensor blocks. The XB sensor block in top-left side of the figure senses the custom joint connecting the body and the ball in *x*-plane. The sensor measures relative angular position and velocity of the body respect to the ball in *x*-plane, ω_{Bbx}. Below the XB sensor is the Xb sensor block which senses angular position and velocity of the ball in *x*-plane, ω_{bx}. The next is continuous angle and add blocks to obtain the body position in continuous angle (the top blocks) and velocity of the body (the lower blocks) in *x*-plane to implement $\omega_{Bx} = \omega_{Bbx} + \omega_{bx}$. Below the Xb sensor is the input signals block. It gives reference inputs or signals desired for the system i.e. $\theta_x \quad \varphi_x \quad \theta_z \quad \varphi_z \quad \dot{\theta}_x \quad \dot{\varphi}_x \quad \dot{\theta}_z \quad \dot{\varphi}_z$ or they named as state vector in previous section 7.3 discussion. Below the input signals block is the ZB sensor block in which senses the custom joint connecting the body and the ball in *z*-plane. The sensor measures relative angular position and velocity of the body respect to the ball in *z*-plane, ω_{Bbz}. Below the ZB sensor is the Zb sensor block which senses angular position and velocity of the ball in *z*-plane, ω_{bz}. The next is an add block to obtain the angular position (top add block) and velocity (the lower add block) in *x*-plane of the body to implement $\omega_{Bz} = \omega_{Bbz} + \omega_{bz}$. Thus the sensor block outputs send signals of reference state variables and actual state variables in *x*-*y* plane and in *z*-*y* plane to the controller block.

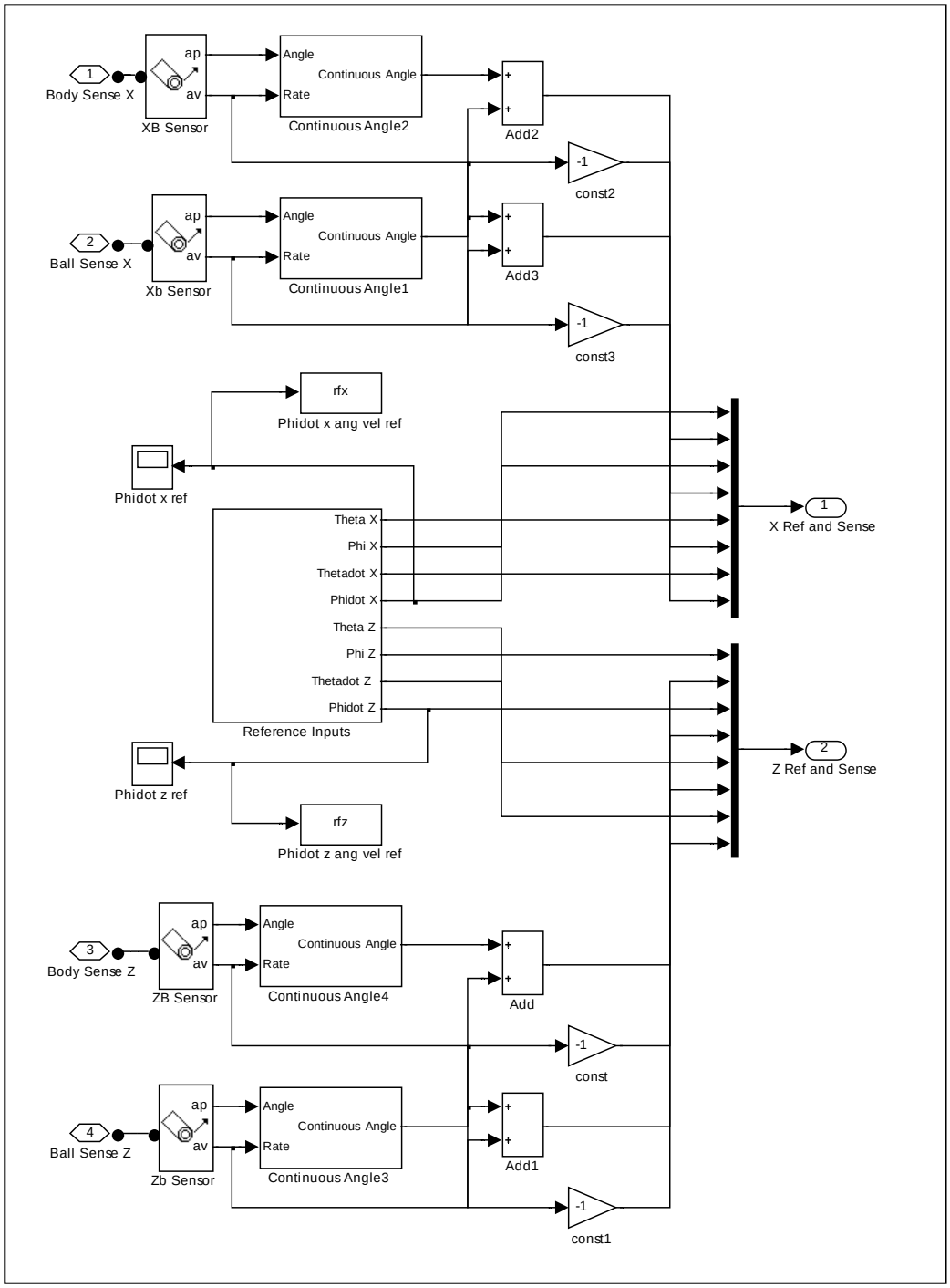


Figure 7.7 Inside sensor block

Figure 7.8 is inside the controller block in Figure 7.4. The block gets all the sensor data described above i.e. the reference state variables and the actual

state variables. They put into two identical controllers, the top one in *x* direction and below one in *z* direction. The outputs of this block are torques that will be applied to the ball. These ball torques is then converted to roller torques in actuator block.

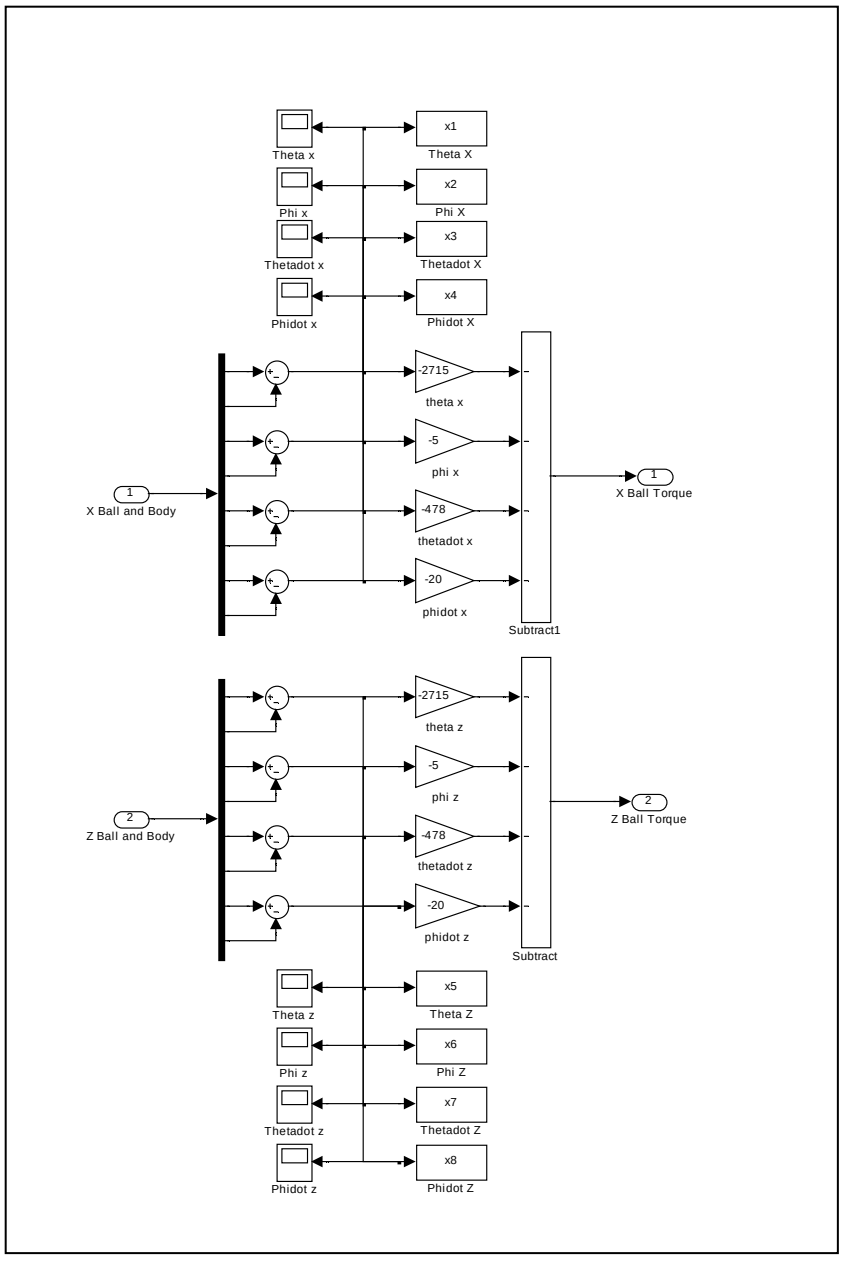


Figure 7.8 Controller block

Figure 7.9 is inside of the actuator block in Figure 7.4. This block contains two motor blocks and body actuator blocks. This actuator block is used to convert the ball torques sent by the controller block to be rollers torques. Then these roller torques are applied to the rollers for both in *x-y* plane and in *z-y* plane.

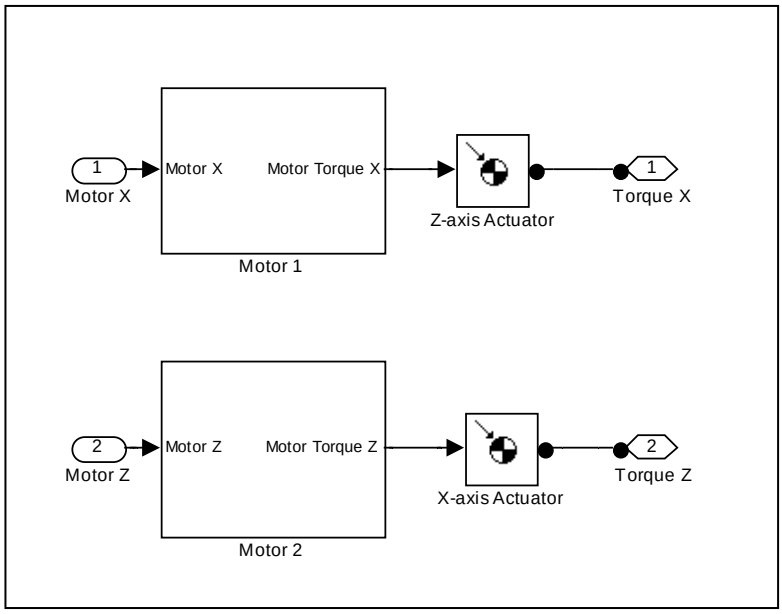


Figure 7.9 The inside of the actuator block

7.5. Simulation Result

The visualisation window of the SIPW modelling in SimMechanics is shown in Figure 7.10.

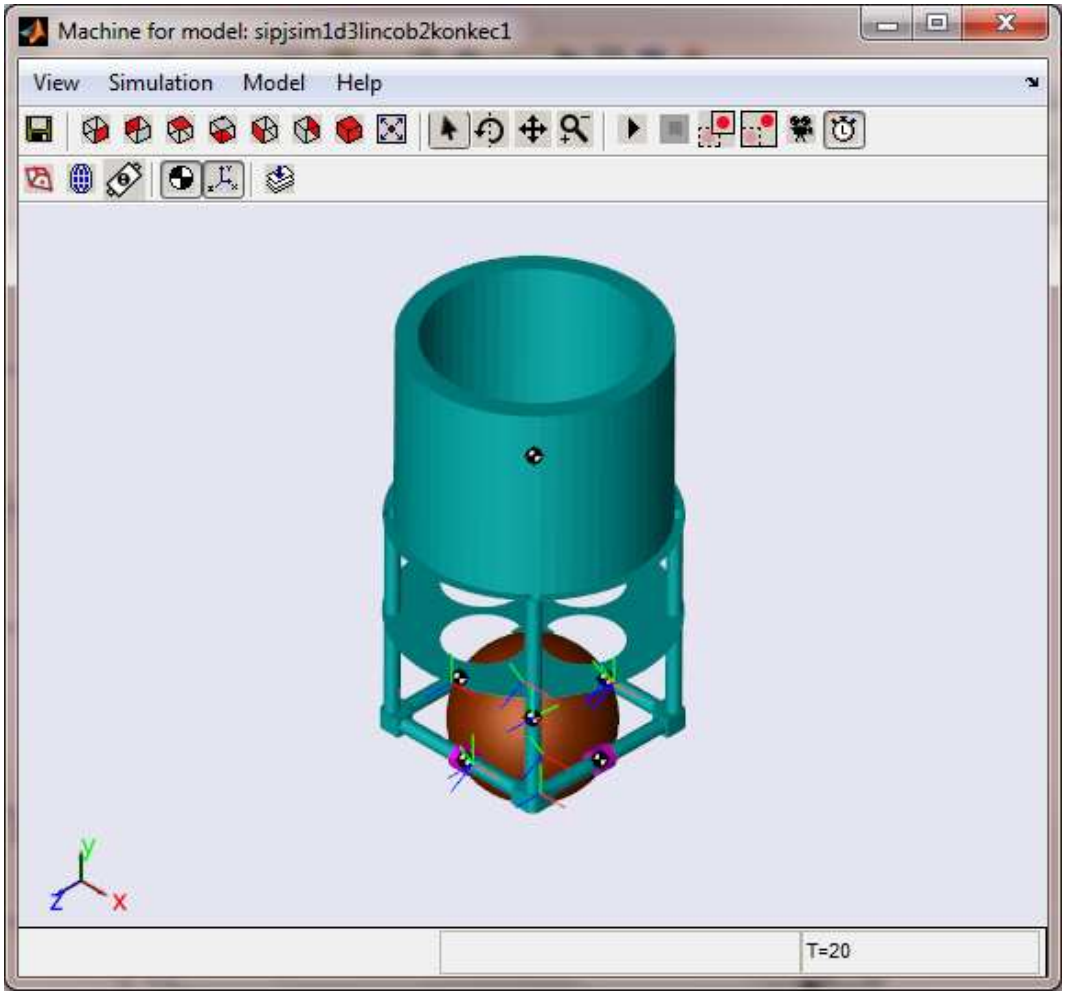
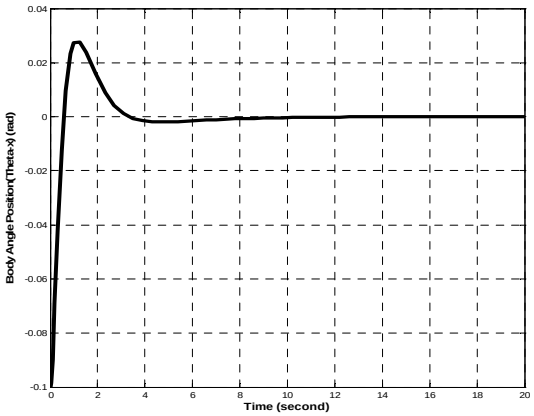
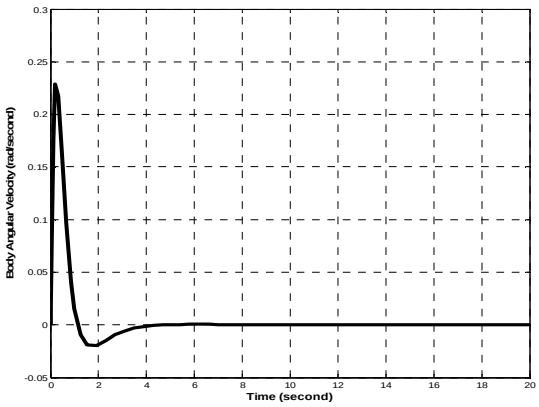


Figure 7.10 SimMechanics visualisation window of the SIPW

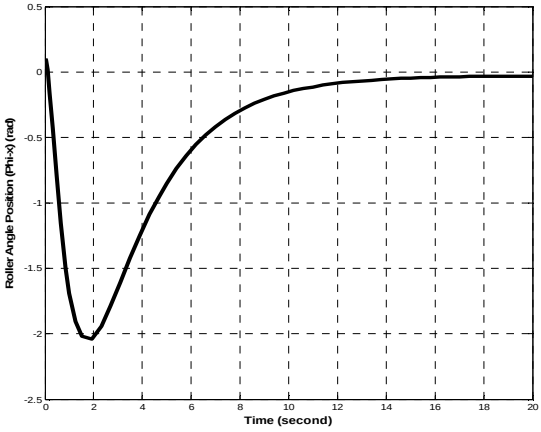
Then for simulation the SIPW system modelling in SimMechanics is set initially by placed the body angle θ_x at 0.1 rad from the vertical. This means that the SIPW needs to move to eliminate the body angle θ_x position to zero radians so that the body that represent chair and people is on vertically straight position and make the stabilisation of the SIPW system. The simulation operates and is shown in the *x*-plane only as the controller operates in the same way in both planes. The result can be seen in Figure 7.11.



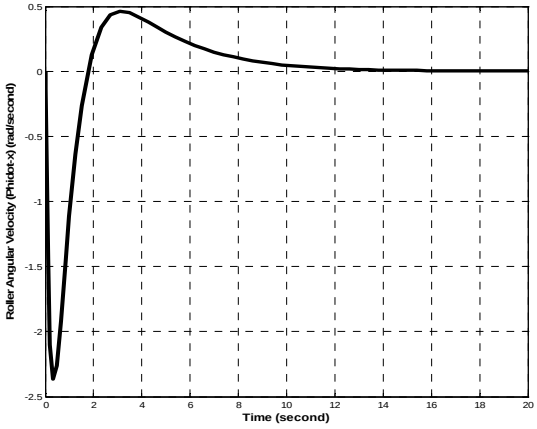
(a) Body position (θ_x)



(b) Body angular velocity ($\dot{\theta}_x$)



(c) Roller Position (φ_x)



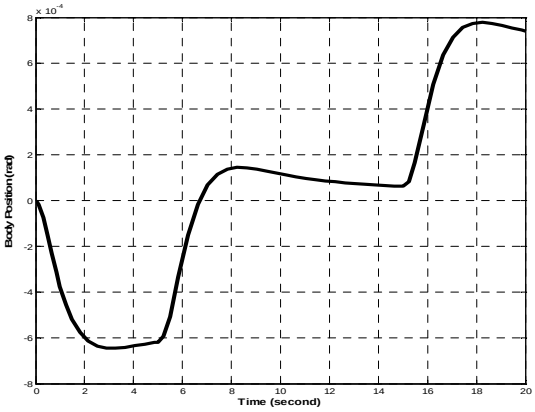
(d) Roller angular velocity ($\dot{\varphi}_x$)

Figure 7.11 Stabilisation of the SIPW

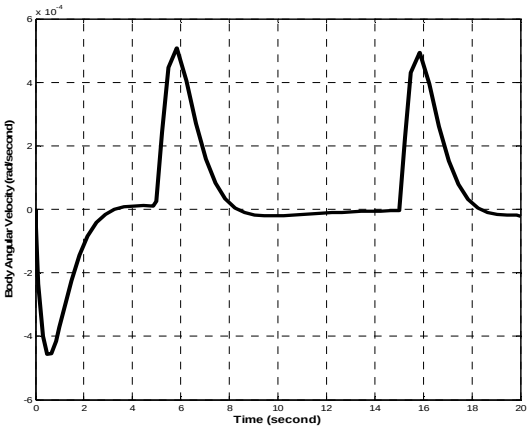
The simulation result in Figure 7.11 shows that the controller using LQR method can stabilise the SIPW system. The graph shows that the body which was initially placed at 0.1 rad from vertical can be stabilised to zero point at about 3 seconds while the roller also can reach steady state error to zero and with small

overshoot only about 2 %. Thus the LQR controller is able to stabilise the spherical inverted pendulum low percentage overshoot

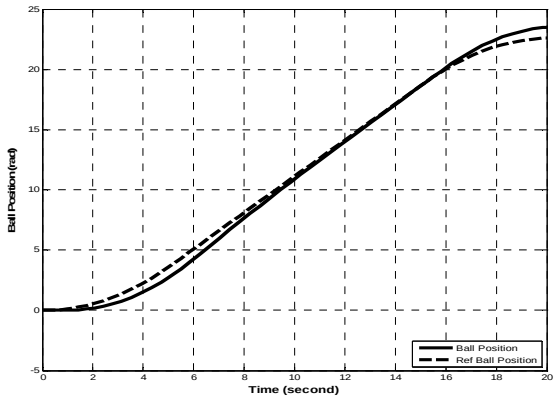
When the system is given a prescribed velocity, the simulation result or the body and the ball responses are as shown in Figure 7.12.



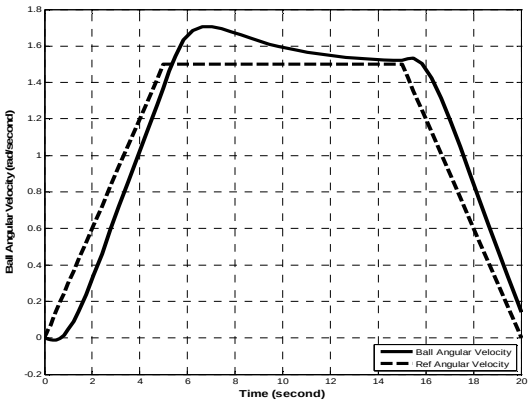
(a) Body position (θ_x)



(b) Body angular velocity ($\dot{\theta}_x$)



(c) Roller Position (φ_x)



(d) Roller angular velocity ($\dot{\varphi}_x$)

Figure 7.12 The SIPW with command tracking

The result with prescribed velocity as can be seen in Figure 7.12 shows that the SIPW systems using LQR controller method has rollers angular velocity

lagging about less than 1 second from the velocity input reference given. However it can be said that the SIPW can be controlled successfully to track the reference input given while at same time stabilized the body which is represent the chair and the people.

7.6. Summary

This chapter has discussed how to derive the mathematical modelling of the Spherical Inverted Pendulum Wheelchair (SIPW) in state space model which is then used to obtain the gains controller for the system. The SIPW is a concept of a wheelchair on spherical based upon inverted pendulum control. In the SIPW concept the wheelchair and the occupant are ride and balance on a ball. Driving the ball and balancing the system is provided through four rollers which is based on mouse ball driving mechanism concept. Thus the SIPW can move in any direction with rotation. The simulation of the SIPW is carried out in 3D simulation in SimMechanics.

Moreover, the chapter also shows and demonstrates that the SIPW which is a dynamics of multibody system is successfully designed within SimMechanics modelling. The SIPW system is stabilised using the LQR feedback method with 8 state variables. The simulation result study of the SIPW shows that LQR control strategy is capable to control multi input and multi output of the systems successfully and gives satisfy of controlling the SIPW. It has shown that the control is capable in stabilising the SIPW and prescribed tracking capabilities.

CHAPTER 8

DISCUSSION

The Spherical Inverted Pendulum Wheelchair (SIPW) is based upon the dynamic and control of spherical inverted pendulum concept which has four DoF i.e. two DoF rotation moving of the pendulum (body which is represent the chair and the user) in *x*-axis and *z*-axis and two DoF horizontally moving of the ball in *x*-axis direction and *z*-axis direction. This spherical inverted pendulum concept is chosen to design and simulate the SIPW as it has the potential to offer improving mobility and maneuverability compared with existing wheelchairs. Thus the outcome of the work is a new wheelchair concept with novel motion capabilities that can provide improvement in mobility and maneuverability.

There are two types of previous work on mobile robot based on inverted pendulum is a two-wheeled robot, like Segway (Nguyen et.al., 2004) and a single spherical wheel robot, like ballbot (Lauwers, Kantor and Hollis, 2006). As the single spherical inverted pendulum is able to move in any direction without changing orientation so the SIPW design proposed is based on a single spherical wheel. Thus the new wheelchair concept on a single spherical wheel, the SIPW, can provide improvement in its mobility and manoeuvrability than on four wheels which is commonly wheelchair designed and operated.

To investigate the dynamic of the SIPW which is designed and simulated based upon spherical inverted pendulum concept some various control methods of inverted pendulum works is reported in this project. The work has demonstrated that three control strategies i.e. pole placement, LQR and PID can be applied and are capable of controlling the inverted pendulum's angular position and the cart's linear position or velocity of the both Simulink model and SimMechanics model in two DoF inverted pendulum.

As stability and control performance of the two DoF inverted pendulum system largely depend on pole locations thus the pole-placement in system design is very important. Therefore, the pole placement control strategy was used to place the poles of the two DoF inverted pendulum closed-loop system in the desired positions by state feedback or output feedback. There are two main steps to carry out. The first step is the placement or assignment of poles and the second step is the identification of the feedback gain matrix. By placing the two DoF inverted pendulum's poles at S_1=-7+7i, S_2=-7-7i, S_3=-7, S_4=-7, S_5=-7, the state feedback gain matrix is given as (K,k_i)=(-4714.2 -1533.4 3347.2 558.3 6595.7). The simulation result can be seen in Figure 4.4. The result shows that the settling time for controller based on state space pole placement control strategy is about one and half second i.e. the controller can stabilise the pendulum within one and half second and has no overshoot. When applying a prescribed velocity input, the inverted pendulum is able to follow it while keep the pendulum stable although there exists a small lagging only about a half second as shown in Figure 4.5. Therefore, based on simulation results, it can be said that the pole placement

control strategy can be useful to design controllers for the two DoF inverted pendulum with satisfactory performance.

Another alternative technique such as LQR control strategy also provides satisfactory result in controlling the two DoF inverted pendulum system. In a LQR design, the gain matrix K for a state feedback control law $u = -Kx$ is found by minimizing a quadratic cost function in the form equation 4.16, where Q and R is weighting parameter. The weighting parameters chosen in the inverted pendulum of the state feedback controller are:

$$Q = \begin{bmatrix} x_w & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & y_w & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } R = 1$$

The element in the (1,1) position will be used to weight the cart's linear position and the element in the (3,3) position will be used to weight the pendulum's angular position and denoted as x_w and y_w weighting variables. The weighting x_w and y_w variables could be changed to see the various responses. If x_w and y_w are increased even higher, improvement to the response should be found. But, in order to satisfy the design requirements of keeping x_w and y_w as small as possible since in this problem the values of x_w and y_w have been used to describe the relative weight of the tracking error in the cart's linear position and pendulum's angular position versus the control effort. The higher the values of x_w and y_w, the more control effort will be required, but the smaller the tracking error. Using all parameter values of the inverted pendulum and choosing x_w = 10000 and y_w = 1000, the following values for controller gains K matrix are determined:

$$[K, k_r] = [k_1, k_2, k_3, k_4, k_r] = [-100.0 \quad -88.9 \quad 545.3 \quad 87.7 \quad -100.0]$$

The simulation result can be seen in Figure 4.12. The result shows that the settling time for controller based on LQR control strategy is about one and half second i.e. the controller can stabilise the pendulum within two and half second and has overshoot about 5 %. When applying a prescribed velocity input, the two DoF inverted pendulum system is able to follow it while keep the pendulum stable although there exists a lagging for about one second as can be seen in Figure 4.13. Compared with the pole placement method, the LQR result gives time to stabilise the system about one second slower but LQR use smaller gains of *K* matrix. It means that the system uses lesser effort or energy for stabilising the system. This is the advantage to use LQR control strategy the poles are placed in such way through the cost function to get optimal gains for not only in stabilising the system but also in controlling effort. Therefore, based on simulation results, it can be said that the LQR control strategy can be useful to determine controller gain values for the two DoF inverted pendulum system with optimal performance.

The conventional controller such as PID is implemented and reported to control the two DoF inverted pendulum in this work. The PID controller is commonly known a good controller to control the single-input-single-output (SISO) system. Thus with only one PID controller cannot be used to control the two DoF inverted pendulum i.e. the cart horizontally moving and the pendulum rotation moving simultaneously at the same time. In another word can be said that single PID cannot control a multi-input-multi-output (MIMO) system like the two DoF inverted pendulum system. Thus there are needed two PID controllers to

control the two DoF inverted pendulum system as can be seen in Figure 4.18. By using two PID controllers are able to control both outputs such the pendulum angular position and the cart linear position at the same time. 'PID Cart' is used to control the cart whereas 'PID Pendulum' is used to control the pendulum. Applying the controller gains for 'PID Cart' as K_p = -414.8, K_i = -5.1, and K_d = -467.4 and for 'PID Pendulum' as K_p = 4157.3, K_i = 3539.5, and K_d = 443.3 gives result as shown in Figure 4.21. The simulation result shows that the two DoF inverted pendulum can be stabilised successfully with rise time about 1 second, settling time about 4 seconds and overshoot about 5 %. Furthermore, when applying a prescribed velocity input, the two DoF inverted pendulum is successfully to follow the prescribed velocity while keep the pendulum stable on upright position. Although there exists a lagging for less than one second but the cart is able to follow the prescribed velocity as shown in Figure 4.22. Thus it can be said that the model of two DoF inverted pendulum system can be stabilised successfully by applying two PID controllers simultaneously if they are tuned properly.

An alternative inverted pendulum design that consists of two carts (bottom cart (vehicle) and top cart) and a pendulum which is controlled using PID has also been investigated and presented in this thesis. This design gives possibility to control the bottom cart (vehicle) to get better characteristic on set point tracking while another cart moving on top to stabilise the pendulum stand upright. Three PID controllers are used to control both carts and the pendulum as can be seen in Figure 5.4. 'PID Vehicle' is used to control bottom cart or vehicle linear position

or velocity, ‘PID Pendulum’ is used to control the pendulum angular position or velocity and the ‘PID Cart’ is used to control the top cart linear position or velocity. Applying the controller gains for ‘PID Vehicle’ as K_p = 617.9, K_i = 0.6, and K_d = 0, for ‘PID Pendulum’ as K_p = -346.7, K_i = 1.4, and K_d = 0.07 and for ‘PID Cart’ as K_p = -82.7, K_i = 12.2, and K_d = 0 gets a result as shown in Figure 5.8. The graph shows that the system can be stabilised successfully while the vehicle follows the prescribed velocity. The vehicle is able to follow the reference very satisfactory with no lagging. From the graph also can be seen that the sudden change of acceleration can lead to instability of the system.

The advantage of the two carts design is the system can follow the prescribed velocity given very satisfactory, with no lagging. This is because the velocity of the system is no longer associated with the pendulum. However, there is disadvantage in this design. As the pendulum is stabilised by the moving second or top cart on the vehicle or bottom cart, thus the movement of the top cart is constrained by the long dimension of the bottom cart. As a result the capability of the system to stabilise the pendulum is highly depend on the long dimension of the bottom cart. If the top cart need to move exceed the length dimension constrain when stabilising the pendulum, the design will fail to stabilise the pendulum.

In this work, the two DoF inverted pendulum is modelled and simulated in both ways Simulink model and SimMechanics model. A SimMechanics model differs significantly from a Simulink model in how it represents the inverted pendulum system. A Simulink model represents mathematics of the inverted

pendulum motion, i.e., the differential equations that predict the inverted pendulum system future state from its present state. The mathematical model enables Simulink to simulate the system. In contrast, a SimMechanics model represents a physical structure of the two DoF inverted pendulum, the geometric and kinematic relationships of its component bodies. Thus mathematical model no longer needed to be developed as SimMechanics converts this structural representation to an internal, equivalent mathematical model. The two DoF inverted pendulum is represented by connected block diagrams. The Physical modelling environment SimMechanics makes the task easier than the Simulink one where the dynamic system equation should be developed first before building the block diagram of the system.

Figure 4.6 is represents the two DoF inverted pendulum model in SimMechanics. From the Figure can be seen obviously, every block corresponds to one mechanical component of the two DoF inverted pendulum. The properties of the blocks can be entered by double-clicking on them. The position, velocity, and acceleration variables of the two DoF inverted pendulum system can be measured by adding sensor blocks. The forces and torques transmitted by the joints can be sensed, too. The simulation results of both models are identical, although the numerical errors may differ slightly as we can see from Figure 4.12 which is simulation result of Simulink model and Figure 4.15 which is simulation result of SimMechanics model. The differences in the computation time are more obvious. Simulation with SimMechanics takes longer, it maybe because of the mathematical model has to be derived by the software before the integration of the

ODE system can begin. With this model and external graphics visualisation facility of SimMechanics it is possible to import Solidworks assembly model and can animate the motion of the pendulum and the cart more realistic as can be seen in Figure 4.7 while this is not possible in only Simulink environment.

The spherical inverted pendulum wheelchair (SIPW) is modelled and simulated in SimMechanics model in this project. The state feedback control with gains control calculated trough LQR method is used to stabilise the SIPW. This control method is selected as from the three methods experienced in this report shows that LQR is capable to stabilise the MIMO systems such as the two DoF inverted pendulum system and also has command tracking capabilities when the system wanted to follow the desired input. As discussed in chapter 7 the SIPW has 8 state variables with two input, thus it is a MIMO system obviously which is the LQR method suitable to be applied.

The simulation result shows that LQR technique can provide satisfactory result in controlling the SIPW system. In a LQR design, the gain controller matrix K for the state feedback controller law as equation 4.28, $u = K\left(x_{ref} - x\right)$ where x_{ref} is the desired state reference and x is actual state of the SIPW, can be found by minimizing a quadratic cost function in the form $J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt$. This cost function J can be interpreted as energy function, x is the state variable which is weighted by Q and u is control input which is weighted by R. Minimisation the cost function J, results in moving x to zero via little control energy and in turn guarantees that the systems will be stable. In the SIPW system, the control signal

input vector $u = \begin{bmatrix} \tau_{bgx} & \tau_{mx} + \tau_{Bbx} & \tau_{bgz} & \tau_{mz} + \tau_{Bbz} \end{bmatrix}^T$ is torque that applied to the ball and the state vector of the system $x = \begin{bmatrix} \theta_x & \varphi_x & \dot{\theta}_x & \dot{\varphi}_x & \theta_z & \varphi_z & \dot{\theta}_z & \dot{\varphi}_z \end{bmatrix}^T$. The matrices *Q* and *R* used to obtain the control gain matrix *K* for stabilising the SIPW system are:

$$Q = \begin{bmatrix} 60000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ And, } R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

These weighting values were selected and assumed only for reasonable balancing the systems and not to achieve a certain goal. These choosing are also intended to give high importance to the body and the ball position and low importance to the torque used. Using all parameter values of the SIPW system as discussed in chapter 7, the following controller gains *K* matrix was determined as:

$$K = \begin{bmatrix} -2715 & -5 & -478 & -20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2715 & -5 & -478 & -20 \end{bmatrix}$$

The simulation result can be seen in Figure 7.11. The result shows that the controller using LQR method can stabilise the SIPW system. The graph shows that the body which was initially placed at 0.1 rad from vertical can be stabilised to zero point at about 3 seconds while the roller also can reach steady state error to zero and with small overshoot only about 2 %. When applying a prescribed velocity input the SIPW system is able to follow it and at the same time keeps the

body remain stable on upright position although there exists a small lagging about less than 1 second from the velocity input reference given. Thus the LQR controller is capable to stabilise the SIPW successfully and has prescribed tracking capabilities.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

9.1. Conclusions

The design and simulation of the new concept of wheelchair based upon spherical inverted pendulum control i.e. the SIPW is reported in this thesis. The thesis has demonstrated some various control strategies of inverted pendulum system can be applied. Three control strategies, pole placement, LQR and PID are capable of controlling the two DoF inverted pendulum's angle and the cart's position of the both Simulink model and SimMechanics model. The PID control approach that was proposed to use two PID controllers solves the problem to control inverted pendulum system which is MIMO system successfully. An alternative inverted pendulum design that consists of two carts (bottom cart (vehicle), top cart with inverted pendulum) is presented in this report. This design gives possibility to control the bottom cart (vehicle) to get better characteristic on set point tracking while another cart moving on top to stabilise the inverted pendulum. The final objective of the project in establishing the design and simulation of a spherical inverted pendulum wheelchair can be realised. The wheelchair on spherical based upon inverted pendulum control is designed and

simulated in SimMechanics with state feedback control using LQR control strategy to find the gain matrix controller. The simulation shows that the body which represent the chair and the occupant is balanced on a spherical ball through four rollers (two driven and two idler) successfully. Thus the new wheelchair concept based upon the dynamic and control of spherical inverted pendulum which has the potential to offer improved mobility compared with exiting wheelchairs in the market can be achieved.

The development of this a new wheelchair concept based have made contributions to the inverted pendulum application knowledge base including: developing simulation for stabilisation of the inverted pendulum system using various control strategies in both Matlab Simulink and SimMechanics, proposing the new alternative inverted pendulum design construction with two carts (bottom cart (vehicle), top cart with inverted pendulum) for improving characteristic on the command tracking and modelling and controlling conceptual wheelchair, the spherical inverted pendulum wheelchair (SIPW), for improving mobility to help wheelchair user.

9.2. Recommendations for Future Work

For future work need to investigate the application of an adaptive suspension system which will be modelled and integrated into the SIPW simulation and the SIPW dynamics investigation in terms of comfort and safety. In term of safety need to be investigated the dynamic of the SIPW on inclined

surface. And in term of comfort improvement need to be designed an optimised package for providing optimal seat performance for the SIPW, to be investigated the driving control of the SIPW trough a mobile phone, and maybe also need to designed turntable mechanism under the seat to provide more degree of freedom to the wheelchair system. Finally need to be determined and developed the parameters of the SIPW systems including the suspension system experimentally. Run test for the SIPW prototype to tune or maybe optimise the physical parameters so that can meet with the real spherical inverted pendulum wheelchair application.

REFERENCES

ADAM B. AND ROBERT R. (2004) *Experimental Verification of the Dynamic Model for a Quarter Size Self Balancing Wheelchair*. Proceeding of American Control Conference June 30- July 2, 2004; Boston, Massachusetts; 488-492.

AKIHIRO N. AND TOSHIYUKI M. (2009), *A Stabilisation Control of Two Wheels Driven Wheelchair*. IEEE/RSJ International Conference on Intelligent Robots and Systems; Oct 11 – 15, 2009; St Louis, USA.

ASCHER U. M. AND PETZOLD L. R. (1998), *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*. Philadelphia, Pennsylvania; SIAM.

ASTROM K.J. AND HAGGLUND T.H. (1995), *New Tunning Method for PID Controllers*. Proceeding of the 3rd European Control Conference: 2456-2462.

BALCELLS A. C. AND GONZLEZ J. A. (1998) *TetraNauta: a Wheelchair Controller for Users with Very Severe Mobility Restrictions*. TIDE, July.

BANKS J. (1999) *Introduction to Simulation*. Proceeding of the 1999 Winter Simulation Conference December 5-8; Phoenix, AZ; 7-13.

BORGOLTE U., HOYER H., BUEHLER C., HECK H., AND HOELPER R. (1998) *Architectural Concepts of a Semi-Autonomous Wheelchair*. J Intell Robotic Syst. 22(3/4):233-53.

BOURHIS G., MOUMEN K., PINO P., ROHMER S., AND PRUSKI A. (1993) *Assisted Navigation for a Powered Wheelchair.*Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. 1993 Oct 17 – 20; Le Toquet, France. Piscat-away (NJ):IEEE; 553-58.

CAGIGAS D. AND ABASCAL J. (2004) *Hierarchical Path Search with Partial Materialization of Costs for A Smart wheelchair*. J Intell Robotic Sys, 39(4): 409-431.

CHANG L. H. AND LEE A. C. (2007), *Design of Nonlinear Controller for Bi-Axial Inverted Pendulum System. IET Control Theory Appl*, 2007, 1(4): 979-986.

CHESNEY D. A., AXELSON P. W., NOON J. H. AND SIEKMAN A. R. (1998) *DSore Butts '98- International Cushion Design Competition.* 21st Annual RESNA Conference, June 26-30.

COOPER R. A., COOPER R., TOLERICO M., GUO S., DING D. AND PEARLMAN J. (2006). *Advances in Electric Powered Wheelchairs.* Top Spinal Cord Inj Rehabil; 11(4): 15-29.

COOPER R., CORFMAN T., FITZGERALD S., BONINGER M., SPAETH D., AMMER W., ARVA J. (2002) *Performance Assesment of a Pushrim Activated Power Assisted Wheelchair*. IEEE Trans Control Syst Technol. 10(1):121-26.

DIAZ-CALDERON A., PAREDIS C. J. J., AND KHOSLA P. K. (2000) *Reconfigurable Models: A Modelling Paradigm to Support Simulation-based Design*, Summer Computer Simulation Conference; Vancouver, Canada.

DING D. AND COOPER R. A., (2005) *Electric Powered Wheelchairs a Review of Current Technology and Insight into Future Directions*, IEEE Control Systems Magazine, 22-34.

DURFEE W. K., WALL M. B., ROWELL D., AND ABBOTT F. K. (1991) *Interactive Software for Dynamic System Modelling Using Linear Graphs,* IEEE Control Systems; vol. 27, 60-66.

FEHR L, LANGBEIN W.E., AND SKAAR S.B. (2000). *Adequacy of Powered Wheelchair Control Interfaces for Persons with Severe Disabilities: a Clinical Survey*. J Rehabil Res Devel. Vol 37(3):353–60.

GOHER K. M. AND TOKHI M. O. (2010), *A New Configuration of Two Wheeled Inverted Pendulum: a Lagrangian Based Mathematical Approach;* JRSC; Dec Edition; 1-5.

GOHER K. M. AND TOKHI M. O. (2010), *Modeling and Control of a Two Wheeled Machine: A Genetic Algorithm Based Optimisation Approach*; JRSC; Dec Edition; 17-21.

HAUSER J., ALESSANDRO S. AND RUGGERO F. (2005), *On the Driven Inverted Pendulum*; 44th IEEE Conference on Decision and Control; Dec 12 – 15 2005; Seville, Spain. 6176-6180.

JUNG S. AND KIM S. S. (2008) *Control Experiment of a Wheel Driven Mobile Inverted Pendulum Using Neural Network.* IEEE Transactions on Control Sys Tech. 16(2): 297 -303.

JIN SUN JU, YUNHEE SHIN AND EUN YI KIM. (2009) *Intelligent Wheelchair (IW) Interface using Face and Mouth Recognition*. International Conference on Intelligent User Interface. Feb 8-11 2009; Sanibel Island, USA. 307-314.

KAMENETZ, H. L. (1969). *A Brief History of the Wheelchair.* Journal of the History of Medecine and Allied Sciences, 24(2): 205-210

KATEVAS N. I., SGOUROUS N. M., TZAFESTAS S. G., PAPAKONSTANTINOU G., BEATTIE P., BISHOP J. M., TSANAKAS P. AND KOUTSOURIS D. (1997) *The Autonomous Mobile Robot SENARIO : A Sensor Aided Inteligent Navigation System for Powered Wheelchairs.* J Intell Robotic Syst. IEEE Robot Autom Mag, 4(4):60-70.

KIM Y., KIM S. H. AND KWAK Y. K. (2006) *Improving Driving Ability for Two Wheeled Inverted Pendulum Type Autonomous Vehicle*. Proc. IMechE. Vol 220; 165-175

KUNO Y., SHIMADA N. AND SHIRAI Y. (2003) *Look Where You're Going [Robotic Wheelchair]*. IEEE Robot Autom Mag. 10(1); 26-34

LANKENAU A. AND ROFER T. (2001) *A Versatile and Safe Mobility Assistant*. IEEE Robot Autom Mag. ;8(1):29–37.

LAUWERS T.B., KANTOR G.A., AND HOLIIS R.L. (2006) *A Dynamically Stable Single Wheeled Mobile Robot with Inverse Mouse-Ball Drive*. Proceedings IEEE International Conference on Robotics and Automation, Orlando, Florida, May 2006. IEEE.

LEVINE S. P., BELL D. A., JAROS L., SIMPSON R. C., KOREN Y. AND BOREINSTEIN J. (1999) *The NavChair Assistive Wheelchair Navigation System*. IEEE Trans Rehabil Eng. 7(4): 43-51.

MAEDA S., FUTATSUKA M., YONESAKI J., AND IKEDA M. (2003) *Relationship between Questionnaire Survey Result of Vibration Complains of Wheelchair Users and Vibration Transmissibility of manual Wheelchair.* Environmental Health and Preventive Medicine. 8(3): 82-89

MATSUMOTO Y., INO T. AND OGASAWARA T. (2001) *Development of Intelligent Wheelchair System with Face and Gaze Based Interface*. Proceeding of the 10th IEEE International Workshop on Robot and Human Interactive Communication (ROMAN) 2001 Sep18-21; Bordeaux-Paris, France. Piscat-away (NJ): IEEE; 2001. 262-267.

MAZO M. (2001) *An Integral System for Assisted Mobility*. IEEE Robot Autom Mag. 8(1): 46-56.

McGUIRE W. R. (1999) *Voice Operated Wheelchair Using Digital Signal Processing Technology for People with Disabilities (RESNA)*; 1999 June 25-29; Long Beach, CA Arlington (VA): RESNA Press; 364-66.

MOON I., LEE M., RYU J., AND MUN M. (2003) *Intelligent Robotic Wheelchair with EMG-, Gesture-, and Voice-based Interfaces*. Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2003 Oct 27–31; Las Vegas, NV. Piscataway (NJ): IEEE; 2003. 3453–58.

NAWAWI S. W., AHMAD N. M. AND OSMAN J. H. S. (2006) *Control of Two Wheels Inverted Pendulum Mobile Robot Using Full Order Sliding Mode Control*. Proceeding of international conference on Man-Machine Systems; Sep 15-16 2006;Langkawi, Malaysia.

NGUYEN H. G., MORRELL J., MULLERS K., BURMEISTER A., MILES S., FARRINGTON N., THOMAS K. AND GAGE D. (2004) *Segway Robotic Mobility Platform*. SPIE Proc.5609: Mobile Robots XVII, Philadelphia, PA, October 2004.

OGATA K. (2002) *Modern Control Engineering.* Publisher: Prentice Hall of India, ISBN-10:812032045X, fourth edition.

PEI FIA, HOUSENG HU, TAO LU AND KUI YUAN. (2007) *Head Gesture Recognition for Hands Free Control of an Intelligent Wheelchair*. Industrial Robot: An International Journal 34(1): 60-68.

RASMUSSEN H. (2002) *Automatic Tuning of PID Regulators.* Text summarize of digital PID Controller, Aalborg University, Departement of Control Engineering, Denmark, 1-21

ROSENBERG R. C. AND KARNOPP D. C. (1983). *Introduction to Physical System Dynamics.* New York, Mc Graw Hill.

SEONG H. J. AND TAKAYUKI T. (2007), *Wheeled Inverted Pendulum Assistant Robot: Inverted Mobile, Standing and Sitting Motions*. IEEE/RSJ Inter Conference on Itelligent robot and systems; Oct 29 – Nov 2007; 1932-1937.

SEIDEL H. (1993) *Selected Health Risk Caused by Long-term Whole Body Vibration.* American Journal of Industrial Medicine, 23(4): 589-604.

SEIDEL H., BLUETEHNER R. AND HINZ B. (1986) *Effect of Sinusoidal Whole Body Vibration on the Lumbar Spine: the Stress-strain Relationship.* International Archives of Occupational and Environmental Health, 57(3): 207-223.

SIMPSON R .C., (2005) *Smart Wheelchairs: A literature Review*. Journal of Rehabilitation Research and Development, 42(4): 423 - 436.

SIMPSON R. C. AND LEVINE S. P., (2002) *Voice Control of a Powered Wheelchair*. IEEE Trans Neural Syst Rehabil Eng 10(2): 122-125.

STRAUSS J. C. AND OTHERS (1967) *The SCI Continues System Simulation Language (CSSL)*. Simulation. Vol 9: 281-303.

VANSICKLE D. P., COOPER R. A., BONINGER M. L., AND DIGIOVINE C. P. (2001) *Analysis of Vibration Induced during Wheelchair Propulsion.* 38(4): 409- 421.

YAMAMOTO Y. (2008) *NXT Way GS Model Based Design Control of Self-balancing Two Wheeled Robot Built with LEGO Mindstorms NXT,* Advanced Support Group 1 Engineering Department, Applied Systems First Division, CYBERNET SYSTEMS CO., LTD.

YOSHIHIKO T., MASAYOSHI K., YU S. & KAZUKI K. (2008) *Psychological Flutter of Robotic Wheelchair with Inverse Pendulum Control*, IEEE, 1318-1323.

ZIMMERMANN C. L., COOK T. M. AND GOELSS V. K. (1993) *Effect of Seated Posture on Erector Spinae EMG Activity During Whole Body Vibration.* Rgonomic, 36(6): 667- 476.

APPENDICES

Appendix 1.

Publication: Journal Paper

Modelling and Simulation of Spherical Inverted Pendulum Based on LQR Control with SimMechanics, Applied Mechanics and Material, Vol. 391 (2013) pp 163-167, Trans Tech Publications, Switzerland.
doi:10.4028/www.scientific.net/AMM.391.163

Appendix 2.

Script M-file to determine matrices of state space and poles

```
clear all;
clc;
M = 13.26;
m = 2.88;
b = 0;
i = 0.04;
g = 9.8;
l = 0.21;
q=i*(M+m)+M*m*l^2; %denominator for the A and B matricies
A=(0      1              0           0;
   0 -(i+m*l^2)*b/q  (m^2*g*l^2)/q   0;
   0      0              0           1;
   0 -(m*l*b)/q       m*g*l*(M+m)/q  0)
B=(     0;
     (i+m*l^2)/q;
          0;
        m*l/q)
C=(1 0 0 0;
   0 0 1 0)
D=(0;
   0)
p=eig(A);
```

Appendix 3.

Script M-file to calculate feedback gains using pole placement method

```
clear all;
clc;
M = 13.26;
m = 2.88;
b = 0;
i = 0.04;
g = 9.8;
l = 0.21;
q=i*(M+m)+M*m*l^2; %denominator for the A and B matricies
A=(0       1             0          0;
   0 -(i+m*l^2)*b/q (m^2*g*l^2)/q  0;
   0       0             0          1;
   0 -(m*l*b)/q      m*g*l*(M+m)/q 0);
B=(0;(i+m*l^2)/q;0;m*l/q);
C=(1 0 0 0);
D=(0);
Ah=(A zeros(4,1);-C 0);
Bh=(B;0);
pl=(-7+i*7 -7-i*7 -7 -7 -7);
Kh=acker(Ah,Bh,pl);
```

Appendix 4.

Script M-file to produce optimal controller using LQR method

```
clear all;
clc;
M = 13.26;
m = 2.88;
b = 0;
i = 0.04;
g = 9.8;
l = 0.21;
q=i*(M+m)+M*m*l^2; %denominator for the A and B matricies
A=(0      1              0           0;
   0 -(i+m*l^2)*b/q  (m^2*g*l^2)/q   0;
   0      0              0           1;
   0 -(m*l*b)/q       m*g*l*(M+m)/q  0)
B=(     0;
     (i+m*l^2)/q;
          0;
        m*l/q)
C=(1 0 0 0;
   0 0 1 0)
D=(0;
   0)
p=eig(A);
R=1;
x=10000;%input ('chart position weighting = ...');
y=1000;%input ('pendulum angel = weighting= ...');
Q=(x 0 0 0;
   0 1 0 0;
   0 0 y 0;
   0 0 0 1);
K=lqr(A,B,Q,R)
te=inv(-A+B*K);
l=C*te*B;
Kr=1/l
```

Appendix 5.

Script M-file to derive Euler-Lagrangian equation of the SIPW and the result

```
%Generating dynamic of the SIPW system using Euler-Lagrangian
Equation
clear all
clc
syms Rb L Mb MB Ib IBx IBy IBz IM G N...
    thtx phix thtxdot phixdot thtxddot phixddot;
%Rb=ball radius,L=height of body CG,Mb MB=mass of (ball,body)
%Ib IBx IBy IBz IM=momen inertia of (ball,body(x,y,z),motor)
%G=momen inertia, N=gear ratio(Rb/Rr(rollerradius))

%%%ball energy
syms balx balxdot Tlbalx Trbalx Vbalx

balx(1)=Rb*(thtx+N*phix);
balx(2)=0;
balx(3)=0;

balxdot(1)=diff(balx(1),thtx)*thtxdot+diff(balx(1),phix)*phixdot
balxdot(2)=diff(balx(2),thtx)*thtxdot+diff(balx(2),phix)*phixdot
balxdot(3)=diff(balx(3),thtx)*thtxdot+diff(balx(3),phix)*phixdot

Tlbalx=simple(Mb/2*(balxdot(1)^2+balxdot(2)^2+balxdot(3)^2))
Trbalx=simple(Ib/2*(thtxdot+N*phixdot)^2)
Vbalx=0

%%%body energy
syms bodix bodixdot bodiosx Tlbodix Trbodix Vbodix

bodiosx(1)=L*sin(thtx);
bodiosx(2)=0;
bodiosx(3)=L*cos(thtx);

bodix(1)=balx(1)+bodiosx(1);
bodix(2)=balx(2)+bodiosx(2);
bodix(3)=balx(3)+bodiosx(3);

bodixdot(1)=simple(diff(bodix(1),thtx)*thtxdot...
    +diff(bodix(1),phix)*phixdot);
bodixdot(2)=simple(diff(bodix(2),thtx)*thtxdot...
    +diff(bodix(2),phix)*phixdot);
bodixdot(3)=simple(diff(bodix(3),thtx)*thtxdot...
    +diff(bodix(3),phix)*phixdot);

Tlbodix=simple(expand(simple((MB/2*(bodixdot(1)^2+bodixdot(2)^2+...
    bodixdot(3)^2)))))
Trbodix=simple(expand(simple((IBx/2*(thtxdot)^2))))
Vbodix=simple(MB*G*bodix(3))
```

```
%%%motor energy
syms Trmotx

Trmotx=simple(IM/2*(thtxdot+phixdot)^2)

%%%Lagrangian function
syms dLthtx dLphix dLdthtxdot dLdphixdot ddLdthtxdotdt
ddLdphixdotdt
syms ELthtx ELphix

Lagrangianx=simple(Tlbalx+Trbalx+Tlbodix+Trbodix+Trmotx-Vbalx-
Vbodix)

%Euler-Lagrangian motion equation
%thetax
dLdthtxdot=simple(expand(simple(diff(Lagrangianx,thtxdot))));
ddLdthtxdotdt=simple(expand(simple(...
    diff(dLdthtxdot,thtxdot)*thtxddot...
    +diff(dLdthtxdot,phixdot)*phixddot...
    +diff(dLdthtxdot,thtx)*thtxdot...
    +diff(dLdthtxdot,phix)*phixdot)));
dLdthtx=simple(expand(simple(diff(Lagrangianx,thtx))));
ELthtx=simple(expand(simple(ddLdthtxdotdt-dLdthtx)))

%phix
dLdphixdot=simple(expand(simple(diff(Lagrangianx,phixdot))));
ddLdphixdotdt=simple(expand(simple(...
    diff(dLdphixdot,thtxdot)*thtxddot...
    +diff(dLdphixdot,phixdot)*phixddot...
    +diff(dLdphixdot,thtx)*thtxdot...
    +diff(dLdphixdot,phix)*phixdot)));
dLdphix=simple(expand(simple(diff(Lagrangianx,phix))));
ELphix=simple(expand(simple(ddLdphixdotdt-dLdphix)))
```

balxdot =

Rb*thtxdot + N*Rb*phixdot

balxdot =

[Rb*thtxdot + N*Rb*phixdot, 0]

balxdot =

[Rb*thtxdot + N*Rb*phixdot, 0, 0]

Tlbalx =

(Mb*Rb^2*(thtxdot + N*phixdot)^2)/2

Trbalx =

(Ib*(thtxdot + N*phixdot)^2)/2

Vbalx =

0

Tlbodix =

(MB*(L^2*thtxdot^2 + 2*cos(thtx)*L*N*Rb*phixdot*thtxdot +
2*cos(thtx)*L*Rb*thtxdot^2 + N^2*Rb^2*phixdot^2 +
2*N*Rb^2*phixdot*thtxdot + Rb^2*thtxdot^2))/2

Trbodix =

(IBx*thtxdot^2)/2

Vbodix =

G*L*MB*cos(thtx)

Trmotx =

(IM*(phixdot + thtxdot)^2)/2

Lagrangianx =

(IM*(phixdot + thtxdot)^2)/2 + (MB*(L^2*thtxdot^2 + 2*cos(thtx)*L*N*Rb*phixdot*thtxdot + 2*cos(thtx)*L*Rb*thtxdot^2 + N^2*Rb^2*phixdot^2 + 2*N*Rb^2*phixdot*thtxdot + Rb^2*thtxdot^2))/2 + (IBx*thtxdot^2)/2 + (Ib*(thtxdot + N*phixdot)^2)/2 + (Mb*Rb^2*(thtxdot + N*phixdot)^2)/2 - G*L*MB*cos(thtx)

ELthtx =

IM*phixddot + IBx*thtxddot + IM*thtxddot + Ib*thtxddot + Ib*N*phixddot + L^2*MB*thtxddot + MB*Rb^2*thtxddot + Mb*Rb^2*thtxddot + MB*N*Rb^2*phixddot + Mb*N*Rb^2*phixddot - G*L*MB*sin(thtx) - L*MB*Rb*thtxdot^2*sin(thtx) + 2*L*MB*Rb*thtxddot*cos(thtx) + L*MB*N*Rb*phixddot*cos(thtx)

ELphix =

IM*phixddot + IM*thtxddot + Ib*N*thtxddot + Ib*N^2*phixddot + MB*N*Rb^2*thtxddot + Mb*N*Rb^2*thtxddot + MB*N^2*Rb^2*phixddot + Mb*N^2*Rb^2*phixddot - L*MB*N*Rb*thtxdot^2*sin(thtx) + L*MB*N*Rb*thtxddot*cos(thtx)

Appendix 6.

Script M-file to derive the SIPW equation of motion and the result

```
%Generating SIPW dynamic equation in matrices form

syms tec Mx Rx Fx

for i=1:2
    switch i
        case 1,
            tec=ELthtx;
        case 2,
            tec=ELphix;
    end
    [c,t]=coeffs(tec,thtxddot);
    Mx(i,1)=0;
    rmd=0;
    for j=1:length(c)
        if t(j)==thtxddot
            Mx(i,1)=c(j);
        elseif t(j)==1
            rmd=c(j);
        end
    end
    [c,t]=coeffs(rmd,phixddot);
    Mx(i,2)=0;
    Rx(i,1)=0;
    for j=1:length(c)
        if t(j)==phixddot
            Mx(i,2)=c(j);
        elseif t(j)==1
            Rx(i,1)=c(j);
        end
    end
end
Mx
Rx

%force matrix
syms voltx ix Kt Kb Rm FBbx FbGx
%Kt Kb Rm=motor parameters(torque, back emf, resistance)
%FBbx=force friction body-ball (x) FbGx=force friction ball-
ground (x)

%motor current dynamic
ix=1/Rm*(voltx-Kb*phixdot);

%force
Fx=[-FBbx*thtxdot;...
    expand(Kt*ix-FbGx*phixdot)]

syms noliqxddot
noliqxddot=inv(Mx)*(Fx-Rx)
```

Mx =

[IBx + IM + Ib + L^2*MB + MB*Rb^2 + Mb*Rb^2 + 2*L*MB*Rb*cos(thtx),
IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb*cos(thtx)]

[IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb*cos(thtx),
IM + Ib*N^2 + MB*N^2*Rb^2 + Mb*N^2*Rb^2]

Rx =

- L*MB*Rb*sin(thtx)*thtxdot^2 - G*L*MB*sin(thtx)

-L*MB*N*Rb*thtxdot^2*sin(thtx)

Fx =

-FBbx*thtxdot

(Kt*voltx)/Rm - FbGx*phixdot - (Kb*Kt*phixdot)/Rm

noliqxddot =

((L*MB*Rb*sin(thtx)*thtxdot^2 - FBbx*thtxdot +
G*L*MB*sin(thtx))*(IM + Ib*N^2 + MB*N^2*Rb^2 +
Mb*N^2*Rb^2))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 +
IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + L^2*MB^2*N^2*Rb^2 +
Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 +
IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 - 2*IM*MB*N*Rb^2 -
2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 -
L^2*MB^2*N^2*Rb^2*cos(thtx)^2 + 2*IM*L*MB*Rb*cos(thtx) -
2*IM*L*MB*N*Rb*cos(thtx)) + ((FbGx*phixdot - (Kt*voltx)/Rm +
(Kb*Kt*phixdot)/Rm - L*MB*N*Rb*thtxdot^2*sin(thtx))*(IM + Ib*N +
MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb*cos(thtx)))/(IBx*IM + IM*Ib -
2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 +
IM*Mb*Rb^2 + L^2*MB^2*N^2*Rb^2 + Ib*L^2*MB*N^2 +
IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 +
IM*Mb*N^2*Rb^2 - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 +
L^2*MB*Mb*N^2*Rb^2 - L^2*MB^2*N^2*Rb^2*cos(thtx)^2 +
2*IM*L*MB*Rb*cos(thtx) - 2*IM*L*MB*N*Rb*cos(thtx))

- ((L*MB*Rb*sin(thtx)*thtxdot^2 - FBbx*thtxdot + G*L*MB*sin(thtx))*(IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb*cos(thtx)))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + L^2*MB^2*N^2*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - L^2*MB^2*N^2*Rb^2*cos(thtx)^2 + 2*IM*L*MB*Rb*cos(thtx) - 2*IM*L*MB*N*Rb*cos(thtx)) - ((FbGx*phixdot - (Kt*voltx)/Rm + (Kb*Kt*phixdot)/Rm - L*MB*N*Rb*thtxdot^2*sin(thtx))*(IBx + IM + Ib + L^2*MB + MB*Rb^2 + Mb*Rb^2 + 2*L*MB*Rb*cos(thtx)))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + L^2*MB^2*N^2*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - L^2*MB^2*N^2*Rb^2*cos(thtx)^2 + 2*IM*L*MB*Rb*cos(thtx) - 2*IM*L*MB*N*Rb*cos(thtx))

Appendix 7.

Script M-file to linearise the SIPW equation of motion and the result

```
%Linearising SIPW dynamic equation

syms qx qxdot qxddot...
     x xbar nolissmatx nolissmatxbar...
     jacobianmatx jacobianmatxbar liqxdot;

%nonlinear statespace
qx=[thtx phix];
qxdot=[thtxdot phixdot];
qxddot=[thtxddot phixddot];

x=[qx qxdot];
xbar=[0,phix,0,0];

nolissmatx=[ thtxdot;...
             phixdot;...
             noliqxddot];
nolissmatxbar=subs(nolissmatx,x,xbar);

jacobianmatx=jacobian(nolissmatx,x);
jacobianmatxbar=subs(jacobianmatx,x,xbar);

liqxdot= nolissmatxbar+jacobianmatxbar*(transpose(x));

%create State matrices
%state is trans([qx qxdot])
syms Ax Bx
for i=1:4
    rmd=liqxdot(i);
    for j=1:4
        [c,t]=coeffs(rmd,x(j));
        Ax(i,j)=0;
        for k=1:length(c)
            if t(k)==1
                rmd=c(k);
            else
                Ax(i,j)=c(k);
            end
        end
    end
```

```
[c,t]=coeffs(rmd,voltx);
    Bx(i,1)=0;
    rmd=0;
    for k=1:length(c)
        if t(k)==1
            rmd=c(k);
        else
            Bx(i,1)=c(k);
        end
    end
end
Ax
Bx
rmd
```

Ax =

[
0, 0,
1,
0]

[
0, 0,
0,
1]

[(G*L*MB*(IM + Ib*N^2 + MB*N^2*Rb^2 + Mb*N^2*Rb^2))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb), 0, -(FBbx*(IM + Ib*N^2 + MB*N^2*Rb^2 + Mb*N^2*Rb^2))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb), ((FbGx + (Kb*Kt)/Rm)*(IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb -

2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb)]

[-(G*L*MB*(IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb), 0, (FBbx*(IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb), -((FbGx + (Kb*Kt)/Rm)*(IBx + IM + Ib + L^2*MB + MB*Rb^2 + Mb*Rb^2 + 2*L*MB*Rb))/(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb)]

Bx =

0

0

-(Kt*(IM + Ib*N + MB*N*Rb^2 + Mb*N*Rb^2 + L*MB*N*Rb))/(Rm*(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 + Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 + IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb - 2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 - 2*IM*L*MB*N*Rb))

(Kt*(IBx + IM + Ib + L^2*MB + MB*Rb^2 + Mb*Rb^2 + 2*L*MB*Rb))/(Rm*(IBx*IM + IM*Ib - 2*IM*Ib*N + IBx*Ib*N^2 + IM*Ib*N^2 + IM*L^2*MB + IM*MB*Rb^2 + IM*Mb*Rb^2 +

```
Ib*L^2*MB*N^2 + IBx*MB*N^2*Rb^2 + IM*MB*N^2*Rb^2 +
IBx*Mb*N^2*Rb^2 + IM*Mb*N^2*Rb^2 + 2*IM*L*MB*Rb -
2*IM*MB*N*Rb^2 - 2*IM*Mb*N*Rb^2 + L^2*MB*Mb*N^2*Rb^2 -
2*IM*L*MB*N*Rb))


rmd =


     0
```

Appendix 8.

Script M-file to derive gain controller *K* for the SIPW

```
%Calculating the controller, Trying with parameter values

%System parameters
Rb=0.2;
Mb=2.3529;
Ib=0.0612;
Rr=0.03;
MB=37.3419;
IBx=3.9002;
L=0.7372;
IM=0;
G=9.81;
FBbx=0;
FbGx=0;
Kb=0.4;
Kt=0.3;
Rm=6;


Asub=subs(Ax);
Bsub=subs(Bx);
Csub=eye(4);
Dsub=zeros(4,1);

%Controllability
con=ctrb(Asub,Bsub);
uncost=length(Asub)-rank(con)
kondis=cond(con)

%tambah 2 reference states for thtx and thty
c=[0 1 0 0];
Abar=[Asub zeros(4,1);...
        c  zeros(1,1)];
Bbar=[Bsub;...
      zeros(1,1)];
con1=ctrb(Abar,Bbar);
unconst1=length(Abar)-rank(con1)
kondis1=cond(con1)

%LQR control
Q=eye(5);
Q(1,1)=6e4;
Q(5,5)=4e2;
R=(1e3)*eye(1);
K=lqr(Abar,Bbar,Q,R);
Kf=K(1:4)
Ki=K(5)

olpoles=eig(Abar)
clpoles=eig(Abar-Bbar*K)
```