CONTROL OF A CLASS OF UNSTABLE NONLINEAR SYSTEMS

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech

in

Electrical and Electronics Engineering

By

Ananda Rangan N (107116009)

Bhavyasree Aruljothi (107116015)

ELECTRICAL AND ELECTRONICS ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

TIRUCHIRAPPALLI – 620015

JUNE 2020

BONAFIDE CERTIFICATE

This is to certify that the project titled **CONTROL OF A CLASS OF UNSTABLE NONLINEAR SYSTEMS** is a bonafide record of the work done by

Ananda Rangan N (107116009)

Bhavyasree Aruljothi (107116015)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Electrical** in **Electronics Engineering** of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI**, during the year 2019- 20.

Dr. V. Sankaranarayanan
Guide

Dr. V. Sankaranarayanan
Head of the Department

Project Viva-voce held on _____

Internal Examiner

External Examiner

ABSTRACT

Control of nonlinear system receives more attention from control researchers due to its importance and applications. Many nonlinear control methods are available based on Lyapunov design, sliding mode control, back-steeping, passivity and energy shaping methods. Especially controlling an unstable nonlinear system is more challenging and there is no unique control design method available in literature.

This thesis presents a novel control algorithm for state tracking for a class of unstable underactuated nonlinear systems subject to nonholonomic constraints. The class of unstable nonlinear systems possess similar zero dynamics; a set of stable states and a set of unstable states. This project considers two different systems belonging to this class of unstable underactuated nonlinear systems – mobile inverted pendulum (MIP) and quadcopter. The nonlinear dynamics of both the systems are derived mathematically. The stable and unstable states are derived from the dynamics equation of the system and verified from zero dynamics analysis through simulation. The proposed control algorithm for the class of nonlinear systems is applied to MIP and quadcopter and the validity of the proposed control algorithm is verified through simulation.

Keywords: Nonlinear control; underactuated systems; mobile inverted pendulum (MIP); quadcopter; zero dynamics

ACKNOWLEDGEMENTS

We would like to thank a number of people who have encouraged and helped us in writing this thesis without whom the completion of this project would not have been possible. We are very proud of the achieved results and we appreciate the support we received from all sides.

We are much obliged to **Dr. V. Sankaranarayanan**, Professor, Head of the Department, Department of Electrical and Electronics Engineering, for helping us and guiding us in the course of this project. We are extremely thankful for his support and confidence in our team and for providing the opportunity to be part of such a far-reaching project. His broad knowledge in system modeling and control and his very professional and friendly way of support had a wide influence on the results and also on our education. He persistently helped us with many complex questions and motivated us to extend the scope of the project into the right directions.

Many thanks to **Dr. G. Saravana Ilango** and **Dr. S. Senthil Kumar**, our internal reviewers, for their insight and advice on this project during the reviews.

We are very thankful to all members of this team and are very happy to have spent a semester working together in this project. We have formed a very capable team and helped each other to realize our ideas.

Most importantly, we are very grateful to all our friends and family members who supported us during the whole project and helped us to achieve our aims.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATIONS

τ_r, τ_l	Torques provided by wheel actuators acting on left and right wheel
ϕ_r , ϕ_l	Rotation angles of the left and right wheels
x_0, y_0	The coordinates in X-Y plane
α	Tilt angle of the pendulum
θ	Heading angle of the MIP
v	Forward velocity of the MIP
M_w	Mass of each wheel
r	Radius of each wheel
M_B	Mass of the MIP body
g	Acceleration due to gravity
l	Distance from the point O to the center of gravity of the MIP body
2b	Distance between the two wheels of the MIP
I_w	Moment of inertia of the wheel along the wheel axis direction
I_{Wd}	Moment of inertia of the wheel about Z-axis through the center of wheel
I_B	Moment of inertia of the MIP body along the wheel axis direction
I_Z	Moment of inertia of the MIP body about the Z-axis through the center
T	Input thrust force
τ_ϕ , τ_θ, τ_ψ	Input torques along roll, pitch and yaw
x,y,z	Coordinates of quadcopter in XYZ cartesian space
ϕ, θ, ψ	Roll, pitch and yaw angles measured w.r.t body frame
V_x, V_y, V_z	Linear velocities in XYZ space (Inertial frame)

p, q, r	Angular velocities in XYZ space (Inertial frame)
m	Mass of the quadcopter
ω_i	Angular velocity of ith propeller
g	Acceleration due to gravity
I_x, I_y, I_z	Inertia of Quadcopter along the X, Y and Z axes

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

1.1.1 Nonlinear Control Systems

Control systems are prevalent in nature and in man-made systems. Natural control exists in biological and chemical processes, which may help, for example, to maintain the various constituents at their correct levels. Governors were designed to regulate the speed of steam engines in the early days of the industrial revolution, while in modern times computerized control systems have become ubiquitous in industrial plants, robot manipulators, aircraft and spacecraft etc.

Control theory has been crucial in NASA's Apollo and Space Shuttle program. Control systems such as in these examples use an essential way the idea of feedback. Control theory is the engineering / science branch concerned with designing and analyzing control systems. Linear control theory deals with systems for which an inherent linear model is assumed and is a relatively mature subject, complete with firm theoretical framework and a broad range of powerful and applicable design methodologies, see [1]. Nonlinear control theory, on the other hand, deals with systems for which linear models are not adequate, and is relatively immature, particularly in respect of applications. In addition, given the existence of nonlinearities, linear system techniques are still employed.

Real world systems are inherently nonlinear in nature at least when viewed over wide operating range although under certain assumptions many of the systems are assumed to 'behave' as linear in the neighbourhood of a certain operating point at low speeds. Many physical processes are represented by nonlinear models. Examples include; coulomb friction, gravitational and electrostatic attraction, voltage-current characteristics of most electronic systems and drag on a vehicle in motion. Recently, many researchers from such broad areas like process control, biomedical engineering, robotics, aircraft and spacecraft control have shown an active interest in the design and analysis of nonlinear

control strategies. Thus, most of the real problems necessitate invariably bumping into nonlinearities [2].

Nonlinear control system deals with dynamical systems that are modelled by a finite number of coupled first-order ordinary differential equations as shown in Eq 1.1

$$\begin{aligned}
\dot{x_1} &= f_1(t, x_1, x_2, ...x_n, u_1, ...u_p) \\
\dot{x_2} &= f_2(t, x_1, x_2, ...x_n, u_1, ...u_p) \\
. \quad & \quad . \\
. \quad & \quad . \\
. \quad & \quad . \\
\dot{x_n} &= f_n(t, x_1, x_2, ...x_n, u_1, ...u_p) \qquad (1.1)
\end{aligned}$$

Where the left hand term of the i^{th} equation denotes the derivative of x_i. With respect to time t, and u_1, u_2, …, u_p are represented by input variables. x_1, x_2, …, x_n are usually called as state variables. The parameters of eqn 1.1 can be written in vector notation to write the equations in compact form as shown below

$$x = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ . \\ u_p \end{bmatrix}, f(t, x, u) = \begin{bmatrix} f_1(t, x, u) \\ f_2(t, x, u) \\ . \\ . \\ . \\ f_n(t, x, u) \end{bmatrix}$$

And rewrite n-first order differential equations of eqn 1.1 as one n-dimensional first order vector differential equation

$$\dot{x} = f(t, x, u) \qquad (1.2)$$

Eqn 1.2 is called as state equation, x is the state and, u is the input.

$$y = h(t, x, u) \qquad (1.3)$$

Eqn 1.3, called as the output equation, defines a q-dimensional output vector y, that comprises variables of particular interest in the analysis of the dynamical system (e.g, variables that can be physically measured or variables that are required to behave in a specified manner). Eqns 1.1 and 1.2 together are known as the state-space model.

1.1.2 Need for Nonlinear Control Systems

Primary reasons behind growing interest in nonlinear control include [3]; improvement of linear control systems, analysis of hard nonlinearities, need to deal with model uncertainties and design simplicity. Nonlinear strategies improve trivial approaches by taking into accounts the dynamic forces like centripetal and Coriolis forces which vary in proportion to the square of the speed. So, the linear control laws put a serious constraint on speed of motion to achieve a specified accuracy. However, simple nonlinear controller can reasonably compensate the nonlinear forces thus achieving high speed in an ample workspace. Also, hard nonlinearities like dead-zones, hysteresis, Coulomb friction, stiction, backlash and saturation do not permit linear approximation of real-world systems. After predicting these nonlinearities, nonlinear approaches properly compensate these to achieve unmatched performance.

Moreover, real systems often exhibit uncertainties in the model parameters primarily due to sudden or slow change in the values of these parameters. A nonlinear controller through robustness or adaptability can handle the consequences due to model uncertainties [4]. The systems that are not linearly controllable or observable may be controllable/observable in a nonlinear sense. The linear control of the manipulators assumes each joint to be independent and considers the inertia 'seen' by each joint as a constant. This approximation leads to non-uniform damping throughout the work envelope and results other undesirable effects [5]. Here, a nonlinear controller would reach the desired reference faster and the undesirable effects are removed.

1.1.3 Stability of Nonlinear Systems

In simple terms, a system is said to be mall perturbations give rise to small variations which damp down to negligible vales and a system is called unstable if small perturbations give rise to large variations. However, the stability of a system is defined more technically based on the stability of its equilibrium points [6].

A state x_e is an equilibrium point of a system as described in eqn (1.2), if

$$f(t, x_e, 0) = 0 \qquad (1.4)$$

An equilibrium point x_e is stable according to Lyapunov if:

$$\forall \epsilon > 0, \exists \delta(\epsilon) : ||x_0 - x_e|| < \epsilon, \forall t > 0 \qquad (1.5)$$

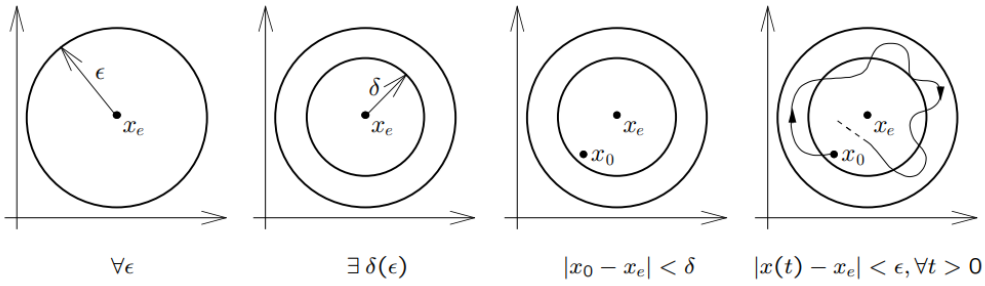


Figure 1.1 Graphical representation of Lyapunov stability

An equilibrium point x_e is defined as asymptotically stable if it is stable according to Lyapunov and if

$$||x_0 - x_e|| \to 0, t \to \infty$$

Here, $\| \cdot \|$ denotes 2-norm of a vector, also known as the magnitude of a vector i.e. the square root of the sum of squares of each element.

If an equilibrium point does not satisfy the above conditions, then it is defined as an unstable equilibrium point. In general, when a system is said to be unstable, the equilibrium points, in the subspace in which the system is being considered, are unstable equilibrium points.

1.1.4 Class of Nonlinear Systems

This thesis presents a control algorithm for a class of unstable nonlinear systems that have similar properties. The control algorithm is applicable to nonlinear systems that are

(i) underactuated (ii) subject to nonholonomic constraints

Underactuated systems are those systems that have fewer number of actuators than the actual Degrees of Freedom of the system. Nonholonomic constraints refer to the

constraints on the position (configuration) of a system of particles. The thesis takes two such systems, mobile inverted pendulum (MIP) and quadcopter.

1.2 PROBLEM FORMULATION

Consider the nonlinear affine in control system

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \qquad (1.6)$$

Where $f(x)$ and $g(x)$ are assumed to be a smooth vector field in the domain R^n. Let x_e be the equilibrium, that is $f(x_e) = 0$. The system is unstable around the equilibrium point x_e. The control objective is to asymptotically stabilize the system around its (one of) equilibrium point.

1.3 SOFTWARE USED

The project verifies the proposed control algorithm on the two systems MIP and quadcopter on the MATLAB/SIMULINK software. The zero dynamics analysis and state tracking are verified on the same platform. MATLAB (*matrix laboratory*) is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. Although MATLAB is intended primarily for numerical computing, an additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

MATLAB and SIMULINK offer [7]:

(i) A multi-domain block diagram environment for modeling plant dynamics, designing control algorithms, and running closed-loop simulations

(ii) Plant modeling using system identification or physical modeling tools

(iii) Prebuilt functions and interactive tools for analyzing overshoot, rise time, phase margin, gain margin, and other performance and stability characteristics in time and frequency domains

(iv) Automatic tuning of PID, gain-scheduled, and arbitrary SISO and MIMO control systems

Hence, SIMULINK is the preferred simulation software to simulate the control algorithm for this project.

1.4 PROPOSED WORK

In Chapter 2, the various literature review on the dynamics of the MIP and the quadcopter are discussed. The two systems are explained and the derivation of the affine in control equation of the dynamics of both the systems are derived mathematically. In Chapter 3, the novel control algorithm is explained and the mathematical equations are derived, and the control algorithm is implemented on the two test systems; the MIP and the quadcopter. In Chapter 4, the simulation and results are discussed. The zero dynamics of the two systems are verified through simulation. The control algorithm is then added to the simulation and the results are verified on the simulation platform SIMULINK.

CHAPTER 2

LITERATURE REVIEW

2.1 MOBILE INVERTED PENDULUM

2.1.1 Introduction

The mobile inverted pendulum (MIP) is a robotic system with nonholonomic constraint due to no-slip condition on the wheel. In particular, it has four configuration variables to be controlled using only two control inputs. Hence, it is an underactuated system. It is a well-known nonlinear control systems problem and is widely used as a benchmark for testing control algorithms, where the goa is to stabilize the pendulum and move the configuration from point to point. [8], [9], and [10] show some existing work on MIP. Segway, self-balancing hoverboard and self-balancing unicycle are some commercial examples of classic mobile inverted pendulum.

A normal pendulum is stable when hanging downwards, whereas an inverted pendulum is inherently unstable, and must be actively balanced in order to remain upright; this can be done by applying a torque at the pivot point, by moving the pivot point horizontally as part of a feedback system, changing the rate of rotation of a mass mounted on the pendulum on an axis parallel to the pivot axis and thereby generating a net torque on the pendulum. A simple demonstration of moving the pivot point in a feedback system is achieved by balancing an upturned broomstick on the end of one's finger.

Several control strategies for the MIP have been developed over the past few decades in the literature, based on stabilization principle of inverted pendulum [11], and pole placement ([10], [12]), backstepping ([13], [14]), sliding mode [15], Lyapunov redesign [16], adaptive control, fuzzy control ([17], [18]), neural networks ([19], [20], [21], [22], [23]), neuro-fuzzy [24], energy based controllers [25] and model based controllers ([26], [27], [28], [29]) are proposed. Most of the control techniques discussed in the literature focus on linear controllers.

2.1.2 Dynamics Equation of the MIP

Figure 2.1 shows the model of MIP, where the pendulum is anchored to a base chassis with wheels mounted on each side. The MIP has two components, the wheels, and the body of the MIP. The body of the MIP includes the pendulum, pendulum bob, chassis and every component apart from the wheels of the MIP. The parameters and variables of the MIP are described in Table 2.1

Let q_0 represent the vector of the MIP. Then,

$$q_0 = \begin{bmatrix} x_0 \\ y_0 \\ \theta \\ \phi \\ \theta_r \\ \theta_l \end{bmatrix}$$

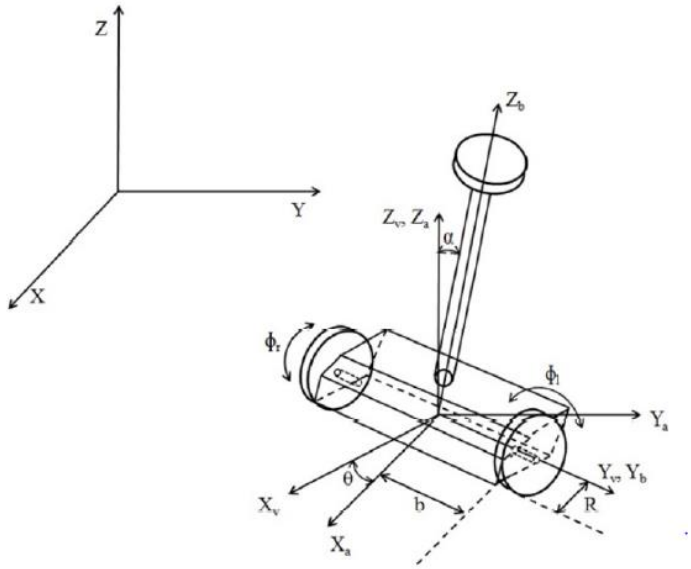


Figure 2.1 Schematic Diagram of an MIP

Considering pure rolling and non-slipping between the wheels and the ground, the MIP is subjected to the following nonholonomic constraints

$$
\begin{aligned}
&-\dot{x_0}sin\theta + \dot{y_0}cos\theta = 0 \\
&\dot{x_0}cos\theta + \dot{y_0}sin\theta + d\frac{\dot{\theta}}{2} - r\dot{\theta_r} = 0 \qquad (2.1) \\
&\dot{x_0}cos\theta + \dot{y_0}sin\theta - d\frac{\dot{\theta}}{2} - r\dot{\theta_r} = 0
\end{aligned}
$$

Notation	Definition
τ_l, τ_r	Torques provided by wheel actuators acting on left and right wheels
ϕ_l, ϕ_r	Rotation angles of the left and right wheels
x_0, y_0	The coordinates of point in X-Y plane
α	Tilt angle of the pendulum
θ	Heading angle of the MIP
v	Forward velocity of the MIP
M_w	Mass of each wheel
r	Radius of each wheel
M_b	Mass of the MIP body
g	Acceleration due to gravity
l	Distance from the point O to the centre of gravity of the MIP body
$2b$	Distance between the two wheels of the MIP
I_w	Moment of inertia of the wheel along the wheel axis direction
$I_w d$	Moment of inertia of the wheel about the Z-axis through the center of wheel
I_B	Moment of inertia of the MIP body along the wheel axis direction
I_Z	Moment of inertia of the MIP body about the Z axis through the centre

Table 2.1 The parameters and variables of MIP

Let ***v*** be the vector of derivatives of tilt angle of the pendulum, displacement of the MIP and heading angle of the MIP,

$$\mathbf{v} = (\dot{\phi}, v, \dot{\theta})$$

The nonholonomic constraints of the MIP can be written such that

$$\mathbf{F}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}_{3\times 1} \qquad (2.2)$$

Where,

$$\mathbf{F}(\mathbf{q}) = \begin{bmatrix} -sin\theta & cos\theta & 0 & 0 & 0 & 0 \\ cos\theta & sin\theta & \frac{d}{2} & 0 & -r & 0 \\ cos\theta & sin\theta & -\frac{d}{2} & 0 & 0 & -r \end{bmatrix}$$

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{v} \qquad (2.3)$$

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} 0 & cos\theta & 0 \\ 0 & sin\theta & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & \frac{1}{r} & \frac{d}{2r} \\ 0 & \frac{1}{r} & -\frac{d}{2r} \end{bmatrix}$$

Here, d refers to the distance between the two wheels of the MIP along the axis of the wheels.

The sum of the translational and rotational kinetic energy of the wheels of the MIP given by

$$T_w = \frac{1}{2}M_w r^2 \dot{\phi_l}^2 + \frac{1}{2}M_w r^2 \dot{\phi_r}^2 + \frac{1}{2}I_w r^2 \dot{\phi_l}^2 + \frac{1}{2}I_w r^2 \dot{\phi_r}^2 + + \frac{1}{2}I_{wd} r^2 \dot{\theta}^2 \qquad (2.4)$$

The rotational and translational kinetic energy of the body of the MIP are given by

$$T_{B,translational} = \frac{1}{2}M(\dot{\theta} l sin\alpha)^2 + \frac{1}{2}M[\dot{\alpha} l cos\alpha + \frac{1}{2}r(\dot{\theta_l} + \dot{\theta_r})]^2 + \frac{1}{2}M(-\dot{\alpha} l sin\alpha)^2 \quad (2.5)$$

$$T_{B,rotational} = \frac{1}{2}I_B \dot{\alpha}^2 + \frac{1}{2}I_Z \dot{\theta}^2 \qquad (2.6)$$

The moment of inertia I_z depends on the pendulum tilt angle. However, for very small tilt angles, it is assumed constant.

The potential energy of the MIP is given by

$$P = Mgl cos\alpha \qquad (2.7)$$

The Lagrangian function is the difference between the total kinetic energy and total potential energy of the system,

$$L = T_w + T_{B,translational} + T_{B,rotational} - P \qquad (2.8)$$

Applying Euler-Lagrange equation, the dynamic equations of the MIP become,

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\mathbf{q}}}) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{E}(\mathbf{q})\mathbf{T} + \mathbf{F}^T(\mathbf{q})\lambda \qquad (2.9)$$

Where, **T** is the input motor torque vector and **E(q)** is the matched matrix, and lambda is the Lagrangian multiplier,

$$\mathbf{T} = \begin{bmatrix} T_r \\ T_l \end{bmatrix}, \mathbf{E}(\mathbf{q}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Substituting Eqns (2.4-2.8) into Eqn(2.9) and collecting similar terms, we get,

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}(\mathbf{q})\mathbf{T} + \mathbf{F}^T(\mathbf{q})\lambda$$

Where,

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & m_{44} & m_{45} & m_{46} \\ 0 & 0 & 0 & m_{54} & m_{55} & m_{56} \\ 0 & 0 & 0 & m_{64} & m_{65} & m_{66} \end{bmatrix}, \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 \\ 0 \\ C_{31} \\ C_{41} \\ C_{51} \\ C_{61} \end{bmatrix}$$

$$m_{33} = Ml^2 sin^2\alpha + 2I_{wd} + I_Z$$
$$m_{44} = Ml^2 + I_B$$
$$m_{45} = m_{54} = m_{46} = m_{64} = \frac{1}{2}Mrlcos\alpha$$
$$m_{55} = m_{66} = \frac{1}{4}Mr^2 + I_w + M_w r^2$$
$$m_{56} = m_{65} = \frac{1}{4}Mr^2$$
$$C_{31} = Ml^2\dot{\theta}^2\dot{\alpha}sin2\alpha$$
$$C_{41} = -Mlsin\alpha(\dot{\theta}^2 lcos\alpha + g)$$
$$C_{51} = C_{61} = -\frac{1}{2}Mrl\dot{\alpha}^2 sin\alpha$$

By differentiating both sides of Eqn (2.3) and substituting in the equation above, we get,

$$\mathbf{AS}\dot{\mathbf{v}} + \mathbf{A}\dot{\mathbf{S}}\mathbf{v} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}(\mathbf{q})\mathbf{T} + \mathbf{F}^T(\mathbf{q})\lambda \qquad (2.10)$$

Premultiplying S^T (q) to both sides of the eqn (2.10), and substituting $S^T F^T = 0$, we get,

$$\mathbf{S}^T\mathbf{AS}\dot{\mathbf{v}} + \mathbf{S}^T(\mathbf{A}\dot{\mathbf{S}}\mathbf{v} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})) = \mathbf{S}^T\mathbf{E}(\mathbf{q})\mathbf{T} \qquad (2.11)$$

$$Since, \mathbf{S}^T\mathbf{A}\dot{\mathbf{S}} = 0,$$

$$\dot{\mathbf{v}} = -\bar{\mathbf{A}}^{-1}\mathbf{S}^T\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \bar{\mathbf{A}}^{-1}\mathbf{S}^T\mathbf{E}(\mathbf{q})\mathbf{T} \qquad (2.12)$$

$$Where, \bar{\mathbf{A}} = \mathbf{S}^T\mathbf{AS}$$

Substituting the value for the matrices on the right hand side of eqn(2.12) and expanding, we get a part of the dynamic equations of the MIP as follows:

$$\ddot{\alpha} = \frac{-Ml(\dot{\theta}^2 lcos\alpha + g)[Mr^2 sin\alpha + 2I_w sin\alpha + 2M_w r^2 sin\alpha] + M^2 l^2 r^2 \dot{\phi}^2 sin\alpha cos\alpha}{\Delta + M^2 l^2 r^2 cos^2\alpha}$$
$$+ \frac{Mr^2 + 2I_w + 2M_w r^2 + Mlrcos\alpha}{\Delta + M^2 l^2 r^2 cos^2\alpha}(\tau_l + \tau_r)$$

$$\dot{v} = \frac{Mlr^2[Mlsin\alpha cos\alpha(\dot{\theta}^2 lcos\alpha + g) - \dot{\alpha}^2 sin\alpha Ml^2 - \dot{\alpha}^2 sin\alpha I_B]}{\Delta + M^2 l^2 r^2 cos^2\alpha} - \frac{Mlr^2 cos\alpha + Mrl^2 + I_B r}{\Delta + M^2 l^2 r^2 cos^2\alpha}(\tau_l + \tau_r) \qquad (2.13)$$

$$\ddot{\theta} = \frac{2Mr^2 l^2 \dot{\theta}\dot{\alpha} sin2\alpha}{\Omega + 2Ml^2 r^2 cos^2\alpha} + \frac{rd}{\Omega + 2Ml^2 r^2 cos^2\alpha}(\tau_l - \tau_r)$$

Where,

$$\Delta = -M^2 l^2 r^2 - 2Ml^2 I_w - 2Ml^2 M_w r^2 - I_B Mr^2 - 2I_B I_w - 2M_w r^2 I_B,$$
$$\Omega = -M_w d^2 r^2 - 4I_{wd} r^2 - d^2 I_w - 2Ml^2 r^2$$

Let us consider the state vector x given by

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ \alpha \\ \dot{\alpha} \\ v \\ \dot{\theta} \end{bmatrix}$$

Then the affine in control equation of the MIP as described in eqn (1.6) can be written as

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g_1(\mathbf{x})\tau_r + g_2(\mathbf{x})\tau_l \qquad (2.14)$$

Where,

$$f(\mathbf{x}) = \begin{bmatrix} vcos\theta \\ vsin\theta \\ \dot{\theta} \\ \dot{\alpha} \\ f_5(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_6(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_7(\alpha, \dot{\alpha}, \dot{\theta}) \end{bmatrix}; g_1(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{g_5(\alpha)}{RT_1(\alpha)} \\ \frac{g_6(\alpha)}{RT_1(\alpha)} \\ \frac{b}{RT_1(\alpha)} \end{bmatrix}; g_1(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{g_5(\alpha)}{RT_1(\alpha)} \\ \frac{g_6(\alpha)}{RT_1(\alpha)} \\ -\frac{b}{RT_1(\alpha)} \end{bmatrix}$$

The functions appearing in the state equations can be defined as

$$T_1(\alpha) = b_2 b_3 - b_5^2 cos^2(\alpha) > 0$$

$$T_2(\alpha) = b_4 + b_1 sin^2\alpha > 0$$

$$g_5(\alpha) = b_3 R + b_5 cos\alpha > 0$$

$$g_6(\alpha) = b_2 + R b_5 cos\alpha > 0$$

$$f_5(\alpha, \dot{\alpha}, \dot{\theta}) = \frac{sin2\alpha(b_3 b_1 \dot{\theta}^2 - b_5^2 \dot{\alpha}^2) + 2b_5 b_3 g sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)}$$

$$f_6(\alpha, \dot{\alpha}, \dot{\theta}) = \frac{sin2\alpha(-b_1 b_5 \dot{\theta}^2 cos\alpha - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)}$$

$$f_6(\alpha, \dot{\alpha}, \dot{\theta}) = \frac{-b_1 \dot{\alpha} \dot{\theta} sin2\alpha}{b_4 + b_1 sin^2\alpha}$$

Where,

$$b_1 = M_b l^2 + I_x - I_z$$
$$b_2 = M_b l^2 + I_y$$
$$b_3 = \frac{M_b R^2 + 2(I_{wa} + M_w R^2)}{R^2}$$
$$b_4 = \frac{R^2(I_z + 2I_{wd} + 2b^2 M_w) + 2b^2 I_{wa}}{R^2}$$
$$b_5 = M_b l$$

2.2 QUADCOPTER

2.2.1 Introduction

A quadcopter is a multirotor helicopter that is lifted and propelled by four rotors, also called a quadrotor helicopter or quadrotor. Quadcopters are defined as rotorcraft as opposed to fixed-wing aircraft, as a group of rotors (vertically oriented propellers) produces their lift.

Quadcopters usually use two pairs of identical fixed pitch propellers; two in the clockwise direction and two in the opposite direction. To achieve control, these use independent variation of the speed of each rotor. By adjusting the speed of each rotor, it

is possible to produce a desired total thrust; to determine both laterally and longitudinally for the center of thrust; and to generate a desired total torque, or turning force [30].

Quadcopters are different from conventional helicopters that use rotors that can dynamically vary the pitch of their blades as they move around the hub of the rotor. Quadcopters (then referred to either as 'quadrotors' or simply as 'helicopters') were seen as possible solutions to some of the persistent problems in vertical flight in the early days of flight. Torque-induced control problems (as well as efficiency problems arising from the tail rotor, which does not generate a useful lift) can be eliminated by counter-rotation, and relatively short blades are much easier to design. In the 1920s and 1930s a variety of manned designs emerged. These vehicles were among the first successful heavier-than-air VTOL (Vertical Take Off and Landing) vehicles [31]. However, due to poor stability increase and limited control authority, early prototypes suffered from poor performance, and latter prototypes needed too much pilot work load.

Between 2005 and 2010, developments in electronics permitted cheap lightweight flight controllers, accelerometers (IMU), global positioning system, and cameras to be manufactured. This led to success of the quadcopter system for small unmanned aerial vehicles. Quadcopters can be flown both indoors and outdoors with their small size and maneuverability ([32], [33]).

Due to their mechanical simplicity, quadcopters are cheaper and lasts longer than conventional helicopters at a small size [[34]]. Also, their smaller blades are advantageous because they have less kinetic energy, which reduces their ability to cause damage. [35] Fixed propeller quadcopters, however, develop drawbacks relative to conventional helicopters as the size increases. Increasing size of blade increases its momentum. This means that it takes longer to change blade speed which has a negative impact on control. Helicopters do not encounter this issue because the ability to control blade pitch does not have a large effect on increasing the size of the rotor disk.

Quadcopter aircraft are frequently used as amateur model aircraft projects, due to their ease of construction and control. Quadcopter also has a wide variety of applications in military and law enforcement, photography, journalism, drone delivery, art, sport apart from being an interesting research topic.

2.2.2 Flight Dynamics

Each rotor produces both a torque and a thrust about its center of rotation, as well as a drag force opposite the direction of flight of the vehicle. If all rotors spin at the same angular velocity, with rotors one and three in the clockwise direction and rotors two and four in the counterclockwise direction, the total aerodynamic torque and thus the angular acceleration about the yaw axis is exactly zero, which means that a tail rotor is not necessary as in traditional helicopters. Yaw is induced by mismatching the aerodynamic torque balance (i.e. by offsetting the cumulative thrust commands between counter-rotating pairs of blades) [36]

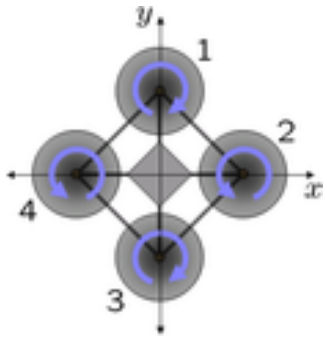


Figure 2.2 Schematic diagram of a quadcopter

Before understanding the flight dynamics of the quadcopter there are 4 terms associated with the quadcopter. These terms refer to the dimensions that an aircraft in flight is free to move about.

- *Pitch* : Pitch refers to the nose of the aircraft going up or down. You could think of it as climbing or diving
- *Yaw* : Yaw on the other hand refers to the nose of the aircraft turning left or right. You could simply think of this as turning.
- *Roll* : To understand roll think of an axis running from the front to the back of the aircraft. When an aircraft rolls it is turning on this access. You can also think of roll as a tilt.

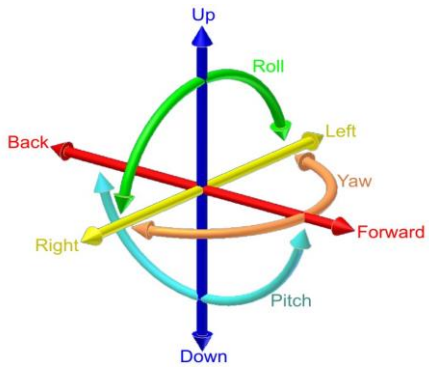


Figure 2.3 Representation of roll, pitch and yaw

Altitude: A quadrotor hovers or adjusts its altitude by applying equal thrust to all four rotors. To increase the altitude, the rotors rotate faster, generating more thrust and they rotate slower to decrease altitude

Yaw: A quadrotor adjusts its yaw by applying more thrust to the pair of rotors rotating in the same direction and lower thrust to the pair rotating in the other direction so that the yaw torque changes, but the thrust remains the same.

Roll or pitch: The roll or pitch of a quadcopter is controlled by applying more thrust to one rotor and less thrust to its diametrically opposite rotor, while the other pair rotors rotate at the same speed.

(a) (b) (c)

Figure 2.4 Schematic diagram to control altitude, yaw and pitch or roll of quadcopter

2.2.3 Dynamics Equations of the Quadcopter

The structure of the quadcopter is shown in Figure 2.4 including the corresponding angular velocities, torques and forces created by the four rotors (numbered from 1 to 4). The parameters and variables related to the dynamic equations are listed in Table 2.2.

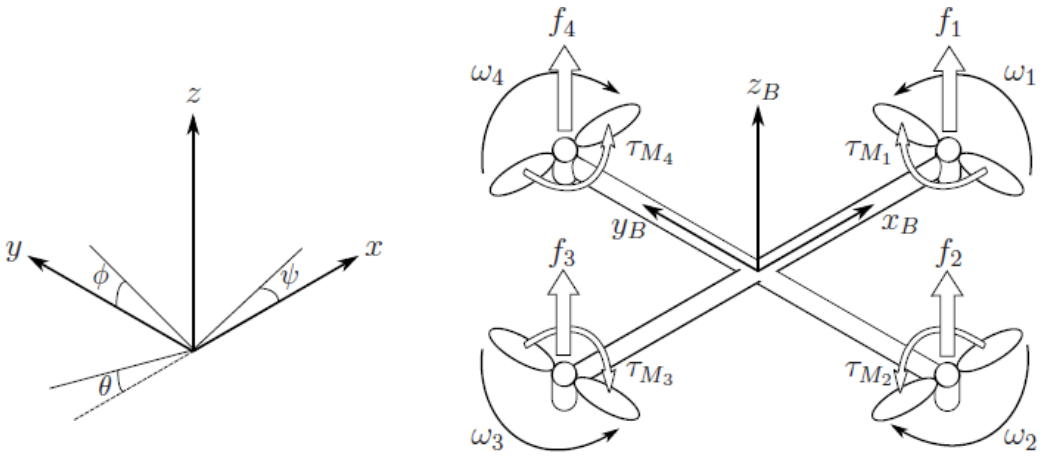


Figure 2.5 The inertial and body frames of the quadcopter

Notation	Definition
T	Input thrust force
$\tau_\phi, \tau_\psi, \tau_\theta$	Input torques along roll, pitch and yaw
$x,\ y,\ z$	Coordinates of quadcopter in XYZ Cartesian space
ϕ, ψ, θ	Roll, pitch and yaw angles measured w.r.t inertial frame
v_x, v_y, v_z	Linear velocities in XYZ space
p, q, r	Angular velocities in body frame
m	Mass of the quadcopter
ω_i	angular velocity of i^{th} propeller
g	Acceleration due to gravity
I_x, I_y, I_z	Inertia of the quadopter along the X, Y, and Z axes
I_r	Inertia of the quadopter along the axis of the propeller
ω_t	total angular velocity of the rotors $(\omega_1 + \omega_2 + \omega_3 + \omega_4)$

Table 2.2 Parameters and variables of quadcopter

We take the state vector as the position coordinates, body frame angles (phi – roll, psi – pitch, theta – yaw) and the linear and angular velocities of the quadcopter. The state vector then, is given by

$$\xi = \begin{bmatrix} x \\ v_x \\ y \\ v_y \\ z \\ v_z \\ \phi \\ p \\ \theta \\ q \\ \psi \\ r \end{bmatrix}$$

To derive the nonlinear dynamic equation of the state vector, we need to write the derivative of the state vector defined above as the function of state variables and the input as discussed in eqn (1.6).

The linear velocities are the derivatives of their corresponding positions. Hence,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \qquad (2.15)$$

Neglecting aerodynamic drag effects, the linear accelerations are affected only by the gravity and the corresponding projections of the input thrust.

$$\begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{v_z} \end{bmatrix} = g \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} + \frac{T}{m} \begin{bmatrix} cos\psi sin\theta cos\phi + sin\psi sin\phi \\ sin\psi sin\theta cos\phi - cos\psi sin\phi \\ cos\theta cos\phi \end{bmatrix} \qquad (2.16)$$

Angular acceleration with respect to the body frame can be calculated by summing the corresponding projections of the angular acceleration along inertial frame. [37] i.e,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -sin\theta \\ 0 & cos\phi & cos\theta sin\phi \\ 0 & -sin\phi & cos\theta cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \qquad (2.17)$$

Conversely,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & \frac{sin\phi}{cos\theta} & \frac{cos\phi}{cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad (2.18)$$

Now with this last transformation we can change from the body axis rotational rates to Euler angle rates. The Euler angle rates can be integrated with respect to time to yield the Euler angles, thus specifying the quadcopter's orientation.

In the body frame, the sum of the angular acceleration of the inertia, the centripetal forces and the gyroscopic forces is equal to the external torque.

$$\mathbf{I}\alpha + \mathbf{\Omega} \times (\mathbf{I}\mathbf{\Omega}) + \mathbf{\Gamma} = \tau \qquad (2.19)$$

Where,

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}, \alpha = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}, \mathbf{\Omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \mathbf{\Gamma} = I_r \begin{bmatrix} q \\ -p \\ 0 \end{bmatrix} \omega_t$$

So eqn (2.19) could be expanded into,

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_y \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_y \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + I_r \begin{bmatrix} q \\ -p \\ 0 \end{bmatrix} \omega_t = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \qquad (2.20)$$

Note that the X mark represents the cross product of angular velocity and the angular momentum of the quadcopter to give the centrifugal forces.

Solving the equations for the angular accelerations, we get

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{(I_y - I_z)}{I_x} qr \\ \frac{(I_z - I_x)}{I_y} pr \\ \frac{(I_x - I_y)}{I_z} pq \end{bmatrix} + \begin{bmatrix} \frac{q}{I_x} \\ \frac{p}{I_y} \\ 0 \end{bmatrix} I_r \omega_t + \begin{bmatrix} \frac{\tau_\phi}{I_x} \\ \frac{\tau_\theta}{I_y} \\ \frac{\tau_\psi}{I_z} \end{bmatrix} \qquad (2.21)$$

From eqns (2.15), (2.16), (2.18) and (2.21) we get the dynamic equation of the quadcopter

$$\dot{\xi} = f(\xi) + g_1(\xi)u_1 + g_2(\xi)u_2 + g_3(\xi)u_3 + g_4(\xi)u_4 \qquad (2.22)$$

Where,

$$f(\xi) = \begin{bmatrix} v_x \\ 0 \\ v_y \\ 0 \\ v_z \\ -g \\ p + sin\phi tan\theta q + cos\phi tan\theta r \\ \frac{I_r}{I_x} q\omega_t + \frac{I_y - I_z}{I_x} qr \\ cos\phi q \ - sin\phi r \\ \frac{I_r}{I_y} p\omega_t + \frac{I_z - I_x}{I_y} pr \\ sin\phi sec\theta q + cos\phi sec\theta r \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix}$$

$$g_1(\xi) = \begin{bmatrix} 0 \\ g_{11} \\ 0 \\ g_{12} \\ 0 \\ g_{13} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, g_2(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_x} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, g_3(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_y} \\ 0 \\ 0 \end{bmatrix}, g_4(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_z} \end{bmatrix}$$

$$g_{11} = \frac{1}{m}(cos\psi sin\theta cos\phi + sin\psi sin\phi)$$

$$g_{12} = \frac{1}{m}(sin\psi sin\theta cos\phi - cos\psi sin\phi)$$

$$g_{13} = \frac{1}{m}(cos\phi cos\theta)$$

$$u_1 = T; u_2 = \tau_\psi; u_3 = \tau_\theta; u_4 = \tau_\phi$$

CHAPTER 3

METHODOLOGY

3.1 PROPOSED CONTROL LAW

3.1.1 Class of unstable system

The aim of the project is to design a controller for the nonlinear affine in control system defined in eqn (1.6). Furthermore, the system for which the controller is designed for, is unstable around its equilibrium point x_e. The control objective is to asymptotically stabilize the system around its equilibrium point.

Let us assume that the system in eqn (1.6) could be written in the following regular form

$$\begin{aligned} \dot{\mathbf{z_c}} &= f_1(\mathbf{z_c}, \mathbf{z_2}) \\ \dot{\mathbf{z_2}} &= f_2(\mathbf{z_c}, \mathbf{z_2}) + \sum_{i=1}^{m} g_i(\mathbf{z_c}, \mathbf{z_2}) u_i \end{aligned} \qquad (3.1)$$

By using an invertible coordinate transformation $\mathbf{z} = T\boldsymbol{x}$, where $\mathbf{z} = (\mathbf{z_c}, \mathbf{z_2})^T$.

If $\mathbf{z}$ is represented as $\mathbf{z} = (\mathbf{z_{c1}}, \mathbf{z_{c2}}, \mathbf{z_2})$, then the equilibrium point $\mathbf{z_e} = (*, 0, 0)$.

The zero dynamics of the system in eqn (3.1) with the following output y is stable.

$$y = \mathbf{z_{c2}}$$

If $\mathbf{z_{c2}}$ dynamics is written in the following form, then

$$\ddot{\mathbf{z}}_{c2} = l(\mathbf{z}) + \sum_{i=1}^{q} h_i(\mathbf{z}) u_i \qquad (3.2)$$

Where q < m, the system in eqn (3.2) is controllable

In this system,

$\mathbf{z_2}$ represents the vector of state variables that represent the velocity or the derivatives of the variables in $\mathbf{z_c}$.

$\mathbf{z_{c1}}$ represents the set of variables that are stable during the zero dynamics analysis, i.e. when the input is zero, there is no change in the values of $\mathbf{z_{c1}}$ regardless of the initial value, whereas when the input is zero, the dynamics of the vector $\mathbf{z_{c2}}$ depends on its initial value.

$\mathbf{z_{c2}}$ will be stable during zero input only if its initial value is zero.

In other words, the system will be in equilibrium for all values of $\mathbf{z_{c1}}$, given $\mathbf{z_{c2}}$ and $\mathbf{z_2}$ are zero.

3.1.2 Controller Design

Step 1: Design a controller such that $\mathbf{z_{c2}}$, is asymptotically stable around the equilibrium point $\mathbf{x_e}$.

Step 2: Write the dynamics of zc1 in terms of z and the designed controller in the following form

$$\ddot{\mathbf{z}}_{c1} = m(\mathbf{z}) + \sum_{i=q+1}^{m} n_i(\mathbf{z})u_i + o(\mathbf{z}_{c2}) \qquad (3.3)$$

Step 3: Design the trajectory $\mathbf{z_{c1}}$ should take.

Step 4: Calculate the input required for the trajectory of $\mathbf{z_{c1}}$ from eqn (3.3).

Step 5: Design a controller for zc2, such that the torque derived in Step 4 is required to stabilize it asymptotically according to eqn (3.2).

3.2 MOBILE INVERTED PENDULUM

In this subsection the proposed control algorithm is implemented to the MIP. Initially, the system is checked if it satisfies the conditions of the class of unstable nonlinear systems described earlier in Section 1.1.4. Then the zero dynamics of the system is analyzed to distinguish the stable and unstable state variables. Finally, the proposed control algorithm is implemented for the MIP.

3.2.1 Class of the system

The nonlinear affine in control model of the MIP as derived in eqn (2.14) is given by

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g_1(\mathbf{x})\tau_r + g_2(\mathbf{x})\tau_l \qquad (2.14)$$

Where,

$$f(\mathbf{x}) = \begin{bmatrix} vcos\theta \\ vsin\theta \\ \dot{\theta} \\ \dot{\alpha} \\ f_5(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_6(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_7(\alpha, \dot{\alpha}, \dot{\theta}) \end{bmatrix}; g_1(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{g_5(\alpha)}{RT_1(\alpha)} \\ \frac{g_6(\alpha)}{RT_1(\alpha)} \\ \frac{b}{RT_1(\alpha)} \end{bmatrix}; g_1(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{g_5(\alpha)}{RT_1(\alpha)} \\ \frac{g_6(\alpha)}{RT_1(\alpha)} \\ -\frac{b}{RT_1(\alpha)} \end{bmatrix}$$

The MIP is a system with 4 degrees of freedom: the coordinates of the position of the MIP (given by x and y), the orientation of the MIP (given by theta), and the angle of the inverted pendulum (given by alpha). However, the system could be controlled by two input torques to the wheels. Hence the system is underactuated.

Moreover, the MIP system is subjected to nonholonomic constraints due to no-slip conditions imposed on the wheels.

The mobile inverted pendulum thus, satisfies the properties of the class of the unstable nonlinear systems considered for this algorithm.

3.2.2 Zero Dynamics

Zero dynamics of a system refers to the response of the system to different initial states when the input is zero. The proposed control algorithm works for system with unstable equilibrium points having stable and unstable state variables at these equilibrium points. If the unstable equilibrium points could be distinguished from the state vector, then a suitable controller could be design to stabilize these unstable state variables. The other states could be controlled by the control method proposed in this thesis. The velocity components are considered zero throughout the analysis

Therefore, $(v, \dot{\alpha}, \dot{\theta} = 0)$

Hence, for the MIP system, we have four initial condition cases.

$$\mathbf{x}_1 = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 0 \\ y_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 0 \\ 0 \\ \theta_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha_0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Substituting the first three initial conditions in equation (2.14), the dynamics of the system is zero.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\alpha} \\ \ddot{\alpha} \\ v \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Hence the system is stable for all initial values of x, y, and theta. Hence these are stable state variables. However, on substituting the fourth initial condition case in equation (2.14) the dynamics of the system is not zero.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\alpha} \\ \ddot{\alpha} \\ v \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \dfrac{b_5 b_3 g sin\alpha_0}{b_2 b_3 - b_5^2 cos\alpha_0} \\ \dfrac{2b_2 b_5 \dot{\alpha}^2 sin\alpha_0 - b_5^2 g sin2\alpha_0}{2(b_2 b_3 - b_5^2 cos\alpha_0)} \\ 0 \end{bmatrix}$$

Here, if the initial value of the pendulum is nonzero, then the system falls out of the equilibrium point and becomes unstable. Hence the pendulum angle (alpha) is an unstable state variable. The simulation of zero dynamics analysis of the MIP is shown in

Section 4. Table 3.1 shows the non-velocity state parameters of the MIP and their stability around an equilibrium point.

State	Stability
x	Stable
y	Stable
θ	Stable
α	Unstable

Table 3.1 State variables and their stability of the MIP

3.2.3 Controller Design

Step 1:

$$\dot{\mathbf{z}}_c = f_1(\mathbf{z}_c, \mathbf{z}_2)$$
$$\dot{\mathbf{z}}_2 = f_2(\mathbf{z}_c, \mathbf{z}_2) + \sum_{i=1}^{m} g_i(\mathbf{z}_c, \mathbf{z}_2)u_i$$

Where, $\mathbf{z_2}$ represents the velocity components and $\mathbf{z_{c1}}$ represents the stable state variables and $\mathbf{z_{c2}}$ represents the unstable state variables

$$\mathbf{z}_c = (\mathbf{z}_{c1}, \mathbf{z}_{c2}) = \begin{bmatrix} x \\ y \\ \theta \\ \alpha \end{bmatrix} ; \mathbf{z}_{c1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} ; \mathbf{z}_{c2} = \alpha$$

$$\mathbf{z}_2 = \begin{bmatrix} \dot{\alpha} \\ v \\ \dot{\theta} \end{bmatrix}$$

The state functions f_1 and f_2 and the input functions g_1 and g_2 are given by

$$f_1(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} vcos\theta \\ vsin\theta \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix}$$

$$f_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} f_5(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_6(\alpha, \dot{\alpha}, \dot{\theta}) \\ f_7(\alpha, \dot{\alpha}, \dot{\theta}) \end{bmatrix}$$

$$g_1(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} -\dfrac{g_5(\alpha)}{RT_1(\alpha)} \\ \dfrac{g_6(\alpha)}{RT_1(\alpha)} \\ \dfrac{b}{RT_2(\alpha)} \end{bmatrix}; g_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} -\dfrac{g_5(\alpha)}{RT_1(\alpha)} \\ \dfrac{g_6(\alpha)}{RT_1(\alpha)} \\ -\dfrac{b}{RT_2(\alpha)} \end{bmatrix}$$

Step 2:

If z is represented as $\mathbf{z_e} = (\mathbf{z_{c1}}, \mathbf{z_{c2}}, \mathbf{z_2})$, then the equilibrium point $\mathbf{z_e} = (*,0,0)$. Therefore $f_1(\mathbf{z_c},\mathbf{z_2})$ and $f_2(\mathbf{z_c},\mathbf{z_2})$ dynamics now becomes,

$$f_1(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} 0cos(0) \\ 0sin(0) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$f_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} \dfrac{sin2(0)(b_3 b_1(0)^2 - b_5^2(0)^2) + 2b_5 b_3 g sin(0)}{2(b_2 b_3 - b_5^2 cos^2(0))} \\ \dfrac{sin2(0)(-b_1 b_5(0)^2 cos(0) - b_5^2 g) + 2b_2 b_5(0)^2 sin(0))}{2(b_2 b_3 - b_5^2 cos^2(0))} \\ \dfrac{-2b_1(0)(0) sin2(0)}{b_4 + b_1 sin^2(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 3:

The zero dynamics of the system is stable with the following output *y* is stable.

$$y = \mathbf{z}_{c2} = \alpha$$

Substituting alpha = 0, *f1*($\mathbf{z_c}$,$\mathbf{z_2}$) and *f2*($\mathbf{z_c}$,$\mathbf{z_2}$) dynamics now becomes

$$f_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} \frac{sin2(0)(b_3 b_1 \dot{\theta}^2 - b_5^2 \dot{\alpha}^2) + 2b_5 b_3 g sin(0)}{2(b_2 b_3 - b_5^2 cos^2(0))} \\ \frac{sin2(0)(-b_1 b_5 \dot{\theta}^2 cos(0) - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin(0))}{2(b_2 b_3 - b_5^2 cos^2(0))} \\ \frac{-2b_1 \dot{alpha} \dot{\theta} sin2(0)}{b_4 + b_1 sin^2(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 4:

If $\mathbf{z_{c2}}$ dynamics is written in the following form, then

$$\begin{aligned} \ddot{\mathbf{z}}_{c2} &= l(\mathbf{z}) + \sum_{i=1}^{q} h_i(\mathbf{z}) u_i \\ &= \ddot{\alpha} \\ &= \frac{sin2\alpha(b_3 b_1 \dot{\theta}^2 - b_5^2 \dot{\alpha}) + 2b_5 b_3 g sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)} \\ &+ \begin{bmatrix} -\frac{b_3 R + b_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} & -\frac{b_3 R + b_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \end{aligned} \quad (3.1)$$

Where q<m, the system in eqn (3.1) is controllable

$$\begin{aligned} \ddot{\mathbf{z}}_{c1} &= l(\mathbf{z}) + \sum_{i=q+1}^{m} n_i(\mathbf{z}) u_i + O(\mathbf{z}_{c2}) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} \\ &= \begin{bmatrix} (\frac{sin2\alpha(b_1 b_5 \dot{\theta}^2 cos\alpha - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)})(cos\theta) - v sin\theta\dot{\theta} \\ (\frac{sin2\alpha(b_1 b_5 \dot{\theta}^2 cos\alpha - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)})(sin\theta) + v cos\theta\dot{\theta} \\ \frac{-b_1 \dot{\alpha}\dot{\theta} sin2\alpha}{b_4 + b_1 sin^2\alpha} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} (\frac{sin2\alpha(-b_1 b_5 \dot{\theta}^2 cos\alpha - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)})(cos\theta) - vsin\theta\dot{\theta} \\ (\frac{sin2\alpha(-b_1 b_5 \dot{\theta}^2 cos\alpha - b_5^2 g) + 2b_2 b_5 \dot{\alpha}^2 sin\alpha}{2(b_2 b_3 - b_5^2 cos^2\alpha)})(sin\theta) + vcos\theta\dot{\theta} \\ \frac{-b_1 \dot{\alpha}\dot{\theta} sin2\alpha}{b_4 + b_1 sin^2\alpha} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{b_2 + Rb_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} & \frac{b_2 + Rb_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} \\ \frac{b_2 + Rb_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} & \frac{b_2 + Rb_5 cos\alpha}{R(b_2 b_3 - b_5^2 cos^2\alpha)} \\ \frac{b}{R(b_4 + b_1 sin^2\alpha)} & \frac{-b}{R(b_4 + b_1 sin^2\alpha)} \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \qquad (3.2)$$

Eqn (3.2) defines the dynamics of $\mathbf{z_{c1}}$, where the input torque asymptotically stabilizes $\mathbf{z_{c2}}$ according to (3.1).

A three-step control method is used track the desired trajectory of $\mathbf{z_{c1}}$.

(i) Homing towards the target:

The aim of this homing control is to rotate the MIP to the target position such that the wheel axis is perpendicular to the line connecting the target and the MIP's current position. A PD control method is used to calculate the required torque.

K_p and K_d are positive constants representing proportional and derivative gains respectively.

$$\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} = \begin{bmatrix} K_p(\theta - tan^{-1}\frac{y_d}{x_d}) - K_d(\dot{\theta}) \\ -K_p(\theta - tan^{-1}\frac{y_d}{x_d}) + K_d(\dot{\theta}) \end{bmatrix}$$

(ii) Pedalling action of the MIP:

The pendulum of the MIP can be used as a pedal to move the pendulum forward or backwards. Since the pedalling action allows the MIP to move only in a straight line, it is therefore, important that the heading (theta) of the MIP is along the straight line joining the destination and the current position. This is where the proposed control algorithm is used.

According to the control algorithm, a controller is designed to asymptotically stabilize the unstable state variable, the pendulum angle in this case. A simple PD controller is used to bring the system into stabilization.

$$\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} = \begin{bmatrix} K_p(\alpha) + K_d(\dot{\alpha}) + k_d(v) \\ K_p(\alpha) + K_d(\dot{\alpha}) + k_d(v) \end{bmatrix}$$

Here, K_p is a positive proportional gain constant and K_d is a negative derivative gain constant. k_d is a velocity derivative constant, needed to bring the system to a halt. If k_d is not present, then the MIP would keep on moving past the destination with the same velocity due to its inertia.

Note that if the pendulum angle is positive, the MIP moves forward while stabilizing the pendulum and the MIP moves backward to stabilize it if the pendulum angle is negative. Conversely, to move the system forward or backward the pendulum angle could be adjusted as positive or negative correspondingly. For simplicity, a proportional controller with saturation is used to design the dynamics of $\mathbf{z_{c2}}$, pendulum angle in this case, based on the trajectory of $\mathbf{z_{c1}}$, x, y states in this case.

$$\mathbf{z}_{c2} = \alpha = \begin{cases} k_p e; & |e| < e_t \\ e_t; & e > e_t \\ -e_t; & e < -e_t \end{cases}$$

Where, k_p is the positive proportional gain constant, et is the error threshold within which the proportional gain works, e is the error given by,

$$e = \sqrt{x_d^2 + y_d^2} - \sqrt{x^2 + y^2}$$

It is to be noted that this proportional controller a simple controller designed for simulation purposes to prove the validity of the control algorithm. Hence, this algorithm will effectively track the desired position. However, a more sophisticated controller could be designed to track the desired trajectory entirely.

(iii) Correcting the heading of MIP

Once the MIP reaches its desired position, it still faces the same direction that it travelled through. The MIP could be rotated to reach any desired heading (theta) once it has reached its desired coordinates. The control is similar to the first technique as both methods control the heading of the MIP, which is given by,

$$\begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} = \begin{bmatrix} K_p(\theta - \theta_d) - K_d(\dot{\theta}) \\ -K_p(\theta - \theta_d) + K_d(\dot{\theta}) \end{bmatrix}$$

The controller could be proved to stabilize the system asymptotically, the proof is out of scope of this thesis. The system and the controller are simulated on the simulation platform SIMULINK and the results are shown in Section 4.

3.3 QUADCOPTER

In this subsection the proposed control algorithm is implemented to the quadcopter. Initially, the system is checked if it satisfies the conditions of the class of unstable nonlinear systems described earlier in Section 1.1.4. Then the zero dynamics of the system is analyzed to distinguish the stable and unstable state variables. Finally, the proposed control algorithm is implemented for the quadcopter.

3.3.1 Class of the system

The nonlinear affine in control model of the quadcopter as derived in eqn (2.22) is given by

$$\dot{\xi} = f(\xi) + g_1(\xi)u_1 + g_2(\xi)u_2 + g_3(\xi)u_3 + g_4(\xi)u_4 \qquad (2.22)$$

Where,

$$f(\xi) = \begin{bmatrix} v_x \\ 0 \\ v_y \\ 0 \\ v_z \\ -g \\ p + sin\phi tan\theta q + cos\phi tan\theta r \\ \frac{I_r}{I_x} q\omega_t + \frac{I_y - I_z}{I_x} qr \\ cos\phi q - sin\phi r \\ \frac{I_r}{I_y} p\omega_t + \frac{I_z - I_x}{I_y} pr \\ sin\phi sec\theta q + cos\phi sec\theta r \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix}$$

$$g_1(\xi) = \begin{bmatrix} 0 \\ g_{11} \\ 0 \\ g_{12} \\ 0 \\ g_{13} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, g_2(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_x} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, g_3(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_y} \\ 0 \\ 0 \end{bmatrix}, g_4(\xi) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_z} \end{bmatrix}$$

The quadcopter is a system with 6 degrees of freedom: the coordinates of the position of the quadcopter (given by x, y, and z), the orientation of the quadcopter (given by phi, theta and psi). However, the system could be controlled by four input torques generated by four propellers. Thus, the system is underactuated.

3.3.2 Zero Dynamics

The state vector contains twelve state variables, six of them are velocity components, linear velocity and angular velocity. The zero dynamics analysis hence could be tested for six initial condition cases given by,

$$\xi_1 = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \xi_2 = \begin{bmatrix} 0 \\ 0 \\ y_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \xi_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ z_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \xi_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \phi_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \xi_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \theta_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \xi_1 = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \psi_0 \\ 0 \\ 0 \end{bmatrix}$$

Substituting the initial conditions of x_0, y_0 and psi_0 (yaw) in eqn (2.22), the dynamics of the system is zero. Hence the system is stable for all values of x and y. The quadcopter is not stable for every value of z without an external thrust force input. But, a suitable thrust force controller could directly control the altitude (z) of the quadcopter. Assuming that a suitable thrust force is always applied to stabilize the altitude of the quadcopter, z is considered stable in this thesis.

Considering thrust force sufficient to maintain the quadcopter at z being the only input, i.e.

$$T = mgsec\theta sec\phi$$

The nonlinear system model then becomes,

$$\xi = f(\xi) + g_1(\xi)T = \begin{bmatrix} v_x \\ (cos\psi tan\theta + sin\psi sec\theta tan\phi)g \\ v_y \\ (sin\psi tan\theta - cos\psi sec\theta tan\phi)g \\ v_z \\ 0 \\ p + sin\phi tan\theta q + cos\phi tan\theta r \\ \frac{I_r}{I_x} q\omega_t + \frac{I_y - I_z}{I_x} qr \\ cos\phi q - sin\phi r \\ \frac{I_r}{I_y} p\omega_t + \frac{I_z - I_x}{I_y} pr \\ sin\phi sec\theta q + cos\phi sec\theta r \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix} \qquad (3.3)$$

Substituting a non-zero initial condition for phi and theta, the two equations are shown in equations (3.4) and (3.5) respectively

$$\xi = f(\xi_4)+g_1(\xi_4)T = \begin{bmatrix} 0 \\ (cos(0)tan(0) + sin(0)sec(0)tan\phi_0)g \\ 0 \\ (sin(0)tan(0) - cos(0)sec(0)tan\phi_0)g \\ 0 \\ 0 \\ (0) + sin\phi_0 tan(0)(0) + cos\phi_0 tan(0)(0) \\ \frac{I_r}{I_x}(0)\omega_t + \frac{I_y - I_z}{I_x}(0)(0) \\ cos\phi_0(0) \ -sin\phi_0(0) \\ \frac{I_r}{I_y}(0)\omega_t + \frac{I_z - I_x}{I_y}(0)(0) \\ sin\phi_0 sec(0)(0) + cos\phi_0 sec(0)(0) \\ \frac{I_x - I_y}{I_z}(0)(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -tan\phi_0 g \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.4)$$

$$\xi = f(\xi_4)+g_1(\xi_4)T = \begin{bmatrix} 0 \\ (cos(0)tan\theta_0 + sin(0)sec\theta_0 tan(0))g \\ 0 \\ (sin(0)tan\theta_0 - cos(0)sec\theta_0 tan(0))g \\ 0 \\ 0 \\ (0) + sin(0)tan\theta_0(0) + cos(0)tan\theta(0) \\ \frac{I_r}{I_x}(0)\omega_t + \frac{I_y - I_z}{I_x}(0)(0) \\ cos\phi_0(0) \ -sin\phi_0(0) \\ \frac{I_r}{I_y}(0)\omega_t + \frac{I_z - I_x}{I_y}(0)(0) \\ sin(0)sec\theta_0(0) + cos(0)sec\theta_0(0) \\ \frac{I_x - I_y}{I_z}(0)(0) \end{bmatrix} = \begin{bmatrix} 0 \\ tan\theta_0 g \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.5)$$

Hence, the roll and pitch angles are unstable state variables. Table 3.2 shows the non-velocity state parameters of the quadcopter and their stability around an equilibrium point.

Parameter	Stability
x	Stable
y	Stable
z	Stable
ϕ	Unstable
θ	Unstable
ψ	Unstable

Table 3.2 State variables and their stability of the quadcopter

3.3.3 Controller Design

Step1:

$$\dot{\mathbf{z}}_c = f_1(\mathbf{z}_c, \mathbf{z}_2)$$
$$\dot{\mathbf{z}}_2 = f_2(\mathbf{z}_c, \mathbf{z}_2) + \sum_{i=1}^{m} g_i(\mathbf{z}_c, \mathbf{z}_2)u_i$$

Where,

$$mathbf z_c = (\mathbf{z}_{c1}, \mathbf{z}_{c2}); \mathbf{z}_{c1} = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}; \mathbf{z}_{c2} = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$$

$$\mathbf{z}_2 = \begin{bmatrix} v_x \\ v_y \\ v_z \\ p \\ q \\ r \end{bmatrix}$$

The state functions f_1 and f_2 and the input functions g_1 and g_2 are given by

$$f_1(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ p + sin\phi tan\theta q + cos\phi tan\theta r \\ cos\phi q - sin\phi r \\ sin\phi sec\theta q + cos\phi sec\theta r \end{bmatrix}$$

$$f_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} -K_{vx} + \frac{T}{m}(cos\psi sin\theta cos\phi + sin\psi sin\phi) \\ -K_{vy} + \frac{T}{m}(sin\psi sin\theta cos\phi - cos\psi sin\phi) \\ -K_{vz} - g + \frac{T}{m}(cos\theta cos\phi) \\ \frac{I_r}{I_x} q\Omega_r + \frac{I_y - I_z}{I_x} qr \\ \frac{I_r}{I_y} p\Omega_r + \frac{I_z - I_x}{I_y} pr \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix}$$

$$\sum_{i=1}^{m} g_i(\mathbf{z}_c, \mathbf{z}_2) u_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{I_x} \\ 0 \\ 0 \end{bmatrix} \tau_\phi + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_y} \\ 0 \end{bmatrix} \tau_\theta + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{I_z} \end{bmatrix} \tau_\psi$$

Step 2:

If **z** is represented as **z** = (**z**$_{c1}$, **z**$_{c2}$, **z**$_2$), then the equilibrium point **z**$_e$ = (*,0,0). Substituting the variables **z**$_{c2}$ and **z**$_2$ = 0, f_1(**z**$_c$,**z**$_2$) and f_2(**z**$_c$,**z**$_2$) dynamics now become

$$f(\xi) = \begin{bmatrix} 0 \\ 0 \\ (0) + sin(0)tan(0)(0) + cos(0)tan(0)(0) \\ cos(0)(0) - sin(0)(0) \\ sin(0)sec(0)(0) + cos(0)sec(0)(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$f(\xi) = \begin{bmatrix} -K_{vx} + \frac{T}{m}(cos\psi sin(0)cos(0) + sin\psi sin(0)) \\ -K_{vy} + \frac{T}{m}(sin\psi sin(0)cos(0) - cos\psi sin(0 \\ -K_{vz} - g + \frac{T}{m}(cos(0)cos(0)) \\ \frac{I_r}{I_x}(0)\Omega_r + \frac{I_y - I_z}{I_x}(0)(0) \\ \frac{I_r}{I_y}(0)\Omega_r + \frac{I_z - I_x}{I_y}(0)(0) \\ \frac{I_x - I_y}{I_z}(0)(0) \end{bmatrix} = \begin{bmatrix} -K_{vx} \\ -K_{vy} \\ -K_{vz} - g + \frac{T}{m} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 3:

The zero dynamics of the system with the following output y is stable

$$\mathbf{y} = \mathbf{z}_{c2} = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$$

Substituting y = 0, $f_1(\mathbf{z_c},\mathbf{z_2})$ and $f_2(\mathbf{z_c},\mathbf{z_2})$ dynamics now become

$$f_1(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ p + sin(0)tan(0)1 + cos(0)tan(0)r \\ cos(0)q - sin(0)r \\ sin(0)sec(0)q + cos(0)sec(0)r \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ p \\ q \\ r \end{bmatrix}$$

$$f_2(\mathbf{z}_c, \mathbf{z}_2) = \begin{bmatrix} -K_{vx} + \frac{T}{m}(cos\psi sin(0)cos(0) + sin\psi sin(0)) \\ -K_{vy} + \frac{T}{m}(sin\psi sin(0)cos(0) + cos\psi sin(0)) \\ -K_{vz} + \frac{T}{m}(cos(0)cos(0)) \\ \frac{I_r}{I_x}q\Omega_r + \frac{I_y - I_z}{I_x}qr \\ \frac{I_r}{I_y}p\Omega_r + \frac{I_z - I_x}{I_y}pr \\ \frac{I_x - I_y}{Iz}pq \end{bmatrix} = \begin{bmatrix} -K_{vx} \\ -K_{vy} \\ -K_{vz} - g + \frac{T}{m} \\ \frac{I_r}{I_x}q\Omega_r + \frac{I_y - I_z}{I_x}qr \\ \frac{I_r}{I_y}p\Omega_r + \frac{I_z - I_x}{I_y}pr \\ \frac{I_x - I_y}{Iz}pq \end{bmatrix}$$

Step 4:

If $\mathbf{z_{c2}}$ dynamics is written in the following form, then

$$\begin{aligned} \ddot{\mathbf{z}}_{c2} &= l(\mathbf{z}) + \sum_{i=1}^{q} h_i(\mathbf{z})u_i \\ &= \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} \\ &= \begin{bmatrix} \dot{p} + qtan\theta cos\phi\dot{\phi} + qsin\theta(tan\theta^2 + 1)\dot{\theta} + \dot{q}sin\phi tan\theta \\ -sin\phi tan\theta r\dot{\phi} + cos\theta(tan\theta^2 + 1)\dot{\theta}r + cos\phi tan\theta\dot{r} \\ \\ -sin\phi p\dot{h}iq - cos\phi\dot{\phi}r + cos\phi\dot{q} - sin\phi\dot{r} \end{bmatrix} \end{aligned}$$

$$\ddot{\mathbf{z}}_{c2} = \begin{bmatrix} (p + sin\phi tan\theta q + cos\phi tan\theta r) + qtan\theta cos\phi\dot{\phi} + qsin\theta(tan\theta^2 + 1)\dot{\theta} \\ +(cos\phi q - sin\phi r)sin\phi tan\theta - sin\phi tan\theta r\dot{\phi} + cos\theta(tan\theta^2 + 1)\dot{\theta}r \\ +cos\phi tan\theta(sin\phi sec\theta q) \\ \\ -sin\phi\dot{\phi}q - cos\phi\dot{p}hir + cos\phi(cos\phi q - sin\phi r) - sin\phi(sin\phi sec\theta q + cos\phi sec\theta r) \end{bmatrix} + \begin{bmatrix} \frac{1}{I_x} & \frac{1}{I_y} & \frac{1}{I_z} \\ 0 & \frac{1}{I_y} & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (3.6)$$

Here, q<m. Hence, the system is controllable

$$\ddot{\mathbf{z}}_{c1} = m(\mathbf{z}) + \sum_{i=q+1}^{m} n_i(\mathbf{z})u_i + O(\mathbf{z}_{c2}) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\psi} \end{bmatrix}$$

$$\ddot{\mathbf{z}}_{c1} = \begin{bmatrix} -K_{vx} + \frac{T}{m}(cos\psi sin\theta cos\phi + sin\psi sin\phi) \\ -K_{vy} + \frac{T}{m}(sin\psi sin\theta cos\phi + cos\psi sin\phi) \\ -K_{vz} - g + \frac{T}{m}(cos\phi cos\phi) \\ \\ cos\phi\dot{\phi}sec\theta q - sin\phi\dot{p}hisec\theta r + sin\phi sec\theta tan\theta\dot{\theta}q \\ +cos\phi sec\theta tan\theta\dot{\theta}r + \dot{q}sin\phi sec\theta + \dot{r}cos\phi sec\theta \\ +(sin\phi sec\theta q + cos\phi sec\theta r)cos\phi sec\theta \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{I_y} & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \quad (3.7)$$

The controller equation to control the altitude is given as follows

$$T = mgcos\theta\cos\phi + K_p(z_d - z) - K_d(v_z)$$

Where, Kp and Kd are positive proportional and derivative gain constants and the other term compensates the force due to gravity.

The roll or pitch angle can be used as a pedal to move the quadcopter in the y or x axis direction respectively. Since the pedalling action allows the quadcopter to move only in 2 different axes, the quadcopter does not need to rotate to face the direction of the line joining its current position and the target destination. This is where the proposed control algorithm is used.

The controller to asymptotically stabilize $\mathbf{z_{c2}}$ is designed as follows

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \end{bmatrix} = \begin{bmatrix} K_p(\phi) + K_d(\dot{\phi}) + k_d v_y \\ K_p(\theta) + K_d(\dot{\theta}) + k_d v_x \end{bmatrix}$$

A PD controller is designed to stabilize the unstable state variables where K_p is the positive proportional gain constant, K_d is the negative derivative gain constant and k_d is the negative derivative gain constant for the linear velocity.

From the dynamics equation it is clear that if the roll angle (phi) is positive then the quadcopter moves along the positive y direction and if it is negative the quadcopter moves in the negative y direction. Also, if the pitch angle (theta) is positive then the quadcopter moves along the positive x direction. Conversely, to move the quadcopter along the XY axis, given a constant z value, the roll and pitch angles could be controlled suitably so the desired trajectory is achieved. Similar to the MIP, a proportional controller with saturation is used to design the dynamics of $\mathbf{z_{c2}}$, roll and pitch angles in this case, based on the trajectory of $\mathbf{z_{c1}}$,

$$\mathbf{z}_{c2} = \begin{bmatrix} \phi \\ \theta \end{bmatrix} = \begin{bmatrix} \phi = \begin{cases} k_p e_\phi; & |e_\phi| < e_t \\ e_t; & e_\phi > e_t \\ -e_t; & e_\phi < -e_t \end{cases} \\ \\ \theta = \begin{cases} k_p e_\theta; & |e_\theta| < e_t \\ e_t; & e_\theta > e_t \\ -e_t; & e_\theta < -e_t \end{cases} \end{bmatrix}$$

Where, k_p is the positive proportional gain constant, et is the error threshold within which the proportional gain works, e is the error given by,

$$e_\phi = x_d - x;$$
$$e_\theta = y_d - y;$$

3.4 SUMMARY

This section presented the novel control algorithm and the implementation of the control algorithm on two example cases, the mobile inverted pendulum and the quadcopter. For each system, the dynamics equations of the system were revisited, the zero dynamics of the system was analyzed and stable and unstable state parameters were

distinguished. The proposed control algorithm for each of the system was discussed. The stability of the control algorithm could be proved through Lyapunov stability criterion, but it is out of scope of this thesis. The next section deals with the testing and simulation results and the inferences. The systems and the proposed controller are simulated on SIMULINK platform.

CHAPTER 4

SIMULATION AND RESULTS

4.1 OVERVIEW

This chapter presents the simulation software, parameters, simulation results and the inference of each result. The first subsection presents the simulation of the MIP and the second subsection presents the simulation of the quadcopter on SIMULINK. The results of zero dynamics analysis of the MIP is shown and the results are verified with the theoretical derivations. The implementation of the proposed control algorithm and the results are then presented in this section.

4.2 MOBILE INVERTED PENDULUM

4.2.1 Zero Dynamics Analysis

Figure 4.1 shows the simulation model of the mobile inverted pendulum. The zero dynamics analysis involves the study of the dynamics of the system, when there is no input. In every case, all the state variables, except one, has been assigned zero. One state variable is given an initial value, 1 in this simulation.

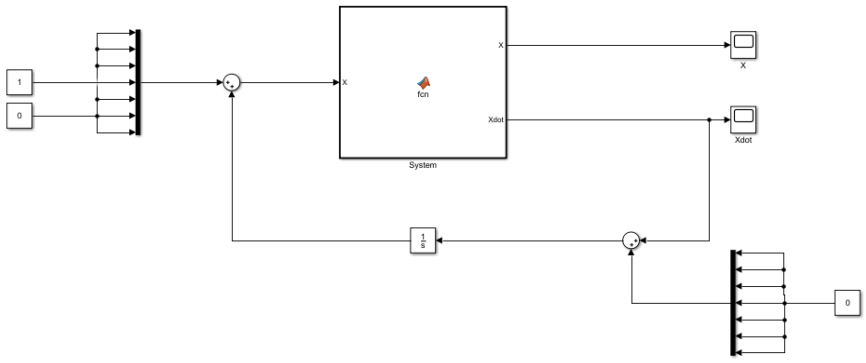


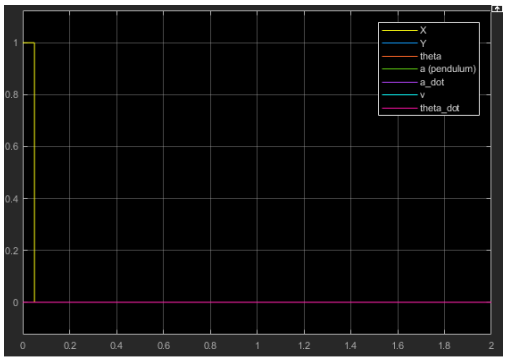
Figure 4.1 Zero Dynamics simulation of MIP on SIMULINK

Table 4.1 describes the different parameters of the MIP and their values considered for the purpose of simulation.

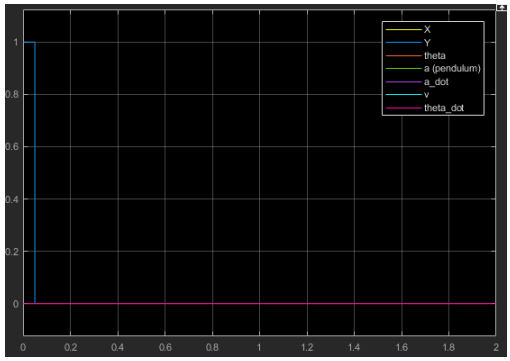
Parameter	Value	Unit
M_b	3.9	kg
M_w	0.375	kg
R	0.1025	m
C_z	0.1	m
b	0.1620	m
g	9.81	ms^{-2}
I_x	0.02	kg m^2
I_y	0.015	kg m^2
I_z	0.01	kg m^2
I_w	0.002	kg m^2
$I_w a$	0.001	kg m^2

Table 4.1 Simulation parameters of MIP

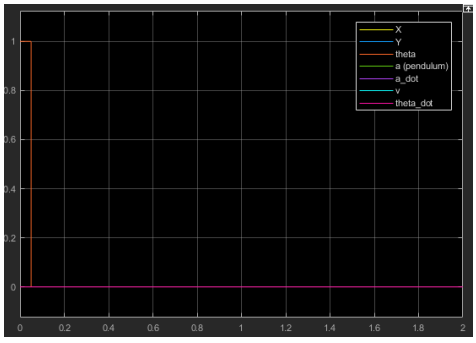
Figures 4.2a, 4.2b and 4.2c represent the zero dynamics of the system for initial values of x. y and theta respectively.



(a) X_initial



(b) Y_initial



(c) theta_initial

Figure 4.2 Zero dynamics of MIP for different state variables

The results show that there is no change in the dynamics of the system for different initial values of x, y and theta. Therefore, the state variables x, y and theta are stable state variables. These state variables could be classified under $\mathbf{z_{c1}}$ of the control algorithm. Figure 4.3 shows the zero dynamics of the MIP when alpha_initial is non zero. The system becomes unstable if the initial value of the pendulum is non zero. This proves that the pendulum angle (alpha) is an unstable state.

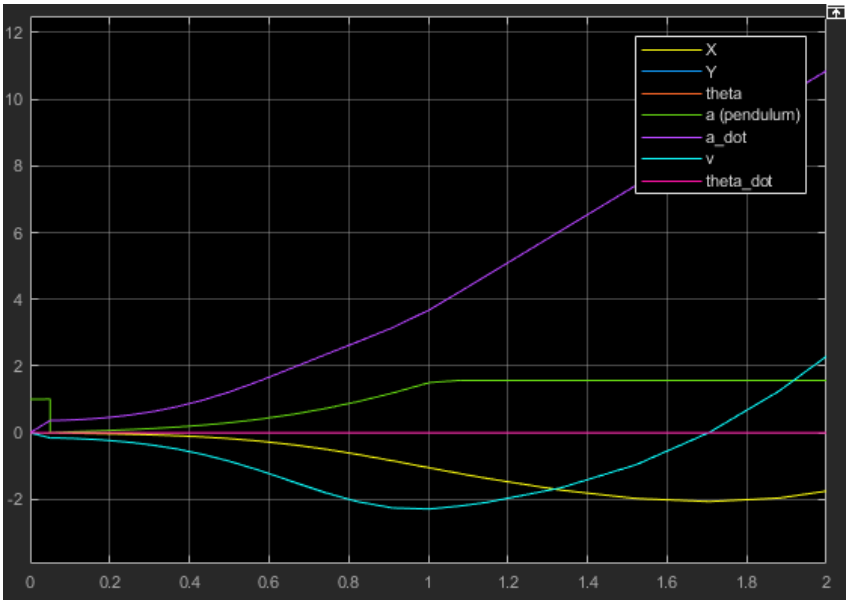


Figure 4.3 Zero Dynamics of MIP when alpha_initial is non zero

The results of the Zero Dynamics analysis of the MIP is shown in the Table 4.2. It is noted that the Table 4.2 is identical to Table 3.1. This proves the validity of the dynamics equation and also proves that the simulated system follows the dynamics of a typical mobile inverted pendulum.

State	Stability
x	Stable
y	Stable
θ	Stable
α	Unstable

Table 4.2 State variables of MIP and their simulation stability

4.2.2 State Tracking

State tracking refers to when a reference state is chosen and the system controller is designed to track the reference state. The control algorithm discussed in the section 3.2 is implemented to track a desired state vector. Figure 4.4 shows the simulation of the system-controller of the MIP.

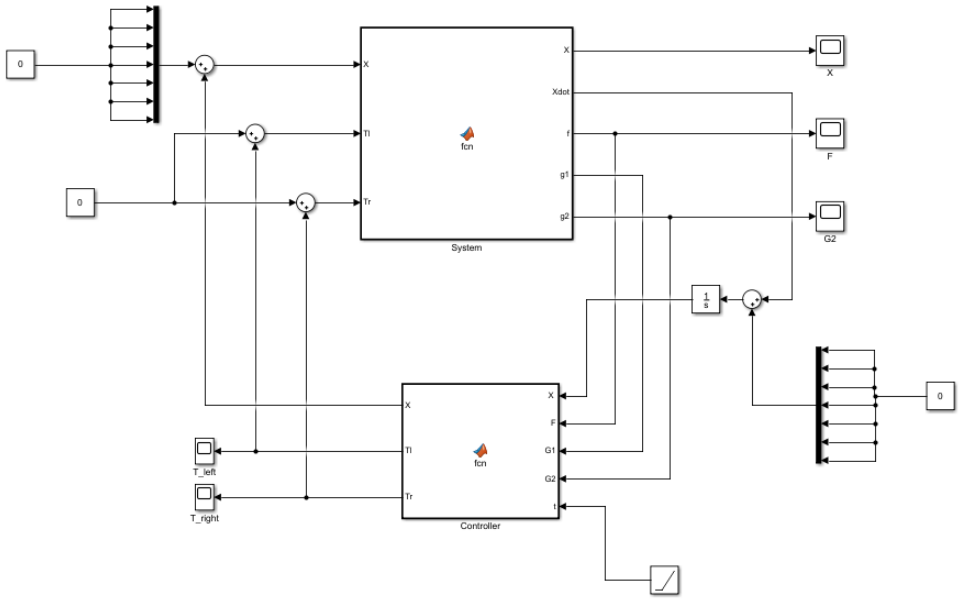


Figure 4.4 System – Controller Simulation of the MIP

Control	Gain	Value
Homing	K_p	5
	K_d	10
Pedalling	K_p	5
	K_d	-1
	k_p	1
	k_d	-5
	e_t	0.001
Heading	K_p	5
	K_d	10

Table 4.3 Gain parameters of the controller of MIP

Table 4.3 shows the PD controller parameters of the controller and their values. Figure 4.5 shows the entire dynamics of the system when the desired target are (x, y,

theta) = (0.2, 0.4 , 0.6). Figure 4.6a shows the trajectory of x, y and theta, figure 4.6b shows the trajectory of the pendulum angle, alpha.

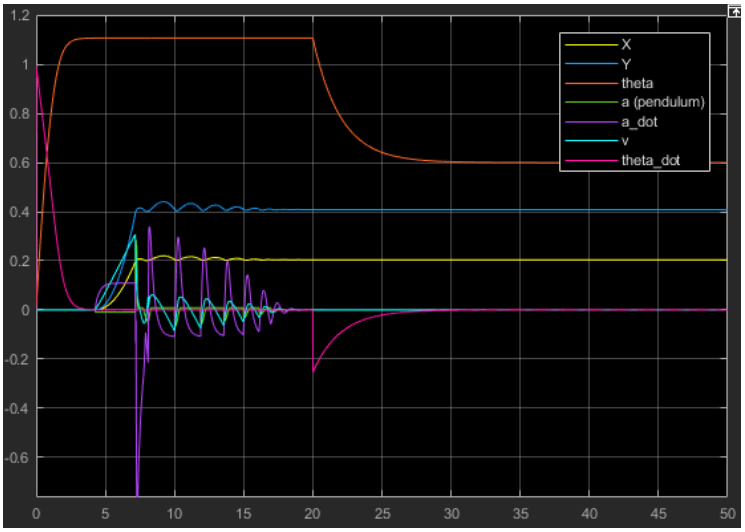
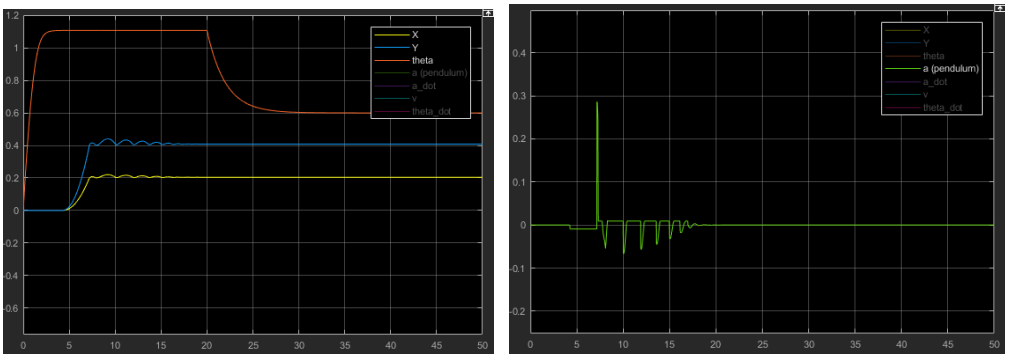


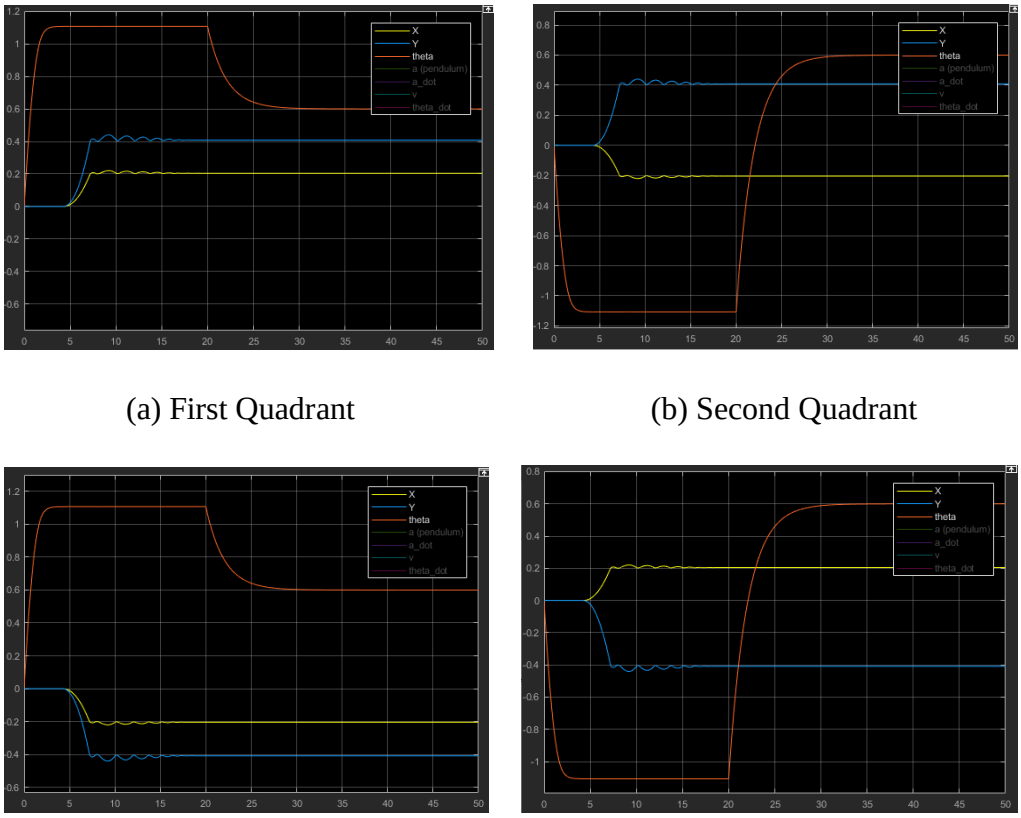
Figure 4.5 System state tracking of MIP



(a) (b)

Figure 4.6 (a)Trajectory of $\mathbf{z_{c1}}$ (x,y, theta) and (b)trajectory of $\mathbf{z_{c2}}$ (alpha)

The figures 4.5 and 4.6 show that the system tracks the desired state and the system always remains stable. This proves the validity of the proposed control law on the MIP. The controller works for all values of x, y and theta regardless of the quadrant of the desired coordinates. Figure 4.7 displays the trajectory of zc1 (x, y, theta) for desired coordinates in each quadrant



(a) First Quadrant (b) Second Quadrant

(c) Third Quadrant (d) Fourth Quadrant

Figure 4.7 Trajectory of MIP for a desired target in each quadrant

The three-step controlling technique could easily be visualized from the results. The heading of the MIP (theta) changes initially, not towards the desired value, but towards the slope of the line connecting the desired point and the system's current location. Once the heading of the system is fixed, the bot moves in a straight line (pedalling). This could be seen from the fact that the x and y coordinates are changing, but theta remains constant. This could also be seen from the values of x and y coordinates. Event when the x and y coordinates are changing, their ratio remains constant. Finally, the system heading reaches the desired value.

4.3 QUADCOPTER

4.3.1 Zero Dynamics Analysis

The zero dynamics of the quadcopter deals with zero input forces, considering a thrust force suitable enough to negate the gravity effects is available. Six different initial cases exist for six degrees of freedom. In each case, one state variable is set as non zero while the other state variables have zero initial value. Figure 4.8 presents the simulation model for zero dynamics analysis of the quadcopter.

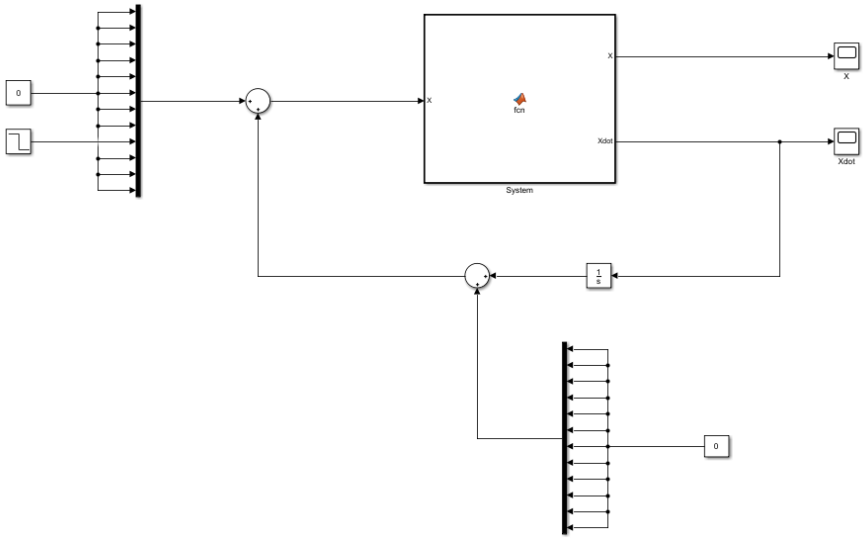


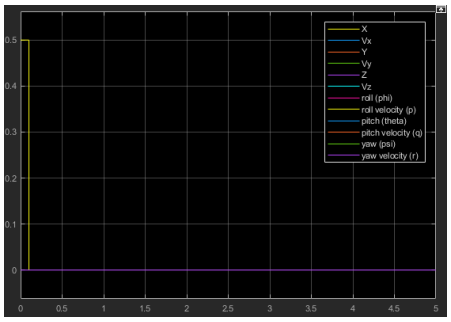
Figure 4.8 Zero Dynamics simulation model of Quadcopter on SIMULINK

Table 4.4 presents the different gain constants discussed during the controller derivation and their values used for the simulation.

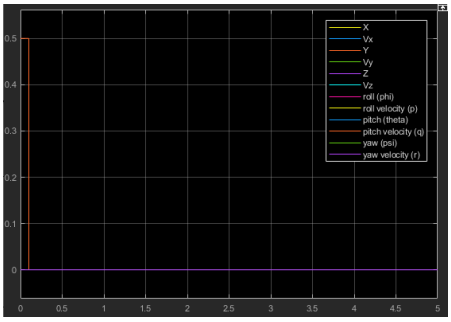
Parameter	Value	Unit
m	0.468	kg
I_x	4.856×10^{-3}	kg m^2
I_y	4.856×10^{-3}	kg m^2
I_z	8.801×10^{-3}	kg m^2
I_r	3.357×10^{-5}	kg m^2
Ω_r	50	rad/s
g	9.81	ms^{-2}

Table 4.4 Simulation parameters of quadcopter

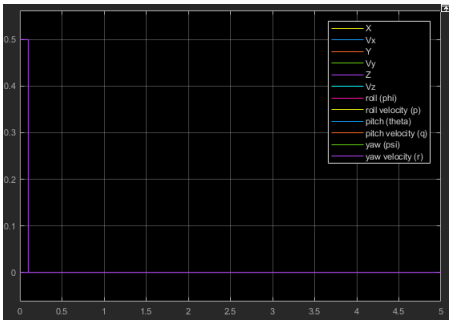
Figures 4.9a, 4.9b, 4.9c and 4.9d represents the zero dynamics of the system for initial values of x. y, z and yaw respectively. It could be seen from the results that there is no change in the dynamics of the system for any initial value of x,y,z or yaw. Hence, this proves that x, y, z and yaw angle (psi) are stable state variables. These state variables could be classified under $\mathbf{z_{c1}}$ of the control algorithm.



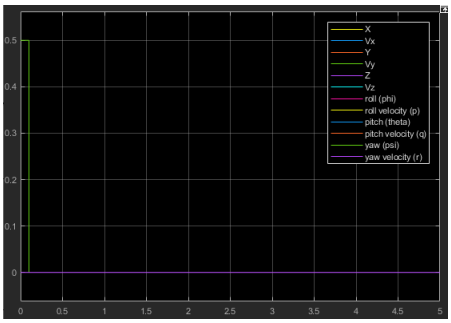
(a) x_initial



(b) y_initial



(c) z_initial



(d) yaw_initial

Figure 4.9 Zero dynamics of quadcopter for different state variables

Figure 4.10 and 4.11 show the zero dynamics of the quadcopter when roll angle (phi) is non zero and when pitch angle (theta) is non zero respectively. It could be seen that the dynamics of the system becomes unstable when the value of phi or theta is non-zero. Hence, these are considered as unstable state variables and they could be classified under $\mathbf{z_{c2}}$ of the control algorithm.

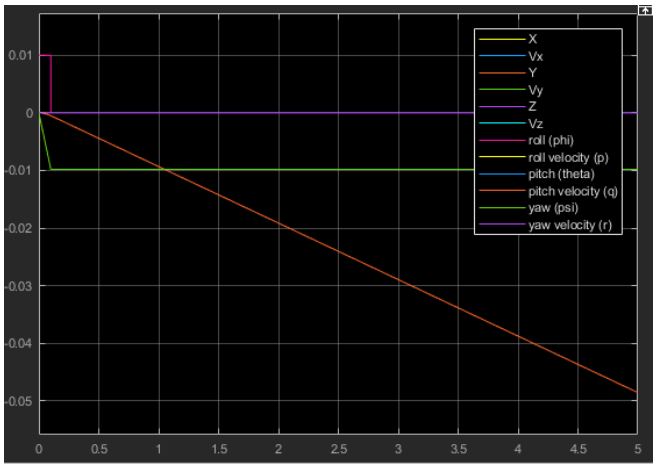


Figure 4.10 Zero dynamics of quadcopter for non zero roll angle (phi)

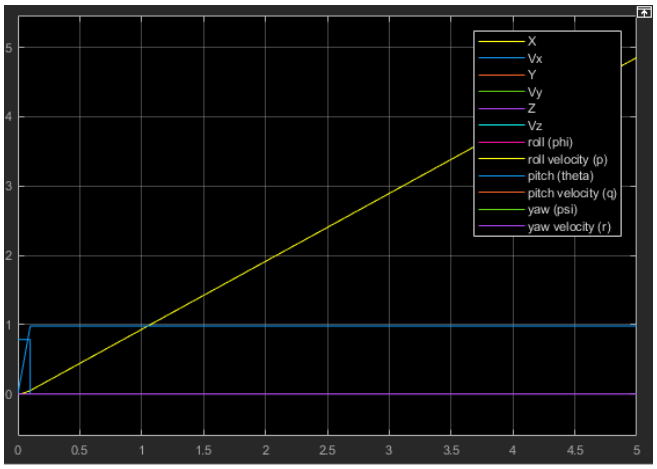


Figure 4.11 Zero dynamics of quadcopter for non zero pitch angle (theta)

In the first case, the initial roll angle is taken to be 0.01 and in the second case, the initial pitch angle is taken as 0.78539 (approximately pi/4). It is interesting to note that the dynamics of the state x with respect to phi_initial and the dynamics of the state y with respect with theta_initial are linear. Furthermore, it could be seen that the slope of the line in Figure 4.10 is approximately –0.01 and the slope of the line in Figure 4.11 is approximately 1. The slope of the trajectory of the state with time, which is linear velocity is similar to the trigonometric tan of the angle of the corresponding state variable. This follows the equations (3.4) and (3.5).

The results of the Zero Dynamics analysis of the quadcopter is shown in the Table 4.5. It is noted that the Table 4.5 is identical to Table 3.2. This proves the validity of the

dynamics equation and also proves that the simulated system follows the dynamics of a typical quadcopter.

Parameter	Stability
x	Stable
y	Stable
z	Stable
ϕ	Unstable
θ	Unstable
ψ	Unstable

Table 4.5 State variables of quadcopter and their simulated stability

4.3.2 State Tracking

A coordinate in the XYZ space is chosen as desired target and the controller is designed to move the system to track the particular state. The control algorithm discussed in Section 3.3 is implemented to track the desired state. Figure 4.12 simulation model of the controller-system of the quadcopter. Table 4.6 shows the gain constants used in the control algorithm and their values.

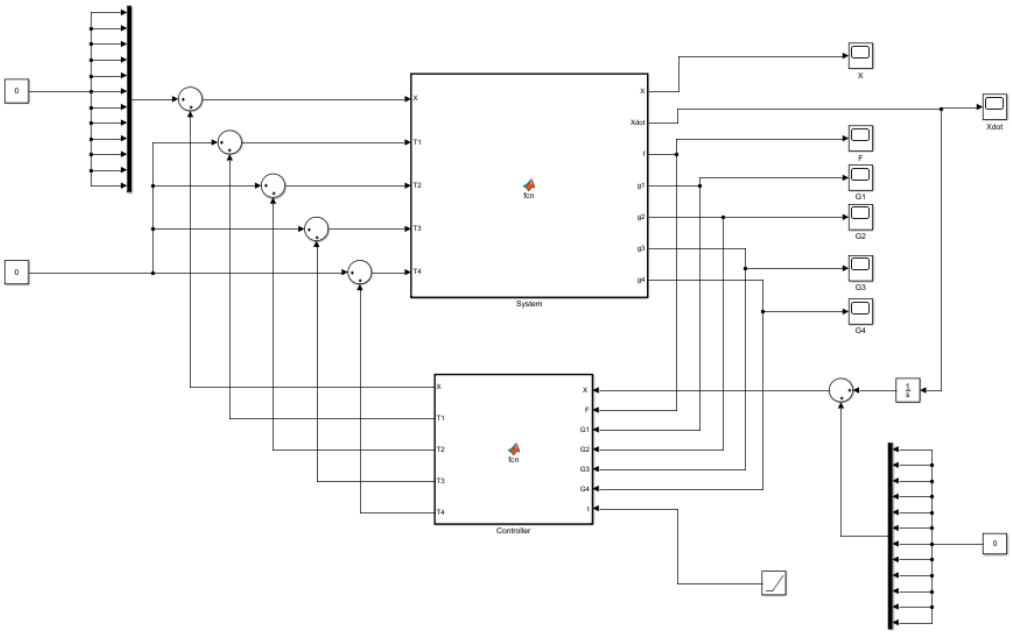


Figure 4.12 System – Controller Simulation of the quadcopter

Input	Gain	Value
T	K_p	25
	K_d	20
τ_ϕ	K_p	5
	K_d	-1
	k_p	1
	k_d	-2.4
	e_t	0.1
τ_θ	K_p	5
	K_d	-1
	k_p	1
	k_d	-2
	e_t	0.1

Table 4.6 Gain parameters of the controller of the quadcopter

Figure 4.13 shows the entire dynamics of the system when the desired target is (x, y, z) = (0.4, 0.6, 0.5). Figure 4.14a shows the trajectory of $\mathbf{z_{c1}}$ (x,y,z, and psi), Figure 4.14b shows the trajectory of $\mathbf{z_{c2}}$ (phi, theta).

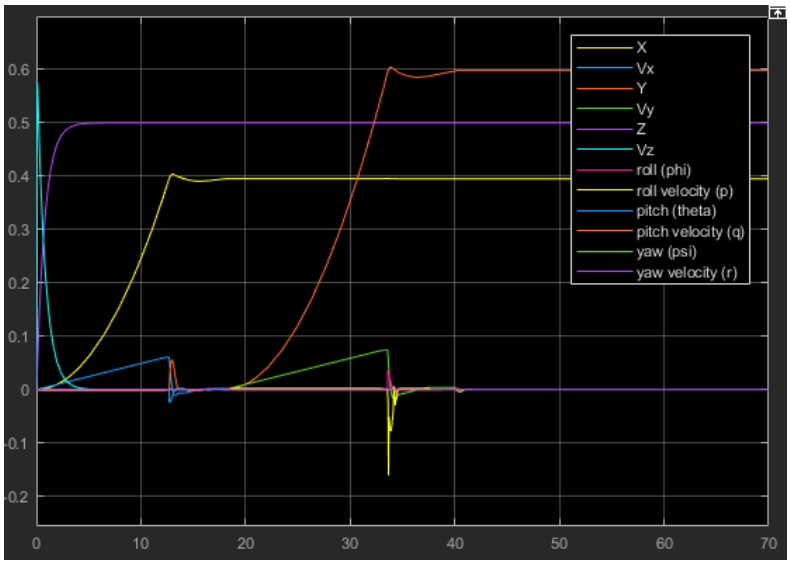
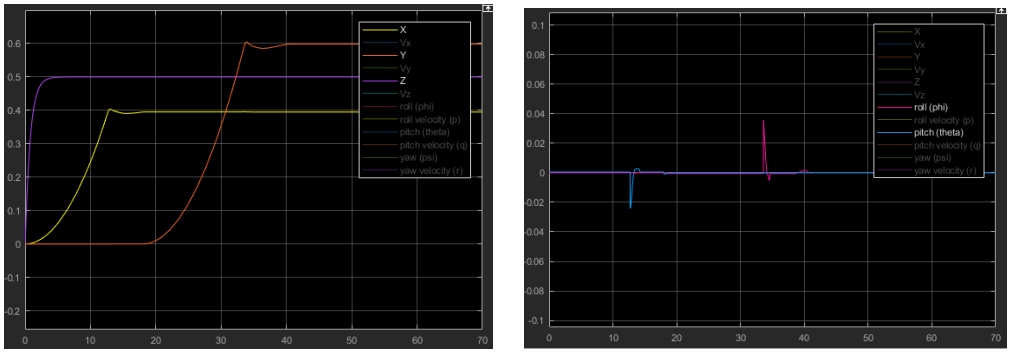


Figure 4.13 System state tracking of quadcopter



(a) (b)

Figure 4.14 (a)Trajectory of $\mathbf{z_{c1}}$ (x,y,z, psi) and (b)trajectory of $\mathbf{z_{c2}}$ (phi, theta)

The controller allows stable state tracking for all possible values in the XYZ space. Furthermore, the state could be tracked faster by increasing the value of sum of angular velocities of the rotors. For simplicity, in this simulation, first the x coordinate is tracked by controlling pitch angle, and then y coordinate is tracked by controlling the roll angle. However, both the roll and pitch angles could be controlled simultaneously to reach the desired target faster. A more sophisticated control algorithm than the PD algorithm would ensure a faster convergence of the value. The main aim of this simulation is to prove the validity of the proposed control algorithm, not to provide the fastest solution. Hence, a simple controller is used.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This thesis provided a novel control algorithm to control a class of unstable nonlinear systems that are underactuated and subjected to nonholonomic constraints. The proposed control law is implemented on two systems under the specified class of nonlinear systems – the mobile inverted pendulum and the quadcopter. The control algorithm for the mobile inverted pendulum and the quadcopter is common and there exist a number of control algorithms ranging from simple on-off control method to heavily complex reinforcement learning. But the dynamics of the mobile inverted pendulum and the quadcopter have never been compared or categorized into the same class of unstable nonlinear systems sharing common properties.

Comparing the common features of the mobile inverted pendulum and the quadcopter and their control algorithms reveals a bigger connection in their nonlinear affine in control equations. Both the systems have distinct stable and unstable state variables throughout the domain. Both the systems control their stable state variable by pushing a corresponding state variable out of the equilibrium and a controller designed to bring the latter into equilibrium asymptotically. In general, any system that falls under the specified class of nonlinear systems could be controlled to achieve perfect state tracking or even path tracking, by controlling its unstable state variables such that the torque developed to drive the unstable state into stability, also drives the system towards the desired path.

The class of unstable nonlinear systems mentioned in this thesis is not exhaustive to two particular cases. Numerous real life and practical systems would come under this class and the control algorithm proposes a general method to control any of these systems provided the dynamics equations of the system is available. For example, Ballbot or the ball balancing bot is a recently emerging system of robots developed by the Carnegie Mellon [38] is a dynamically-stable mobile robot designed to balance on a single

spherical wheel (a ball). The Ballbot works has 6 degrees of freedom and is actuated by 3 omnidirectional wheels. The Ballbot is subject to nonholonomic constraints due to the no slip constraints between the ball and the ground. Hence the ballbot falls under the specified class of unstable nonlinear systems. The control of ballbot could be easily achieved by the proposed control methodology.

5.2 FUTURE WORK

Future work includes implementing the proposed control algorithm into actual systems of the mobile inverted pendulum and the quadcopter since the simulations work as expected. The stability of the system with the proposed control law could be proven mathematically using Lyapunov stability criterion. The simulation in this thesis shows state tracking only. The scope of the thesis could be extended to prove path tracking for these systems.

REFERENCES

1. Anderson, B.D.O., and Moore, J.B. (1990), “Optimal Control: Linear Quadratic Methods”, Prentice-Hall, Englewood Cliffs NJ.
2. Ajwad SA, Iqbal J, Ullah MI, Mehmood A. “A systematic review of current and emergent manipulator control approaches”. Front Mech Eng China. 2015, 10:198–210.
3. Slotine J-JE, Li W. “Applied nonlinear control”, Prentice-Hall Englewood Cliffs, NJ, 1991.
4. Menini L, Zaccarian L, Abdallah CT. “Current trends in nonlinear systems and control” - In Honor of Petar Kokotovic and Turi 55 Nicosia: Springer, 2006.
5. Ajwad SA, Ullah M, Khelifa B, Iqbal J. “A comprehensive stateof-the-art on control of industrial articulated robots”. J Balk Tribol Assoc. 2014, 20:499–521.
6. H.-K. Khalil, “Nonlinear Systems”, Prentice-Hall, New Jersey, NJ, USA, 2002.
7. MATLAB and Simulink for Control System - Mathworks Resources, https://in.mathworks.com/solutions/control-systems.html
8. Y.S. Ha, S. Yuta, Trajectory tracking control of the inverse pendulum type self-contained mobile robot, Robot. Auton. Syst. 17 (1-2) (1996) 65-80.
9. Dean L. Kamen, J. Douglas Field, John B. Morrell, US6779621 B2, August 24, 2004.
10. F. Grasser, A.D. Arrigo, S. Colombi, A.C. Rufer, JOE: A mobile inverted pendulum, IEEE Trans. Ind. Electron. 49 (1) (2002) 107-114.
11. Boubaker, Olfa. “The Inverted Pendulum Benchmark in Nonlinear Control Theory: A Survey.” *International Journal of Advanced Robotic Systems* 10 (2013): 1-9
12. Nawawi, S. W. et al. “Development of a Two-Wheeled Inverted Pendulum Mobile Robot.” *2007 5th Student Conference on Research and Development* (2007): 1-5.
13. Kang, Ming Tao, Hoang Duy Vo, Hak Kyeong Kim and Sang Bong Kim. “Control System Design for a Mobile Inverted Pendulum via Sliding Mode Technique.” *2007 IEEE International Conference on Mechatronics* (2007): 1-6.

14. Huang, Jian, Hongwei Wang, Takayuki Matsuno, Toshio Fukuda and Kousuke Sekiyama. “Robust velocity sliding mode control of mobile wheeled inverted pendulum systems.” *2009 IEEE International Conference on Robotics and Automation* (2009): 2983-2988.
15. Nomura, Tetsuroh, Yuta Kitsuka, Haruo Suemitsu and Takami Matsuo. “Adaptive backstepping control for a two-wheeled autonomous robot.” *2009 ICCAS-SICE* (2009): 4687-4692.
16. Ravichandran, Maruthi T. and Arun D. Mahindrakar. “Robust Stabilization of a Class of Underactuated Mechanical Systems Using Time Scaling and Lyapunov Redesign.” *IEEE Transactions on Industrial Electronics* 58 (2011): 4299-4313.
17. Chang, Wen-Jer, Chia-Hao Chang and Cheung-Chieh Ku. “Fuzzy control with relaxed nonquadratic stability conditions for inverted pendulum robot system with multiplicative noise.” *IEEE ICCA 2010* (2010): 1019-1024.
18. Yi, Jianqiang and Naoyoshi Yubazaki. “Stabilization fuzzy control of inverted pendulum systems.” *Artif. Intell. Eng.* 14 (2000): 153-163.
19. Li, Zhijun and Chenguang Yang. “Neural-Adaptive Output Feedback Control of a Class of Transportation Vehicles Based on Wheeled Inverted Pendulum Models.” *IEEE Transactions on Control Systems Technology* 20 (2012): 1583-1591.
20. C. Yang, Z. Li, R. Cui and B. Xu, "Neural Network-Based Motion Control of an Underactuated Wheeled Inverted Pendulum Model," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2004-2016, Nov. 2014, doi: 10.1109/TNNLS.2014.2302475.
21. Jung, Seul and Sung Su Kim. “Control Experiment of a Wheel-Driven Mobile Inverted Pendulum Using Neural Network.” *IEEE Transactions on Control Systems Technology* 16 (2008): 297-303.
22. Chiu, Chih-Hui, Ya-Fu Peng and You-Wei Lin. “Intelligent backstepping control for wheeled inverted pendulum.” *Expert Syst. Appl.* 38 (2011): 3364-3371.
23. Chiu, Chih-Hui, You-Wei Lin and Chun-Hsien Lin. “Real-time control of a wheeled inverted pendulum based on an intelligent model free controller.” (2011).

24. Li, Zhijun and Chunquan Xu. “Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums.” *Fuzzy Sets Syst.* 160 (2009): 1787-1803.
25. Siuka, Andreas and Markus Schöberl. “Applications of energy based control methods for the inverted pendulum on a cart.” *Robotics Auton. Syst.* 57 (2009): 1012-1017.
26. Huang, Jian, Zhi-Hong Guan, Takayuki Matsuno, Toshio Fukuda and Kousuke Sekiyama. “Sliding-Mode Velocity Control of Mobile-Wheeled Inverted-Pendulum Systems.” *2016 12th World Congress on Intelligent Control and Automation (WCICA)* (2010): 461-466.
27. Muralidharan, Vijay and Arun D. Mahindrakar. “Position Stabilization and Waypoint Tracking Control of Mobile Inverted Pendulum Robot.” *IEEE Transactions on Control Systems Technology* 22 (2014): 2360-2367.
28. Adhikary, Nabanita and Chitralekha Mahanta. “Integral backstepping sliding mode control for underactuated systems: swing-up and stabilization of the Cart-Pendulum System.” *ISA transactions* 52 6 (2013): 870-80 .
29. Guo, Zhao-Qin, Jian-Xin Xu and Tong Heng Lee. “Design and implementation of a new sliding mode controller on an underactuated wheeled inverted pendulum.” *J. Frankl. Inst.* 351 (2014): 2261-2282.
30. Leishman, J.G. (2000). Principles of Helicopter Aerodynamics. New York, NY: Cambridge University Press.
31. Anderson, S.B. (1997). "Historical Overview of V/STOL Aircraft Technology". NASA Technical Memorandum
32. Hoffmann, G.M.; Rajnarayan, D.G.; Waslander, S.L.; Dostal, D.; Jang, J.S.; Tomlin, C.J. (November 2004). "The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC)". *Proceedings of the 23rd Digital Avionics System Conference*. (2004) 1–10
33. Büchi, Roland (2011). Fascination Quadrocopter.
34. Pounds, P.; Mahony, R.; Corke, P. (December 2006). "Modelling and Control of a Quad-Rotor Robot". *Proceedings of the Australasian Conference on Robotics and Automation*.

35. Hoffman, G.; Huang, H.; Waslander, S.L.; Tomlin, C.J. (20–23 August 2007). "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment". *Conference of the American Institute of Aeronautics and Astronautics*.
36. Andrew Hobden. "Quadcopters: Yaw". Hoverbear.org
37. T. S. Alderete, “Simulator aero model implementation.” NASA Ames Research Center, Moffett Field, California,
38. Ankit Bhatia, Carl Curren, Mark Dudley, “Dynamically Stable Mobile Robots in Human Environments”, http://www.msl.ri.cmu.edu/projects/ballbot/