Dynamics and Control of a Single Legged Offset Mass Hopping System

A Pre-synopsis Report

Submitted in partial fulfillment of the award of the degree of

DOCTOR OF PHILOSOPHY (Ph.D.)

by

Ajij D. Sayyad

Research Scholar

Roll No. : 03423001

Under the Guidance of

Prof. Bhartendu Seth (Guide)

Prof. K. Kurien Issac (Co-Guide)

Interdisciplinary Programme in Systems and Control Engineering,

Indian Institute of Technology Bombay,

Mumbai-400076, Maharashtra, India

2007-08

Abstract

In comparison to multi-legged robots, single-legged robots have only one type of locomotion gait, i.e., hopping, which represents a highly non-linear dynamical behavior consisting of alternating flight and stance phases. Hopping motion has to be dynamically stabilized and presents challenging control problems. In the proposed study, an asymmetrical planar one-legged hopping robot - a *Springy-Legged Offset-Mass* robot - is investigated. The offset-mass hopper poses a ability to compensate for forward pitching. A mathematical model is derived using a Lagrangian method.

An under-actuated hopping system mainly uses two actuators; linear actuator and rotary actuator. Both the actuators are active during the flight phase. The silent features of these actuators and the approach of applying these actuators during the flight phase is highlighted in the thesis. A 3-dimensional Poincaré return map is used to construct a reduced order implicit model representing the non-linear dynamics of the robot. Further, Using a *Taylor series approximation* technique, we formulated the explicit linear and non-linear approximate models of the proposed robot.

We explored the use of discrete-time classical, optimal and sliding mode based state feedback stabilization strategies to accomplish sustained hopping behaviour. The control input is derived based on the linear approximate model and applied to stabilize the nonlinear hopping model. The performance of the hopper with the use of a single measured state variable to stabilize the system is also explored in this thesis. The present work also enables us to estimate the maximum step disturbance admissible while achieving the stable hopping gait. The basin of attraction of proposed stabilizing controllers is stated in terms of the magnitude of step size introduced during motion. The investigation in the thesis will provide the evidence that a very simple control technique can control and stabilize a dynamic activity which is complex and non-linear.

Keywords : hopping robot, Poincaré map, Taylor series approximation, Discrete-time sliding mode, under-actuated model.

Table of Contents

List of Figures

List of Tables

Nomenclature and Acronyms

ENGLISH LETTERS

A	State matrix
B	Input matrix
C	Output matrix, Matrix containing elements related to coriolis/centrifugal forces (Lagrangian Formulation)
D	Inertia matrix (Lagrangian Formulation)
d	Mass offset
E	Total energy
G	Gravitational force vector (Lagrangian Formulation), Switching vector
g	Acceleration of gravity
h	Leg CG measured from foot tip
I_b	Body inertia
I_l	Leg inertia
$\mathbf{J}$	Jacobian (linear deviational model)
J	Jacobian (Dynamic Modelling)
K_e	State Observer gain matrix
K_p	State feedback controller gain matrix
K_S	Leg spring stiffness
k	Sampling interval, Number of complete hop
L	Lagrangian function
L_0	Length l when the spring force is zero
L_L	Rigid length of leg measured from the lower spring end w.r.t. foot tip
L_{LB}	Lower body length measured from the mechanical stop w.r.t. offset link
L_{UB}	Upper body length measured from the top of the hopper w.r.t. offset link
L_d	Length l after shortening of spring to desired amount
L_{to}	Length l when the body hits the leg
l	Length of springy leg measured from the foot w.r.t. offset link

m_b Body mass

m_l Leg mass

n Number of DOF of system

$\mathbf{P}$ Poincaré mapping function

P Pole (controller design specification)

q Generalized coordinates vector

S_0 Free length of spring

s Length of spring measured from the top of the hopper w.r.t. offset link

T Kinetic energy of the hopper

t Time

$\mathbf{u}$ Input

V Potential energy of the hopper

$\mathbf{x}$ State vector related to discrete return map

$\dot{q}$ Generalized velocity vector

x Cartesian horizontal position

$\mathbf{y}$ System output states

y Cartesian vertical position

GREEK LETTERS

δ Leg spring pre-compression

θ Leg orientation

λ Eigenvalue, Constraint force vector

ψ Constraint

τ External force or torque, sampling time

ζ State vector in new co-ordinate

SUBSCRIPTS

$+$ Immediately after a specified event

$-$ Just before a specified event

b Body

f Foot

l Leg

SUPERSCRIPTS

$+$	Immediately after a specified event
$-$	Just before a specified event
T	Transpose operator

ACRONYMS

CG	Center of Gravity
CSMC	Continuous-time Sliding Mode Control
DOF	Degrees of Freedom
DSM	Discrete-time Sliding Mode
DSMC	Discrete-time Sliding Mode Control
EL	Euler-Lagrangian
QSMB	Quasi Sliding Mode Band
SLOM	Springy Legged Offset Mass

Chapter 1

An Introduction

Inspired by agility of animal and human locomotion, the number of researchers studying and developing legged robots has been increasing at a rapid rate over a century [1]. The reason behind such sustained interest in legged robots is due to the fact that most of the earth's land surface is inaccessible to wheeled or tracked systems. Legged animals can, however, be found everywhere. Thus, mankind has been fascinated with the idea of a mobile legged robot that can handle difficult terrain and be useful in the fields of transportation, forestry, agriculture, fire fighting, hazardous areas, defense (carrying weapons to soldiers, de-mining), police purposes, assistive devices for walking, entertainment (toy production), robotic pets, ocean and space exploration.

1.1 Why to Study Single-legged Robots?:

The main advantages of legged locomotion can be summarized as [2]

:-: Active suspension of body
:-: Use of isolated footholds
:-: Adaptation to uneven terrain
:-: Less damage to soil and vegetation as compared to wheeled or tracked vehicles

Some of the challenges in developing legged robots are:

- Legged robots (part of mobile robots) have to carry the entire weight of the machine including the weight of all the actuators. This leads to the requirement of stronger and heavier legs, requiring bigger actuators. This has a multiplying effect on the weight of the machine.
- Payload-to-machine-weight ratio is less as compared to wheeled or tracked vehicles.

- Dynamics of legged robots tends to be more complex than stationary and wheeled mobile robots especially due to the presence of impact with ground and distinct swing and stance phases.
- Control of walking is complex as legged robots are inherently nonlinear and each leg works in two distinct regimes where the speed and force/torque requirements are very different.
- There is energy loss due to joint friction and intermittent impacts. As mobile robots have to carry their power pack, energy efficiency is very important.
- In multi-legged robots, coordination of legs is complex. Different gait patterns are admissible in multi-legged systems and depending on the number of legs, terrain and speed of locomotion; particular gaits are most efficient. Thus, the coordination pattern of legs may change as speed or terrain changes.

For all these reasons, building and analyzing legged robotic systems is relatively complex. Therefore, a large number of researchers have focused on single legged systems. They have simulated passive and controlled monopedal hopping motion, analyzed stability of such systems and developed hopping robot hardware. A comprehensive survey of the extensive research work done on the single legged hopping robots is presented in [3], one of the contribution of our research. In [3], we have attempted to cover developments of prototype models as well as theoretical models of such hopping systems.

1.2 Motivation:

In this thesis, we deal with a special kind of geometric model of hopping robot using an *asymmetrical* leg configuration and more interesting it uses one leg for locomotion. Single legged systems have simpler configurations and admit only one gait, namely, hopping. These robots require dynamic balancing. The following issues appear to be important in understanding mobility by hopping:

- Active balance and dynamic stability
- Use of elastic members (such as muscles and tendons) in enabling resonant mechanical oscillations associated with hopping.

Research in single legged robots has been focused in understanding the dynamics, control strategies and balancing principles. The major motivation behind our study is to model and control the single legged robot in order to understand the basic operating

principle for locomotion of legged systems. Being focussed on a single legged robot, the study is motivated by, both the configuration of leg and the design of the leg. An interesting feature of one-legged robot is that in general they are naturally always dynamically stable, unless a broad foot is used. Hence it is challenging to control its locomotion even though its dynamics is relatively simple.

Almost all one-legged robots [3] have symmetric configuration where the center of gravity (CG) of the body is along the line of action of the spring. These symmetric configurations are commonly referred as *Spring-Loaded Inverted Pendulum*(SLIP) models. The inherent stability of SLIP model is described with reference of *neutral point* and *symmetric stance phase* [4]. The SLIP model does not take into account the body pitch stabilization problem that any physical hopping system have to deal with [5], [6]. This is one of the reason why we decided to analyze a special kind of asymmetrical configuration to study the hopping motion.

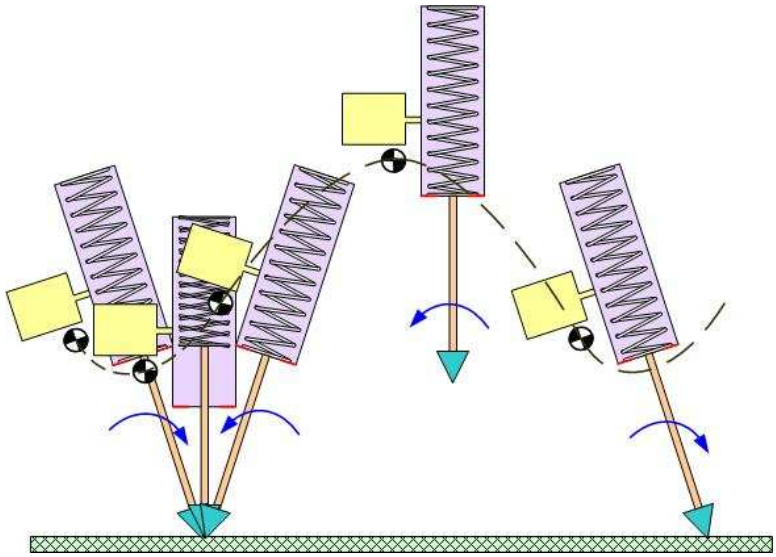

Figure 1.1: Schematic diagram illustrating the forward pitching compensation ability of passive asymmetrical configuration

The proposed configuration have CG of body at fixed distance (offset) from the geometrical axis of the leg. This configuration shows an interesting aspect of how physical asymmetry can be exploited for achieving dynamic stability. During the stance phase, length of leg changes through compression and extension phases. So a force couple arises due to the spring force because of the offset mass of the body. It results in anticlockwise force couple and body takes off with anticlockwise angular velocity. This rotation in anticlockwise direction remains continuous in flight duration in such a way that the angular momentum is conserved. Hopper body travels along the parabolic

trajectory and lands with some anticlockwise angular velocity. But due to the landing impact and the forward speed of propulsion, the direction of angular velocity changes. It pivots on foot with clockwise angular velocity for some duration before take-off. Thus, compensation of forward pitching during motion can be achieved through the use of an offset mass and springy leg as shown in the **Fig. 1.1**.

1.3 Brief Survey About 'Offset-Mass Hoppers':

Matsuoka [7] may be the first researcher to formulate a linearized two-link model of a hopping mechanism. Raibert is a major contributor in the field of hopping robot research [1] , [8], [9]. The mechanism designed and built by Raibert and co-researchers became the standard for several researchers, especially focused on telescopic leg configuration. The telescopic leg functioned as an energy restoring as well as a ground interacting element. After Raibert's work, there are significant developments in proposing the controller algorithms and analysis tools. In these models, the hopper had either one or two actuated joints: a springy prismatic leg and a revolute hip joint. The actuated rotary joint propelled the leg to and fro, and thus, controlled the direction of motion. A linear actuator located at the prismatic joint controlled the hopping height.

Most of the researchers have used relatively light telescopic leg mechanism attached to a massive body. In planar configurations, a revolute joint is used between the body and the leg. Some researchers have extended this to hopping robots in 3D by using spherical joint between the body and the leg. Such configurations have been mostly symmetric about the axis of the telescopic leg. Some researchers have departed from symmetric configurations. Asymmetric 2D configurations have been considered by Shanmuganathan [10], Kuswadi et al. [11], and Wei et. al. [12]. While, Chew et.al. [13] have considered 3D asymmetric configuration. We referred these models as "Offset Mass Hoppers", wherein the spring or actuator force was offset from the center of mass, which gave rise to restoring rotational moments. It has been claimed that asymmetric configuration is common in biological quadrupeds and has potentially beneficial effects on the dynamics of bounding [14]. In single-legged configuration, it may lead to a natural hopping motion as compared to a natural juggling motion (hopping in place) in symmetric configurations.

Shanmuganathan [10] considered the configuration of single- legged hopping robot shown in **Fig. 1.2**. He termed it the "Springy Leg Offset Mass (SLOM) Hopper". Through simulation and phase-plane analysis, he demonstrated the possibility

of achieving continuous jump due to rotational moments arising around CG during each cycle. He found that, due to an asymmetry in the configuration, the hopper was

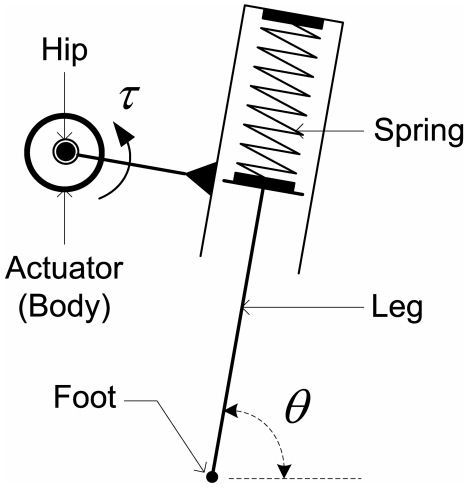

Figure 1.2: Under-actuated (rotary actuator at hip) *SLOM* hopper (adapted [10])

able to perform forward periodic sustained oscillations for some finite duration without any external actuation. He realized the need of starting the motion with proper initial conditions, which seemed to be practically difficult. So, he used a single rotary actuator (as shown in **Fig. 1.2**) to stabilize the passive forward-hopping motion of SLOM hopper. He reported the possibility of *p-hop* stabilization (periodic repetition after every 'p' hopping cycles) and was able to extend the duration of hopping motion.

Independently during same research tenure, Kuswadi *et al.* [11], [15] proposed the asymmetric model similar to Shanmuganathan's SLOM model [10]. They realized that a sustained jumping motion was possible using a telescopic pneumatic leg and single pneumatic linear actuator. They also constructed the physical prototype (shown in **Fig. 1.4**). The robot consisted of an offset body and a leg, which were in contact with a sufficiently wide horizontal ground surface. They produced partially elastic landing using counter weight arrangement, sand filled surface (covered with rubber sheet) and pneumatic cylinder (operated like a damper). Force actuation affected the orientation of the body by a moment of force that arose due to the mass of the body and the offset of the leg. Hence, both the orientation of the body and the height of a jump could be controlled by only one actuator. They [15] experienced some hardware difficulties and conceptual problems in this prototype. They found difficulty to get enough data to validate their model of the robot. They implemented a new prototype, which had the original asymmetrical configuration except changes in robot parameters that were optimally determined from the "transition map" [16].

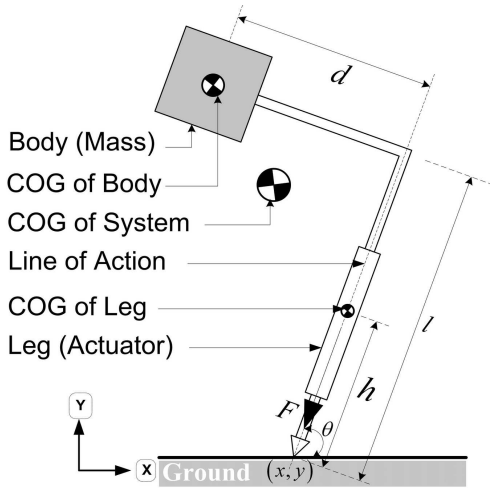

Figure 1.3: An asymmetrical hopping model proposed by Kuswadi *et al.* (adapted [11])

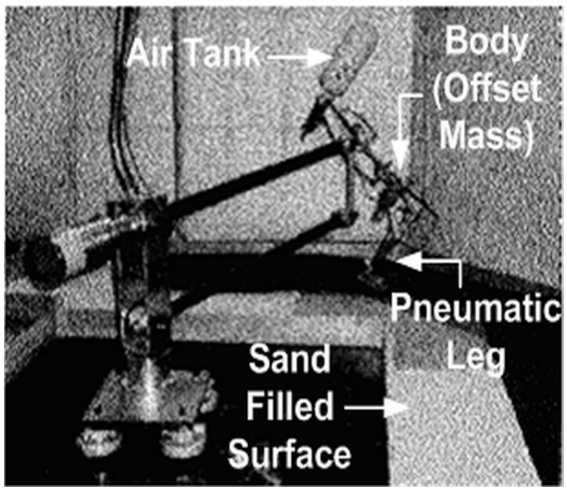

Figure 1.4: Improved prototype of hopping robot developed by Kuswadi *et al.* [14]

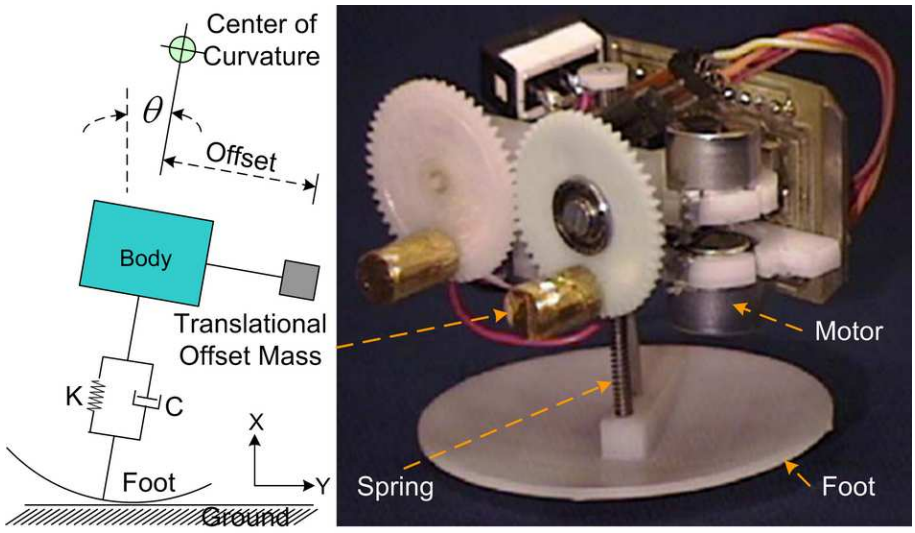

Figure 1.5: Leg schematic and a 5-cm hopping Monopod built by Wei *et al.* [12]

Wei *et al.* [12] constructed an autonomous one-legged hopper that could fit into a 5 cm cube. It was statically stable and passive-dynamically stable. **Fig. 1.5** shows the leg structure and robot prototype. The hopper was restricted to move in a plane and defined the motion with 5 DOF. Hopping was achieved by exciting a spring-mass system at its resonant frequency. 2D simulations were extensively used to choose the design parameters, before the construction of the robot. This hopper included a translational offset mass in order to cause the tipping action that helped in forward propulsion. The hopper did not have active directional control. It could travel at a rate of 7.75 cm/s or 1.5 body lengths per second. It could cover a distance of over 225 cm without

tipping over in one direction and climb a step of 1 mm. It could operate up to 45 minutes before depleting the energy stored in the batteries. Through experiments on the prototype, they observed that the performance could be improved if the hopper included some mechanism of directional control.

Based on the same mechanism of self-stabilization with a mass-offset, Chew *et al.* [13] constructed a prototype of a 3D hopping robot. It was able to hop in any direction and reject horizontal disturbances. As shown in **Fig. 1.6**, the prototype consisted of a springy leg, which was actuated electrically with a DC motor placed at the top of the leg. The rotary motion of the motor was converted to linear motion with the help of a slider-crank mechanism. Another servomotor along with a spur gear could rotate the weights, thus, influencing the offset mass system.

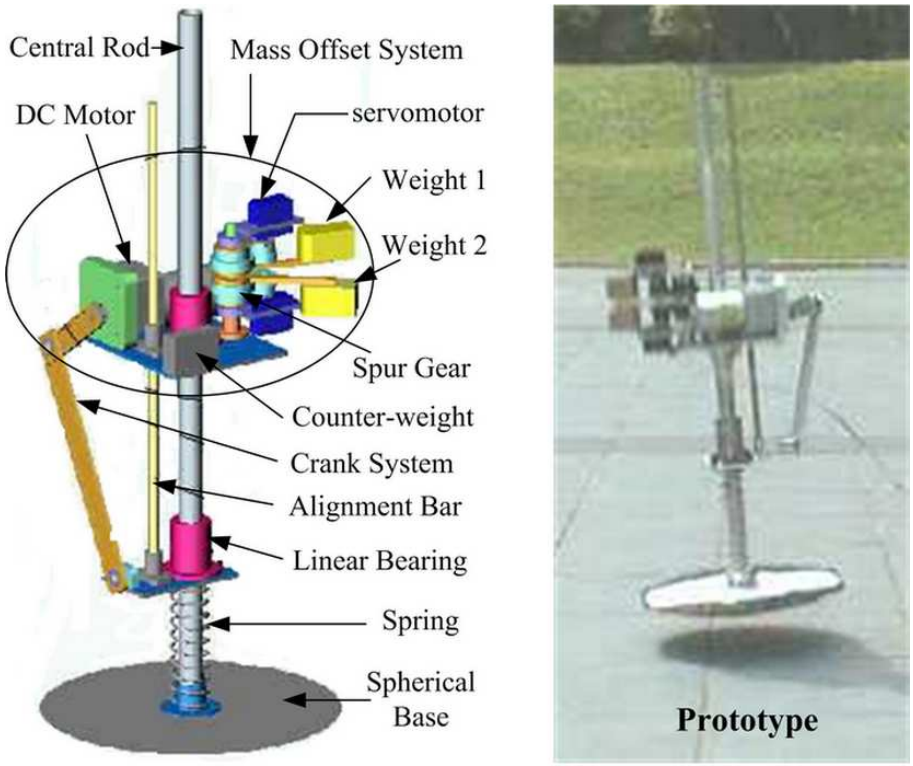

Figure 1.6: An overall design and prototype of hopping robot built by Chew *et al.* [13]

The comparative study of these robots with respect to different silent features is shown in Table 1.1 (**Note**: A term "NA" stands for non-availability of details in the said literature).

Table 1.1: The comparative study of 'Offset-Mass Hoppers'

Features	**P. V. S.** [10]	**Kuswadi** [11]	**Wei** [12]	**Chew** [13]
Body Mass	Yes	Yes	Yes	Yes
Leg Mass	No	Yes	No	Yes
Offset Link	Fixed	Fixed	Prismatic	Rotary
Actuator/s	1	1	1	2
Actuation	Rotary	Linear	Rotary	Both
Foot Type	Point	Point	Spherical	Spherical
Motion Plane	2D	2D	2D	3D
Prototype	No	Yes	Yes	Yes
Spring Use	Yes	No	Yes	Yes
Damper Use	No	Yes	Yes	No
Modelling	Lagrangian	Lagrangian	Newtonian	NA
Control at	Flight	Stance	Stance	NA
Control i/p	Continuous	On-Off	Discrete	NA
Controller Type	Adhoc	State F/B, Intelligent	Phase Lock Loop	NA

1.4 Control Strategies and Gait Stability

Hopping is a special type of running gait in multi-legged systems but is the only way a one-legged robot can locomate. Some variations are, however, possible. For example, hopping need not be simple periodic phenomenon and researchers have reported multi-hop periodic gaits, as well as, flipping hopping gaits. In general, single-legged robots require dynamic stabilization with the exception of robots equipped with a large foot. Several researchers have made use of this feature; however, this feature tends to negate an important potential advantage, namely, higher speed of locomotion, if the hopper is statically stabilized at each hop.

Single-legged robots have to be dynamically stabilized and pose a challenging control problem. Numerous attempts have been reported that aim to maintain some desired hopping height, forward speed, body orientation, etc., at certain phase of the hopping cycle. Stabilization of such robots may include active or passive dynamics. Also, the derived controller/s may be active either during stance phase or flight phase or both phases. From a control perspective, classification of such strategies may be

given as a state feedback strategy, multi-rate feedback strategy, neural-fuzzy-genetic strategy, adaptive feedback strategy, dead-beat control, or hybrid control strategies. Some literature also mentions some open-loop control, energy efficient control and motion planning strategies. In this section, we are focusing on different approaches to stabilize the single-legged hopping robot.

Generally, the hopping robot with non-telescopic leg (spring-less articulated leg) configuration use revolute active joint. The overall objective is to achieve a soft-landing trajectory by suppressing the energy loss due to hard landing by shaping the flight trajectory. The approach is based on the principle of conservation of kinetic energy. The hip compliance is also used to ensure soft landing. While, for controlling hopping height and velocity, they steered the posture during the flight phase based on law of conservation of angular momentum [17].

Animals use muscles whose elastic properties help in conserving energy and providing shock tolerance. The springy articulated leg models are mostly bio-inspired mechanisms. The models utilizing telescopic and springy articulated leg configurations, use similar to Raibert's three-part controller for controlling the forward speed, hopping height, foot placement, and transition between various gaits. The computer algorithms that controlled the motion of Raibert's one-legged robot [1] were decomposed into two basic control problems. The control objectives of the Raibert's "vertical motion controller" were to initiate and terminate the hopping and to control its height. Lengthening and shortening the length of the leg using leg position actuator accomplished these objectives. This leg actuation was based on changing the energy (based on maximum height error as feedback) of the resonant mass-spring system formed by the leg and the body in order to maintain the predetermined height. The control objectives of the Raibert's "horizontal controller" were to control the forward velocity and to maintain the balance of body posture or body attitude to prevent the robot from tipping.

Hopping motion is repetitive and therefore several researchers have made use of poincaré sections in analyzing the dynamics. Hopping dynamics is also highly nonlinear as there are stance and flight phases with sudden transitions. The dynamics of the two phases are also very different. Some researchers have studied the richness of the dynamics of these systems by analyzing chaos and bifurcation diagrams [18], [19]. For studying the controllability of these systems, some researchers have used a Lie algebra approach [20], [21]. This approach establish the stability of the system, which is based on a two-part Lyapunov function corresponding to the two phases of motion.

Purely passive systems cannot sustain hopping motion on level ground indefinitely due to energy losses. Energy loss occurs due to impacts, as well as, due to friction in joints. Several researchers have used open loop energy compensation mechanism. Use of an actuator to pre-charge the leg spring appears a popular way of achieving this in a juggler. The latch mechanism used to store energy in the spring is released when the leg hits the ground leading to the additional pre-charge energy becoming available for the next cycle. It is conceivable that such energy-compensated systems would undergo sustained hopping even in the absence of active control.

Different approaches have been used in designing controllers for single legged robots. A popular approach in practical systems has been based on heuristics. The concepts of neutral points and zero moment points have been successfully exploited. Many researchers have designed separate controllers for height control and speed control. Some researchers have designed the controller based on neural, fuzzy logic and genetic algorithms.

1.5 Gap in the Literature and Objectives of Investigation:

It is observed that the offset-mass hoppers posses the ability to compensate for forward pitching. This configuration shows an interesting aspect of how physical asymmetry can be exploited for achieving the dynamic stability. Also, there is no need to consider a revolute hip joint for forward locomotion. The overall objectives of investigation is to propose a springy-legged offset-mass (SLOM) hopper and devise control strategies for making it hop with stability.

Based on the fundamental principle of conservation of angular momentum, many researchers proposed a "closed-loop" strategy that could optimally control the body orientation of one-legged-robot during flight phase using the internal motion of the leg. Stretching or shortening the leg during the flight could alter the moment of inertia and effectively vary the angular motion of the body.

Most of the configurations analyzed and built have utilized separate actuators for controlling leg bounce and for controlling leg orientation. Both these actuators are required to achieve hopping motion for symmetric configurations whereas it has been shown that a single actuator could suffice for asymmetric configurations. Asymmetric models have been demonstrated with a single solenoid actuator as well as with a single

reaction wheel actuator.

Kuswadi [11] used a linear actuator (on-off) that shortens the leg length during flight phase and lengthens it during stance phase. This actuation may lead to hard landing and high energy losses. consequently, the control effort required will be more. Also, in Kuswadi's model, the control input is the thrust time during stance after the vertical position of CG of hopper is exactly above the vertical foot position. Shanmuganathan [10] used only a single rotary actuator and not the linear leg actuator. Since, it is claimed that only the leg actuation is sufficient to control jumping height and leg orientation, it is worthwhile to investigate the model performance with controlled leg length. In this study, we decided to input energy by compressing spring during flight phase. The stance phase follows the natural dynamics. There are several advantages of this approach as described in the next Section.

Control can be applied during the stance phase, during the flight phase or during both phases of motion. The stance phase control can utilize the reaction forces for reorientation, which is not available in the flight phase control. No work has been reported on use of reaction jets as actuators. Flight phase motion of the center of mass of the system, as a whole, is ballistic. However, leg reorientation in flight phase can be achieved by relative motion of different parts of the robot. A single reaction wheel actuator is also investigated in this thesis.

Shanmuganathan [10] used a single rotary actuator to stabilize the passive forward-hopping motion of SLOM hopper. He reported the possibility of *p-hop* stabilization and was able to extend the duration of hopping motion. This composite stabilization strategy was based on the alternative use of desired forward and vertical velocity determined from the nominal hop. This strategy realized limping gait and despite the back-and-forth hopping, there was a net forward motion. But the limitation of this stabilization approach was that the angular velocity of the rotary actuator became unbounded. Also, this stabilization strategy is *adhoc*. So, while designing controller to stabilize hopping model with rotary actuator, we have kept these constraints in mind. Also, we have attempted to apply both linear and rotary actuator simultaneously. In real systems, it is not always possible to measure all variables of interest for feedback. In such a situation, generally it is recommended to develop feedback controller based on estimator or observer. Also, it is noticed that there is no literature, for single-legged configuration, which focus on the control design based on observer. So, we have also explored observer based controller to stabilize the proposed model.

The SLOM hopper model is first investigated for fixed energy input. Based on

local stability analysis, we propose a design which has a simple method for energy injection, with the hope that a physical model can be designed and made to hop. In devising a feedback control strategy, our approach is to use simple inputs and minimal sensing. We intend to a devise a control system which include practical advantages for a two-dimensional hopper.

It is well known that, poincaré map is one of the popular analysis tools for investigating stability of nonlinear periodic systems, like the hopping systems. Our investigation is based on the approximate models derived using this tool. With the help of a discrete poincaré map, an implicit nonlinear model can be derived. Further, by linearising it about the fixed point, the explicit linear model is used to derive the feedback control law such that it could stabilize (at least locally) the model with minimum actuation. The stability analysis can be carried for particular type of the structured disturbances. Eventhough this will not give complete stability analysis, the robustness of these stabilization strategies can be illustrated via simulation.

It is well known that the hopping robotic system is highly nonlinear hybrid system requiring dynamic stabilization . Also, these systems are highly sensitive to the parameter variations and perturbations during locomotion. Further, any physical system can not be accurately modeled from control design perspective. It is therefore the controller designer's role to ensure that the required performance levels exist despite the existence of expected model mismatches. This concern can be solved by developing a robust control system. One of the robust controller design approach is the sliding mode control methodology. Regarding the sliding mode based control theory, it is based on the concept of varying the structure of the controller based on the state variation of the system in order to obtain a desired response [22]. So, one of our objectives is to develop the sliding mode controller to control the nonlinear hopping model.

The evolved dynamics of the hopping robot by choosing appropriate Poincaré section is discrete time system. So, our investigation is focused on a Discrete-time Sliding Mode Control (DSMC) system. Here the 'time' refers to the hop to hop interval. Recently, many studies are tackling the discrete-time sliding mode (DSM) controller design problems [23]. A typical characteristics of DSMC systems is that the control input is applicable only at certain sampling instants and the control effort is constant over the entire sampling period. However, the system states are no longer constrained to lie upon the confined sliding surface. As a result, DSM can undergo only quasi-sliding mode.

1.6 Thesis Contributions:

In an attempt to understand the basic operating principles and dynamic balancing for locomotion of single legged hopping robots, we carried a comprehensive survey of developments in the field of single-legged hopping robots. A large fraction of studies on legged robots has focused on modeling and control of single-legged hopping machines. In the survey paper [3], we have attempted to cover development of prototype models, as well as, theoretical models of such hopping systems. In order to focus on the area of the core objectives of investigations, we have not included the literature dealing with these hopping systems in depth in the thesis.

We proposed a 2D asymmetrical single -legged hopping robot, referred as "Springy Legged Offset-Mass" hopper. Accounting for the asymmetry in single-legged hopping robot is a new contribution along with a few studies [10]- [13]. It consisted of a body and a leg, which are in contact with a sufficiently wide horizontal ground surface. The linear spring is incorporated between leg and body. The major contributions of thesis can be summarized as below:

- A generic kinetic model of the asymmetrical one-legged hopping robot was formulated, which is different from Shanmuganathan's geometrical model [10].
- Based on constrained Euler-Lagrangian methodology, a mathematical model of SLOM hooping robot is presented for passive as well as active configurations. We defined a 3-dimensional Poincaré map in order to study the dynamic behaviour of proposed model. we validated the model based on basic principles of physics.
- We determined periodic motions by solution of equations for fixed point of a Poincaré map and local stability of periodic motion.
- We proposed three different under-actuated stabilization strategies, utilizing: (i) the simple one linear actuator, (ii) the reaction wheel and (iii) both linear and rotary actuators, to realize the continuous hopping motion.
- We developed linear state feedback controller as well as different state based observers and studied stability using simulations.
- An approximated non-linear model is formulated, which is an extension of the approach followed for formulating linear model using linearized Poincaré map. This study uses a Taylor series approximation technique. This model is used to solve the optimal state feedback control problem.

- We developed a discrete-time sliding mode full state feedback controller based on both linear and non-linear models. Moreover, to our knowledge, this is the first attempt to apply DSMC for single-legged hopping robot.

- In an attempt to set the basis for determining the basin of attractions of proposed controller, we suggested a simple approach to introduce the structured disturbance (fixed sized step) during motion.

1.7 Organization of the Thesis:

This section outlines the remaining five chapters of the thesis.

- **Chapter 2: Mathematical Modelling of SLOM Hopper and One of its Periodic Motion:** In this Chapter, we describes the proposed planar one-legged asymmetrical hopping model under investigation. The description is focused to understand the geometrical arrangement of the proposed model. The mathematical model is based on Euler-Lagrangian dynamic equations. This chapter also described the use of energy pumping mechanism in order to seek the existence of any periodic motion. The study makes use of a 3-Dimensional Poincaré map to arrive at some fixed points exhibiting cyclic motion. Ensuring the existence of one of the periodic motion, the local stability is analysed with the help of a linearized Poincaré map. The validation of model and the simulation results are also presented at the end of this Chapter.

- **Chapter 3: State Feedback Control of SLOM Hopper with a Linear Actuator:** This Chapter mainly focuses on the stabilization strategy based on a linear leg actuator. This chapter describes the full-order observer based state feedback control for stabilizing the proposed hopping system. The key goal is to use single observed state variable to estimate all state variables. We have used a current observer to induce the dynamic stability when the system is subjected to small disturbance from the fixed point. Optimal state based control problem is formulated and solved. We presented two DSMC approaches. The present study uses the Gao's reaching law [24] and the saturation type reaching law [25]. The proposed DSM state feedback based control law is applied for the nonlinear hopping robot that guarantees discrete sliding mode. The idea is to devise the

control input based on approximated models and apply to actual nonlinear hopping model,

- **Chapter 4: State Feedback Control of SLOM Hopper with a Rotary Actuator:** In the Chapter 4, we presents the use of a rotary actuator to stabilize the motion. The rotary actuator exerts the torque to swing the robot leg permitting it to reorient at landing. This under-actuated model utilize the fixed input injection via spring compression during flight. The rotary actuator is active only during fixed fraction of flight duration. Different state feedback based control strategies are applied as described in Chapter 3. The angular velocity of actuator is bounded. The performance is much better than utilizing only the linear actuator.

- **Chapter 5: State Feedback Control of SLOM Hopper with Two Actuators: Linear and Rotary:** This chapter focuses on utilizing two actuators simultaneously for stabilizing the hopping motion. The model is still under-actuated. The Chapter focuses on the application of DSMC approach for stabilizing a hopper considering linear uncertain (unmatched type uncertainty) model with two inputs. Various state feedback control strategies are applied. Regarding the performance level, it is shown by simulation that the controller hopper motion is stabilizable with in few number of hops. Also, the basin of attraction is larger than the previous un-actuated systems.

- **Chapter 6: Conclusions and Recommendations:** The results of the investigation are concluded and a list of suggestions for future work are suggested.

Chapter 2

Mathematical Modelling of SLOM Hopper and One of its Periodic Motion

2.1 Description of the Proposed Hopping System:

The proposed hopping model under investigation is an asymmetrical configuration, referred as "*springy-legged offset-mass*" (SLOM) hopper. The proposed robot propels through two major regimes as stance phase and flight phase. The beginning of the flight and stance phases are termed *take-off* and *touchdown*, respectively. The robot is subjected to two impacts as landing and take-off which are discrete events. The posture of proposed model during stance phase is shown in the **Fig. 2.1**. During flight phase the robot's center of gravity (CG) follows a parabolic trajectory. Angular momentum is conserved during this phase. Landing takes place at the end of flight phase, where the foot touches the ground. At the instant of landing, the foot is assumed to rest suddenly so it creates an impact between body and leg. Stance phase is followed by the landing, where springy leg experiences compression and decompression. The spring stiffness affects the stance duration. At the end of stance phase, the robot takes-off either naturally (when vertical reaction force becomes zero) or when the mechanical stop restricts further lengthening of the spring. In the second case, the body hits the leg, so there exist another impact. In both the type of impacts there is impulsive change in velocities and energy loss.

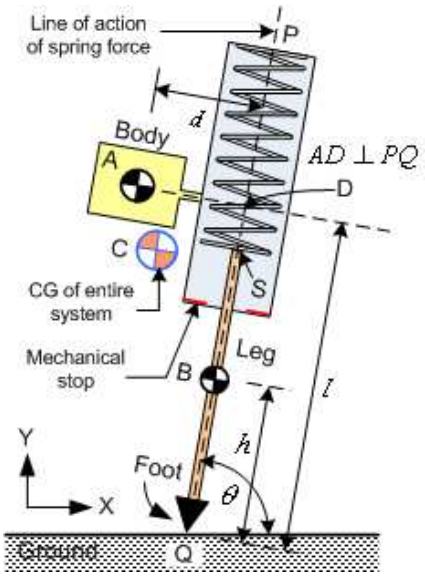

Figure 2.1: The proposed one-legged hopping robot (SLOM)

2.1.1 Kinematic Arrangement:

The one-legged hopping robot used in this study is shown in **Fig. 2.1**. This is a special kind of geometric model of hopping robot that uses an asymmetrical leg configuration. The linear spring is incorporated between leg and body. Shanmuganathan [10] studied a similar configuration where it was assumed that the leg is massless and inertialess. From practical aspect, the leg with mass and inertia is included in our modelling. Kuswadi et al. [11] have also considered the massive leg.

The detailed insight of kinetic arrangement of proposed robot is shown in **Fig. 2.2**. The SLOM hopper consist of a body of mass m_b and moment of inertia I_b about its center of gravity, located at A, and a leg, PQ, of mass m_l and moment of inertia I_l about its center of gravity, located at B. The leg is connected to the body through a massless spring PS of stiffness K_s. It may be noted that the action of the leg spring is offset by a distance d from the center of mass of the body located at A, and hence, the spring force does not act along a line passing through the CG (located at C). The leg length (l) is defined to be the distance DQ measured along PQ. The leg motion in downward direction is restricted by a mechanical stop. It is to be noted that, δ is the spring pre-compression. This parameter is important to express the free length of leg spring. The spring and mechanical stop arrangement is such that spring is in pre-compressed state (δ) when the body hits the leg. So, at the end of stance phase (de-compression phase) body hits the leg, so there exist the impact and due to spring pre-compression the hopper reaches to the take-off state.

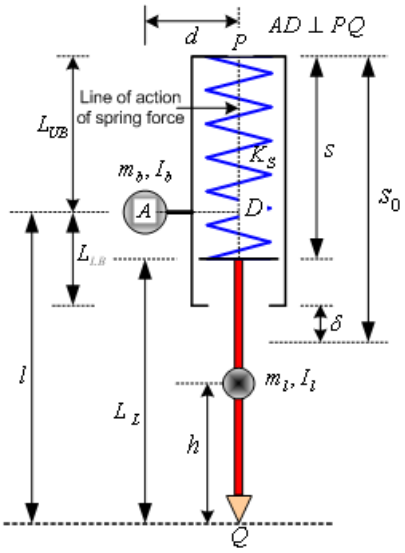

Figure 2.2: A SLOM model describing the kinematic arrangement

Table 2.1 summarizes the typical mechanical specifications of the system considered in this study. Table 2.2 describes few variables of interest and geometrical relationships between them, used in the further study.

2.1.2 Assumptions:

The following assumptions are made in the present study of SLOM hopper:

- Hopping motion is restricted to a vertical plane.
- The ground surface is horizontal and rigid.
- The leg does not slip at landing.
- The leg is without ankle and the foot is pointed.
- Spring is massless and spring force is axial.
- The relative motion during stance between the body and the leg can be modeled as a prismatic joint.
- Hopper takes off as soon as the body hits the leg and the mechanical stop restricts further stretching of springy leg.
- There are no internal friction forces due to contact between leg and body.

Table 2.1: The typical specifications of hopping robot

Parameter	Description	Unit
m_b	Body mass	3.0 kg
m_l	Leg mass	0.3 kg
I_b	Body inertia	0.8 kgm^2
I_l	Leg inertia	0.008 kgm^2
d	Mass offset	0.1 m
h	Leg CG measured from foot tip	0.25 m
K_S	Leg spring stiffness	2000 N/m
L_{UB}	Upper body length measured from the top of the hopper to point D	0.25 m
L_{LB}	Lower body length measured from the mechanical stop to point D	0.15 m
L_L	Rigid length of leg measured from the lower spring end to the foot tip	0.5 m
δ	Pre-compression of leg spring	0.002 m
g	Acceleration of gravity	9.81 m/s^2

Table 2.2: Description of some other variables of interest of hopping robot

Parameter	Description	Unit
l	Length of springy leg measured from the foot to point D (Fig. 2.2)	m
s	Length of spring measured from the top of the hopper to point S (Fig. 2.1)	m
S_0	Free length of spring	m
L_0	Length l when the spring force is zero	m
L_{to}	Length l when the body hits the leg	m
$\triangle l$	Change in the length l	m
	$S_0 = L_{UB} + L_{LB} + \delta$	m
	$L_0 = S_0 + L_L - L_{UB} = L_{LB} + \delta + L_L$	m
	$L_{to} = L_{LB} + L_L = L_0 - \delta$	m
	$\triangle l = l - L_0 = s - S_0$	m

2.2 Euler-Lagrangian (EL) Equations of Motion:

The robot model has three motion degrees of freedom (DOF) during the flight phase. These DOF includes x_b, y_b, the Cartesian, planar Body (hip) position; and the θ, the leg orientation with respect to horizontal as shown in the **Fig. 2.3**. During the stance phase the number of DOF is two, l, the length of the springy leg measured from foot to the foot tip and θ. There are two possible actuation: the leg actuator (linear), and the hip actuator (rotary). The investigation in the current and next Chapter assumes only linear actuation. The use of rotary actuator is described in detail in Chapter 4. Use of both linear and rotary actuators has been explored in this thesis and described in Chapter 5. In this Section, we formulate the dynamic model of proposed hopping system. The core objective of modelling is to derive a mathematical model (set of equations) in order to simulate the dynamical behaviour of the robot.

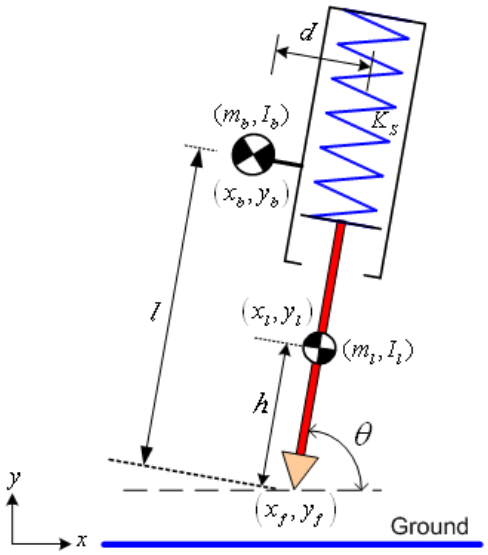

Figure 2.3: An un-actuated hopping model

2.2.1 The Un-actuated Hopping System:

The equations that govern the motion of the un-actuated (passive) SLOM hopping robot, shown in the **Fig. 2.3**, are developed using the Lagrangian approach. In the un-actuated configuration, it is assumed that there is no relative motion between the leg and the body during flight. A hopping robot moves with alternating flight and stance phases. In the EL formulation, during such phase change, the dynamics of the robot does not vary, but only the constraints imposed on the robot changes. We choose $q = [\ x_b \quad y_b \quad l \quad \theta\ ]^T$ as the set of generalized coordinates. Here x_b and y_b are the co-

ordinates of CG of body.

The standard form of a robot model (for no external forces) is,

$$\begin{aligned} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) &= -J^T(q)\lambda_r \\ J(q)\dot{q} &= 0 \end{aligned} \tag{2.1}$$

Where, n is the number of DOF of the system. $D \in R^{n\times n}$, $C \in R^{n\times n}$, and $G \in R^n$ are called the inertia matrix, the "coriolis and centrifugal forces" matrix and gravitational force vector respectively, and given as,

$$D = \begin{bmatrix} M & 0 & -m_l\cos\theta & m_l A(l,\theta) \\ 0 & M & -m_l\sin\theta & -m_l B(l,\theta) \\ -m_l cos\theta & -m_l\sin\theta & m_l & -dm_l \\ m_l A(l,\theta) & -m_l B(l,\theta) & -dm_l & I \end{bmatrix}, \tag{2.2}$$

$$C = \begin{bmatrix} 0 & 0 & m_l\sin\theta\dot{\theta} & m_l\dot{l}\sin\theta + m_l B(l,\theta)\dot{\theta} \\ 0 & 0 & -m_l\cos\theta\dot{\theta} & -m_l\dot{l}\cos\theta + m_l A(l,\theta)\dot{\theta} \\ 0 & 0 & 0 & m_l(l-h)\dot{\theta} \\ 0 & 0 & m_l(l-h)\dot{\theta} & m_l(l-h)\dot{l} \end{bmatrix}, \tag{2.3}$$

$$G = \begin{bmatrix} 0, & Mg, & -m_l g sin\theta + K_s(l-L_0), & -m_l B(l,\theta) \end{bmatrix}^T. \tag{2.4}$$

Note that, $M = m_b + m_l$, $I = I_b + I_l + m_l\Big((l-h)^2 + d^2\Big)$, $A(l,\theta) = (l-h)sin\theta + dcos\theta$, $B(l,\theta) = (l-h)cos\theta - dsin\theta$ and L_0, is the length l when the spring force is zero. λ_r is the generalized constraint force. J is Jacobian matrix, determined based on the constraints. The subscripts b and l stands for parameters related to body and leg of hopper respectively.

Due to sudden change in the constraints during the motion, impulsive forces arise in the system. Such impulses, λ_I, are assumed to exist for an infinitesimally small duration. Due to impact there is sudden discontinuous change in the generalized velocities, while the generalized coordinates and input change continuously. The change of the derivative of the generalized co-ordinates ($\dot{q}$) can be determined by integrating equation (4.1) with respect to time across the impulse.

$$\begin{aligned} D(q)(\dot{q}_+ - \dot{q}_-) &= -J^T(q)\lambda_I \\ J(q)\dot{q}_+ &= 0 \end{aligned} \tag{2.5}$$

where, $\dot{q}_+$ and $\dot{q}_-$ are respectively the post and prior generalized velocities across the impulse. Refer **Appendix A** for the constraints and Jacobian during different phases and events. These constraints are holonomic in nature.

2.2.2 Energy Compensated System using Linear Actuator:

As explained in Section 1.3, it is known that the proposed model poses the ability to compensate for forward pitching, due to presence of mass offset. This property of proposed hopping system can be exploited for achieving dynamic stability. But, it is also known that the SLOM hopping robot exhibits intermittent discrete impacts and highly nonlinear dynamics. As a result, the systems's motion exhibit very different dynamics applied at different phases of the hoping gait. There is loss of energy due to impacts and frictional affects, hence such motion is unsustainable indefinitely. As reported in Chapter 1, our core objective is to obtain the periodic motion with the help of energy compensation mechanism. This energy compensation mechanism can yield extra "stabilization". It may be remembered that the term 'stabilization' used here stands for prevention of hopper from falling and extension of hopping motion.

In that perspective, we proposed energy compensating mechanism using a linear actuator. Also, it is possible that both the orientation of the body and the height of a jump could be controlled by using only one actuator [11]. The actuator force (spring defection) affected the orientation of the body by a moment of force that arose due to the mass of the body and the offset of the leg. A periodic hopping motion is possible only if the lost energy is compensated periodically. In our model, energy is compensated in each hop by pre-compressing the leg spring in the flight phase. There are several advantages of this approach:

- The flight duration being generally longer than the stance duration, more time is available for spring compression requiring a lower power actuator.
- Imparting energy in spring during flight requires a smaller actuator force as the spring is least compressed to begin with.
- The actuator is active only during compression of the spring and is unloaded by the use of a latch, which is released with the landing impact force.

All these features lead to small energy requirement for the actuator, which is very important for any autonomous hopping robots. However, the capability of this input in stabilizing and controlling the SLOM, needs to be examined. So, we looked for the

existence of periodic motion of hopping model by adding fixed amount of energy during each hopping cycle.

2.2.3 Selection of Leg Length Trajectory (l) during Actuation:

The proposed linear actuator attached to the springy leg is shown in **Fig. 2.4**. The role of the linear actuator is to retract the springy leg to shape the energy utilized for locomotion and to achieve a continuous periodic gait. It is desirable to choose smooth trajectory for variation of leg length, l, during actuation phase. The chosen trajectory

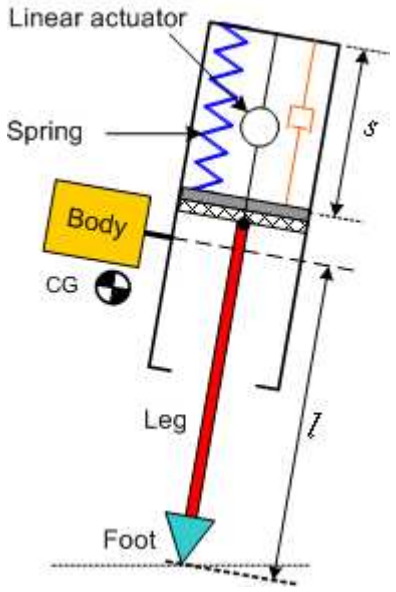

Figure 2.4: Schematic diagram of hopping robot with linear leg actuator

can be obtained from a continuous actuator force profile. The boundary conditions are enumerated below:

1. Leg length (l) at the instant when retraction phase starts and when retraction phase ends should be equal to L_{to}, i.e. desired retraction.

2. Linear leg velocity ($\dot{l}$) at the beginning and end of retraction phase should be zero.

3. Linear leg acceleration ($\ddot{l}$) at the beginning and end of retraction phase should be zero.

Since desired trajectory should satisfy six conditions, we have to select quintic trajectory as,

$$l(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 = f_1(t) \tag{2.6}$$

$$\dot{l}(t) = 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1 = f_2(t) \tag{2.7}$$

$$\ddot{l}(t) = 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2 = f_3(t) \tag{2.8}$$

Solving these equations for known boundary conditions at time $t \in [\ t_{start},\ \ t_{end}\ ]$, we can determine unknowns $[\ a_5,\ \ a_4,\ \ a_3,\ \ a_2,\ \ a_1,\ \ a_0\ ]$. The suffixes t_{start} and t_{end} indicates the pre-known time instants when retraction starts and ends during actuation phase.

During this linear actuation phase, all four general coordinates x_b, y_b, l and θ are varying continuously. It is to be noted that there is no external force applied during this phase ($\tau = 0$). As described above, we can pre-determine $f_1(t)$, $f_2(t)$ and $f_3(t)$ profiles. So, we can express the motion of the system during this phase using equation (4.1), where,

$$D = \begin{bmatrix} M & 0 & 0 & m_lA(l,\theta) \\ 0 & M & 0 & -m_lB(l,\theta) \\ 0 & 0 & 1 & 0 \\ m_lA(l,\theta) & -m_lB(l,\theta) & 0 & I \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & m_l\sin\theta\dot{\theta} & m_l\dot{l}\sin\theta + m_lB(l,\theta)\dot{\theta} \\ 0 & 0 & -m_l\cos\theta\dot{\theta} & -m_l\dot{l}\cos\theta + m_lA(l,\theta)\dot{\theta} \\ 0 & 0 & 0 & m_l(l-h)\dot{\theta} \\ 0 & 0 & m_l(l-h)\dot{\theta} & m_l(l-h)\dot{l} \end{bmatrix},$$

$$G = \begin{bmatrix} -m_lf_3(t)\cos\theta \\ Mg - m_lf_3(t)\sin\theta \\ -f_3(t) \\ -m_lB(l,\theta) - dm_lf_3(t) \end{bmatrix},$$

$$J(q) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.9}$$

2.3 Validation of Mathematical Model:

In order to validate the mathematical model of SLOM hopper, we simulated the unactuated model to generate multiple 11 hopping cycles. The initial conditions are set such that the hopper body is dropped from some height and it will perform the in-place hopping, i.e. the offset is zero ($d = 0\ m$) and horizontal velocity is zero. The

other robot parameters are chosen as given in Table 2.1. The initial conditions are: $x_b = 0\ m$, $y_b = 0.8\ m$, $l = 0.65\ m$,$\theta = \pi/2\ rad$, $\dot{x_b} = \dot{y_b} = \dot{l} = \dot{\theta} = 0$.

Typical simulation results are shown in **Fig. 2.5** to **Fig. 2.11**. These results are used to validate the model. These results are for in-place hopping for complete 11 hops.

Fig. 2.5 indicates the y-coordinate position of hopper body, CG and foot with respect to the time respectively. It can also be seen that, there is continuous decrease in subsequent heights due to energy loss associated with impacts.

Fig. 2.6 presents the variation of leg length with respect to time.

As the simulation is performed assuming model in vertical plane, it is observed in **Fig. 2.7** that the hopper leg has constant vertical orientation.

Fig. 2.8 shows that the total energy is decreasing subsequently in each hop, which is likely due to energy loss associated with each landing and take-off events.

Fig. 2.9 gives one of the evidence of valid model that the angular momentum is being conserved during each flight phase.

Fig. 2.10 shows the variation of leg length during one complete hop in case of the natural takeoff (in absence of mechanical stop). The justification is given in next under Test 5 caption.

Fig. 2.11 presented here to compare analytical and numerical simulation results during flight. In both cases, the robot motion follows ballistic flight motion.

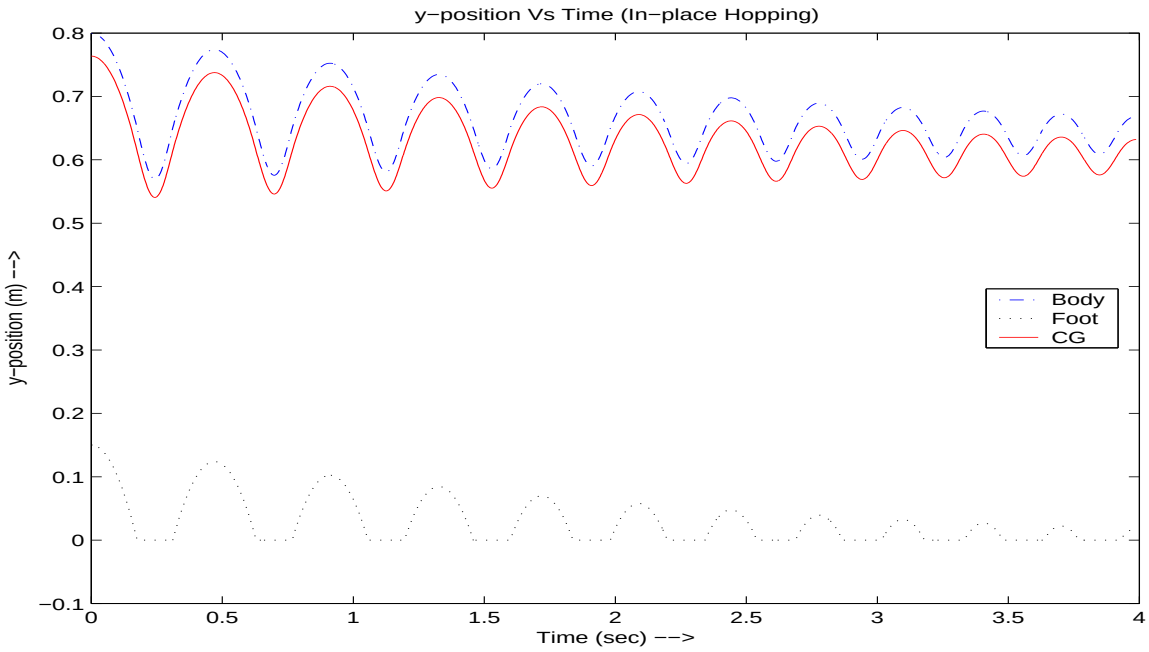

Figure 2.5: Vertical displacement (In-place hopping)

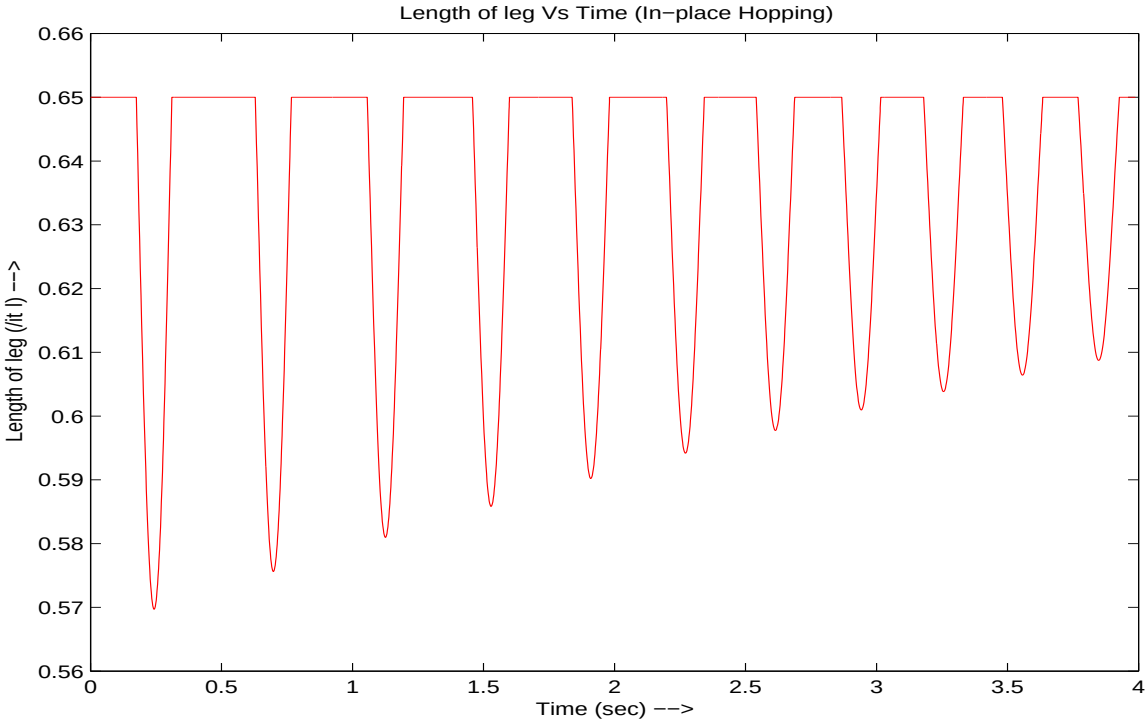

Figure 2.6: Variation in length of hopper leg (In-place hopping)

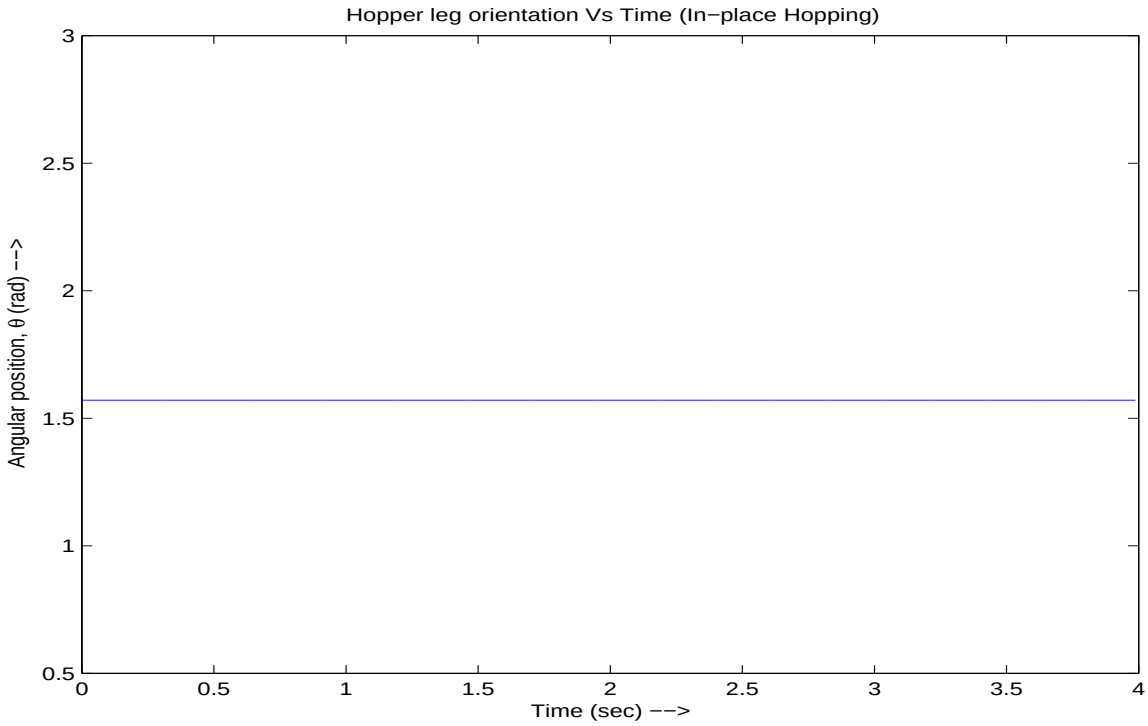

Figure 2.7: Leg orientation (In-place hopping)

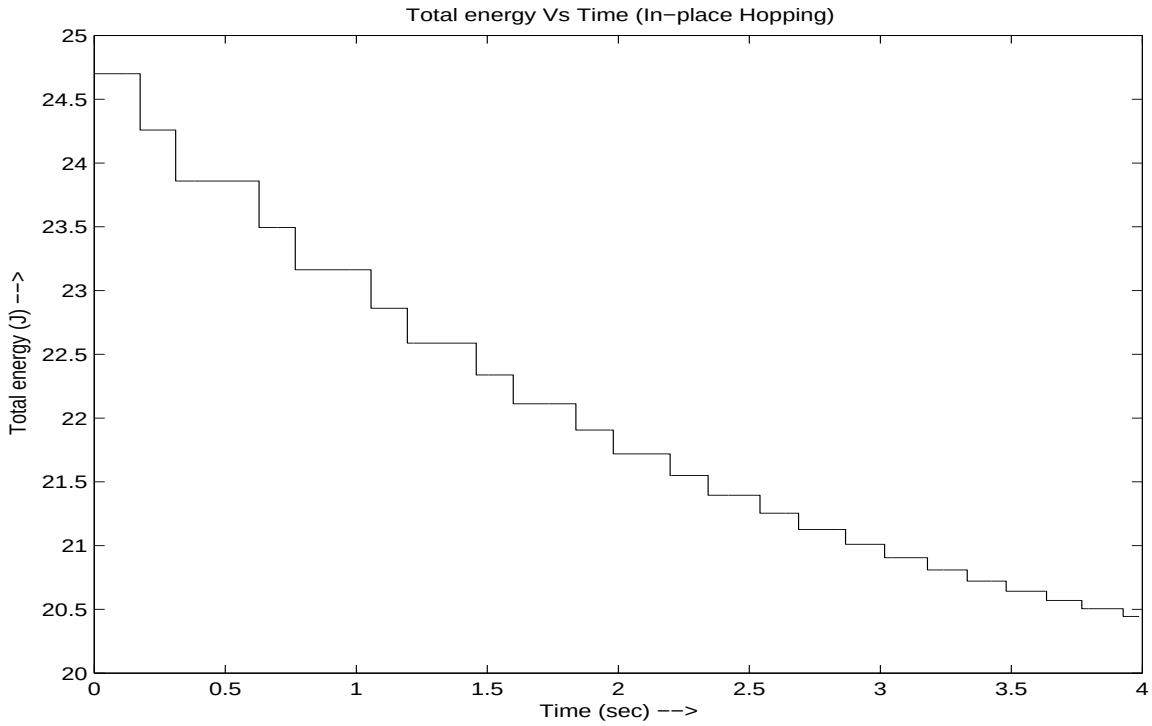

Figure 2.8: Total energy profile (In-place hopping)

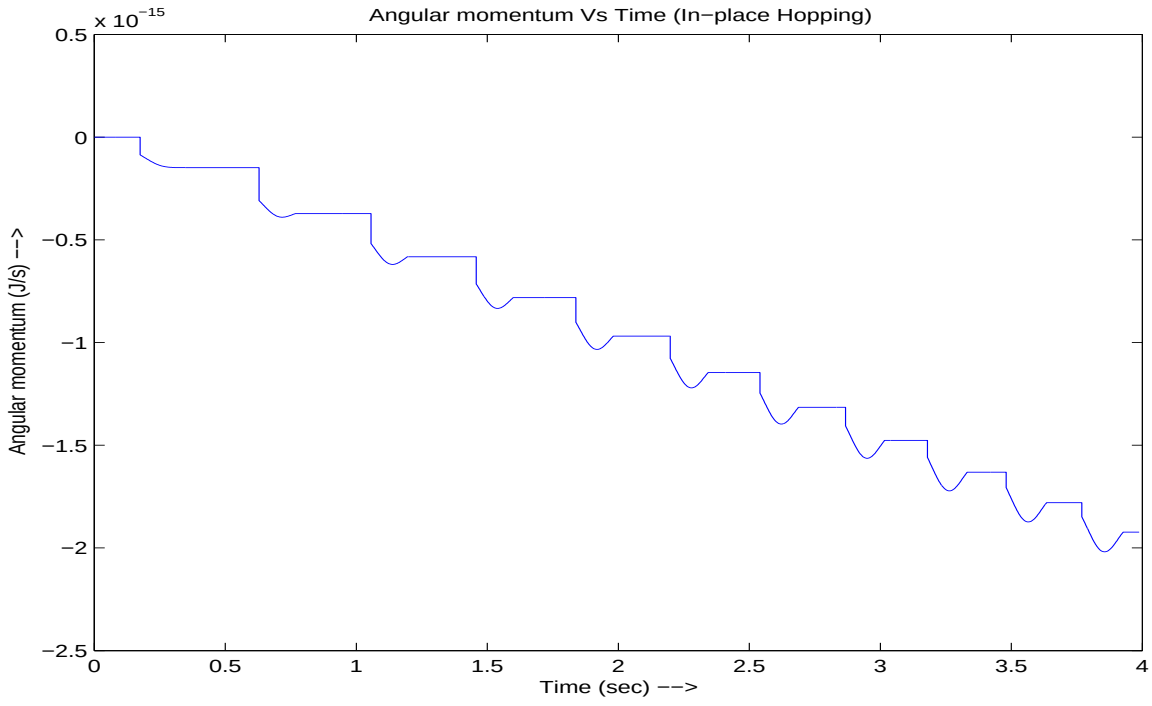

Figure 2.9: Angular momentum profile (In-place hopping)

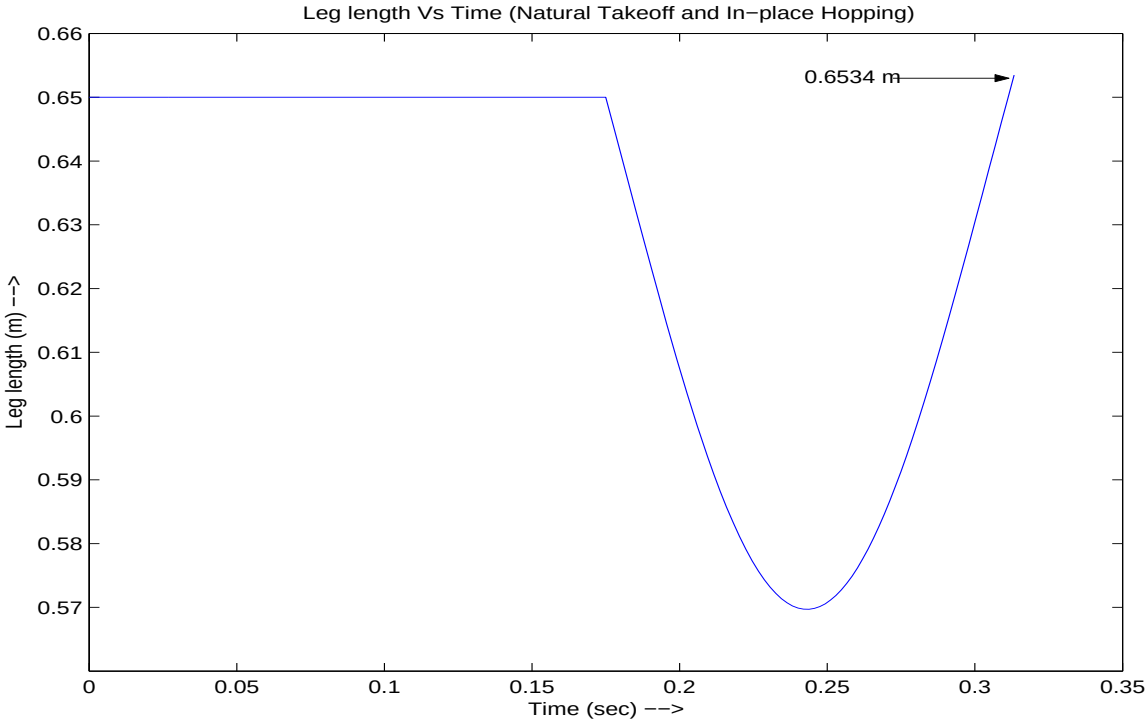

Figure 2.10: Variation in length of hopper leg due to natural takeoff (In-place hopping)

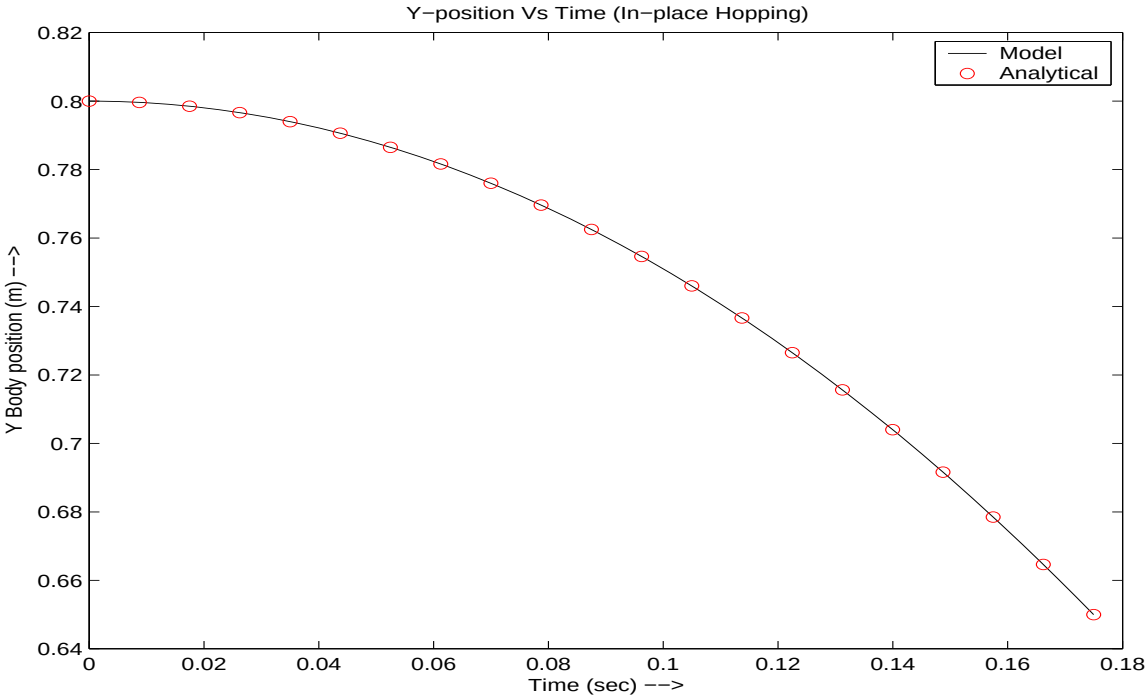

Figure 2.11: Flight trajectory (In-place hopping)

We have validated model based on the basic principles of physics.

***Test 1*:** One of the arguments in validating the model is that the total energy during the stance and flight phase should be constant in case of passive system i.e. without actuation. From **Fig. 2.8**, it is observed that the total energy during each phase remains constant. There is descending step like variation due to the energy loss associated with impacts.

***Test 2*:** It is known that hopping robot loses its energy due to landing impact. During the first flight phase it is observed that the total energy before landing is 24.7 J. While just after impact it is observed that the energy decreases to the value 24.259 J. Hence the net loss during that event is 0.441 J. During flight phase both leg and body travels with same vertical velocity. But at the event of landing the leg comes to rest and body moves with the same vertical velocity. So we can say that the energy loss is equivalent to the kinetic energy of the leg just before the impact. From the numerical data, it is noted that the velocity just before landing (same for both leg and body) is -1.7146 m/s^2. So kinetic energy of the leg just before the impact is,

$$\begin{aligned} T_{td^-} &= (1/2) \times m_l \times \dot{y}_{td^-}^2 \\ &= 0.5 \times 0.3 \times (-1.7146)^2 = 0.441J \end{aligned}$$

***Test 3*:** At the instant of take-off, the body hits the leg (since we are assuming the presence of mechanical stop and robot takes off as soon as the springy leg attains the length L_{to} during stance phase). Due to this impact the vertical velocity of body changes significantly. Just after take-off both the body and leg starts traveling in air with same velocity. We can verify such change in velocity using the law of conservation of linear momentum. From the numerical data,

Velocity just before take-off (body) = 1.7146 m/s^2

Velocity just after take-off (both) = 1.55876 m/s^2.

According to the law of conservation of linear momentum,

$$\begin{aligned} m_b \times \dot{y}_{to^-} &= (m_b + m_l) \times \dot{y}_{to^+} \\ (3.0) \times (1.7146) &= (3.0 + 0.3) \times (1.55876) \\ 5.1438 &\cong 5.1439 \end{aligned}$$

So, we can say that the result obtained from the model are consistent with the assumptions.

***Test 4*:** It is known that the rigid body should follow the parabolic trajectory during free fall in air. **Fig. 2.11** shows the result comparing the model response and the analytical data. Analytical expression for the vertical displacement during flight is,

$$y(t) = y_0 + \dot{y}_0(t - t_0) + (1/2)g(t - t_0)^2$$

where, subscript 0 stands for initial time. It is observed from the simulation result that error between the model response and analytical data is of order 10^{-8} m.

***Test 5*:** In the proposed model, we assumed that the robot takes off as soon as the body hits leg after extension phase of stance. If robot is set free from this constraint, it will take-off naturally when the vertical reaction force (R_y) becomes zero and its relative velocity ($\dot{R}_y$) is negative. In this situation it is possible that the body may stretch the spring attached to the leg i.e. there may be tensile spring force. When we simulated the model for this situation, we observed that the effective length of leg l is greater than the free length, L_0. From simulation result shown in **Fig. 2.10**, the arrow-marked value is, $l = 0.6534\ m$. Analytically, at the moment of take-off the vertical robot posture should satisfy following condition,

$$\begin{aligned} R_y = 0 \Rightarrow -m_l g + F_s = 0 \Rightarrow F_s = m_l g \\ \because F_s = k(l - L_0), \quad \therefore 2000 \times (l - 0.652) = (0.3) \times 9.8 \\ l = \frac{(0.3) \times (9.8)}{2000} + 0.652 = 0.6534\ m. \end{aligned}$$

Thus, this result also support for the validity of our assumptions.

***Test 6*:** It is known that the angular momentum is conserved during flight phase. Even though the variation in angular velocity is not significant in this case (In-place hopping), it is observed in the **Fig. 2.9** that during each flight phase the angular momentum is constant. The moment of inertia of CG is a function of length of leg and is derived as,

$$I_{CG} = I_b + I_l + \frac{m_b m_l}{m_b + m_l}\left[(l - h)^2 + d^2\right].$$

Thus, conservation of angular momentum during flight is also validated.

2.4 Periodic Motion and Definition of the Poincaré Map for Fixed Energy Input:

In dynamically stable legged robots, the motion of the system i.e. its trajectory, repeats itself periodically. A very useful and classical tool to study the existence and stability of periodic orbits is the Poincaré map, [26]. Poincaré map or return map, which, in the context of legged locomotion, is also called the *stride function*. After the initial work of Koditschek and Buehler, [27], a number of authors have used this tool to study the properties of the vertical and forward dynamics of simplified models of monopods, e.g. [28], [19], [18], [29], [5], [30], where they demonstrated emergent behaviours that corresponded to animal gaits.

An important aspect of Poincaré map is that it replaces an n^{th} order continuous-time autonomous system by an $(n-1)^{th}$ order discrete-time system (refer **Appendix B**). For a comprehensive introduction to discrete dynamical systems see [26] and [31]. In order to define the return map for a such system a reference point in the cyclic motion must be selected and then the dynamic equations must be integrated starting from that point until the next cycle.

2.4.1 The Poincaré Section:

In defining the 3-dimensional Poincaré section, we use the posture just before take-off in the stance. Since the posture just before take-off is considered the reference point, let us call it as the *standard position*, we seek a function that maps the standard position of the n^{th} stride to the standard position of the $(n+1)^{th}$ stride i.e. the return map. The states at the n^{th} standard position constitute the initial conditions for the cycle. By successively integrating forward the dynamic equations of all the phases according to the events that happen and with fixed energy pumping (retraction by an amount $\triangle s$), we calculate the value of the state vector at the $(n+1)^{th}$ standard position. If the state vector at the new standard position is identical to the original at n^{th} standard position, then the cycle is repetitive. We seek for such "re-entry" conditions.

We define the Poincaré section to be the hyperplane

$$\Sigma = \{\mathbf{x} \in R^3\} \tag{2.10}$$

where,

$$\mathbf{x} = \begin{bmatrix} \theta & \dot{l} & \dot{\theta} \end{bmatrix}^T \tag{2.11}$$

The return map is defined as a vector function $\mathbf{P} : R^3 \times R^1 \rightarrow R^3$ mapping standard position conditions from stride n to stride $n+1$,

$$\begin{bmatrix} \theta \\ \dot{l} \\ \dot{\theta} \end{bmatrix}_{n+1} = \mathbf{P} \left\{ \begin{bmatrix} \theta \\ \dot{l} \\ \dot{\theta} \end{bmatrix}_{n}, [\mathbf{u}]_n \right\} \tag{2.12}$$

The 3-dimensional closed-form return-map describing the robot's state at the next hop as a function of that at the current hop is derived. As the discussion started in terms of discrete-time space, in order to avoid confusion, the robot's state at the current hop is termed as $\mathbf{x}(k)$ and the state at the next hop is used to refer as $\mathbf{x}(k+1)$.

2.4.2 Different Phases during a Complete Hopping Cycle:

To define the Poincaré section from current standard position to next standard position, one cycle of the hopping motion is further described by the sequence of the following 7 phases as shown in **Fig. 2.12**.

Standard position: One cycle hopping motion begins and ends with the standard position. This is the posture of the robot just before the take-off, where the leg is in contact with the ground. Hence the effective length of leg is equal to L_{to}. The state variables for discrete system of k^{th} step are defined as in equation (2.11) or explicitly as,

$$\mathbf{x}(k) = \begin{bmatrix} \theta(k) & \dot{l}(k) & \dot{\theta}(k) \end{bmatrix}^T \tag{2.13}$$

Take-off Impact: At the moment the leg is fully stretched, the robot takes off and there arises an impact on the length of the leg.

Flight phase 1: We assume that the robot takes off from the ground at the moment the leg is fully stretched to length L_{to}. Sequentially, the length of the leg is again constrained to L_{to} and there is no more constraint on the position of the tip. With this constraint the robot travels in air for a period equal to 20% of estimated flight duration (based on vertical take-off velocity of body assuming forward motion).

Flight phase 2: Right after flight phase 1, the leg begins to shorten (by an amount $\triangle s$) to the desired length, hereafter referred as L_d. During this phase the length of the springy leg should follow the pre-defined trajectory as explained in the Section 2.2.3, such that it does not create impact at the moment when the leg is fully retracted to

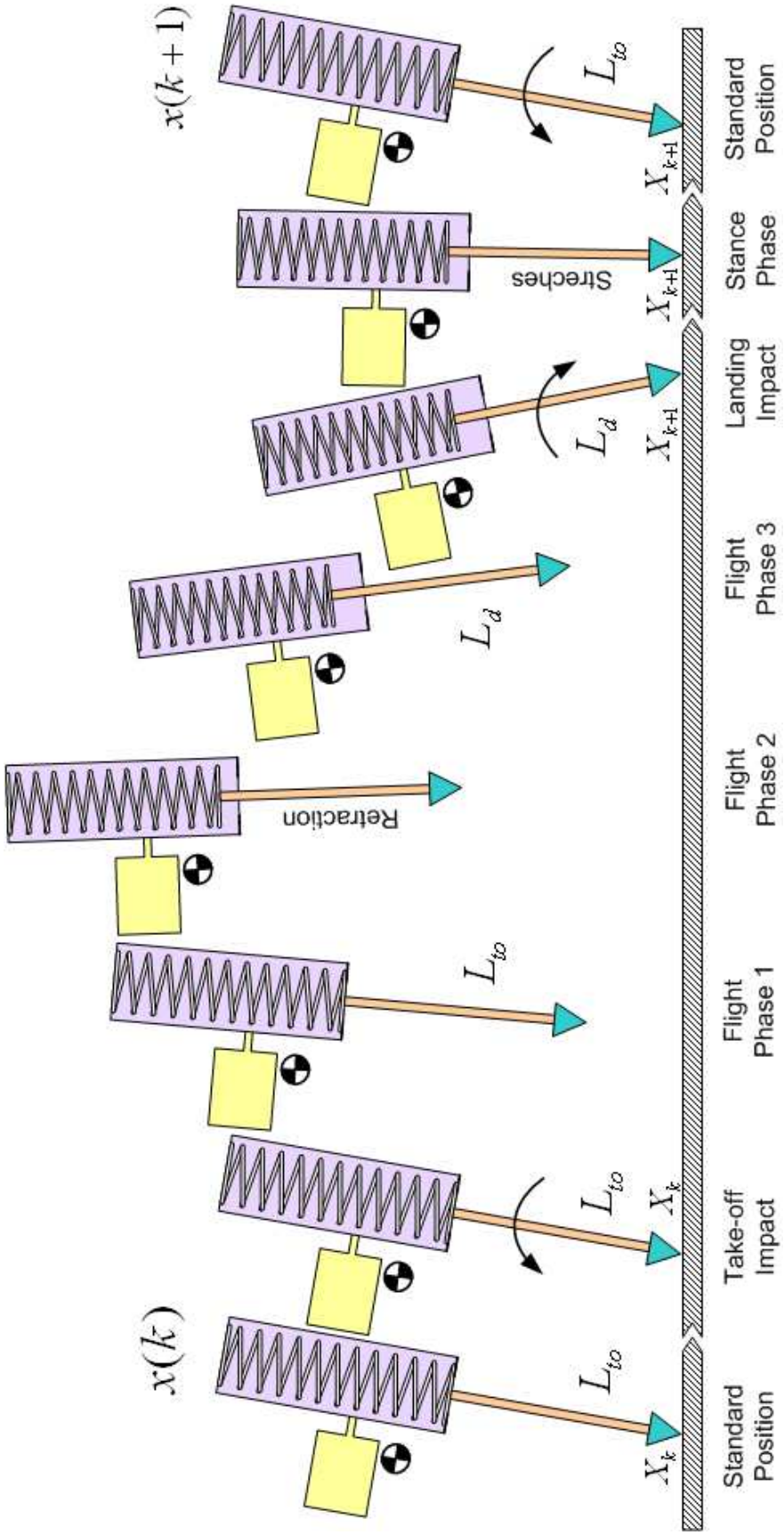

Figure 2.12: One complete hopping cycle divided into different 7 phases

desired value.

Flight phase 3: If the leg shortened to desired value, the length is again constrained to remain fixed to L_d up to landing event.

Landing Impact: When the height of the foot reaches zero, an impact occurs on the foot. The constraint is on the position of the foot (foot tip arrested to ground).

Stance phase: After landing, the position of the foot is constrained. The leg experiences the compression and extension phases. The body pivots on the tip with a varying angular velocity $\dot{\theta}$. The leg stretches until it is fully stretched to the maximum length L_{to}. Therefore, there is no constraint on the length of the leg, but is only on the position of the foot. We assume that all the impacts are inelastic. Just before the take-off, the robot again returns to the standard position.

In summary, the state of the system is $\mathbf{x}(k) = \begin{bmatrix} \theta(k) & \dot{l}(k) & \dot{\theta}(k) \end{bmatrix}^T$ at k^{th} step and the input to the system is the magnitude of retraction of the leg during flight phase 2 as $\triangle s$. Let, us call this as input $\mathbf{u}(k)$.

2.4.3 Search for Fixed Point:

To determine one of the periodic motion, we search for fixed points of the function $\mathbf{P} : R^3 \times R^1 \rightarrow R^3$ as expressed in equation (2.12). Searching the fixed points means we want to find an argument $\mathbf{x}$ of equation (2.12) that maps onto itself, i.e. we want to solve the equation

$$\mathbf{F}(\mathbf{x}) \triangleq \mathbf{x} - \mathbf{P}(\mathbf{x}) = 0 \tag{2.14}$$

To calculate an individual solution one has to specify the values of $\triangle s$ along with the values of the model parameters and solve equation (2.14). To describe $\mathbf{P}$ as a nonlinear function by analytically integrating the dynamic equations over this space is complex, so the search will be conducted numerically. We will use the *quasi-Newton-Raphson* method (fsolve utility of MatLab©), where an initial guess for the fixed point is given and then updated based on the gradient matrix (Jacobian) of the return map.

2.5 Fixed Points Obtained and their Simulation:

Simulation results presented in this chapter used the SLOM hopper specification given in Table 2.1. For different values of $\triangle s$, we numerically found the fixed points using the in-built "fsolve" utility of MatLab©, with the termination tolerances set to 10^{-16} for both $\mathbf{x}$ and $\mathbf{F}$. Table 2.4 shows these fixed points, i.e. the initial desired state, $\overline{\mathbf{x}} = \left[\begin{array}{ccc} \overline{\theta}, & \overline{\dot{l}}, & \overline{\dot{\theta}} \end{array} \right]^T$, necessary for existence of the periodic motion. We simulated model setting initial conditions at fixed point obtained for different spring retraction, $\triangle s = [0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15]$ m.

Table 2.3: Various fixed points at different spring shortenings

Spring Retraction, $\triangle s$ (m)	$\overline{\theta}$ (rad)	$\overline{\dot{l}}$ (m/s)	$\overline{\dot{\theta}}$ (rad/s)
0.01	1.41944	0.61390	0.05168
0.02	1.41488	1.19596	0.07931
0.03	1.40994	1.77437	0.10261
0.04	1.40472	2.34894	0.12440
0.05	**1.39924**	**2.91937**	**0.14544**
0.06	1.39350	3.48536	0.16603
0.07	1.38751	4.04657	0.18630
0.08	1.38127	4.60265	0.20631
0.09	1.37478	5.15322	0.22607
0.10	1.36803	5.69787	0.24562
0.11	1.36103	6.23618	0.26493
0.12	1.35379	6.76769	0.28400
0.13	1.34631	7.29191	0.30281
0.14	1.33860	7.80835	0.32136
0.15	1.33068	8.31645	0.33961

The hopper body and foot trajectories for 2 subsequent hops are shown in **Fig. 2.13** and **Fig. 2.14** respectively. We observed that horizontal and vertical distance covered during a hop and flight duration increases as magnitude of $\triangle s$ increases. Whereas, stance duration decreases as magnitude of $\triangle s$ increases. It is not desirable to have larger hopping height as it produces larger energy losses and in result requires more energy feed. Also, this may damage prototype. It is well known that the flight duration

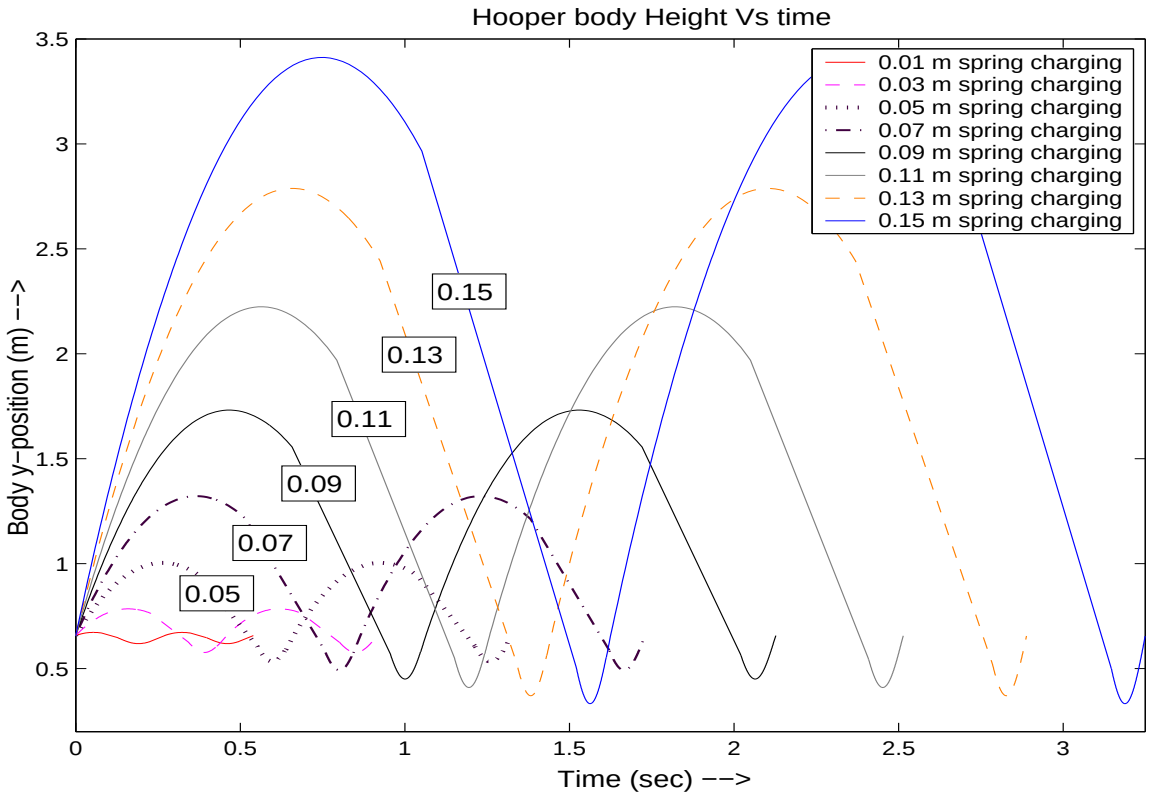

Figure 2.13: Hopper body vertical displacement (Various fixed points)

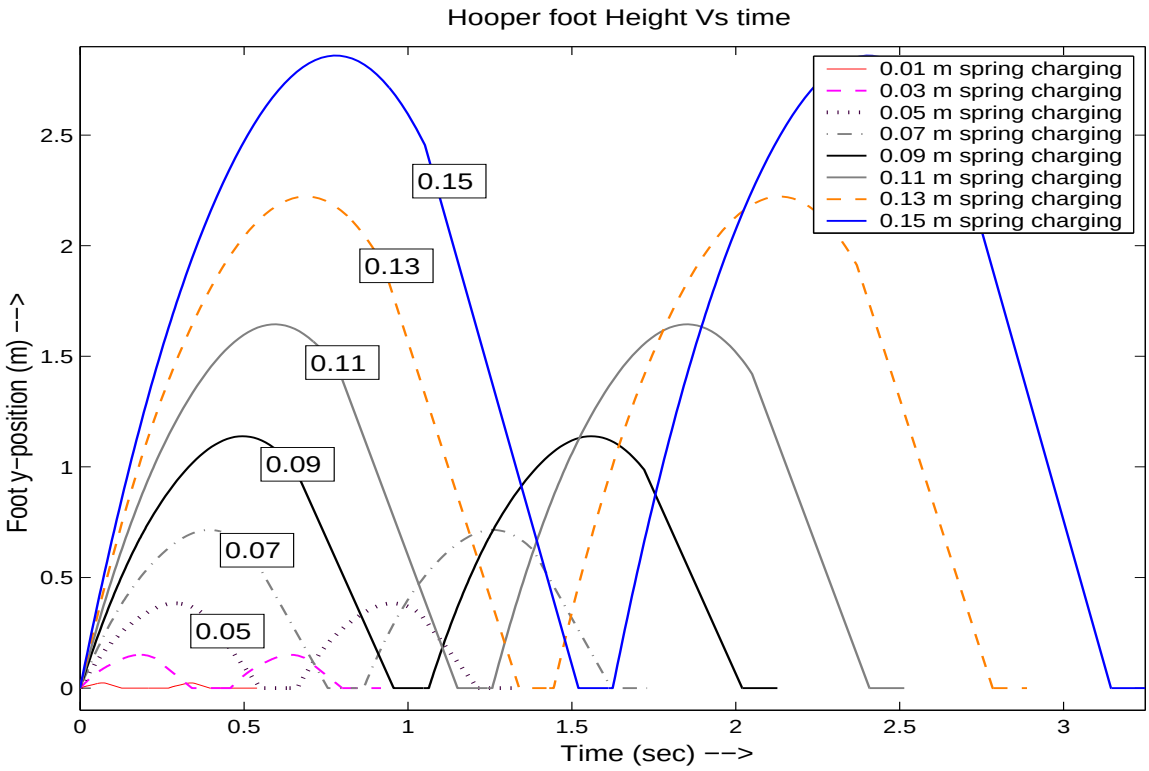

Figure 2.14: Hopper foot vertical displacement (Various fixed points)

should be 4 to 5 times larger than the stance duration. In the case of $\triangle s = 0.05$ m, we found that the flight duration is about 5 times larger than the stance duration. Also, the hopping height of foot is moderate, about 0.4 m. Considering all this aspects, we finally chosen fixed point obtained at $\triangle s = 0.05$ m for further investigation of proposed hopping model. **Fig. 2.15** to **Fig. 2.19** are reported here justifies the sustained motion for 6 hops, initiated with the fixed point obtained at $\triangle s = 0.05$ m (highlighted in Table 2.4).

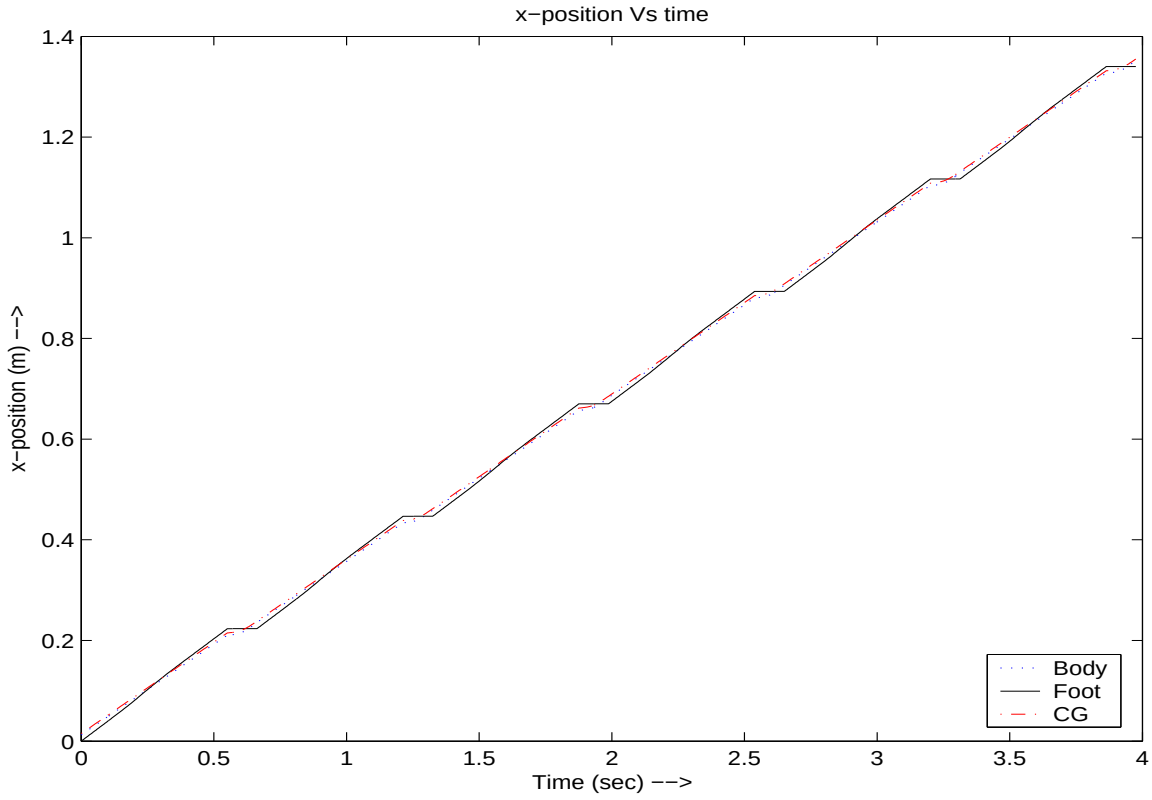

Figure 2.15: Horizontal displacement (Fixed point, $\triangle s = 0.05$ m)

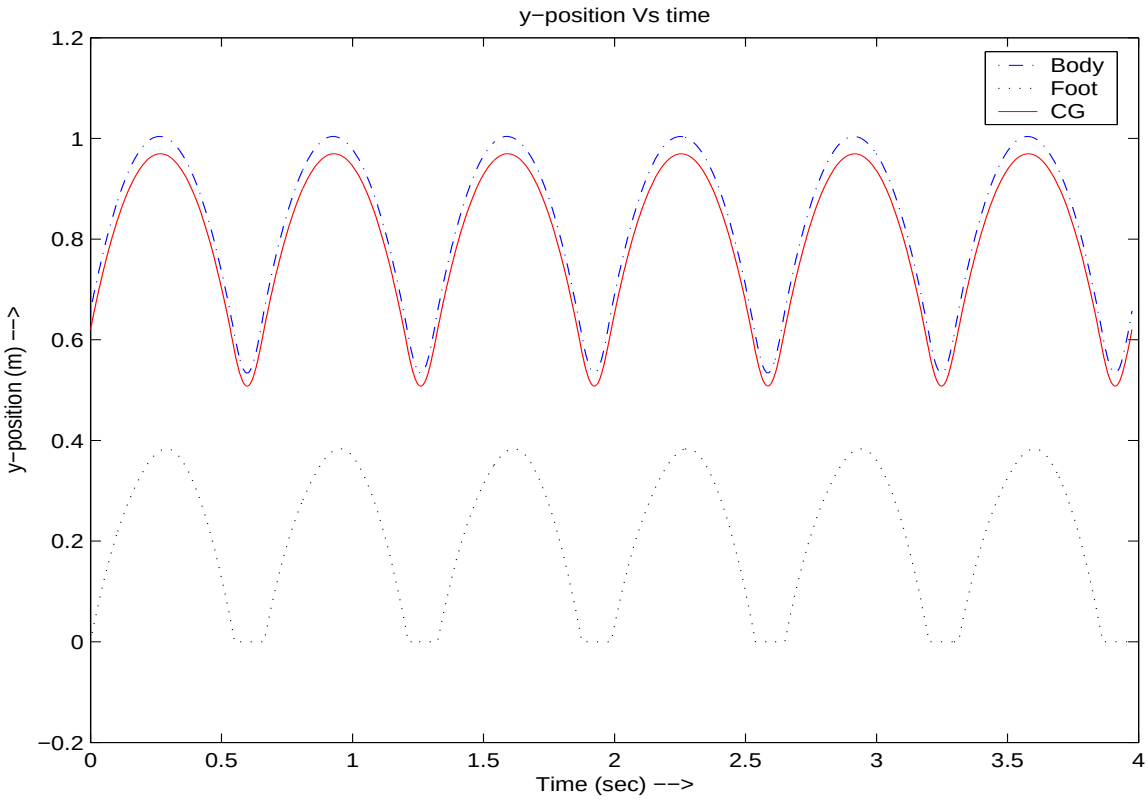

Figure 2.16: Vertical displacement (Fixed point, $\triangle s = 0.05$ m)

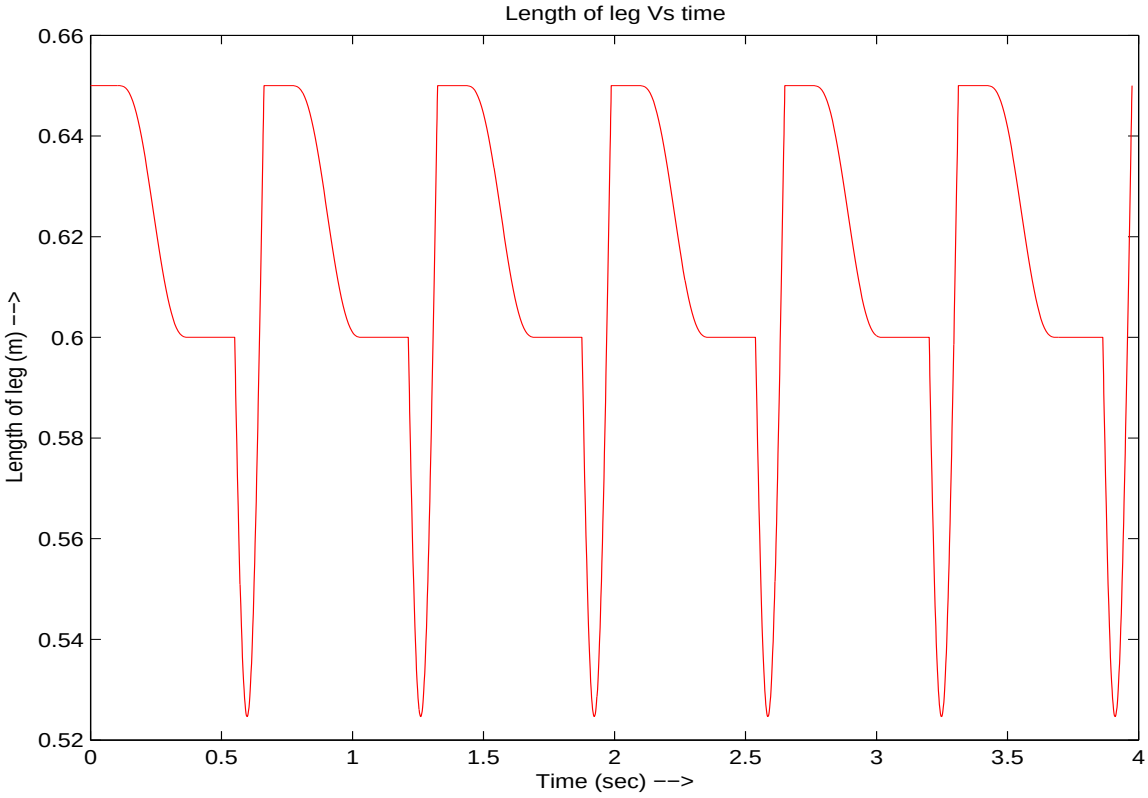

Figure 2.17: Length of hopper leg, l (Fixed point, $\triangle s = 0.05$ m)

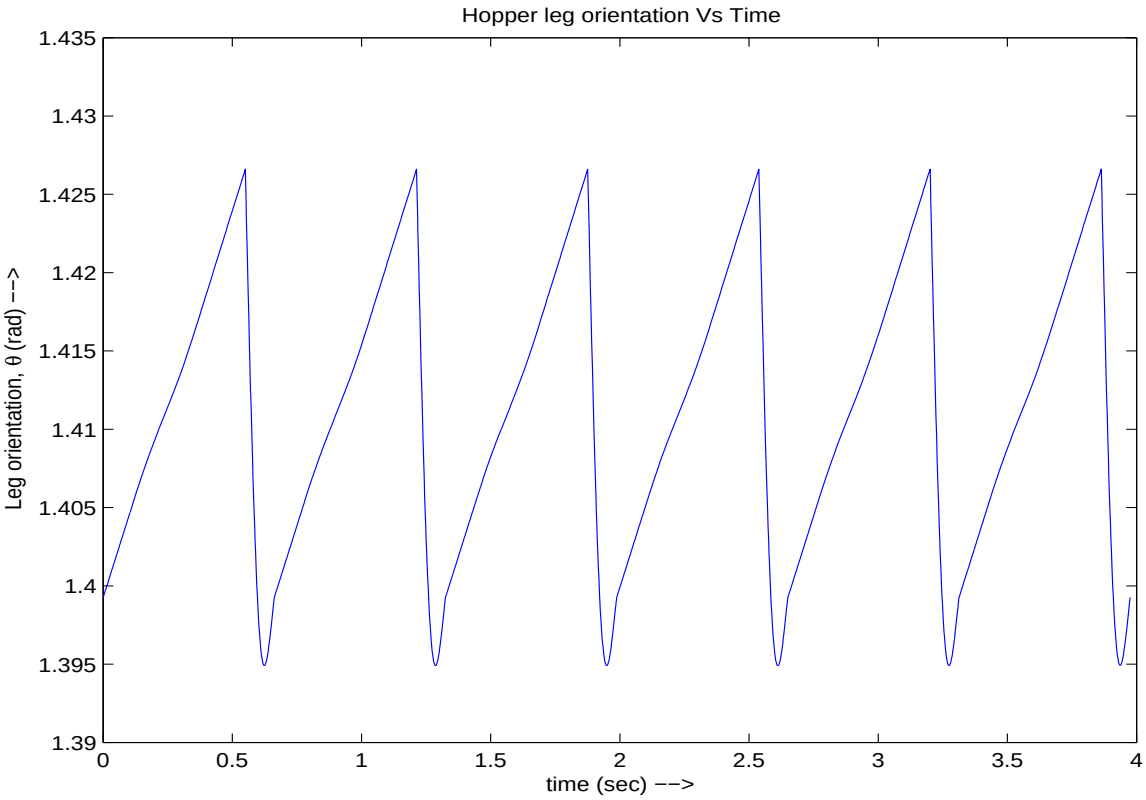

Figure 2.18: Leg orientation, θ (Fixed point, $\triangle s = 0.05$ m)

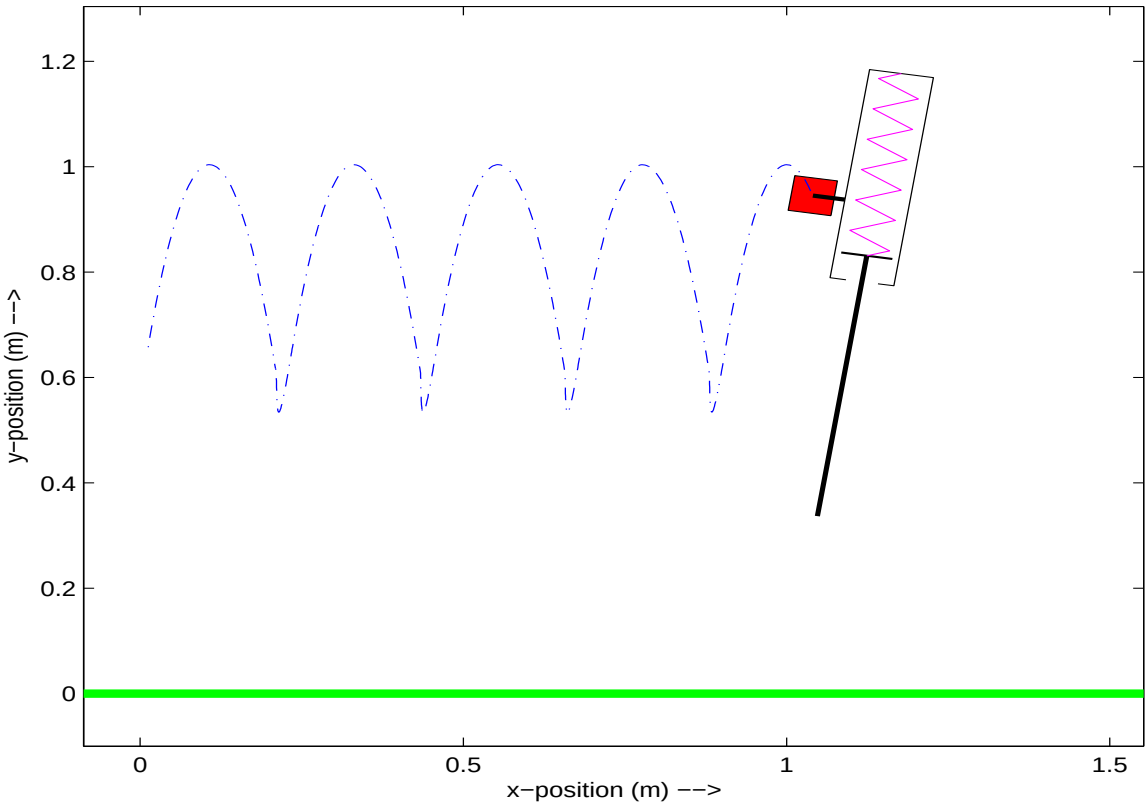

Figure 2.19: Animated posture of hopping robot (Fixed point, $\triangle s = 0.05$ m)

2.6 Stability of the Fixed Point:

The existence of the periodic hopping motion in above simulation results is by itself a very important result since it shows that a energy compensated system initiated with fixed point values $\overline{\mathbf{x}}$ turns to a simply a natural motion of the system. However, in real situations the robot is continuously perturbed, therefore, if the fixed point were unstable, then the periodic motion would not be practically sustainable. It would therefore be important to study the stability properties of the fixed point found above and to design controllers to improve the robustness of the system against perturbations.

In this section, we characterise the stability of one fixed point using local stability analysis i.e. using the eigenvalues of the linearised return map. The stability analysis is based on linearising the nonlinear map about a fixed point. A set of linearised equations specifies how a deviation on the steady cycle propagates from one cycle to the next. We will formulate this set of linearised equations by deviating the system from fixed point or nominal point and hereafter referred as *linear deviational model*. The problem of studying the stability properties of a periodic solution of a hybrid system (like the hopper having continuous stance and flight phases and intermittent discrete impacts) is thus reduced to the problem of studying the stability of the periodic points of the Poincaré map.

2.6.1 Linear Deviational Model:

In order to formulate the linear deviational model, to investigate stability, we assume that the *Standard position* states are deviated from their steady-cycle values $\overline{\mathbf{x}}$, by some small amount $\triangle\mathbf{x}$. The model that relates the deviations from steady state, i.e. the *deviational* model, formulated using *Taylor's series approximation* and defining gradient matrix (Jacobian), is

$$\triangle\mathbf{x}_{k+1} = \left.\frac{\partial\mathbf{P}(\mathbf{x},\mathbf{u})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\overline{\mathbf{x}},\mathbf{u}=\overline{\mathbf{u}}}\triangle\mathbf{x}_k + \left.\frac{\partial\mathbf{P}(\mathbf{x},\mathbf{u})}{\partial\mathbf{u}}\right|_{\mathbf{x}=\overline{\mathbf{x}},\mathbf{u}=\overline{\mathbf{u}}}\triangle\mathbf{u}_k \tag{2.15}$$

$$\triangle\mathbf{x}_{k+1} = \mathbf{A}\triangle\mathbf{x}_k + \mathbf{B}\triangle\mathbf{u}_k \tag{2.16}$$

where, $\mathbf{P}$ is the Poincaré map and $\triangle\mathbf{x} = \mathbf{x} - \overline{\mathbf{x}}$, $\triangle\mathbf{u} = \mathbf{u} - \overline{\mathbf{u}}$. For small deviations, the standard position states at the next hop can be calculated by equation (2.16), which is a linear difference equation.

More explicitly, by referring equation (2.14), in our case we can write,

$$\mathbf{A} = \begin{bmatrix} \dfrac{\partial\mathbf{P}}{\partial x_1} & \dfrac{\partial\mathbf{P}}{\partial x_2} & \dfrac{\partial\mathbf{P}}{\partial x_3} \end{bmatrix} \tag{2.17}$$

$$\mathbf{B} = \left[\frac{\partial \mathbf{P}}{\partial u}\right] \tag{2.18}$$

where, x_i are the state variables and u, is the control input (the spring retraction, $\triangle s$). To calculate the components $\partial \mathbf{P}/\partial x_i$ of the gradient matrix, we need to evaluate $\mathbf{P}$ at $\overline{\mathbf{x}} - d\mathbf{x}$ (fore of the nominal point) and at $\overline{\mathbf{x}} + d\mathbf{x}$ (aft of the nominal point), where $d\mathbf{x}$ is referred as the "linear band" and is obtained by deviating each of components of $\mathbf{x}_i$, $i = 1, 2, 3$, and $\mathbf{u}$ (single input) by some small scalar quantity. Hence we have the linear band comprised of four elements as dx_1, dx_2, dx_3, du.

The next Section described the method used to determine the linear band in our study. Then central difference approximation is used to evaluate numerically the corresponding derivatives, e.g. for some component x_2 of the state vector $\mathbf{x}$ we have,

$$\frac{\partial \mathbf{P}}{\partial x_2} = \frac{\mathbf{P}(x_1, x_2 + dx_2, dx_3, u) - \mathbf{P}(x_1, x_2 - dx_2, dx_3, u)}{2dx_2} \tag{2.19}$$

2.6.2 Linear Band Selection:

It is observed that the generally the linear band selected based on the criteria that the deviation in each state variable is very small and as good as in the range of relative tolerance specified for numerical selection. For example, $dx = 1e^{-6}$ is mostly specified deviation value [5]. This Section describes the procedure adapted by us to select the linear band.

We selected one of the state variables, say x_i to determine the deviation dx_i. Then, for the range of $-1e^{-6}$ to $1e^{-6}$ variation ($\pm dx_i$) from the nominal value, we numerically calculated $\mathbf{P}$. So, obviously we have 3 state behaviour each for x_1, x_2, x_3 during each selection. As shown in **Fig. 2.20**, let us assume that the state variable 2, x_2, varies due to the ($\pm dx_1$) variation from the nominal point. Then, we have selected the effective linear band dx_{12} such that the slope B/C of the dotted line will be constant thereafter for further small deviation in dx_1. Here dx_{12} refers as deviation in state variable 1 which affects the state variable 2. The quantity B/C (slope) is analogous to the term $\partial \mathbf{P}/\partial x_i$ in equation (2.19). The quantity B is the difference of x_i's value at fore and aft of nominal point. It should be noted that $C = 2dx_i$. **Fig. 2.21** illustrates how the slope B/C changes with respect to dx_1. For example, here in state variable 2, further less deviation than 0.002 in variable 1, doesn't make any significant change in value of the state variable 2, means it remains constant further. Likewise we can determine dx_{11} and dx_{13}. The same procedure is repeated for all state variables and control input. So, apparently we have 3 different values of dx_i during each selection of the i^{th} variable.

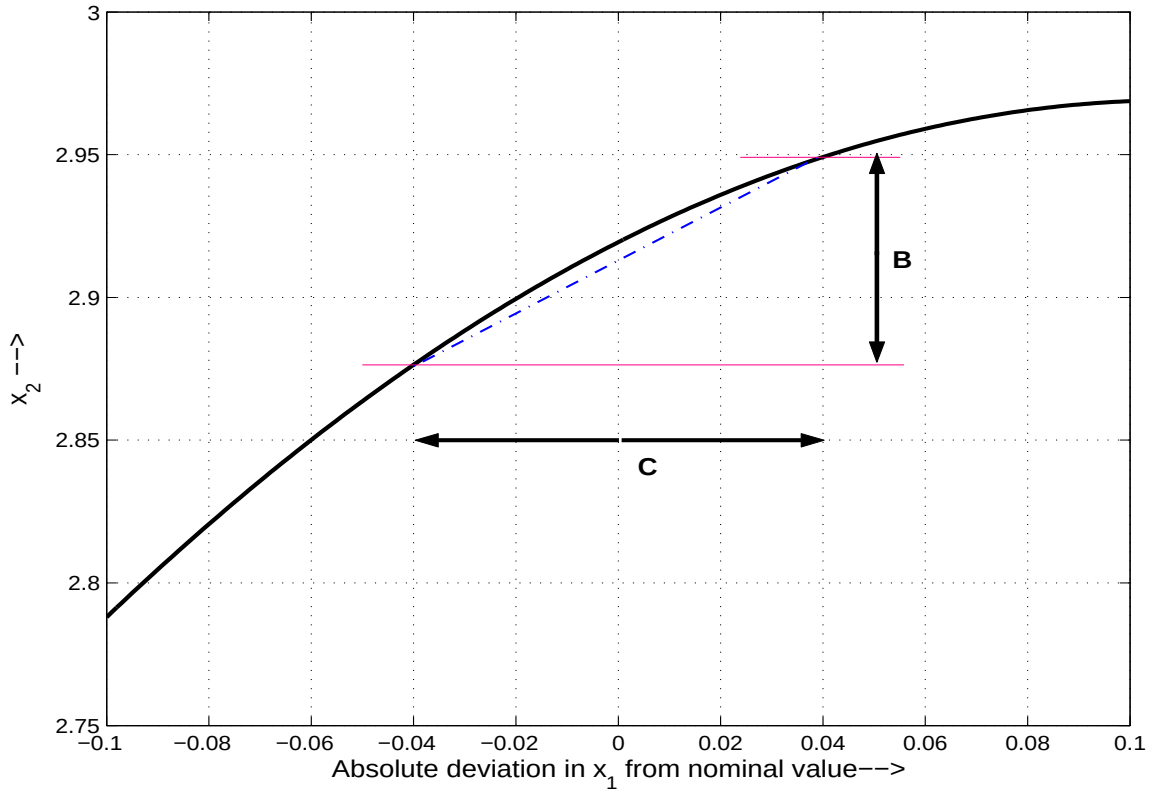

Figure 2.20: Pictorial illustration of linear band selection

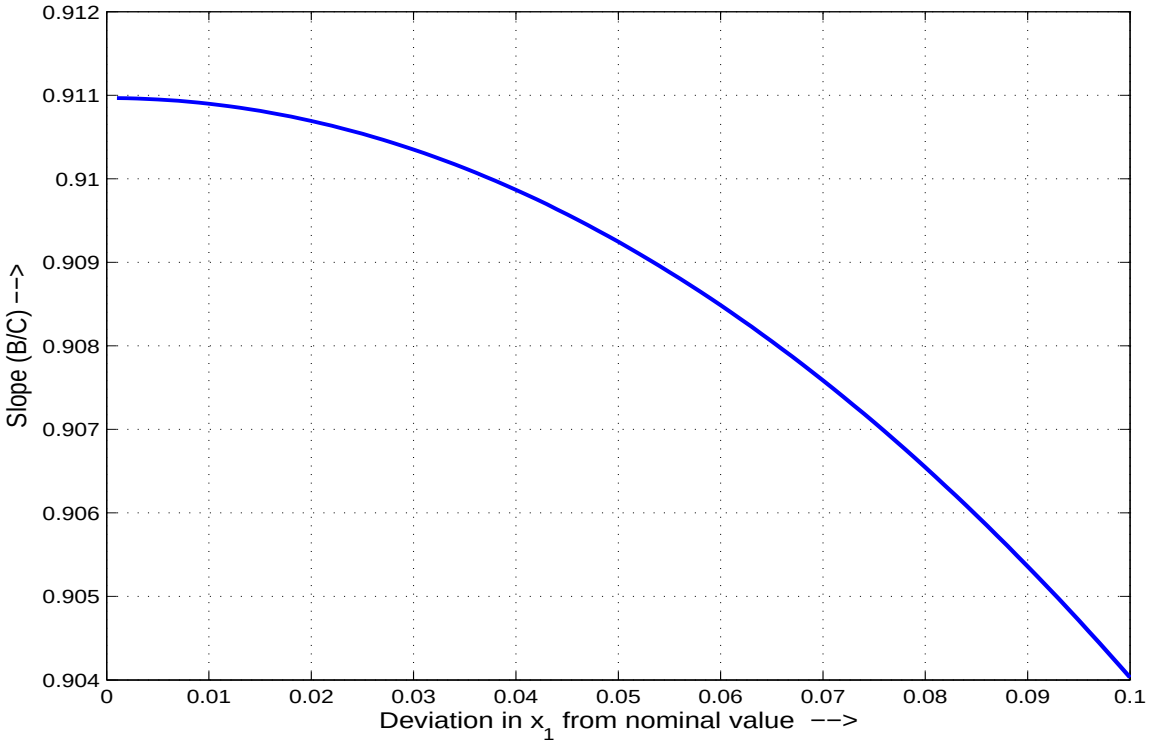

Figure 2.21: Slope B/C variation with respect to the $\pm dx_i$

We have selected the largest out of these 3 values in order to finalize the value of dx_i, in context with linear band selection in the i^{th} variable.

After selecting the linear band values, we can determine the state matrix **A** and the control vector **B** for the linear deviational model using equation (2.17) and equation (2.18). If all the eigenvalues of the system matrix **A** have magnitude less than one, then the periodic solution is stable and disturbances decay in subsequent steps. If not, then disturbances grow and eventually repetitive motion is lost. The next Section includes numerical results which will be useful for further stability analysis.

2.6.3 Numerical Results:

To simulate the proposed model, integrations of the equations of motion use the adaptive step Dormand-Price integration method (MatLab©'s `ode45` function) with $1e^{-6}$ as relative and absolute tolerances. It is worth mentioning that MatLab© offers a very useful feature for event-based integration of differential equations. But, we found some difficulty to detect the end of the stance phase so used the "*interp1*" utility of MatLab© which helps in detecting the event by interpolating the variables of interest.

The Table 2.4 lists the deviation $d\mathbf{x}$ (with appropriate units) for linearising the nonlinear map about a fixed point at $\triangle s = 0.05$ m (refer Table 2.2).

Table 2.4: Linear band selection ($d\mathbf{x}$)

Variable	Individual Value dx_i		Final $d\mathbf{x}$
x_1	dx_{11}	0.2×10^{-4}	1.2×10^{-4}
	dx_{12}	1.2×10^{-4}	
	dx_{13}	0.2×10^{-4}	
x_2	dx_{21}	2.0×10^{-4}	2.0×10^{-4}
	dx_{22}	2.0×10^{-4}	
	dx_{23}	2.0×10^{-4}	
x_3	dx_{31}	2.0×10^{-4}	10×10^{-4}
	dx_{32}	10×10^{-4}	
	dx_{33}	2.0×10^{-4}	
u	dx_{u1}	0.2×10^{-4}	0.2×10^{-4}
	dx_{u2}	0.1×10^{-4}	
	dx_{u3}	0.1×10^{-4}	

With the help of equation (2.17)-(2.19) and the final $d\mathbf{x}$ in Table 2.4, we have,

$$\mathbf{A} = \begin{bmatrix} 1.7115 & -0.0325 & 0.9007 \\ 0.9110 & 0.7947 & 0.2425 \\ 6.2283 & -0.0661 & 3.0436 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0.3743 \\ 11.6734 \\ 3.0052 \end{bmatrix} \tag{2.20}$$

The eigenvalues of state matrix $\mathbf{A}$ are,

$$\lambda_1 = 4.8274, \qquad \lambda_2 = -0.0787, \qquad \lambda_3 = 0.8010$$

The behaviour of the these three eigenvalues for different set of linear bands including the current finalized $d\mathbf{x}$ is explored. The Table 2.5 lists the different cases tried to evaluate the eigenvalues at different deviations $d\mathbf{x}$ about a fixed point. Case 6 duplicates our current finalized $d\mathbf{x}$ listed in Table 2.4. The variation in each eigenvalue is tabulated in Table 2.6. Hence, it is clear that these eigenvalues are insensitive to said variation in deviations. The main objective of this discussion is to emphasize that our linear band selection is appropriate.

Table 2.5: Set of deviations selected to evaluate eigenvalue sensitivity

Case	dx_1	dx_2	dx_3	du
	$\times 10^{-4}$	$\times 10^{-4}$	$\times 10^{-4}$	$\times 10^{-4}$
1	0.2	1.0	5.0	0.05
2	0.4	1.2	6.0	0.08
3	0.6	1.4	7.0	0.11
4	0.8	1.6	8.0	0.14
5	1.0	1.8	9.0	0.17
6	**1.2**	**2.0**	**10.0**	**0.20**
7	1.4	2.2	11.0	0.23
8	1.6	2.4	12.0	0.26
9	1.8	2.6	13.0	0.29
10	2.0	2.8	14.0	0.32
11	2.2	3.0	15.0	0.35

Table 2.7 lists the set of eigenvalues determined for various fixed points. From these eigenvalues of matrix $\mathbf{A}$, it is easy to see that one of the eigenvalues (λ_1) is outside the unit circle. This ensures that the proposed energy compensated system is unstable. Therefore, the periodic motion would not be sustainable in practice without

Table 2.6: Eigenvalue sensitivity

Eigenvalue	**Relative Variation from the central case (6) in Table 2.4**
λ_1	+3.994 $\times 10^{-12}$ to -8.347$\times 10^{-13}$
λ_2	+6.188$\times 10^{-8}$ to -8.389$\times 10^{-8}$
λ_3	+3.210$\times 10^{-9}$ to -7.988$\times 10^{-10}$

feedback. Note that from a control point of view two things come in to picture: First issue is to design the feedback law that can change the position of this eigenvalue, to make the system conservative. Another issue would therefore be important to design controllers to improve the robustness of the system against deviations. These issues are discussed in the next Chapter. Another aspect to be noted that the proposed system is completely state controllable, since it is found that, rank $\left([\ \mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B}\ ] \right) = 3$.

Table 2.7: Eigenvalues of **A** at different spring shortenings

Spring Shortening, $\triangle s$ (m)	λ_1	λ_2	λ_3
0.01	2.22296	0.41410	0.80379
0.02	2.72507	0.27021	0.80370
0.03	3.34688	0.13563	0.80300
0.04	4.05128	0.01966	0.80206
0.05	**4.82735**	**-0.07869**	**0.80096**
0.06	5.66859	-0.16194	0.79975
0.07	6.56926	-0.23254	0.79843
0.08	7.52346	-0.29266	0.79702
0.09	8.52493	-0.34411	0.79549
0.10	9.56699	-0.38835	0.79387
0.11	10.64256	-0.42659	0.79213
0.12	11.74425	-0.45981	0.79026
0.13	12.86440	-0.48881	0.78828
0.14	13.99513	-0.51425	0.78616
0.15	15.12842	-0.53666	0.78389

2.7 Conclusions:

In this Chapter, we described the proposed planar one-legged asymmetrical hopping model under investigation. Such hopping systems are highly nonlinear and only dynamically stable. There is significant energy loss due to intermittent impacts and friction. The 'more' specific kinematic model was presented to understand the kinetics of proposed SLOM hopping model. The mathematical model of the SLOM hopping robot is derived using Euler-Lagrangian formulation for un-actuated and a single DOF actuated (linear) configuration. Based on basic principles of physics, we validated this mathematical model.

The energy compensated system (fixed spring charging) is presented to seek cyclic motion. To find the "re-entry" conditions, a 3-dimensional return map describing the hopping gait is defined. In doing so, we pumped the fixed amount of energy during each hopping cycle. The minimalist approach was adopted in this system, by considering (i) retraction of the springy leg as a control or forced input and (ii) the system to be a discrete-time system, in which one cycle motion is regarded as one sampling interval.

Then a fixed point searching procedure based on the *Quasi-Newton-Raphson* algorithm was implemented to find initial conditions, which result in cyclic motion. After identifying different fixed points, one could analyze the cyclic motion of the system subjected to these fixed points. Based on these motions, we finalized the fixed point for further investigation.

Furthermore, local stability analysis showed that although the obtained periodic motion is unstable, the linear deviational model is controllable, which is essential to obtain stable regions. These local stability study show a preference towards increased control action to achieve the sustained motion in presence of deviations.

Chapter 3

State Feedback Control of SLOM Hopper with a Linear Actuator

3.1 Necessity of Active SLOM Hopper:

The existence of periodic hopping motion using energy compensating strategy (with fixed energy input) shows that such system initiated with fixed point values $\overline{\mathbf{x}}$ turns to a simply a natural motion of the system. So, such motion is dependent on the initial conditions. However, in real situations the robot is continuously perturbed, therefore, if the fixed points were unstable, then the periodic motion would not be sustainable in practice. Also, the system is not unactuated, hence the lost energy can be recouped.

Furthermore, local stability analysis showed that eventhough the proposed model is unstable, it is controllable. So, we can design a linear controller to achieve the sustained motion even in the presence of perturbations. Controlling the dynamic motion of hopper is also important in order to maintain desired hopping height, forward speed, upright posture position. The investigation of the SLOM model in the current Chapter focuses only on controllers based on the linear actuated model. As explained previously, the role of the linear actuator is to retract the spring to shape the energy flow during locomotion. Also, it is possible that both the orientation of the body and the height of a jump could be controlled by only one actuator. The actuator force (spring defection) affected the orientation of the body by a moment of force that arose due to the mass of the body and the offset of the leg.

3.1.1 Closed-loop Control System:

This Chapter focuses on a closed-loop control system known as "state feedback" (SFB) control system which steers the states of hopping system with the help of a measurement of the available states. Our core intention is to design the controller for approximate linear deviational (incremental) model, represented by equation (2.16), and then extend it to *actual* hopper model (non-linear model). The schematic diagram is shown in **Fig. 3.1**.

Let us assume that the initial deviation $\triangle \mathbf{x}_0$ is known. Let us refer it as $\triangle \mathbf{x}(k)$. Then, we can compute the hopping state at the *standard position*, as $\mathbf{x}(k) = \bar{\mathbf{x}} + \triangle \mathbf{x}(k)$. It is important to note that $\bar{\mathbf{x}}$ refers to the value determined at fixed point (refer Chapter 1, Table 2.2). Then the general controller algorithm executed for simulating the state feedback control of actual hooper model is as given below:

- **Step 1.**: The control input $\triangle \mathbf{u}(k) = \ldots$ is determined based on the control law selected and utilizing the state information, $\triangle \mathbf{x}(k)$.
- **Step 2.**: The control input for real system is $\mathbf{u} = \bar{\mathbf{u}} + \triangle \mathbf{u}(k)$.
- **Step 3.**: The state vector $\mathbf{x}$ and the control input $\mathbf{u}$ for actual SLOM model is known so integrate the natural dynamic equations until the next standard position is arrived, say $\mathbf{x}_{k+1}$.
- **Step 4.**: Then, compute the new deviation from nominal point as $\triangle \mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \bar{\mathbf{x}}$ and go to **step 1** to repeat the procedure.

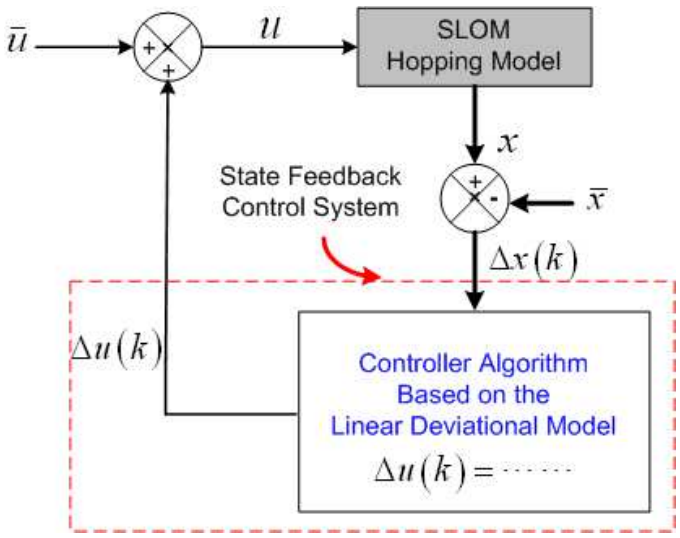

Figure 3.1: Schematic diagram of feedback control system

3.1.2 Performance Evaluation Strategy:

We are looking for global stability analysis of the different feedback control systems. This will help to define the region of attraction of initial deviations for which the system may be stabilized. It is known that, a *cell-to cell mapping* is one of the mathematical tool for analysing the strongly non-linear systems under periodic excitations [32], [33]. Using this tool, we can study effectively bifurcation phenomenon, attractors, basin of attraction, nature of fixed points, etc. But, it is also known that: (i) the efficiency of this tool is greatly dependent on the fine scaling allowed to use in specifying the state variables, (ii) depending upon the parameter variations, the *contiguous cells* may be turn to be neutral or even new *contiguous cells* may arise with the same period or different period [32] and (iii) the analysis is complex and time consuming if the simulation time of one periodic cycle is sufficiently large and finer scaling of sate variables is used. So, in this investigation, we used a simple performance evaluation strategy to determine the efficiency of each controller to define the region of attraction of initial deviations.

We simulated each feedback control system subjected to the four type of structured disturbances. The disturbance incorporated is a step of different vertical height (H_{step}) coming across the system's motion. It may be noted that the step disturbance is sufficient to capture the affect of ground unevenness. The hopper is initiated with the state at the fixed point. These step disturbances are referred as:

1. *Upside as normal* (UAN) step disturbance (**Fig. 3.2**)
2. *Downside as normal* (DAN) step disturbance (**Fig. 3.3**)
3. *Upside down to normal* (UDTN) step disturbance (**Fig. 3.4**)
4. *Downside up to normal* (DUTN) step disturbance (**Fig. 3.5**)

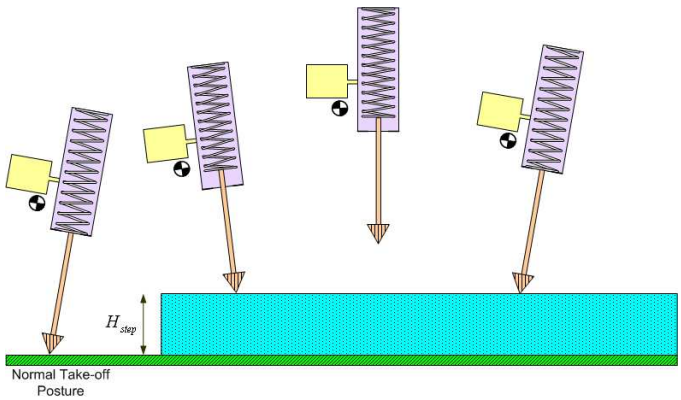

Figure 3.2: Hopper subjected to upside as normal (UAN) step disturbance

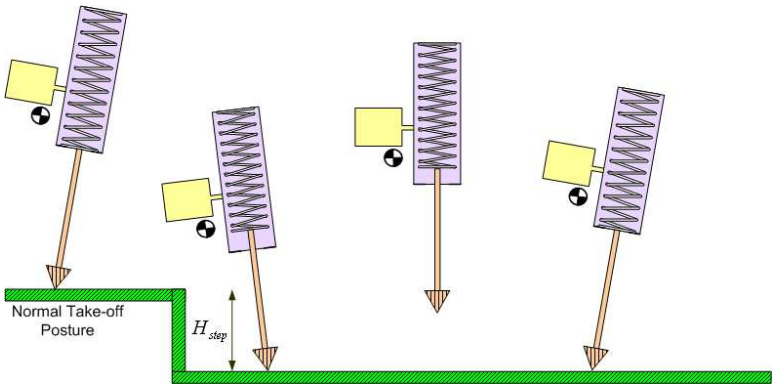

Figure 3.3: Hopper subjected to downside as normal (DAN) step disturbance

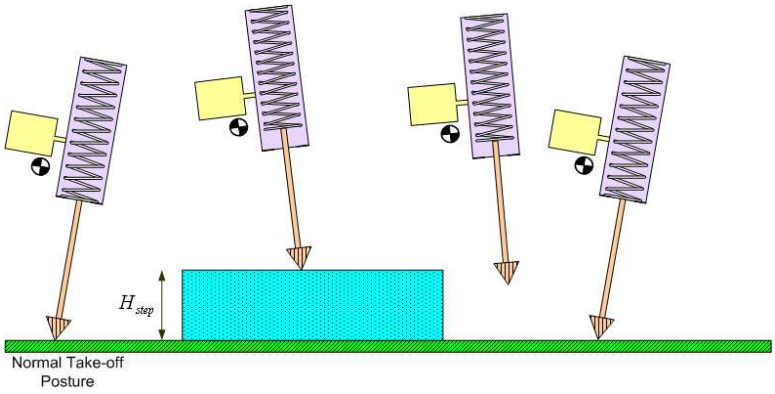

Figure 3.4: Hopper subjected to upside down to normal (UDTN) step disturbance

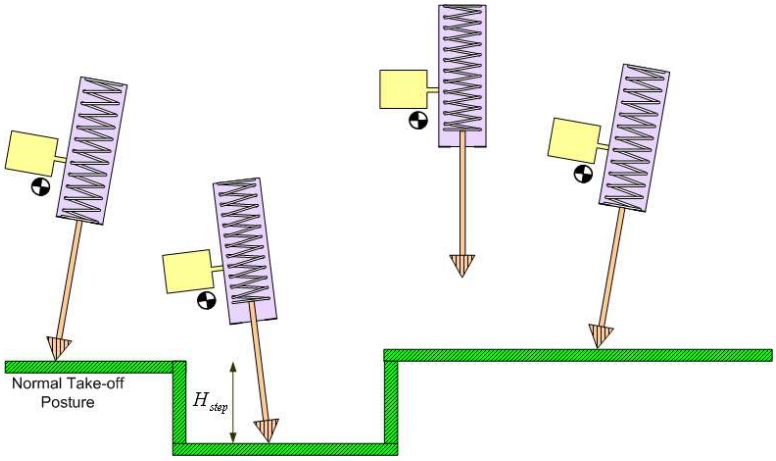

Figure 3.5: Hopper subjected to downside up to normal (DUTN) step disturbance

The remainder of this Chapter is organized as follows: In Section 3.2 we describe a classical full state feedback LQR controller assuming that the all states are available for measurement. In Section 3.3, an observer based full state feedback control system is discussed. In Section 3.4, an optimal state feedback control problem is discussed.In Section 3.5, discrete-time sliding mode based full state feedback control system is dis-

cussed. In Section 3.6, conclusions are drawn based on the study of proposed active configuration of SLOM hopper.

3.2 Classical Full State Feedback Control System Design:

We derived the linear deviational model by linearizing around fixed point and is in discrete form. The usual state space model of a linear perturbed open-loop controlled system (fixed input) system is represented by equation (2.16), and is re-written here as,

$$\triangle \mathbf{x}(k+1) = \mathbf{A}\triangle \mathbf{x}(k) + \mathbf{B}\triangle \mathbf{u}(k) \tag{3.1}$$

From the previous analysis, it is noted that the one of the eigenvalues of matrix $\mathbf{A}$, (λ_1), is outside the unit circle. So, the linear deviational model is unstable. In this Section we present a *pole placement* or *pole assignment technique* to design a classical SFB control to stabilize the proposed hopping model. We assume that all states are measurable and are available for feedback. The necessary condition to design such controller is that the system should be completely state controllable [34], which we already assured in our case since,

$$Rank \left([\ \mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B}\ ] \right) = 3. \tag{3.2}$$

3.2.1 Selection of Closed-loop Poles and Controller Design:

Due to Pioncaré mapping the actual hybrid system is transformed to discrete-time system. In equation (3.1), k refers to the hop number. Our objective of designing the SFB controller for the discrete-time linear deviational model is to evaluate its performance on the actual model. The control law for linear model is of the form,

$$\triangle \mathbf{u}(k) = -\mathbf{K_p}\triangle \mathbf{x}(k). \tag{3.3}$$

We can design $\mathbf{K_p}$, either by pole placement technique using Ackermann's method or optimal control gain that minimizes the cost or performance index. We have to find the control gain matrix, $\mathbf{K_p}$, such that poles of $(\mathbf{A} - \mathbf{BK_p})$ are inside the unit circle.

We wish to design a digital controller, i.e. the constant control gain matrix, $\mathbf{K_p}$, such that the steady state value is zero. It is called the *regulator case*. In our hopping model there is constraint on the control input. From Table 2.1 and Table 2.2, it

can be noted that the free length of spring (S_0) is 0.4002m. We have assumed that the hard spring length is 0.2002m. So, it is assumed that the possible maximum and minimum spring retraction is 0.2m and 0m respectively. In this Section, we describe the optimum constant gain ($\mathbf{K_p}$) selection using the *Linear Quadratic Regulator* (LQR) method. Also, it is found that such control system designed will be asymptotically stable [34].

We desire to determine the control law $\Big(\triangle \mathbf{u}(k)\Big)$ such that a given quadratic performance index is minimized.

$$\mathbb{J} = \sum_{k=0}^{\infty} \left[\triangle x^T(k) Q \triangle x(k) + \triangle u^T(k) R \triangle u(k)\right] \tag{3.4}$$

Q and R are symmetric weighting matrices to be selected by designer. Since we are more concerned about the control input, the second term, R is penalized more in our design.

Here, we selected Q as,

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix} \tag{3.5}$$

and considered the different scaler values of R, i.e, $R = 50$, $R = 100$, $R = 150$, $R = 200$ and $R = 250$. Using MatLab© function *dlqr*, we can determine $\mathbf{K_p}$. Thus, we have 5 different controllers. We simulated the actual hopping model for those SFB controllers. The simulated results (initial condition response) are shown in the **Fig. 3.6**-**Fig. 3.9**. The simulation results presented here are for the initial deviation,

$$\triangle \mathbf{x}_0 = [-0.01rad, \quad 0.1m/s, \quad 0.01rad/s].$$

The convergence criteria to stop the simulation is,

$$max\Big\{abs\Big[\mathbf{x}(k) - \bar{\mathbf{x}}\Big]\Big\} < 10^{-4}. \tag{3.6}$$

We observed that, the control effort is minimizing with respect to nominal value ($\overline{u}$) as value of R increases as expected. But, as we penalize R more, the number of hops required to converge at nominal values is also increasing. So, finally we selected $R = 200$. Also, it follows smoother asymptotic convergence.

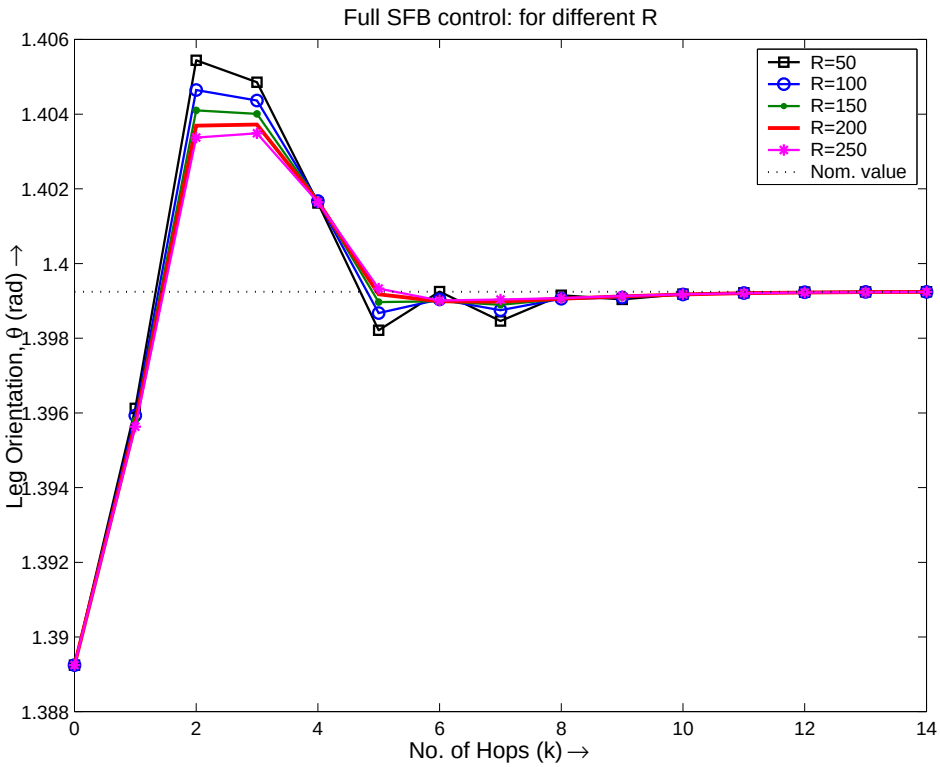

Figure 3.6: Initial condition response of hopping model: State x_1

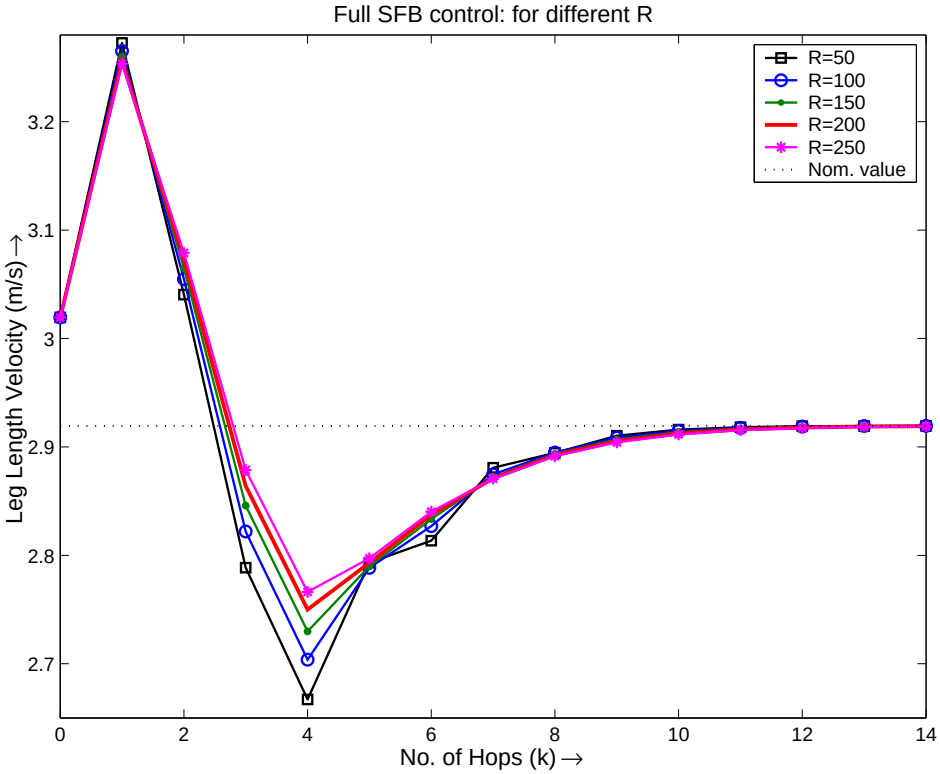

Figure 3.7: Initial condition response of hopping model: State x_2

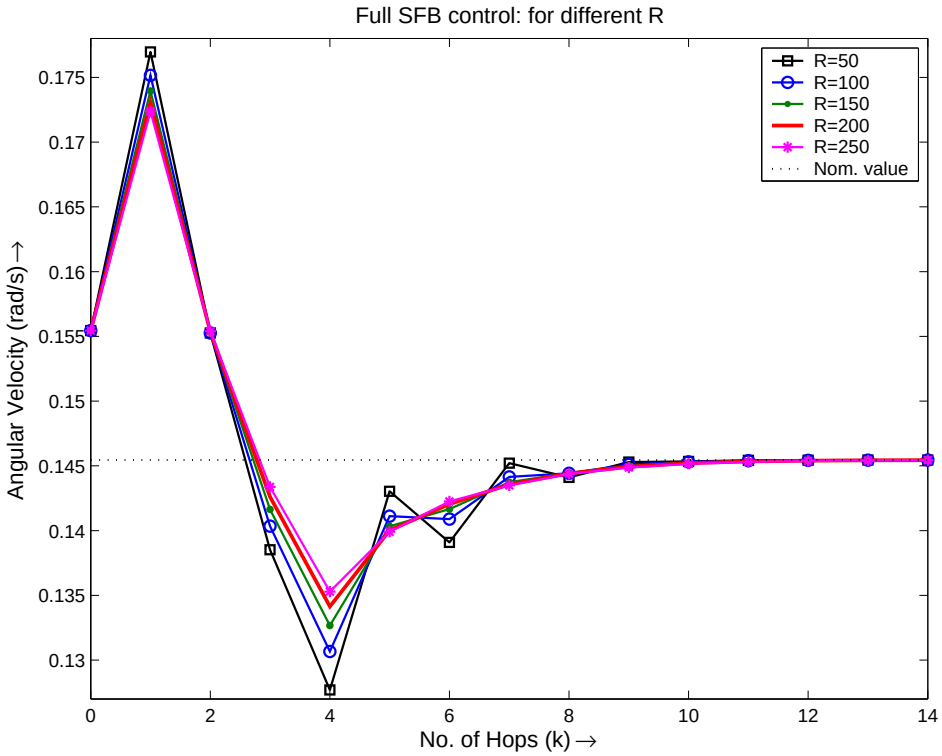

Figure 3.8: Initial condition response of hopping model: State x_3

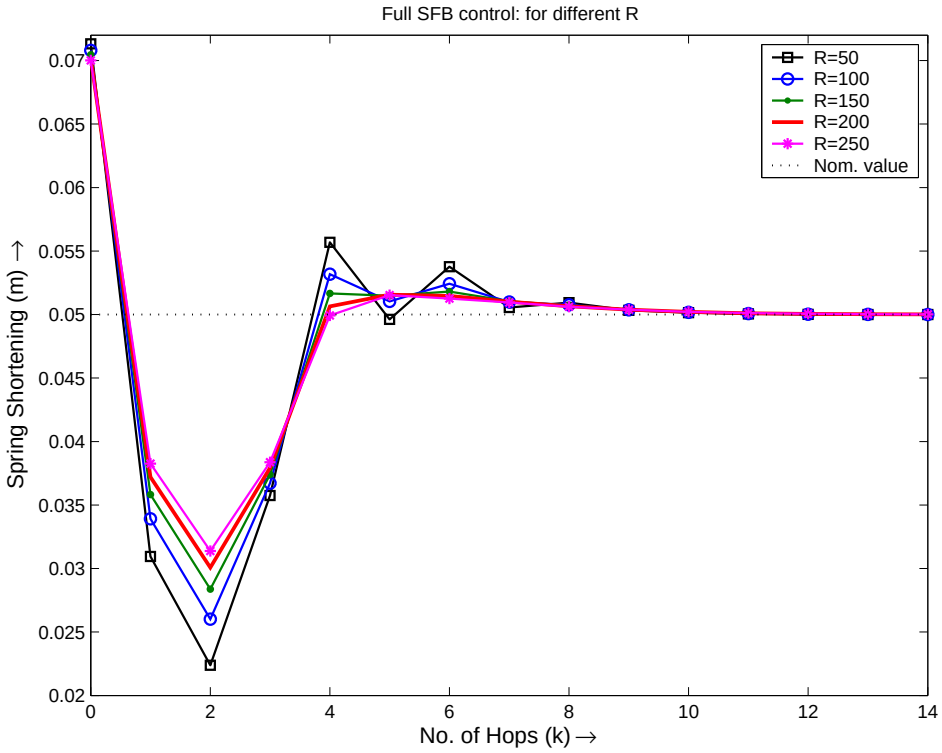

Figure 3.9: Initial condition response of hopping model: State u

Controller design:

The linear deviational model, the pair (**A**,**B**) as shown in equation (3.1) has the original poles at,

$$\lambda_1 = 4.8273, \quad \lambda_2 = -0.0786, \quad \lambda_3 = 0.8009$$

Choosing $Q = \text{diag}\ (1, 1, 10)$ and $R = 200$, and using LQR-optimal method, we obtained constant gain matrix, $\mathbf{K_p}$, using the *dlqr* command as given below.

$$\mathbf{K_p} = [2.9849,\ -0.0537,\ 1.5014] \tag{3.7}$$

The corresponding closed loop poles are at 0.2068, 0.4158 and -0.07488. The obtained $\mathbf{K_p}$, is same when computed by pole placement method if we select these closed loop poles.

3.2.2 Simulation Results:

It is guaranteed that the linear system will become stable with the proposed SFB controller. As we are looking for stabilizing the actual nonlinear hopping model, the control law derived based on linear model is applied to the nonlinear model. Note that the fixed point determined earlier is,

$$\bar{\mathbf{x}} = \begin{bmatrix} 1.3992\ rad \\ 2.9194\ m/s \\ 0.1454\ rad/s \end{bmatrix}, \qquad \bar{\mathbf{u}} = 0.05\ m.$$

In order to determine the basin of attraction of this linear controller, we simulated model considering different type of step disturbances. It is observed that the maximum step height (H_{step}) that can be climbed or stepped-down by hopper is 0.043 m, -0.058 m, 0.030 m and -0.065 m in UAN, DAN, UDTN and DUTN type of step disturbance respectively. As mentioned previously, the hopper is initiated with fixed point. The simulation results of the hopper subjected to these maximum step height is shown in the **Fig. 3.10**. Hopper state, $\mathbf{x}$ and control input, u converges to steady state value. It is observed that in all these cases the control input does not hit the saturation limits. The convergence criteria is as in equation (3.6).

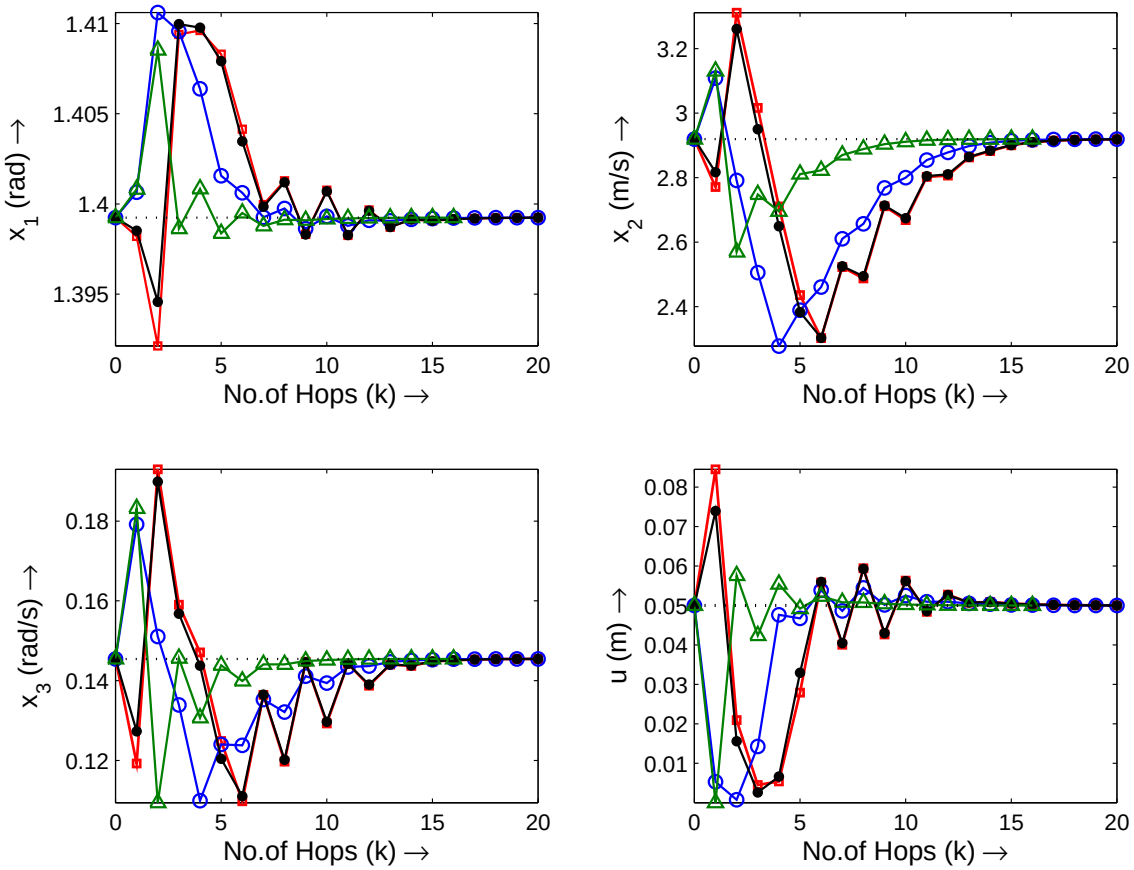

Figure 3.10: Full order classical state feedback controller applied to stabilise the hopping model, subjected to step disturbance [Square-dash: UAN ($H_{step} = 0.043\ m$), Circle-dash: DAN ($H_{step} = -0.058\ m$), dot-dash: UDTN ($H_{step} = 0.030\ m$), Triangle-dash: DUTN ($H_{step} = -0.065\ m$) and Dotted line: Nominal values].

3.3 State Observer Based FSB Control System:

We have designed the full state feedback controller for stabilizing the proposed hopping model. The basic assumption was that the all 3 state variables are available for measurement. Suppose, it is a need to use a minimum information to steer the hopping robot, then a new control problem will pose. We may think for the possibility of designing the controller, refer it as "reduced-order state feedback" controller. In such cases, there may be availability of large region of pole selection or may be none. But, it is possible with the help of state observer.

A state observer is an extension to a state space model that provides feedback to control a system. A state observer is used on a system where direct access to the state is not possible. If the system is observable, then state observers can be designed to estimate the signals that cannot be measured. In that context, it is also known as

an *estimator*. It estimates the states of the system based on: (i) model information ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) and (ii) measurements of past inputs and outputs. There are many types of observers like open-loop, closed-loop, non-linear, etc. But, in our study we are dealing with two types of observers: *Predictor* and *Current* observers.

3.3.1 Current Vs. Predictor Observers:

In the predictor observer, the estimation of states is done based on the measurements of the output vector up to $\mathbf{y}(k-1)$ and control vector up to $\mathbf{u}(k-1)$. So the control law does not use the current state information. While, in the current observer the estimation of states is done based on the measurements of the output vector up to $\mathbf{y}(k)$ and control vector up to $\mathbf{u}(k)$. So the most updated information is utilized to built the control law. The current observer uses the two operations known as *correction* or *update* and *prediction* [34].

Consider a state space model of a linear deviational system as in equation (3.8),

$$\begin{aligned} \triangle\mathbf{x}(k+1) &= \mathbf{A}\triangle\mathbf{x}(k) + \mathbf{B}\triangle\mathbf{u}(k) \\ \triangle\mathbf{y}(k) &= \mathbf{C}\triangle\mathbf{x}(k) \end{aligned} \tag{3.8}$$

If this system is state observable then the output, $\triangle\mathbf{y}(k)$, can be used to steer the state of another state space model, purposely, the observer system. The output of the observer system, $\triangle\tilde{\mathbf{x}}(k)$, termed as *estimated* state, s fedback as the input,

$$\triangle\mathbf{u}(k) = -\mathbf{K}_p\triangle\tilde{\mathbf{x}}(k) \tag{3.9}$$

where matrix $\mathbf{K}_p$ is the feedback gain matrix.

The full-order **predictor type observer** dynamics applicable in our case is,

$$\begin{aligned} \triangle\tilde{\mathbf{x}}(k+1) &= \mathbf{A}\triangle\tilde{\mathbf{x}}(k) + \mathbf{B}\triangle\mathbf{u}(k) + \mathbf{K}_e\Big(\triangle\mathbf{y}(k) - \triangle\tilde{\mathbf{y}}(k)\Big) \\ \triangle\tilde{\mathbf{y}}(k) &= \mathbf{C}\triangle\tilde{\mathbf{x}}(k) \end{aligned} \tag{3.10}$$

In this case, it is to be assured that the $(\mathbf{A}, \mathbf{C})$ pair is state observable.

The full-order **current type observer** dynamics applicable in our case is,

$$\begin{aligned} \triangle\tilde{\mathbf{x}}(k+1) &= \triangle\mathbf{z}(k+1) + \mathbf{K}_e\Big(\triangle\mathbf{y}(k+1) - \mathbf{C}\triangle\mathbf{z}(k+1)\Big) \\ \triangle\mathbf{z}(k+1) &= \mathbf{A}\triangle\tilde{\mathbf{x}}(k) + \mathbf{B}\triangle\mathbf{u}(k) \\ \triangle\tilde{\mathbf{y}}(k+1) &= \mathbf{C}\triangle\tilde{\mathbf{x}}(k+1) \end{aligned} \tag{3.11}$$

In this case, it is to be assured that the $(\mathbf{A}, \mathbf{CA})$ pair is state observable.

It is well-known that, the Predictor observer is preferable if the system is of higher order and hence the control law may be complex. While, the Current observer is preferable if the system is of lower order. Also, Current observers are often more accurate than the Predictor observers. So, we chosen the Current observer in designing the observer based full state feedback controller. We designed three observers as follows:

- Observer 1: It is assumed that θ is only measured variable, i.e. $\mathbf{C} = [\ 1 \quad 0 \quad 0\ ]$
- Observer 2: It is assumed that $\dot{l}$ is only measured variable, i.e. $\mathbf{C} = [\ 0 \quad 1 \quad 0\ ]$
- Observer 3: It is assumed that $\dot{\theta}$ is only measured variable, i.e. $\mathbf{C} = [\ 0 \quad 0 \quad 1\ ]$

For the first case alone, a Predictor observer is also designed and the performance of the model is compared in Section 3.3.3 for a UAN step size of 0.0001m.

3.3.2 Controller Design:

In Section 3.2, the feedback gain matrix ($\mathbf{K_p}$) was determined using LQR-optimal method. The corresponding closed-loop poles are selected here to design $\mathbf{K}_p$, using the *pole placement* method. Recall that $(\mathbf{A}, \mathbf{B})$ is a controllable pair. It is known that there is duality property between controllability and observability pairs. Hence, the observer gain matrix ($\mathbf{K}_e$) can be determined similarly using *pole placement* method, for observer poles using the $\left(\mathbf{A}, \mathbf{C}^T\right)$ and $\left(\mathbf{A}, (\mathbf{CA})^T\right)$ pair in the predictor and current observer respectively. Generally, the closed-loop poles and observer poles are chosen such that the observer system is 4-5 times faster than the feedback system. The observer poles and control gain vectors for different observers are listed in Table 3.1.

3.3.3 Simulation Results:

The simulation result shown in **Fig. 3.11** is related to Observer 1 (θ as measured variable) based control system. This simulation result compare the performance of the predictor and current observer, when the hopper is subjected to the UAN step disturbance ($H_{step} = 0.0001\ m$). It can be clearly noted that the predictor observer based controller converges at 18^{th} hop while current observer based controller converges at 8^{th} hop. Another issue to clarified is that the control effort required in predictor observer is comparatively higher than the current observer as expected. The basin of attraction of these controllers is indicated in Table 3.2.

The simulation result shown in **Fig. 3.12** is also related to Observer 1 based control system. In this case, the hopper is subjected to the DAN step disturbance ($H_{step} =$

Table 3.1: Design parameters for the full order observer based classical SFB controller

Design Parameter	Values
Closed-loop poles	$\begin{bmatrix} 0.2068, & 0.4158, & -0.07488 \end{bmatrix}$
Observer poles	$\begin{bmatrix} 0.05, & 0.01, & 0.10 \end{bmatrix}$
State feedback gain, $\mathbf{K}_p$	$\begin{bmatrix} 2.9849, & -0.0537, & 1.5014 \end{bmatrix}$
Predictor observer gain, $\mathbf{K}_e$ (Observer 1)	$\begin{bmatrix} 5.3896, & 33.8895, & 20.3974 \end{bmatrix}^T$
Current observer gain, $\mathbf{K}_e$ (Observer 1)	$\begin{bmatrix} 1.0002, & 39.8134, & 5.5207 \end{bmatrix}^T$
Current observer gain, $\mathbf{K}_e$ (Observer 2)	$\begin{bmatrix} 2.6118, & 1.0002, & 9.1366 \end{bmatrix}^T$
Current observer gain, $\mathbf{K}_e$ (Observer 3)	$\begin{bmatrix} 0.3390, & -3.5370, & 1.0002 \end{bmatrix}^T$

Table 3.2: Basin of attraction of observer based SFB controller

Type of Step Disturbance	Observer 1 (Meas. Variable, θ)	Observer 2 (Meas. Variable, $\dot{l}$)	Observer 3 (Meas. Variable, $\dot{\theta}$)
UAN	+0.0030 m	$+2 \times 10^{-7}$ m	+0.0018 m
DAN	−0.0037 m	-4×10^{-7} m	−0.0032 m
UDTN	+0.0024 m	$+2 \times 10^{-7}$ m	+0.0018 m
DUTN	−0.0021 m	-2×10^{-7} m	−0.0015 m

$-0.0037\ m$). The simulation result shown in **Fig. 3.13** is related to Observer 2 ($\dot{l}$ as measured variable) based control system. In this case, the hopper is subjected to the UDTN step disturbance ($H_{step} = 2 \times 10^{-7}\ m$). Eventhough the step height considered is comparatively very small, the simulation result conveys the ability of proposed controller to stabilise the hopper. The simulation result shown in **Fig. 3.14** is related to Observer 3 ($\dot{\theta}$ as measured variable) based control system. In this case, the hopper is subjected to the DUTN step disturbance ($H_{step} = -0.0015\ m$).

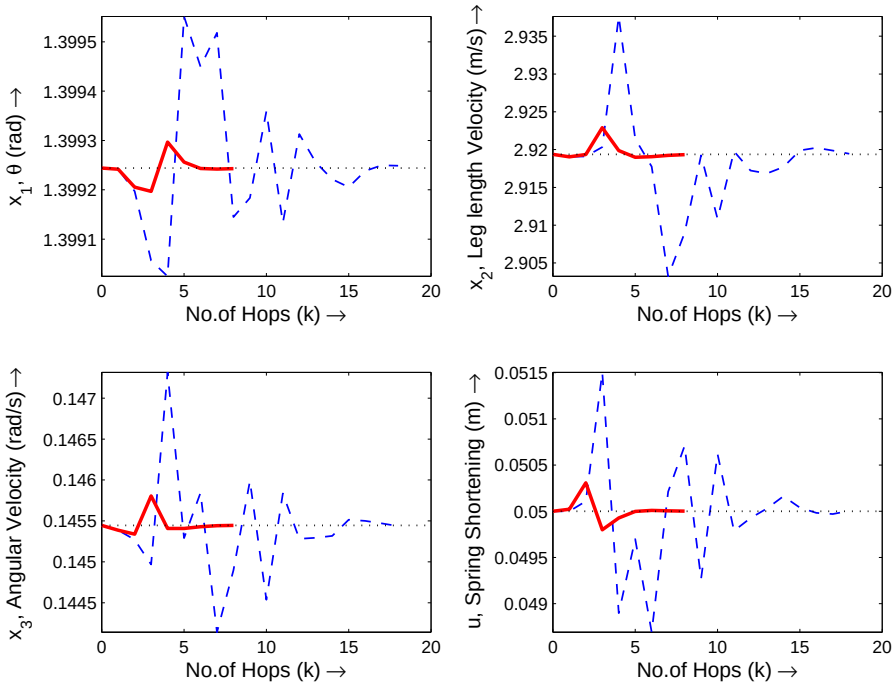

Figure 3.11: Full order observer 1 based SFB controller applied to the hopping model, subjected to UAN type of step disturbance ($H_{step} = 0.0001\ m$). [Solid line: Current observer, Dashed line: Predictor observer, and Dotted line: Nominal values].

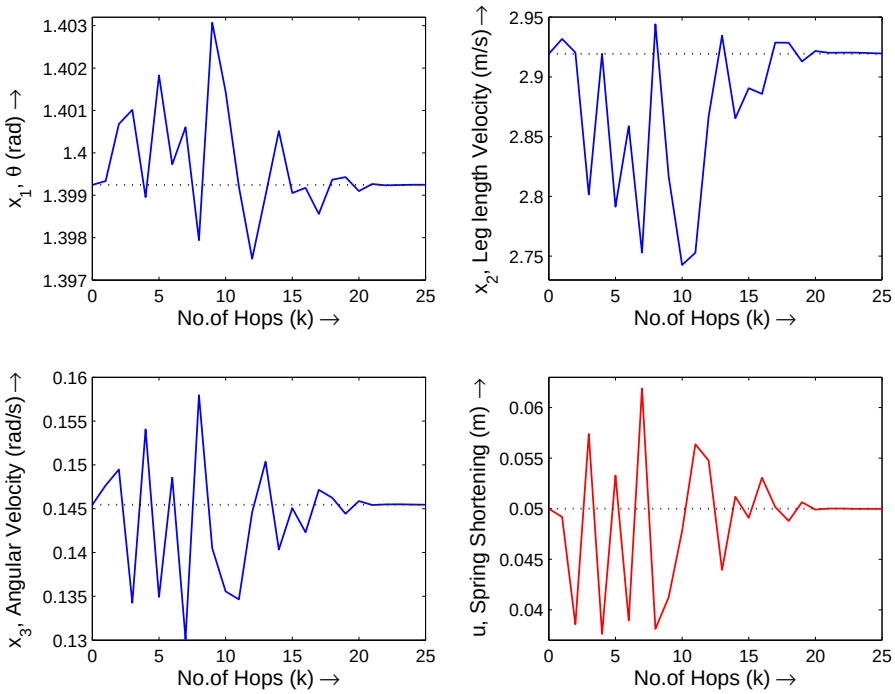

Figure 3.12: Full order Current observer 1 based state feedback controller applied to the hopping model, subjected to DAN type of step disturbance ($H_{step} = -0.0037\ m$). [Dotted line: Nominal values].

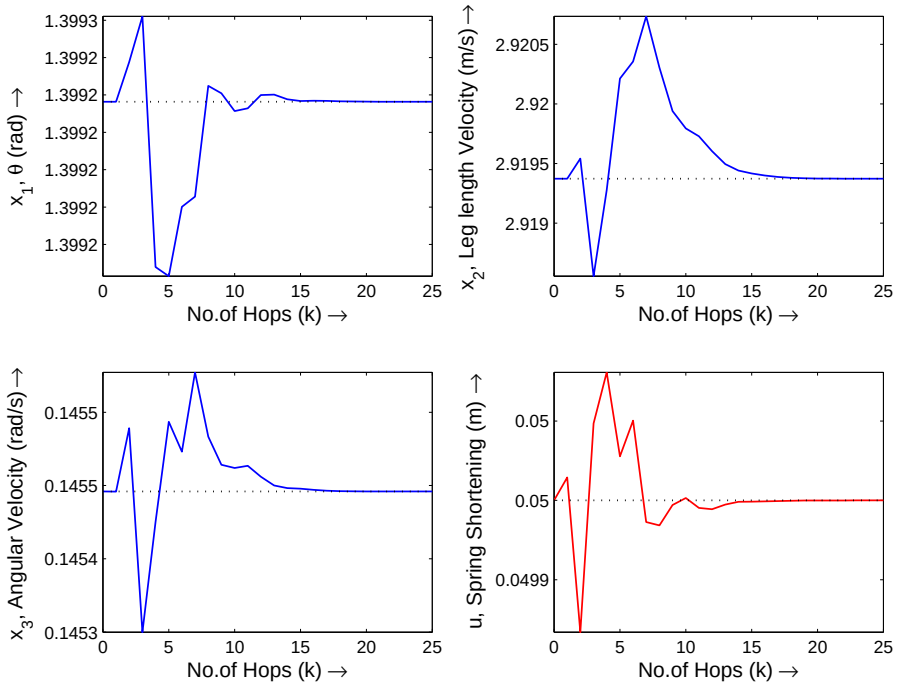

Figure 3.13: Full order Current observer 2 based state feedback controller applied to the hopping model, subjected to UDTN type of step disturbance ($H_{step} = 2 \times 10^{-7}\ m$). [Dotted line: Nominal values].

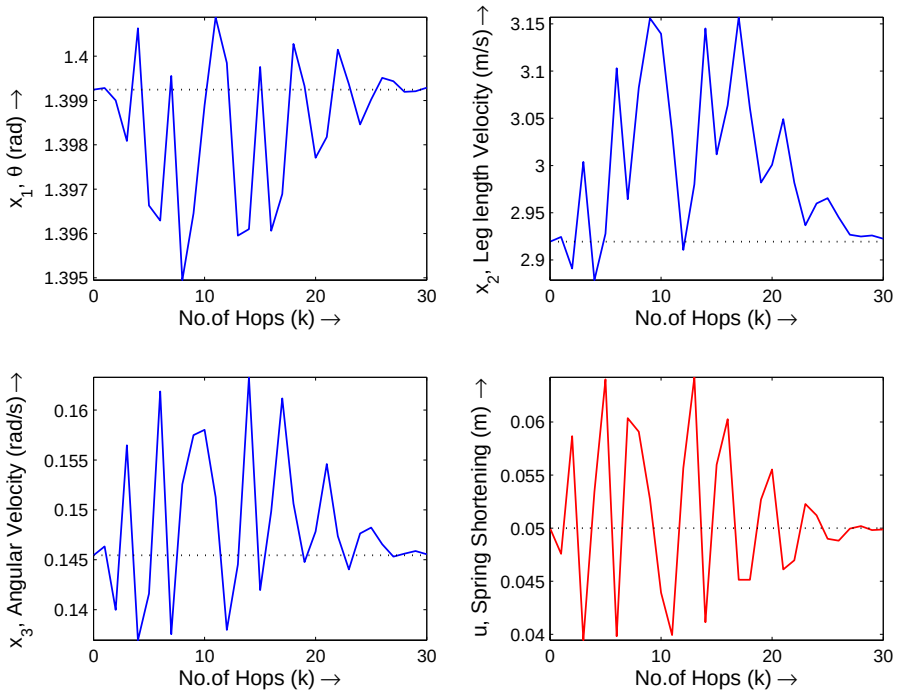

Figure 3.14: Full order Current observer 3 based state feedback controller applied to the hopping model, subjected to DUTN type of step disturbance ($H_{step} = -0.0015\ m$). [Dotted line: Nominal values].

3.4 Objective of constructing non-linear approximated map:

In the previous attempts, we proposed the state feedback and an observer based feedback control strategies for stabilizing the SLOM hopper. From the results it is clear that the basin of attraction of stabilization strategy is 2-4 mm for observer 1, 0.2-0.4 μm for observer 2 and 1-3 mm for observer 3. Since, the system under investigation has highly nonlinear dynamics, the linear model based controller shows poor performance. It is expected that, the controller designed based on a more refined model of the system, capturing some non-linearity will perform better and may increase the basin of attraction. The major objective behind constructing the approximate non-linear model is to seek solutions for an optimal control problems as stated below.

Optimal state feedback control problem:

Suppose an incremental model (approximate non-linear model) of the hopping system is,

$$\triangle x(k+1) = \widehat{f}(\triangle x(k), \triangle u(k)) \tag{3.12}$$

and the associated scalar performance index is a quadratic norm measuring the distance from the origin,

$$||\mathcal{J}|| = ||\widehat{f}^T W \widehat{f}|| \tag{3.13}$$

where, W is symmetric positive definite weighting matrix. The optimal control problem is to find the control $\triangle u(k))^*$ such that the weighted quadratic norm $||\mathcal{J}||$ is minimized subjected to $U_l \leq u(k) \leq U_u$. Here, U_l and U_u are lower and upper bounds of control input respectively.

The overall objective of this optimal feedback control is state regulation, i.e. to drive any initial condition error to zero, thus guaranteeing stability. Since approximate model is *priori* known, we can compute $\triangle u(k))^*$ off-line before applying to system. In that perspective, we have formulated an approximate non-linear map based on the original Poincaré return map. We used two techniques: Least square fitting technique and Taylor series approximation technique for obtaining the nonlinear map. The model originated from former technique is less accurate than the latter technique. So, Taylor series approximation technique is presented in this Section.

3.4.1 Taylor Series Approximation:

The hopper model using the Poincaré map is discrete time model (2.12) as follows,

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), u(k)) \tag{3.14}$$

where, $\mathbf{x} = [\ \theta \quad \dot{l} \quad \dot{\theta}]^T$, $f \in R^3$ and scalar control input, u. More explicitly, $f(\mathbf{x}(k), u(k)) = f(x_1, x_2, x_3, u)$. For this non-linear function, we know one of the fixed point, $(\overline{\mathbf{x}}, \overline{u})$. Let's define incremental variables that simply measures how far we are from $\overline{\mathbf{x}}$ and $\overline{u}$ as below,

$$\triangle\mathbf{x} = \mathbf{x} - \overline{\mathbf{x}},\ \triangle u = u - \overline{u} \tag{3.15}$$

Then using the Taylor series approximation method described in Appendix C, specially, equation (C.10), we can write,

$$f(\mathbf{x}, u) = f(\overline{\mathbf{x}}, \overline{u}) + A\triangle\mathbf{x} + B\triangle u + \frac{1}{2}D_1 f_1(\triangle\mathbf{x}, \triangle u) + \frac{1}{6}D_2 f_2(\triangle\mathbf{x}, \triangle u) \tag{3.16}$$

where, $A \in R^{3\times3}$ and $B \in R^{3\times1}$ are as derived for linear deviational model in Section 2.6. While, $D_1 \in R^{3\times10}$ and $f_1 \in R^{10\times1}$, $D_2 \in R^{3\times20}$ and $f_2 \in R^{20\times1}$ are as given below,

$$D_1{}^T = \begin{bmatrix} \frac{\partial^2 f(\mathbf{X},u)}{\partial {x_1}^2} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial {x_2}^2} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial {x_3}^2} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial u^2} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_1 \partial u} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_2 \partial u} \\ \frac{\partial^2 f(\mathbf{X},u)}{\partial x_3 \partial u} \end{bmatrix}, f_1 = \begin{bmatrix} \triangle {x_1}^2 \\ \triangle {x_2}^2 \\ \triangle {x_3}^2 \\ \triangle u^2 \\ 2\triangle x_1 \triangle x_2 \\ 2\triangle x_1 \triangle x_3 \\ 2\triangle x_1 \triangle u \\ 2\triangle x_2 \triangle x_3 \\ 2\triangle x_2 \triangle u \\ 2\triangle x_3 \triangle u \end{bmatrix}, D_2{}^T = \begin{bmatrix} \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_1}^3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_2}^3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_3}^3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial u^3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_1}^2 \partial x_2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_1}^2 \partial x_3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_1}^2 \partial u} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial {x_2}^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_2}^2 \partial x_3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_2}^2 \partial u} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial {x_3}^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_2 \partial {x_3}^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial {x_3}^2 \partial u} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial u^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_2 \partial u^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_3 \partial u^2} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial x_2 \partial x_3} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial x_2 \partial u} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_1 \partial x_3 \partial u} \\ \frac{\partial^3 f(\mathbf{X},u)}{\partial x_2 \partial x_3 \partial u} \end{bmatrix}, f_2 = \begin{bmatrix} \triangle {x_1}^3 \\ \triangle {x_2}^3 \\ \triangle {x_3}^3 \\ \triangle u^3 \\ 3\triangle {x_1}^2 \triangle x_2 \\ 3\triangle {x_1}^2 \triangle x_3 \\ 3\triangle {x_1}^2 \triangle u \\ 3\triangle x_1 \triangle {x_2}^2 \\ 3\triangle {x_2}^2 \triangle x_3 \\ 3\triangle {x_2}^2 \triangle u \\ 3\triangle x_1 \triangle {x_3}^2 \\ 3\triangle {x_2} \triangle {x_3}^2 \\ 3\triangle {x_3}^2 \triangle u \\ 3\triangle x_1 \triangle u^2 \\ 3\triangle x_2 \triangle u^2 \\ 3\triangle x_3 \triangle u^2 \\ 6\triangle x_1 \triangle x_2 \triangle x_3 \\ 6\triangle x_1 \triangle x_2 \triangle u \\ 6\triangle x_1 \triangle x_3 \triangle u \\ 6\triangle x_2 \triangle x_3 \triangle u \end{bmatrix}$$

(3.17)

Note that, D_1 and D_2 are calculated at $\mathbf{x} = \overline{\mathbf{x}}, u = \overline{u}$. With the help of equation (3.14) and (3.16), we have,

$$\begin{aligned}\mathbf{x}(k+1) &= f(\mathbf{x}(k), u(k)) \\ &= f(\overline{\mathbf{x}}, \overline{u}) + A\triangle\mathbf{x}(k) + B\triangle u(k) + \frac{1}{2}D_1 f_1(\triangle\mathbf{x}(k), \triangle u(k)) + \frac{1}{6}D_2 f_2(\triangle\mathbf{x}(k), \triangle u(k))\end{aligned} \tag{3.18}$$

Using the relation $\overline{\mathbf{x}} = f(\overline{\mathbf{x}}, \overline{u})$, we can write the incremental discrete time model, hereafter referred as *nonlinear approximate model* as,

$$\begin{aligned}\triangle\mathbf{x}(k+1) &= A\triangle\mathbf{x}(k) + B\triangle u(k) + \frac{1}{2}D_1 f_1(\triangle\mathbf{x}(k), \triangle u(k)) + \frac{1}{6}D_2 f_2(\triangle\mathbf{x}(k), \triangle u(k)) \\ &= \widehat{f}(\triangle x(k), \triangle u(k))\end{aligned} \tag{3.19}$$

Recall that the linear approximate model is,

$$\triangle\mathbf{x}(k+1) = A\triangle\mathbf{x}(k) + B\triangle u(k) \tag{3.20}$$

3.4.2 Numerical Results:

As described in Section 2.6, the central difference approximation is used to evaluate numerically the corresponding partial derivatives. We have chosen the linear band values (dx_i) as per Table 2.4. Numerically determined (A, B) pair is as given in equation (2.20). D_1 and D_2 are numerically determined as,

$${D_1}^T = \begin{bmatrix} 0.3340 & -8.2283 & 0.1953 \\ -0.0196 & 0.0478 & -0.0469 \\ 0.0681 & -0.3864 & 0.1439 \\ -1.1366 & 184.4792 & 25.5165 \\ 0.2455 & 0.3571 & 1.7055 \\ 0.2482 & -1.9287 & 0.4868 \\ -2.4387 & -2.9485 & -10.3435 \\ 0.3235 & 0.1713 & 1.1102 \\ 0.0686 & -3.1985 & -0.0456 \\ -0.5362 & -0.5322 & -2.8194 \end{bmatrix} \tag{3.21}$$

$$
{D_2}^T = \begin{bmatrix}
-5.7708 & -4.1363 & -26.465 \\
-0.0048 & -0.03500 & -0.0330 \\
0.0033 & -0.0065 & -0.0524 \\
8.8853 & -2209.8 & -298.95 \\
0.2328 & -2.9639 & 0.2281 \\
-2.8171 & -2.0782 & -10.728 \\
-0.56957 & 34.055 & 1.1922 \\
0.0135 & 0.0083 & -0.0074 \\
0.0524 & 0.0746 & 0.3114 \\
0.0168 & 1.5013 & 0.2919 \\
-0.7360 & -0.5025 & -2.5913 \\
0.0493 & -0.5115 & 0.0716 \\
0.0176 & 1.6039 & 0.2460 \\
21.642 & -28.514 & 22.207 \\
-0.0423 & -24.56 & -3.5441 \\
8.4278 & -8.8263 & 5.5852 \\
0.1591 & -1.4489 & 0.2791 \\
-0.5060 & 0.1950 & -2.6961 \\
-0.2299 & 7.3361 & -0.1130 \\
-0.357 & -0.1845 & -1.6071
\end{bmatrix} \tag{3.22}
$$

Note that, we have to use the relation $\mathbf{x}(k) = \overline{\mathbf{x}} + \triangle\mathbf{x}(k)$, for determining approximate model response. Numerical simulation of comparing linear (3.20) and non-linear (3.19) approximated model with the model originated from Poincaré return map (referred as '*nonlinear model*') is done for various cases. We have presented here two typical simulation results:

1. **Fig. 3.15** shows the $x_2(k+1)$ vs $x_1(k)$, the next hop-state $\dot{l}(k)$ evaluated when only state θ is varied, keeping all other states and input at their fixed point values.

2. **Fig. 3.16** shows the histogram of error calculated between actual and approximated evolved state variables when those next hop-state were evaluated when all states and input are varied randomly between appropriate range.

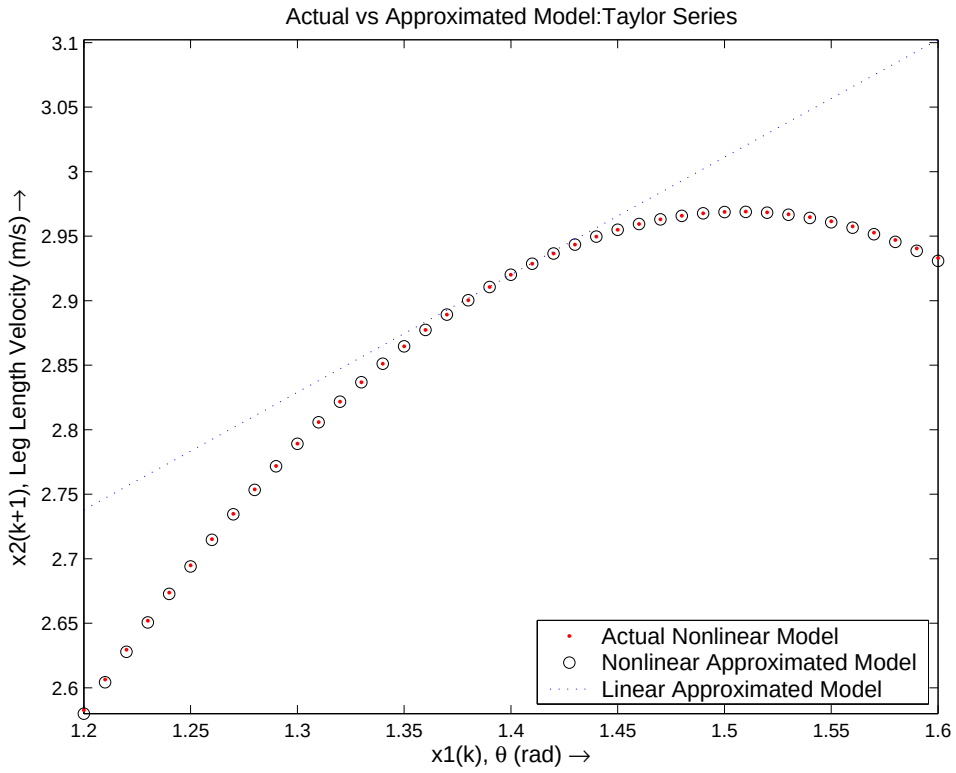

Figure 3.15: Comparing the actual nonlinear and Taylor series approximated models

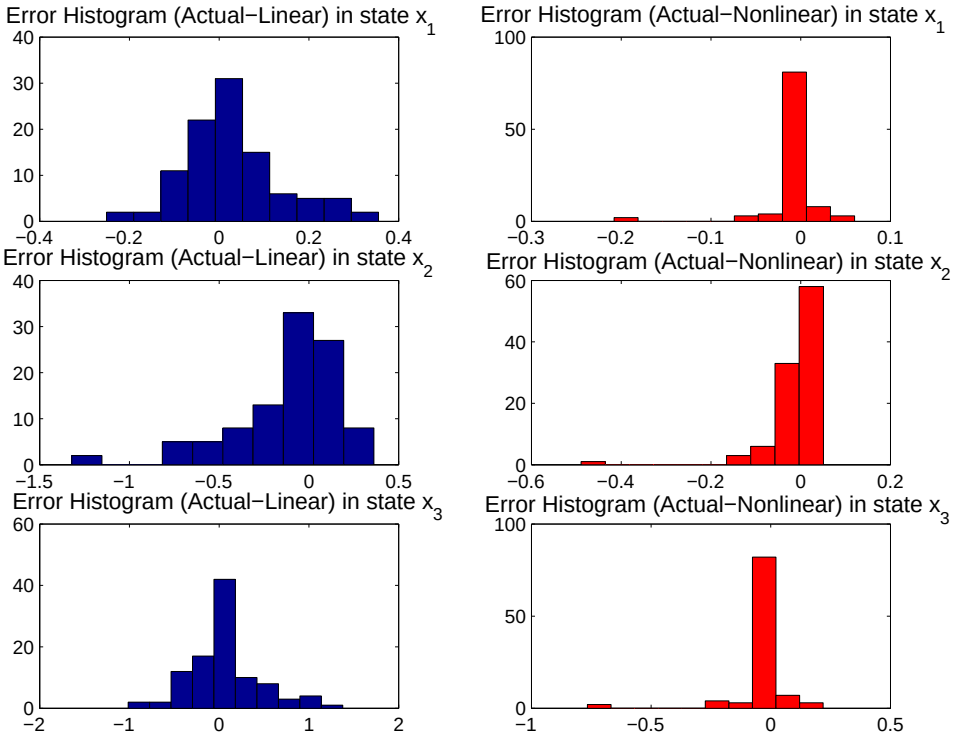

Figure 3.16: Histogram of error (with appropriate units of state) data: (Left- Linear approximate model and Right- Non-linear approximate model)

3.4.3 Solution of Optimal State Feedback Control Problem:

As discussed earlier, the objective of this optimal control problem is to find the control $\triangle u(k))^*$ such that the weighted quadratic norm $||\mathcal{J}||$ is minimized subsequently to regulate state. The nonlinear approximate model $\left(\widehat{f}\right)$ is as given in equation (3.19). We selected W, symmetric positive definite weighting matrix as,

$$W = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 0.00001 & 0 \\ 0 & 0 & 0.022 \end{bmatrix} \tag{3.23}$$

The quadratic norm $\mathcal{J} = \widehat{f}^T W \widehat{f}$ can be evaluated for given initial state deviation $\triangle x$ and different values of $\triangle u$ specified between $[U_l, U_u]$. We selected $U_l = -0.05\ m$ and $U_u = 0.1\ m$. Then optimum scalar value of $\triangle u(k))^*$ is determined based on $\triangle u$, where it gives minimum norm value from available set. The typical $||\mathcal{J}||$ vs $\triangle u$ profile is shown **Fig. 3.17**.

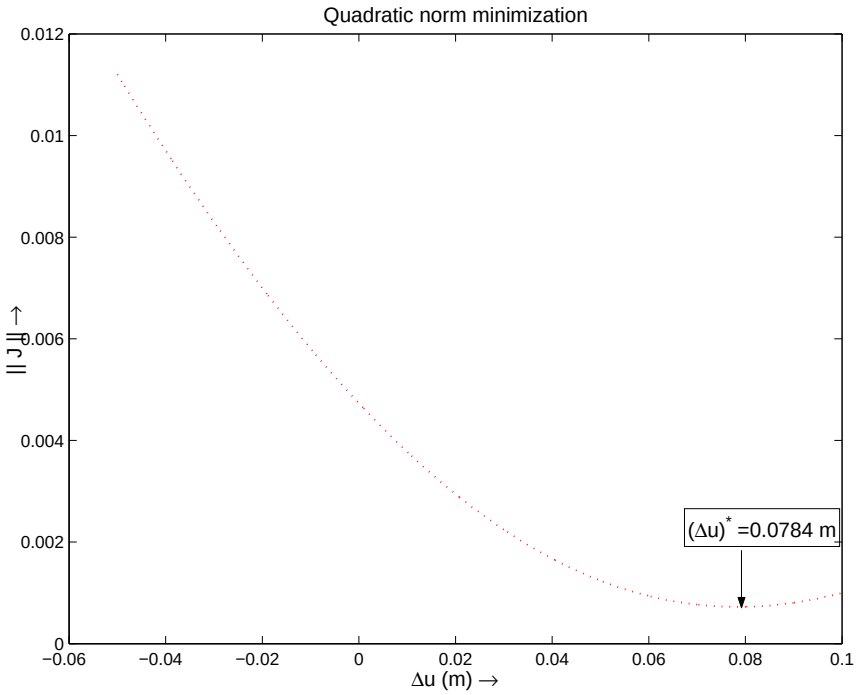

Figure 3.17: The typical $||\mathcal{J}||$ vs $\triangle u$ profile

In order to determine the basin of attraction of this control system we subjected the hopping model to different type of step disturbances. The sequence of evolved state and control input when optimal state feedback control is applied to actual hopping model is presented in **Fig. 3.18**. In this case, we considered UAN type of step disturbance ($H_{step} = 0.22\ m$). The basin of attraction of previously designed full

classical SFB controller and the current optimal controller is reported in the Table 3.3. Here it can be noted that the basin of attraction of optimal SFB controller is larger than the classical SFB controller, but the number of hops required to converge is also significantly more in the optimal controller case.

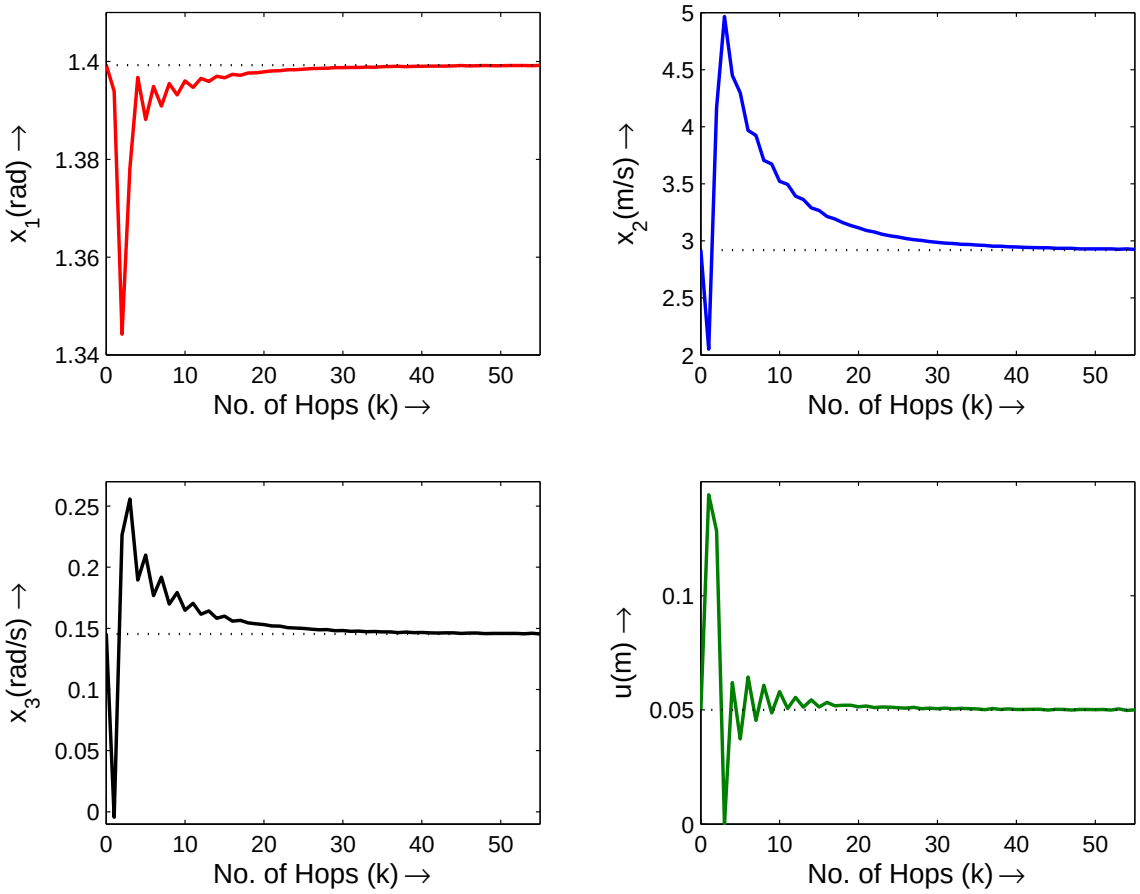

Figure 3.18: Full order optimal SFB controller applied to the hopping model, subjected to UAN type of step disturbance ($H_{step} = 0.22\ m$). [Dotted line: Nominal values].

Table 3.3: Basin of attraction of Classical SFB and optimal controller

Type of Step Disturbance	Classical SFB controller	Optimal controller
UAN	+0.043 m	+0.22 m
DAN	−0.058 m	−0.06 m
UDTN	+0.030 m	+0.05 m
DUTN	−0.065 m	−0.08 m

3.5 Discrete-time Sliding Mode Control:

Recently, many studies have tackling the Digital Sliding Mode (DSM) controller design problems [23]- [25], [35], [36]. Here we propose to study the robust controller design approach using the sliding mode control methodology. In practical systems, a mismatch between actual system and the mathematical model can always be expected. Controllers design based on the mathematical model should be robust to such mismatch/uncertainties. Sliding mode controller is one of the deterministic approach to handle the nonlinear uncertain systems.

A typical characteristics of Discrete Sliding Mode Control (DSMC) systems is that the control input is applicable only at certain sampling instants and the control effort is constant over the entire sampling period. However, the system states are no longer constrained to lie upon the confined sliding surface. As a result, DSM can undergo only quasi-sliding mode, i.e. the system states are allowed to remain in the vicinity of the sliding surface, called the *boundary Layer*. The present study uses the Gao's reaching law [24] and the saturation type reaching law [25]. The proposed DSM state feedback based control law is applied to nonlinear hopping robotic systems that guarantees discrete sliding mode. Moreover, to our knowledge, this is the first attempt to apply DSMC for single-legged hopping robot. This section is focused on the state space based design method.

For discussion we have considered the following system:

$$x(k+1) = Ax(k) + Bu(k)) \tag{3.24}$$

where, $x \in \mathbb{R}^n$ (system state), $u \in \mathbb{R}^m$ (system input), and the matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. The matrix B is assumed to have full rank and the pair (A, B) is assumed to be controllable. The design procedure for a state based discrete-time sliding mode controller (DSMC) can be divided into two steps:

✠**Step 1:** Finding the matrix $G \in \mathbb{R}^{m\times n}$ which defines the switching surface or function:

$$s(k) = Gx(k) \tag{3.25}$$

where, $s \in \mathbb{R}^m$. G is a design parameter, which ensures that the internal dynamics in the sliding mode is stable. Note that, the switching function is dependent only on the system state.

✠**Step 2:** Designing a controller which ensures that the sliding mode is reached and subsequently lie on the sliding surface.

3.5.1 Sliding Surface Design:

Here we introduce a design procedure for Step 1, i.e. finding a switching function with stable zero-dynamics. This design step is the same as designing the sliding surface in continuous-time sliding mode control (CSMC). It is well known that for the controllable system (3.24) there exists an invertible transformation $T \in \mathbb{R}^{n\times n}$ defined as, $x(k) = T\zeta(k)$:

$$\begin{bmatrix} \zeta_1(k) \\ \zeta_2(k) \end{bmatrix} = T^{-1}x(k) \tag{3.26}$$

This transformation turns the system (3.24) into the controllable canonical form.

$$\zeta_1(k) = A_{11}\zeta_1(k) + A_{12}\zeta_2(k) \tag{3.27}$$

$$\zeta_2(k) = A_{21}\zeta_1(k) + A_{22}\zeta_2(k) + B_2u(k) \tag{3.28}$$

where $\zeta_1 \in \mathbb{R}^{n-m}$, $\zeta_2 \in \mathbb{R}^m$, $A_{11} \in \mathbb{R}^{(n-m)\times(n-m)}$, $A_{12} \in \mathbb{R}^{(n-m)\times m}$, $A_{21} \in \mathbb{R}^{m\times(n-m)}$,and $A_{22} \in \mathbb{R}^{m\times m}$. Also, B_2 should be of full rank. According to this transformation the switching function (3.25) becomes:

$$s(k) = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} \zeta_1(k) \\ \zeta_2(k) \end{bmatrix} \tag{3.29}$$

where, $F_1 \in \mathbb{R}^{n-m}$ and $F_2 \in \mathbb{R}^m$.The matrices F_1 and F_2 are the design parameters which define the sliding surface, they should be chosen such that in the case that $s(k) = 0$, all remaining dynamics are stable. In other words, the matrices F_1 and F_2 should be designed such that the zero-dynamics of $s(k)$ are stable. Let us assume that there exists a controller which forces the system into sliding mode (i.e. $s(k) = 0$) and in generality we can assume $F_2 = 1$ in Single Input Single Output (SISO) case. Alternatively, we can select F_2 such that, $F_2 = B_2^{-1}$. Hence, from equation (3.29),

$$\zeta_2(k) = -F_1\zeta_1(k) \tag{3.30}$$

Substituting the above equation into equation (3.27) results in:

$$\zeta_1(k) = (A_{11} - A_{12}F_1)\zeta_1(k) \tag{3.31}$$

The above equation describes all dynamics of the closed-loop system in sliding mode. Hence, stability in the sliding mode is ensured when all eigenvalues of the matrix $(A_{11} - A_{12}F_1)$ are within the unit circle. The problem of finding the design matrix F_1 is in fact a classical state feedback problem. It can be found that if the pair

(A, B) is controllable, then the pair (A_{11}, A_{12}) is controllable as well [36]. Therefore F_1 can be determined by for example pole placement or LQR-design. Remembering that state component $\zeta_2(k)$ can be found from $\zeta_1(k)$ by the algebraic relation (3.30). Stability of $\zeta_1(k)$ implies stability of the closed-loop system in sliding mode.The matrix G in the original coordinates (see transformation (3.26)) can then be found from:

$$G = [F_1 \ 1]T^{-1} \tag{3.32}$$

Summarizing we can redefine the original control problem (3.24)) as:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k)) \\ s(k) &= Gx(k) \end{aligned} \tag{3.33}$$

where G is designed in such a way that the zero dynamics corresponding to $s(k)$ is stable. The design Step 2 is to stabilize $s(k)$ at zero. In the next Section a controller is presented which stabilizes $s(k)$ at zero.

3.5.2 Sliding Mode Control Law:

This section considers Step 2 of the design procedure for a DSMC. In this section we deal with the SISO case as our model is SISO discrete time system. The following controllable system is considered:

$$x(k+1) = Ax(k) + Bu(k) + \widehat{d}(k) \tag{3.34}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\widehat{d} \in \mathbb{R}^n$ is uniformly bounded disturbance, and the matrices (A, B) of appropriate size. It is also assumed that it is a matched type disturbance wherever not specified.

Definition 3.1: (**Matched Uncertainty**) *[23] An uncertain system of the form (3.34) is said to have a matched uncertainties if the condition*

$$\widehat{d}(k) \in \text{Range}(B)$$

is satisfied. Here, $\widehat{d}(k)$ represents the unmodeled dynamics and the external disturbance effect on the system.

There are large number of researchers who have focused on designing the reaching law, $u(k)$. The major objective is to ensure that the sliding mode is reached and subsequently lie on the sliding surface, $s(k)$. switching function is generally defined as in

(3.25). It can be noticed that, the control law in DSMC design is evolved from the CSMC theory. As can be seen in [22], [35], a control law for a CSMC can be attained by the construction of a Lyapunov function. Starting from this Lyapunov function, a reaching law is constructed ($s\dot{s} < 0$) from which the control law automatically follows. Direct discretization of this this reaching law using a first order forward approximation of the derivative $\dot{s}$, the discrete-time reaching law is given by Dote and Hoft [37]. But, Milosavljevic [38] pointed out that the said reaching law does not satisfy the sufficient conditions of convergence to the switching surface. He also suggested the concept of quasi-sliding mode and Quasi-Sliding Mode Band as defined below.

Definition 3.2: (**Quasi-Sliding Mode Band**) *[24] Quasi-Sliding Mode Band (QSMB) is defined as the band or region within the neighborhood of switching surface, where the system trajectories are allowed to oscillate. It is the band where once the system trajectory enters, it will lie inside it if the sliding mode exists.*

Sarpturk et al. [39] suggested the reaching law that directs system states towards the sliding surface, but also the norm of the switching function is defined to be strictly decreasing. Gao et. al. [24] presented the more general general reaching law. This reaching law ensures that the system trajectory stays within a QSMB after hitting the switching manifold. But, this control has the problem of *chattering*, which is more serious if the sampling time is sufficiently small. As our evolved discrete-time model refers sampling time as a hop, effect of chattering is not significant here. Also, this law can be applied with and without consideration of disturbance vector. So, we applied this control law to stabilize our system. Bartoszewicz [40] suggested another reaching law, which is the modification of Gao's reaching law. This law is better in the sense of QSMB which is Less than half of the width of the QSMB in [24]. But, the uncertain linear model of plant is considered with matched uncertainties. Misawa et.al. [41] introduced a reaching law to stabilise the uncertain linear model with unmatched uncertainty and smooth switching function. In this Section, we described these control laws in some detail.

Gao's Reaching Law:

Gao et. al. [24] introduced reaching law based on following three rules:

1. Starting from any initial state, the trajectory will move monotonically towards the switching plane and cross it in finite time.

2. Once the trajectory has crossed the switching plane the first time, it will cross the plane again in every successive sampling period, resulting in a zigzag motion about the switching plane.

3. The size of each successive zigzagging step is nonincreasing and the trajectory stays within a specified band.

This reaching law can be applied for a system with or without disturbance. We observed that the uncertain linear deviational model in our case does not support matched condition. So, here we will discuss the reaching law (for system without disturbance) and is gives as,

$$s(k+1) = s(k) - q\tau s(k) - \epsilon\tau \mathrm{sgn}\Big(s(k)\Big) \tag{3.35}$$

and corresponding control law:

$$u(k) = -(GB)^{-1}\Big[GAx(k) - (1-q\tau)s(k) + \epsilon\tau \mathrm{sgn}\Big(s(k)\Big)\Big] \tag{3.36}$$

Some design guidelines are to be followed as given below:

$$\begin{aligned} \epsilon &> 0 \\ q &> 0 \\ 1 - q\tau &> 0 \end{aligned} \tag{3.37}$$

It is to be noted that, QSMB using Gao's reaching approach is $\frac{2\epsilon\tau}{2-q\tau}$.

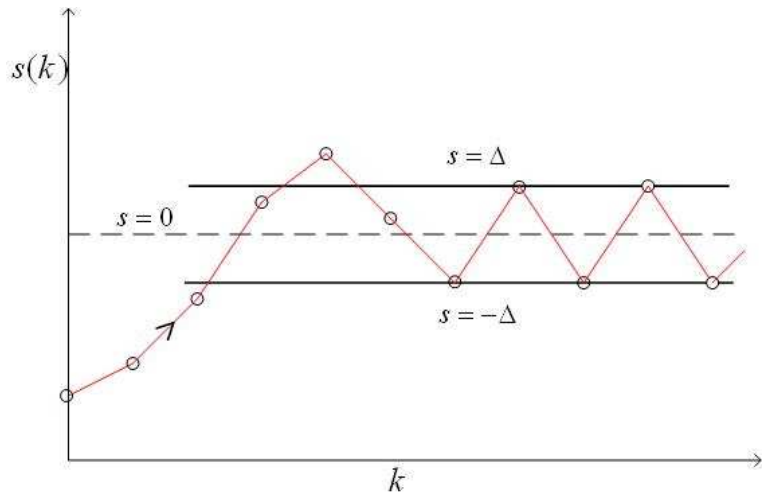

Figure 3.19: Quasi-sliding motion in state space (2Δ = QSMB)

Saturation Type Reaching Law:

The phenomenon known as chattering is evolved when the control signal utilize the relay type approximation mean in order to realize physically the signum function. The typical switching surface utilizing signum function is shown in **Fig. 3.19**. This has a serious effect mostly in CSMC. Many approaches exist to overcome the chattering effect. Here, we discuss one of the approach, where the discontinuous signum function in reaching law is replaced by the saturation function. This function allows the system trajectory to oscillate within neighbourhood of the sliding surface known as *boundary layer* without switching action. This happens due to the low pass behaviour of switching function inside boundary layer. **Fig. 3.20** shows the typical characteristics of switching function with such reaching law. The approach in the SFB based DSMC proposed by Misawa

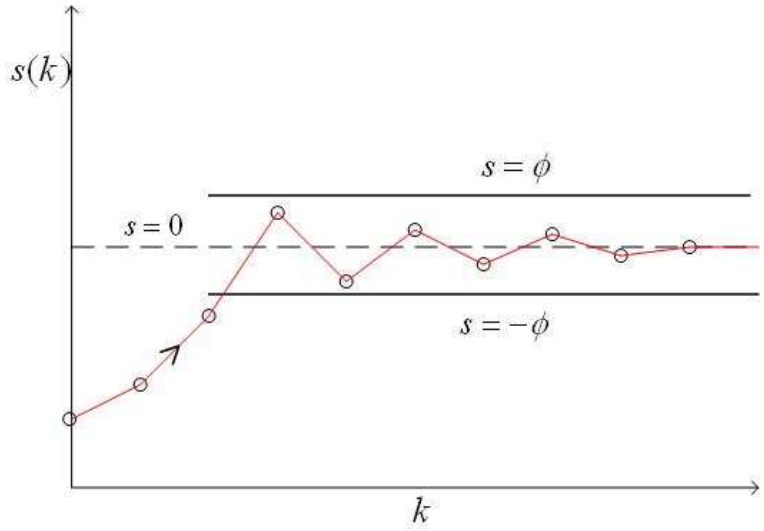

Figure 3.20: Quasi-sliding motion in state space (ϕ = boundary layer thickness)

[41] considered the discrete-time linear dynamic system (3.34), where $\widehat{d}$ is the vector of external disturbances and modeling uncertainties. The objective is to design a DSMC that would force the system state vector $x(k)$ to track the reference trajectory defined by the vector $x_d(k)$, i.e., to drive the tracking error $\tilde{x}(k) = x_d(k) - x(k)$ to zero or to as small as possible [25]. A similar approach for observer-based DSMC can be cited in [42]. The same approach can be used for regulator case i.e. $x_d(k)$ vector is set to 0. Since it is desired to drive the linear deviational model response to zero steady state value in our case (regulator case), we applied the same approach in the present study. In that perspective, we selected the discrete saturation type reaching law and is given as [25]:

$$s(k+1) - s(k) = -K sat\left(\frac{s(k)}{\phi}\right) \tag{3.38}$$

where, K and ϕ are called saturation limit gains and boundary layer thickness respectively. The saturation term is defined as:

$$sat\left(\frac{s(k)}{\phi}\right) = \begin{cases} +1 & \text{if } s(k) > \phi \\ s(k)/\phi & \text{if } |s(k)| \leq \phi \\ -1 & \text{if } s(k) < \phi \end{cases} \tag{3.39}$$

Consider, equation (3.34) and (3.25) and assume that there is no disturbance. From that, the "equivalent control input" (assuming ideal quasi-sliding $s(k+1) = s(k) = 0$) is given as:

$$u_{eq}(k) = -(GB)^{-1}GAx(k) \tag{3.40}$$

The resulting control law that guarantee the attractiveness of the boundary layer [25] is given below after modifying our regulator case:

$$u(k) = u_{eq}(k) - (GB)^{-1}\left(s(k) - Ksat\Big(\frac{s(k)}{\phi}\Big)\right) \tag{3.41}$$

Consider the equation (3.34) with disturbance, which is not necessarily to be satisfying matching conditions but should be bounded in nature as introduced earlier in this section. Since, the vector $\widehat{d}(k)$ is assumed to be bounded, we can write

$$|G\widehat{d}(k)| \leq \gamma \tag{3.42}$$

If we select the design parameters as follows, then the control law (3.41) will drive system trajectory to sliding surface.

$$K = \gamma + 2.\Delta t.\varepsilon,\ \phi \geq \gamma + \Delta t.\varepsilon \tag{3.43}$$

where, Δt is sampling time and ε is arbitrary positive number. In general, we can assume $\phi = \gamma + \Delta t\varepsilon$. By setting $\gamma \gg \Delta t\varepsilon$, the convergence may be made arbitrary fast, but it increases the boundary layer thickness.

3.5.3 SFB Based DSMC Controller Design:

In our case, the linear and uncertain linear discrete time model is given in equation (3.44) and (3.45), respectively.

$$\triangle\mathbf{x}(k+1) = A\triangle\mathbf{x}(k) + B\triangle u(k) \tag{3.44}$$

$$\triangle\mathbf{x}(k+1) = A\triangle\mathbf{x}(k) + B\triangle u(k) + \widehat{d}(k) \tag{3.45}$$

Switching vector design:

In this section, we present two DSMC approaches. In both approaches, the switching vector G is determined using the eigenvalue assignment technique as described in Section 3.5.1. Numerically determined (A, B) pair is as given in equation (2.20). With the transformation 'T',

$$T = \begin{bmatrix} -0.8040 & 0.8906 & 0.3743 \\ -3.9324 & -54.4367 & 11.6734 \\ 1.2594 & -5.9727 & 3.0052 \end{bmatrix}$$

We get,

$$A_{11} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In order to design the stable sliding surface, the closed-loop poles of $(A_{11} - A_{11}F_1)$ (P_{rd}) are selected inside unit circle. We selected two dominant poles of the previously designed classical SFB controller (refer Table 3.1). These reduced order poles and the corresponding F_1 are given below:

$$P_{rd} = [0.2068, \quad 0.4158]$$
$$F_1 = [0.0860 \quad -0.6226]$$

The switching vector is,

$$G = [F_1 \ 1]T^{-1} = \begin{bmatrix} 0.5052 & -0.0173 & 0.3369 \end{bmatrix}$$

DSMC Approach 1 (Using Gao's Reaching Law):

This approach uses the linear model (3.44). The sliding manifold s is defined as:

$$\Psi = \Big\{ \triangle x(k) || s(k) = G \triangle x(k) = 0 \Big\}$$

The Gao's reaching law [24] and the control law is:

$$s(k+1) - s(k) = -\epsilon\tau \text{sgn}(s(k)) - q\tau s(k)$$

$$\triangle u(k) = -(GB)^{-1} \left[GA \triangle x(k) - G \triangle x(k) + q\tau s(k) + \epsilon\tau \text{sgn}(s(k)) \right]$$

We selected,

$$\epsilon = 0.0001, \quad q = 0.7, \quad \tau = 1$$

DSMC Approach 2 (Using Saturation Type Reaching Law):

This approach use the linear model (3.44). The sliding manifold s is defined as the boundary layer,

$$\Psi = \left\{ \triangle x(k) \| s(k) = |G \triangle x(k)| \leq \phi \right\}$$

The approach use saturation type reaching law and control law as:

$$s(k+1) - s(k) = -K \text{sat} \left(\frac{s(k)}{\phi} \right)$$

$$\triangle u(k) = -(GB)^{-1} \left[GA \triangle x(k) - Gs(k) + K \text{sat} \left(\frac{s(k)}{\phi} \right) \right]$$

In our proposed study, $\widehat{d}(k)$ is considered as measurable uncertainty, but we found it as unmatched. In order to find γ that satisfy $|G\widehat{d}(k)| \leq \gamma$, we used following algorithm:

1. Generate vector $H = [-0.75 : 0.01 : 0.33]$ representing the step height (H_{step}) to be introduced during the hopping motion. Thus, the size of vector H is 109. Let $w = 109$.
2. Generate empty two matrices of size $3 \times w$ for storing actual nonlinear model response and deviation from fixed point, i.e. $x(k+1)_{actual}$ and $\triangle x(k+1)$.
3. Consider $p = 1$.
4. Obtain the response of actual nonlinear model ($x(k+1)$) initiated with fixed point i.e. $\triangle x(k) = 0$ and introduce the step of height $H_{step} = H(p)$ during motion. The spring charging during flight phase 2 is fixed point value. i.e. $u = 0.05$ m. The state at the end of hop (next takeoff state) is stored as $x(k+1)_{actual}(p)$.
5. Compute $\triangle x = x(k+1)_{actual}(p) - \overline{x}$. Store it as $\triangle x(k+1)(p)$. It is to be noted that each column this matrix can be referred as $\widehat{d}(k)$. Since $\triangle x(k) = 0$ and $\triangle u(k) = 0$, $\triangle x(k+1)$ presents the spread of uncertainty due to step disturbance.
6. Increment p, i.e. $p = p + 1$. If $p = w + 1$, go to next step otherwise go to step 4.
7. Determine $\gamma = \max \| \left(G \times \triangle x(k+1) \right) \|$.

We followed above algorithm and found that $\gamma = 0.31$. Finally, we selected remaining design parameters as,

$$\phi = 1.701, \quad K = 0.7021, \quad \Delta t = 1, \quad \varepsilon = 0.001$$

3.5.4 Simulation results:

Two numerical simulations are presented in this Section. The simulation result shown in **Fig. 3.21** is comparing the performance of DSMC approach 1 and 2. In this case, the hopper is subjected to the UAN step disturbance ($H_{step} = 0.052\ m$). It can be seen that the control effort required in DSMC approach 2 is smaller than the approach 1. The switching function plot shows the chattering (of amplitude $\frac{2\epsilon\tau}{2-q\tau}$) in approach 1, while asymptotic smooth convergence to $s = 0$ line in approach 2. The same trajectory of the switching function is observed even when it starts outside boundary layer (ϕ). The simulation result in **Fig. 3.22** is presented to prove the robustness of two DSMC approaches. For comparison purpose, the classical FSB control system and DSMC system is simulated with and without uncertainty in matrix A of linear approximate model. In this case, the hopper is subjected to the DUTN step disturbance ($H_{step} = -0.05\ m$). It can be observed that the DSMC is significantly more robust than classical FSB, which is well-known fact. The basin of attraction of these controllers is reported in the Table 3.4.

Table 3.4: Basin of attraction of DSMC SFB controller

Type of Step Disturbance	DSMC approach 1	DSMC approach 2
UAN	+0.052 m	+0.066 m
DAN	−0.052 m	−0.048 m
UDTN	+0.035 m	+0.044 m
DUTN	−0.070 m	−0.075 m

3.6 Conclusions:

In this Chapter we pursued a study of stabilizing controller based on full state feedback and the feedback based on the estimation of states. In case of the full-order observer based feedback control strategy, it utilize the single variable information to steer the hopping robot to a steady state region. Motivated with the control system for a linear system, we extended it to the nonlinear system. Through simulation of controller effect on the actual hopping system (non-linear model), we showed that a very simple control technique can control and stabilize a dynamic activity which is apparently complex and

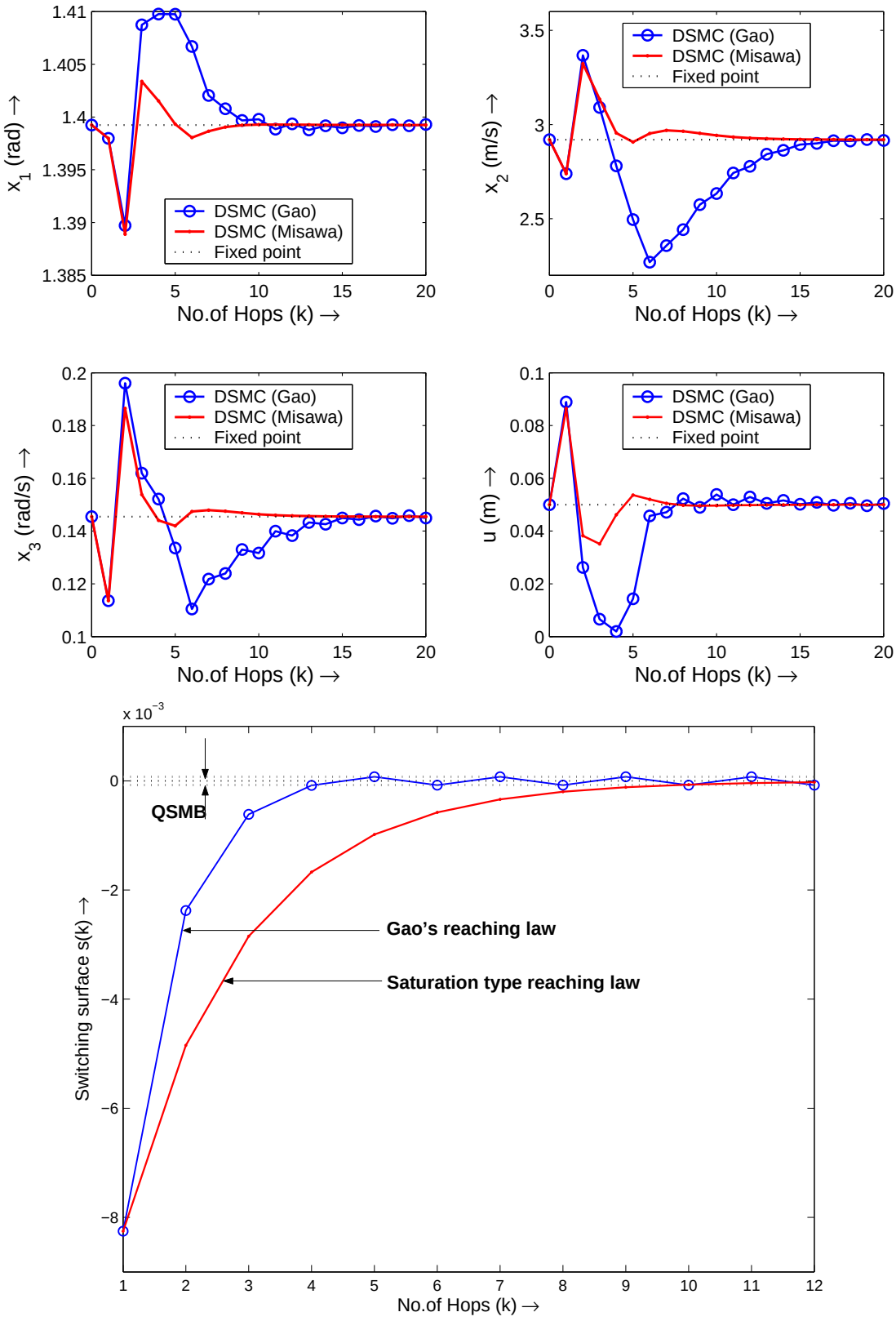

Figure 3.21: FSB DSMC applied to the hopping model, subjected to UAN type of step disturbance ($H_{step} = 0.052\ m$)

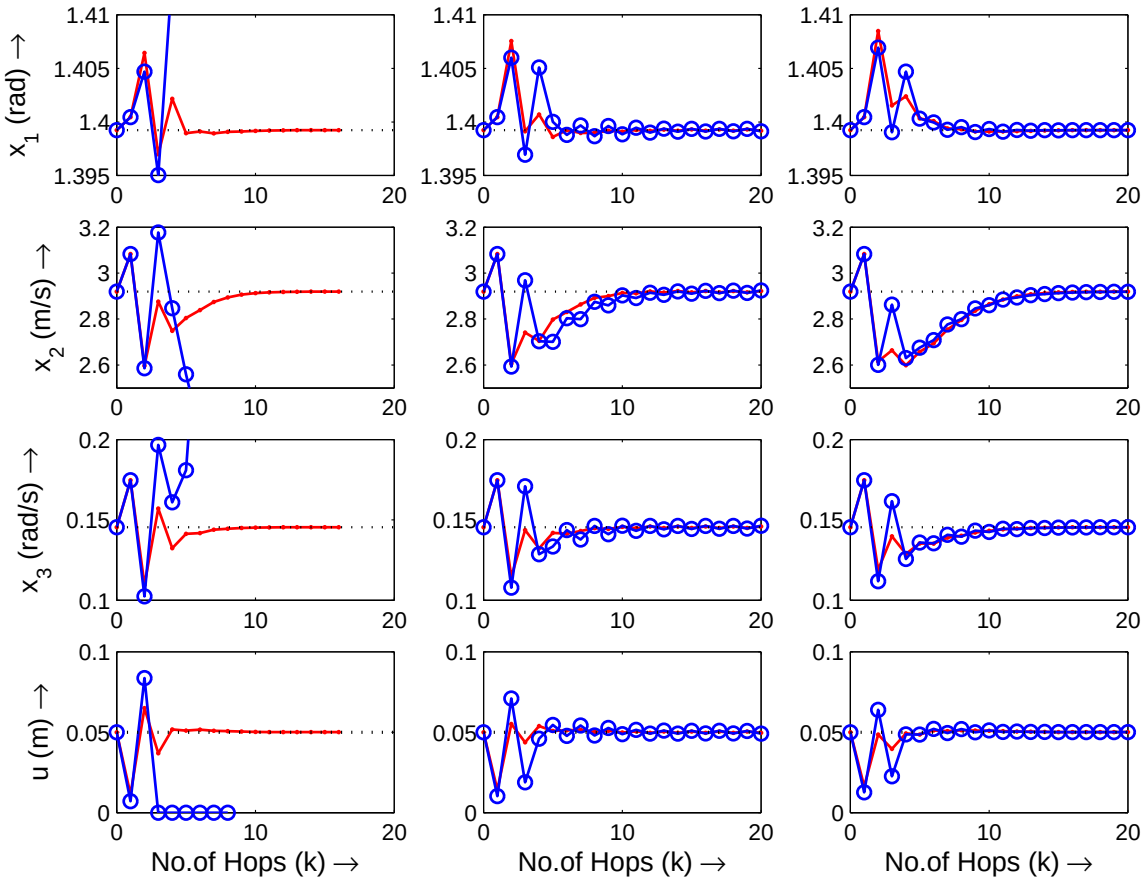

Figure 3.22: Classical FSB and FSB based DSMC applied to the hopping model, subjected to DUTN type of step disturbance ($H_{step} = -0.05\ m$). [Left: Classical FSB, Middle: DSMC appraoch 1, Right: DSMC appraoch 2] [Dot-dash: without uncertainty in A matrix, Circle-dash: with $+10\%$ uncertainty in A matrix, Dotted line: Nominal values].

highly non-linear.

From the simulated results, it is easy to conclude that the non-linear approximate model is better approximation to the *actual system*. We solved the optimal state feedback state regulation problem. It is observed from simulation results that the region of convergence is much larger as compared to other state feedback strategies, but also it requires large number of hops to converge to steady state values.

We presented two full SFB based DSMC strategies, which utilize linear and uncertain linear approximate model which includes unmatched uncertainty. Switching surface design uses eigenvalue assignment technique for the zero dynamics. It is shown that DSMC utilizing the uncertain linear model performs better over classical and DSMC which use only linear certain model. Saturation type reaching law avoids chattering as observed in Gao's reaching law case. In terms of speed of convergence, control

effort required, DSMC approach 2 is better than the DSMC approach 2 (Misawa's approach). It is observed that the DSMC is significantly more robust than classical FSB controller.

Furthermore the present study shows a simple method to find the region of attraction of these control systems. Also, it can be observed that the basin of attraction of these FSB controllers is limited. The reasons behind the limited region of attraction of stabilizing controller can be explained by three facts: (1) There is inherent non-linearity associated with original hopping model, (2) The proposed controller are devised based on the linearized model, which itself is a approximation about the operating point, and (3) The lower limit of actuation in the hopping system where the control effort in the negative direction is limited by the length of the spring cavity. So,it is useful to explore the other actuation strategies without changing the hopper kinematical arrangement. In the next Chapter, we described the another under-actuated model utilizing rotary reaction wheel actuator.

Chapter 4

State Feedback Control of SLOM Hopper with a Rotary Actuator

4.1 Introduction:

The investigation in Chapter 3 use controllers using only a linear actuator to control the SLOM hopper. As explained previously, the role of the linear actuator is to input the energy effectively to overcome energy losses associated with model. It is also used to stabilize the unstable system. Also, it is possible that both the orientation of the body and the height of a jump could be controlled by only one actuator. But, it is found that the basin of attraction of proposed feedback controllers are limited. The basic reason of the limited region of attraction are inherent non-linearity of model and the actuator saturation limits. At first insight, an alternate solution seems to be the use a rotary actuator.

Shanmuganathan [10] used a rotary actuator to stabilize the passive forward-hopping motion of SLOM hopper. He reported the possibility of *p-hop* stabilization (periodic repetition after every 'p' hopping cycles) and was able to extend the duration of hopping motion. The stabilization strategy is based on the *nominal hop* (starting the motion with proper initial conditions) state information. His composite stabilization strategy was based on the alternative use of desired forward and vertical velocity determined from the *nominal hop*. He demonstrated '3-hop' periodic motion in which the forward velocity of the nominal hop was used as the desired value for earlier two hops and the vertical velocity of the *nominal hop* was used as the desired value for every third hop. This strategy realized a limping gait and despite the back-and-forth hopping, there was a net forward motion. But the limitation of this stabilization approach was that

the angular velocity of the rotary actuator became unbounded.

As seen in [10], the rotary actuator has direct action on the hopping model, in the sense of significant influence on the leg orientation, θ and angular velocity, $\dot{\theta}$. The investigation of the SLOM model in the current Chapter focus only on the rotary actuated model. Also, the actuated model uses the fixed energy input via spring retraction during locomotion. The major role of this fixed energy pumping to regain the energy necessary for locomotion subjected to impacts and frictional energy losses. While, the major role of rotary actuator to re-orient the model, such that the motion is sustainable.

Further discussion in this Chapter is organized as follows: In Section 4.2, the proposed rotary actuated hopping model is described. The linear deviational model is derived to investigate the stability. In Section 4.3 we describe a classical FSB controller assuming that the all states are available for measurement. In Section 4.4, the solution of the optimal state feedback control problem, similar to one as discussed in previous Chapter is presented. In Section 4.5, two DSMC based FSB controllers are suggested to stabilize the proposed model. We conclude the investigation presented in this Chapter in Section 4.6.

4.2 Description of the Proposed Model:

The hopping model with the rotary actuator is shown in **Fig. 4.1**. The kinematic arrangement of proposed robot is the same as described in Section 2.1, except the addition of rotary actuator to body (hip).

The rotary actuator is placed such that the effective CG of body and actuator is concentrated at point 'A' (refer **Fig. 2.1**). In this case, the mass m_b is the addition of the body and actuator mass. Also, the parameter I_b is the addition of moment of inertia of body and moving moment of inertia of actuator about the CG, located at 'A'. Except these modifications, there is no change in any other parameters. Also, it is to be noted that, in the present study the specification of mass (m_b) and moment of inertia (I_b) is same as given in the Table 2.1.

The standard form of the EL equations of a robot model (with external forces) is,

$$\begin{aligned} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) &= \Gamma - J^T(q)\lambda_r \\ J(q)\dot{q} &= 0 \end{aligned} \tag{4.1}$$

Where, n is the number of DOF of the system. $q = [\ x_b \quad y_b \quad l \quad \theta\ ]^T$ as the set of

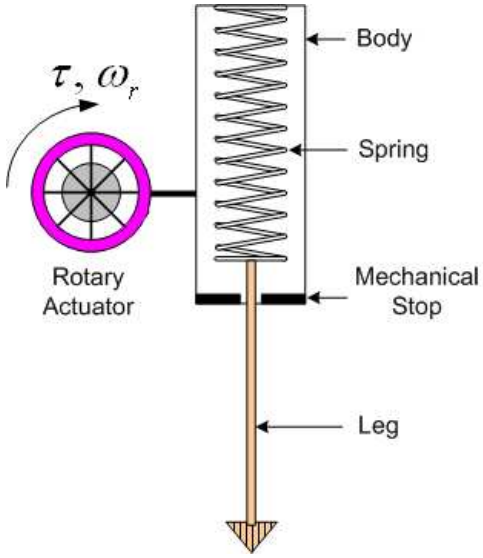

Figure 4.1: A schematic diagram of SLOM model with a rotary actuator

generalized coordinates. Here x_b and y_b are the coordinates of CG of body. $D \in R^{n\times n}$, $C \in R^{n\times n}$, and $G \in R^n$ are called the inertia matrix, the "coriolis and centrifugal forces" matrix and gravitational force vector respectively, and are the same as given in equations (2.2)-(2.4). It is to be noted that, that during actuation phase (flight phase 2),the τ is applied external force. Hence, $\Gamma = [\ 0 \quad 0 \quad 0 \quad \tau\ ]^T$ in flight phase 2 and for rest phases, $\Gamma = [0]_{4\times 1}$.

The actuator dynamics is,

$$I_r\ddot{\theta}_r = I_r\dot{w}_r = -\tau \tag{4.2}$$

where, I_r , θ_r and w_r are moment of inertia, angular position and angular velocity of the actuator respectively.

Also, it is to be noted that the definition of the Poincaré section is same as described in Chapter 2. Using it, the 3-dimensional closed-form return-map describing the robot's state at the next hop as a function of that at the current hop is derived. As the discussion earlier, in terms of discrete-time space, the robot's state at the current hop is termed as $\mathbf{x}(k)$ and the state at the next hop is used to refer as $\mathbf{x}(k+1)$. The the nonlinear discrete-time model (referred to as *actual system*) using Poincaré map is described as, $\mathbf{x}(k+1) = \mathbf{P}\{\mathbf{x}(k), \mathbf{u}(k)\}$, i.e.,

$$\begin{bmatrix} \theta \\ \dot{l} \\ \dot{\theta} \end{bmatrix}_{k+1} = \mathbf{P}\left\{ \begin{bmatrix} \theta \\ \dot{l} \\ \dot{\theta} \end{bmatrix}_k, [\mathbf{u}]_k \right\} \tag{4.3}$$

where, $\mathbf{u} = \tau$ in the present discussion.

4.2.1 Selection of Torque Trajectory (τ) during Actuation:

In the present study, we considered the fixed energy pumping (by compressing spring, $\Delta s = 0.05$ m) during flight phase 2 of the robot. As explained in Chapter 2, it follows the quintic trajectory during the specified actuation time. In the context to rotary actuation, we considered that the rotary actuator is also active during the same flight phase 2. The actuation time (T_a)is fixed during each hop. We considered four different type of torque trajectories during the actuation phase: (1) Step function, (2) Quadratic function, (3) Square wave function and (4) Triangular wave function. The effect of these torque trajectories can be well understood by looking the behaviour of hopping robot by applying them during motion.

The following simulation results is presented here in the **Fig. 4.2**. The model is initiated with $w_r = 0$ rpm, the state at take-off, $\mathbf{x}_0 = [1.5 \text{ rad}, \quad 3.2 \text{ m/s}, \quad 0.1 \text{ rad/s}]$ and $\tau = 1$ N-m.

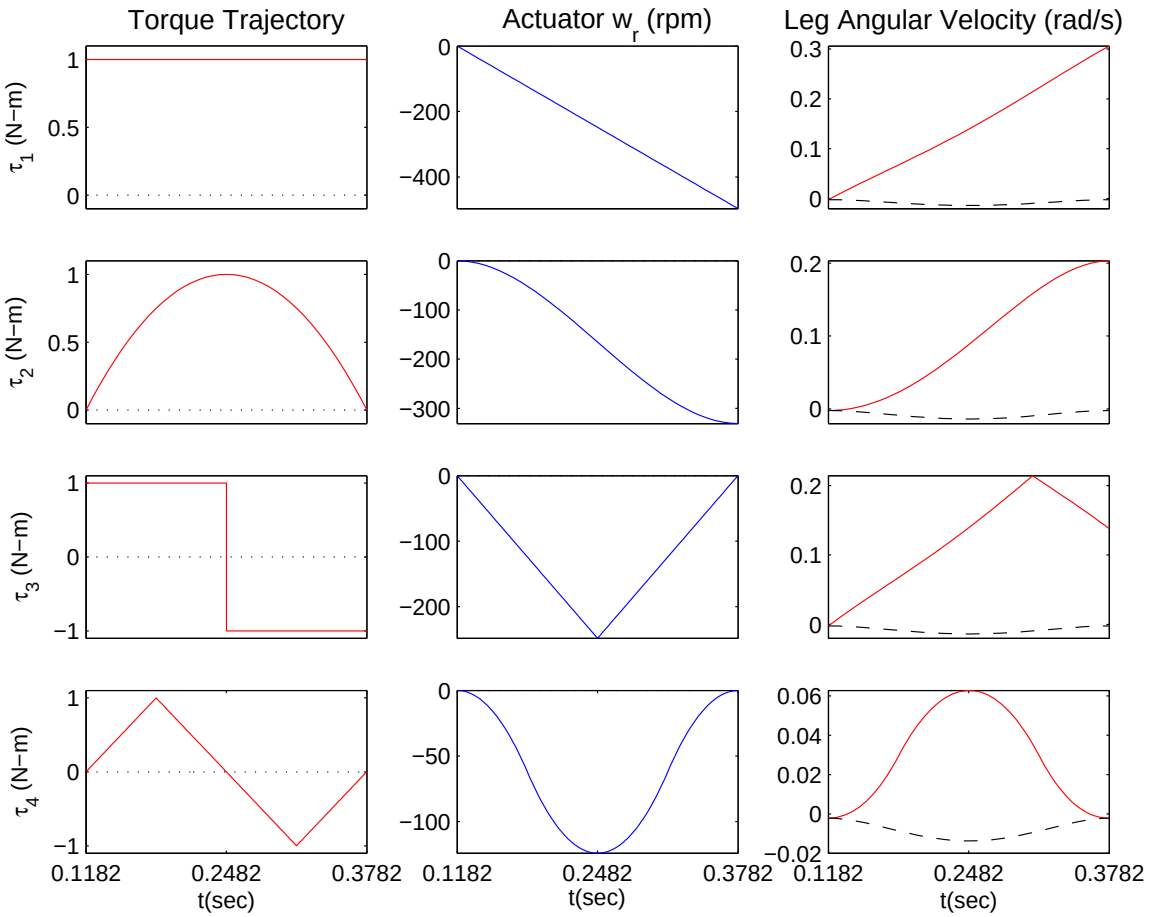

Figure 4.2: Different type of proposed torque trajectories during flight phase 2 (fixed actuation time=0.26 sec) of hopping motion, dashed line represents $\dot{\theta}$ without rotary actuation due to the action of fixed spring retraction

The simulation result shows the different torque trajectories, the corresponding w_r profile and the behaviour of actual hopping model during flight phase 2. It can be seen that the torque trajectories τ_3 and τ_4 bring the actuator angular velocity to zero at the end of actuation time. It is well-known that the angular velocity of leg ($\dot{\theta}$) is playing crucial role in the locomotion of hopping robot [11]. If we look, the net effect of different τ's on $\dot{\theta}$, there is significant change in $\dot{\theta}$ due to τ_1 as compared to other τ's. Eventhough it is expected to select the torque trajectory such that the w_r comes to rest, this issue can be handled by designing a simple feedback controller. Also, from the control point of view, we derived the linear deviational model, (A, B) pair, of the hopping system for different rotary actuations. Since, the derivation of A does not depend upon control input, $\mathbf{u}$, A is same as derived earlier. The vector B obtained for different τ's is as given in Table 4.1.

Table 4.1: B vector obtained in linear deviational model for different rotary actuations

	Torque Trajectory 1 (τ_1)	Torque Trajectory 2 (τ_2)	Torque Trajectory 3 (τ_3)	Torque Trajectory 4 (τ_4)
B	$\begin{bmatrix} 0.1509 \\ 0.0312 \\ 0.4813 \end{bmatrix}$	$\begin{bmatrix} 0.1006 \\ 0.0208 \\ 0.3209 \end{bmatrix}$	$\begin{bmatrix} 0.0798 \\ 0.0191 \\ 0.2411 \end{bmatrix}$	$\begin{bmatrix} 0.0138 \\ 0.0044 \\ 0.0364 \end{bmatrix}$

We have selected the **step function type** torque trajectory for rotary actuation. We considered the Gearhead DC Motor as a rotary actuator. In the context of the selection of the motor and gearhead, it is necessary mostly to consider the torque limit, maximum rated input speed, efficiency of gearhead and cost. In our case, we know the load cycle. Also, we can add the actuator of about 1 kg weight to the hopper. We assumed that the torque limit is 4 N-m. Based on these primary specifications, we selected the gearhead motor appropriately [43]. The Table 4.2 summarizes the typical mechanical specifications of the rotary actuator (Gearhead DC motor) considered in this study. It is to be noted that the drive circuitry is not discussed here.

4.2.2 Augmented Linear Deviational model:

The procedure applied to formulate the linear deviational model is the same as described in case of linear actuated model. The linear deviational model in the present

Table 4.2: The typical specifications of the rotary actuator

DC Motor (Faulhaber made, reversible)	Series 3257 G
Planetary Gearhead (Faulhaber made, steel)	Series 38/1
Reduction ratio	3.71:1
Efficiency	88%
Mass of motor (m_m)	0.242 kg
Mass of gearhead (m_g)	0.166 kg
Mass of flywheel (m_r)	0.5 kg
Length of gearhead motor shaft (r)	0.1 m
Load (flywheel) inertia ($I_r = m_r \times r^2$)	0.005 kgm^2
Maximum input speed at gearhead shaft side (ω_{max})	1080 rpm
Maximum torque at continuous operation (τ_{max})	6 N-m

case has the form,

$$\triangle \mathbf{x}(k+1) = \mathbf{A}\triangle \mathbf{x}(k) + \mathbf{B}\triangle \mathbf{u}(k) \tag{4.4}$$

where, $\triangle \mathbf{x} = \mathbf{x} - \overline{\mathbf{x}}$ and $\triangle \mathbf{u} = \mathbf{u} - \overline{\mathbf{u}}$. The fixed point obtained will be same as earlier except the control input fixed point value. Also, $\mathbf{x} = [\ \theta \quad \dot{l} \quad \dot{\theta}\ ]^T$ and $\mathbf{u} = \tau$. The fixed point is,

$$\bar{\mathbf{x}} = \begin{bmatrix} 1.3992 \text{ rad} \\ 2.9194 \text{ m/s} \\ 0.1454 \text{ rad/s} \end{bmatrix}, \qquad \bar{\mathbf{u}} = 0.0 \text{ N-m} \tag{4.5}$$

By referring equation (4.2), we can write,

$$\dot{w}_r(t) = \left(\frac{-1}{I_r}\right)\tau(t) \tag{4.6}$$

It is to be noted that we have chosen step function type torque trajectory and fixed time actuation. Suppose time $t \in [t_1, t_2]$ is the actuation time interval. Note that, $T_a = t_2 - t_1$, is fixed during each hop. By integrating (4.6) from time t_1 to t_2,

$$\begin{aligned} w_r(t_2) - w_r(t_1) &= -\left(\frac{1}{I_r}\right)\int_{t_1}^{t_2} \tau(t).dt \\ w_r(t_2) - w_r(t_1) &= -\left(\frac{1}{I_r}\right)\tau.(t_2 - t_1) \\ w_r(t_2) &= w_r(t_1) - \left(\frac{T_a}{I_r}\right)\tau \end{aligned} \tag{4.7}$$

Finally, in terms of discrete-time (where, k refers to hop number in our study) form, we get,

$$w_r(k+1) = w_r(k) - \left(\frac{T_a}{I_r}\right)\tau \tag{4.8}$$

By introducing the augmented state vector,

$$\mathbf{x}_e = \begin{bmatrix} \mathbf{x} \\ w_r \end{bmatrix} \tag{4.9}$$

In order to avoid confusion, it is to be noted that the steady state value (nominal value) of w_r is 0, i.e. $\overline{w_r} = 0$. So, $\triangle w_r = w_r - \overline{w_r}$. Hence, with the help of (4.4) and (4.8) we can write,

$$\triangle \mathbf{x}_e(k+1) = \mathbf{A}_e \triangle \mathbf{x}_e(k) + \mathbf{B}_e \triangle \mathbf{u}(k) \tag{4.10}$$

where,

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & 0_{(3\times 1)} \\ 0_{(1\times 3)} & 1 \end{bmatrix}, \qquad \mathbf{B}_e = \begin{bmatrix} \mathbf{B} \\ -(T_a/I_r) \end{bmatrix} \tag{4.11}$$

Since, A in does not depend upon control input, the same A in equation (2.20) is considered here. The vector B for the selected torque trajectory is given in Table 4.1. For convenience we re-written (A, B) pair here,

$$\mathbf{A} = \begin{bmatrix} 1.7115 & -0.0325 & 0.9007 \\ 0.9110 & 0.7947 & 0.2425 \\ 6.2283 & -0.0661 & 3.0436 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0.1509 \\ 0.0312 \\ 0.4813 \end{bmatrix} \tag{4.12}$$

Using $I_r = 0.005$ kg-m^2 and $T_a = 0.26$ s, we have,

$$\mathbf{A}_e = \begin{bmatrix} 1.7115 & -0.0325 & 0.9007 & 0 \\ 0.9110 & 0.7947 & 0.2425 & 0 \\ 6.2283 & -0.0661 & 3.0436 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{B}_e = \begin{bmatrix} 0.1509 \\ 0.0312 \\ 0.4813 \\ -52 \end{bmatrix} \tag{4.13}$$

The eigenvalues of this augmented matrix $\mathbf{A}_e$ are, $\lambda_1 = 4.8276$, $\lambda_2 = -0.0787$, $\lambda_3 = 0.8010$ and $\lambda_4 = 1.0000$.

4.3 Classical Full State Feedback Control System Design:

This Section focuses on the classical state feedback control system. Our core intention is to design the controller for approximate augmented linear deviational model in (4.10),

and then extend it to *actual* hopper (non-linear hybrid model). We assume that all state and angular velocity of actuator are measurable and available for feedback. It is observed that the linear model is completely state controllable i.e.

$$\text{Rank}\left(\begin{bmatrix} \mathbf{B}_e & \mathbf{A}_e\mathbf{B}_e & \mathbf{A}_e^2\mathbf{B}_e & \mathbf{A}_e^3\mathbf{B}_e \end{bmatrix}\right) = 4.$$

The regulatory control law for augmented linear model is,

$$\triangle\mathbf{u}(k) = -\mathbf{K_p}\triangle\mathbf{x}_e(k). \tag{4.14}$$

In the present study, the rotary actuator is operated with torque limit of 4 Nm. So, we designed the optimum constant gain ($\mathbf{K_p}$) using LQR method as discussed earlier.

Controller design:

The linear model, the pair $(\mathbf{A}_e, \mathbf{B}_e)$ as shown in (5.7) has the original poles at,

$$\lambda_1 = 4.8276, \quad \lambda_2 = -0.0787, \quad \lambda_3 = 0.8010, \quad \lambda_4 = 1.0000$$

Choosing $Q = \text{diag}\ (100, 1, 5, 1)$ and $R = 5000$, and using LQR-optimal method, we obtained constant gain matrix, $\mathbf{K_p}$, as,

$$\mathbf{K_p} = [13.3564,\ -0.2181,\ 6.7144,\ 0.0020] \tag{4.15}$$

The corresponding closed loop poles are at 0.2071, -0.0787, 0.8010, and 0.4869.

Simulation results:

For stabilizing the actual nonlinear hopping system, the control law derived based on linear model is applied to the actual model. Note that the fixed point is given in (4.5), and $\overline{w_r} = 0$. In order to determine the basin of attraction of this controller, we simulated model considering different type of step disturbances. It is observed that the maximum step height (H_{step}) that can be introduced during motion is 0.29 m, -0.46 m, 0.33 m and -0.73 m in UAN, DAN, UDTN and DUTN type of step disturbance respectively. The hopper is initiated with fixed point.

During the normal motion, the foot height, y_f at the end of flight phase 2 is 0.335 m. So, we introduced maximum step of 0.33 m size in up direction. The simulation results of the hopper subjected to these maximum step height is shown in the **Fig. 4.3**. Hopper state ($\mathbf{x}$), actuator angular velocity (w_r) and control input ($\mathbf{u}$) converges to steady state value. It is observed that in all neither w_r nor the control input hit the operating limits.

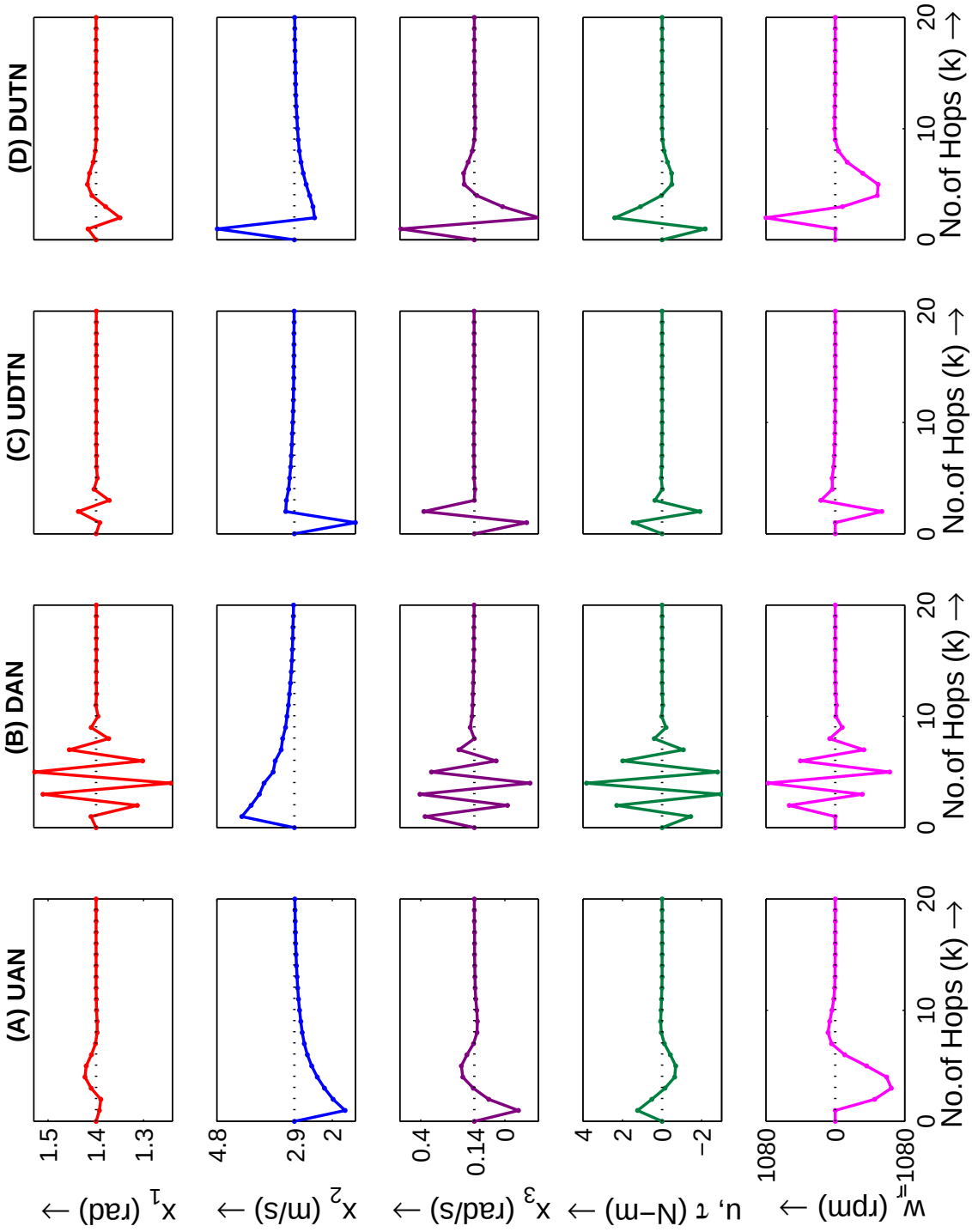

Figure 4.3: Full order classical state feedback controller applied to stabilise the hopping model, with a rotary actuator and subjected to step disturbance [(A): UAN (H_{step} = 0.29 m), (B): DAN (H_{step} = −0.46 m), (C): UDTN (H_{step} = 0.33 m), (D): DUTN (H_{step} = −0.73 m) and Dotted line: Nominal values].

4.4 Solution of Optimal SFB Control Problem:

Here we consider the same optimal SFB control problem as discussed in Section 3.4. The objective of this optimal control problem is to find the control $\triangle u(k))^*$ such that the weighted quadratic norm $||\mathcal{J}||$ is minimized subsequently to regulate state. The nonlinear approximate model $\left(\widehat{f}\right)$ is as given in (3.19). (A, B) pair is as given in equation (4.12). D_1 and D_2 are numerically determined as,

$$D_1{}^T = \begin{bmatrix} 0.3339 & -8.2284 & 0.1950 \\ -0.0197 & 0.0478 & -0.0470 \\ 0.0681 & -0.3864 & 0.1440 \\ 0.0016 & 0.0059 & 0.0063 \\ 0.2455 & 0.3571 & 1.7055 \\ 0.2485 & -1.9287 & 0.4876 \\ 0.0510 & -0.1443 & 0.1353 \\ 0.3242 & 0.1716 & 1.1122 \\ 0.0731 & 0.0342 & 0.2380 \\ 0.0109 & -0.0160 & 0.0313 \end{bmatrix}, \quad D_2{}^T = \begin{bmatrix} -5.7681 & -4.0440 & -26.4422 \\ -0.0058 & -0.0761 & -0.0430 \\ 0.0033 & -0.0066 & -0.0524 \\ -0.0451 & 5.9674 & 0.9272 \\ 0.2299 & -3.0807 & 0.2002 \\ -2.8194 & -2.0850 & -10.7354 \\ -0.4510 & -0.1095 & -1.5150 \\ 0.0156 & 0.0623 & 0.0073 \\ 0.0527 & 0.0764 & 0.3128 \\ 0.0118 & 0.0649 & 0.0833 \\ -0.7371 & -0.5012 & -2.5937 \\ 0.0494 & -0.5128 & 0.0717 \\ 0.0223 & 0.0116 & 0.0609 \\ 0.0035 & -0.7448 & -0.1197 \\ -0.0014 & -0.1638 & -0.0428 \\ 0.0083 & -0.0619 & 0.0066 \\ 0.1593 & -1.4535 & 0.2792 \\ 0.0323 & -0.2556 & 0.0680 \\ -0.0587 & -0.0569 & -0.2180 \\ 0.0088 & -0.0767 & 0.0165 \end{bmatrix}$$

We selected W, symmetric positive definite weighting matrix as,

$$W = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.005 & 0 \\ 0 & 0 & 0.05 \end{bmatrix} \tag{4.16}$$

We selected $U_l = -4$ N-m and $U_u = 4.0$ N-m. Then optimum scalar value of $\triangle u(k))^*$ is determined based on $\triangle u \in [U_l, U_u]$, where it gives minimum norm, $||\mathcal{J}||$, value.

The sequence of evolved state and control input when optimal state feedback control is applied to actual hopping model is presented in **Fig. 4.4**. In this case, we

considered maximum possible step disturbance.

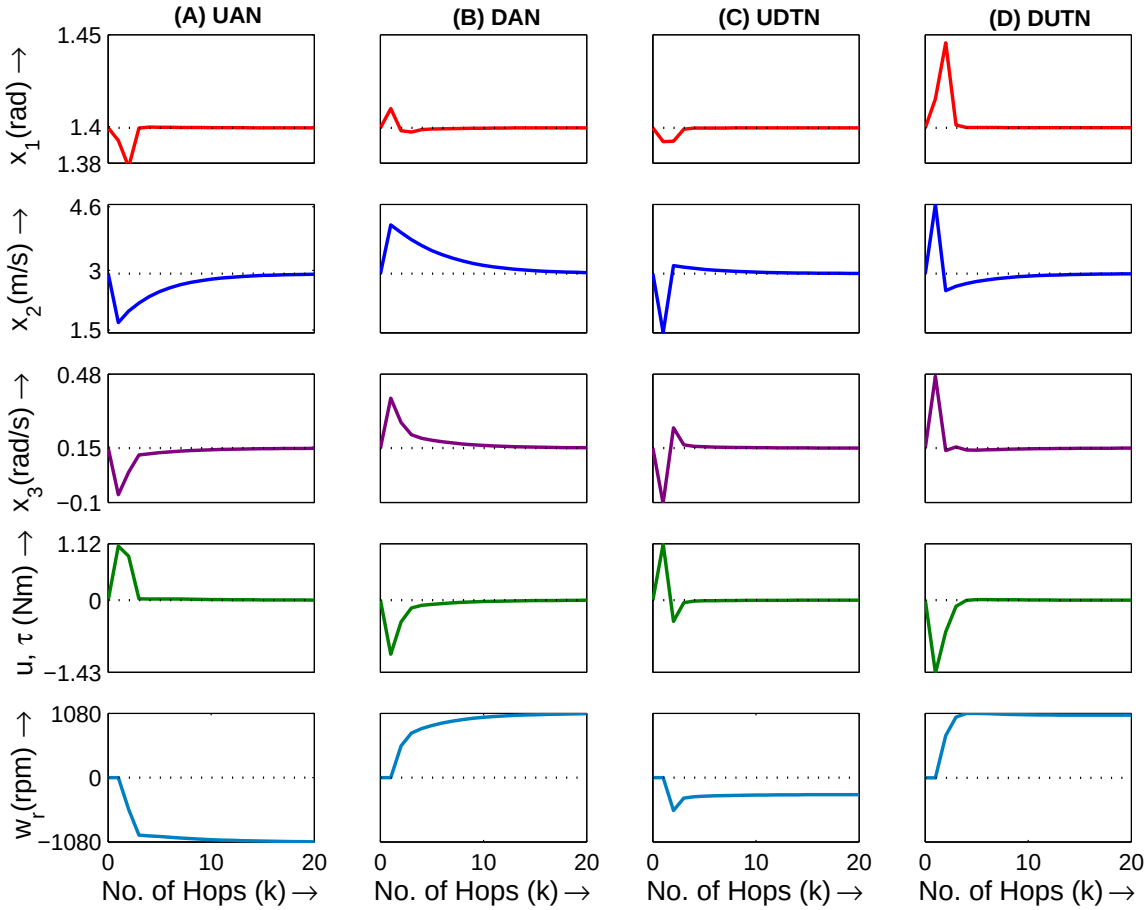

Figure 4.4: Full order optimal SFB controller applied to the hopping model, with a rotary actuator and subjected to step disturbance [(A): UAN (H_{step} = 0.29 m), (B): DAN (H_{step} = −0.44 m), (C): UDTN (H_{step} = 0.33 m), (D): DUTN (H_{step} = −0.67 m) and Dotted line: Nominal values].

The basin of attraction of previously designed full classical SFB controller and the current optimal controller is reported in the Table 4.3. Here it can be noted that the basin of attraction of optimal SFB controller is almost matching to the classical SFB controller. Also, the control effort is minimum and the actuator angular velocity is bounded. But, w_r is not returning to desired 0 steady state value. We also observed similar response, even with considering nonlinear approximate model with augmented state and selecting various combination of $W_{(4\times4)}$ matrix. The actuator dynamics follows its inherent dynamics given in (4.8). The justification goes as below:

Table 4.3: Basin of attraction of Classical SFB and optimal controller (rotary actuator)

Type of Step Disturbance	Classical SFB controller	Optimal controller
UAN	+0.29 m	+0.29 m
DAN	−0.44 m	−0.46 m
UDTN	+0.33 m	+0.33 m
DUTN	−0.67 m	−0.73 m

Consider (4.8), and let, $K_r = T_a/I_r = 52$, and in terms of deviational form,

$$\triangle w_r(k+1) = \triangle w_r(k) - K_r \triangle u(k)$$

Let, $\triangle u(k) = p \triangle w(k)$. Then,

$$\triangle w_r(k+1) = (1 - 52p) \triangle w(k)$$

The actuator dynamics will be asymptotically stable if,

$$0 < p < \left(\frac{1}{52}\right) \tag{4.17}$$

Thus, the condition (4.17) is not satisfied all times during determining optimum $\triangle u(k))^*$ in the solution of SFB control problem. This problem can be solved by using constraint optimization algorithm.

4.5 Discrete-time Sliding Mode Control:

In order to devise DSMC controllers, here we considered, the augmented linear and uncertain linear discrete time model as,

$$\triangle \mathbf{x}_e(k+1) = \mathbf{A}_e \triangle \mathbf{x}_e(k) + \mathbf{B}_e \triangle \mathbf{u}(k) \tag{4.18}$$

$$\triangle \mathbf{x}_e(k+1) = \mathbf{A}_e \triangle \mathbf{x}_e(k) + \mathbf{B}_e \triangle \mathbf{u}(k) + \widehat{d}(k) \tag{4.19}$$

Switching vector design:

In both DSMC approaches, the switching vector G is determined using the eigenvalue

assignment technique as described in Section 3.5.1. We selected three dominant poles of the previously designed classical SFB controller as reduced order poles,

$$P_{rd} = [0.2071, \quad 0.8010, \quad 0.4869]$$

The corresponding switching vector is,

$$G = \begin{bmatrix} 5.2774 & -0.0060 & 0.6285 & 0.0019 \end{bmatrix}.$$

DSMC Approach 1 (Using Gao's Reaching Law):

This approach use the linear model (4.18). The sliding manifold s is defined as:

$$\Psi = \Big\{ \triangle \mathbf{x}_e(k) || s(k) = G \triangle \mathbf{x}_e(k) = 0 \Big\}$$

The Gao's reaching law [24] and the control law is:

$$s(k+1) - s(k) = -\epsilon\tau \text{sgn}(s(k)) - q\tau s(k)$$

$$\triangle \mathbf{u}(k) = -(G\mathbf{B}_e)^{-1} \left[G\mathbf{A}_e \triangle \mathbf{x}_e(k) - G \triangle \mathbf{x}_e(k) + q\tau s(k) + \epsilon\tau \text{sgn}(s(k)) \right]$$

We selected, $\epsilon = 0.0001$, $q = 0.7$, $\tau = 1$.

DSMC Approach 2 (Using Saturation Type Reaching Law):

This approach use the linear model (4.19). The sliding manifold s is defined as the boundary layer,

$$\Psi = \Big\{ \triangle \mathbf{x}_e(k) \| s(k) = |G \triangle \mathbf{x}_e(k)| \leq \phi \Big\}$$

The approach use saturation type reaching law and control law as:

$$s(k+1) - s(k) = -K \text{sat} \left(\frac{s(k)}{\phi} \right)$$

$$\triangle \mathbf{u}(k) = -(G\mathbf{B}_e)^{-1} \left[G\mathbf{A}_e \triangle \mathbf{x}_e(k) - Gs(k) + K \text{sat} \left(\frac{s(k)}{\phi} \right) \right]$$

As mentioned in Section 3.5.3, $\gamma = 0.31$. Finally, we selected remaining design parameters as, $\phi = 1.7601$, $K = 0.3120$, $\Delta t = 1$, $\varepsilon = 0.001$.

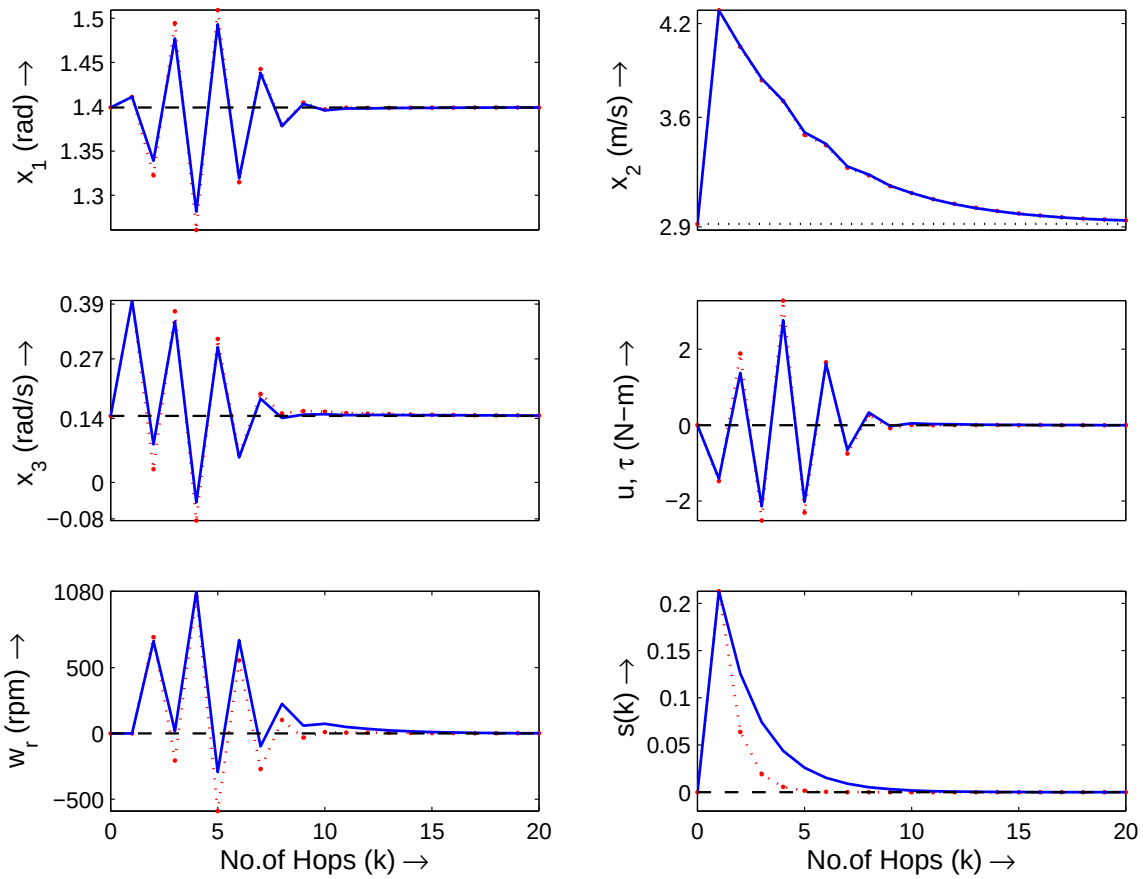

Figure 4.5: FSB DSMC applied to the hopping model, with a rotary actuator and subjected to DAN type step disturbance ($H_{step} = -0.50$ m), [Dot-dash: DSMC appraoch 1, Solid line: DSMC appraoch 2 and Dashed line: Nominal values]

Simulation results:

Two numerical simulations are presented in this Section. The simulation result shown in **Fig. 4.5** is comparing the performance of DSMC approach 1 and 2. In this case, the hopper is subjected to the DAN step disturbance ($H_{step} = 0.50$ m). It can be seen that the control effort required in DSMC approach 2 is lesser than the approach 1. The switching function quickly converges to $s = 0$ line in approach 2.

The simulation result in **Fig. 4.6** is for the case in which the hopper is subjected to DUTN step disturbance. Here possible maximum step size is introduced during motion with DSMC controller 1 and 2. It can be observed here also that the DSMC approach 2 utilizes less control effort. Also it is to be noted that in all cases the actuator angular velocity is within bounds and comes to rest at steady state as expected. The basin of attraction of these DSMC controllers is reported in the Table 4.4.

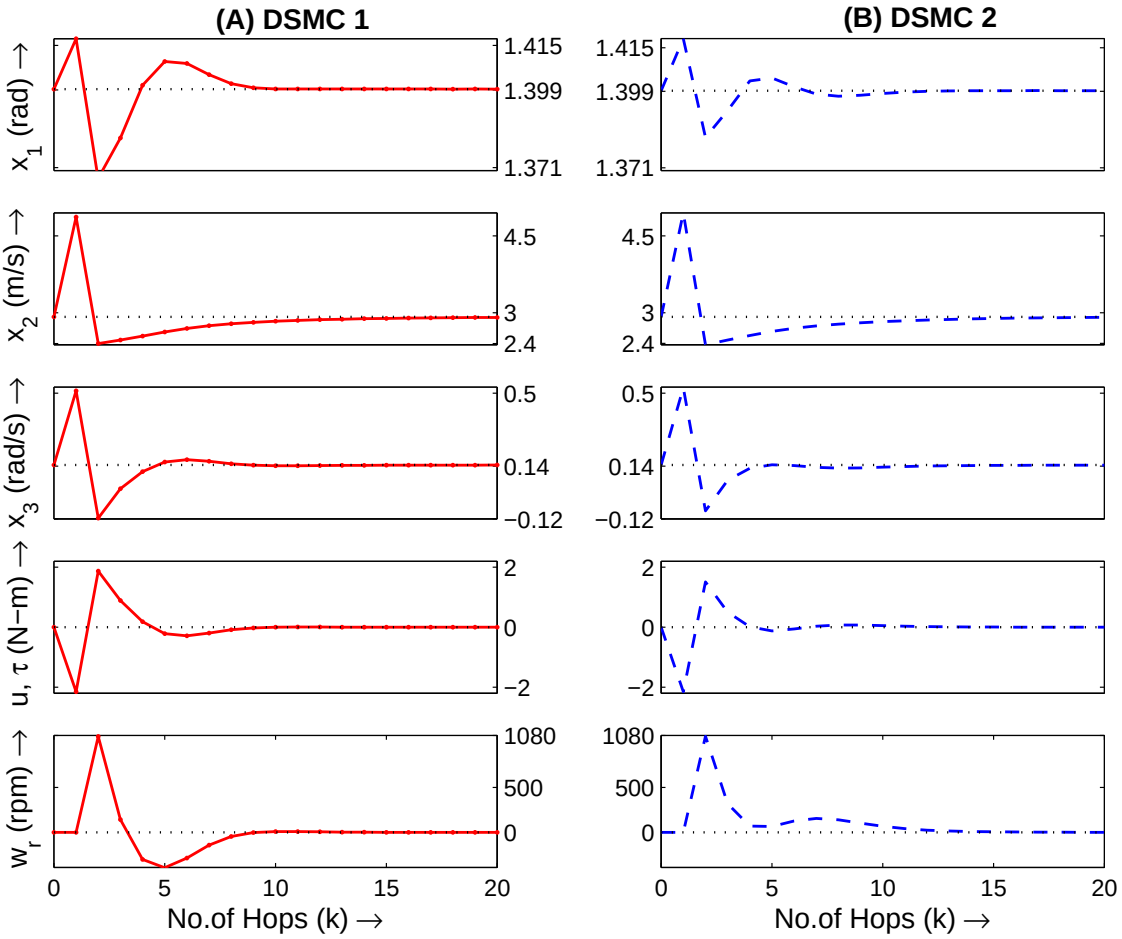

Figure 4.6: FSB based DSMC applied to the hopping model, with a rotary actuator and subjected to DUTN type of step disturbance, [(A): DSMC appraoch 1 ($H_{step} = -0.77$ m), (B): DSMC appraoch 2 ($H_{step} = -0.81$ m) and Dotted line: Nominal values].

Table 4.4: Basin of attraction of DSMC SFB controller with rotary actuator

Type of Step Disturbance	DSMC approach 1	DSMC approach 2
UAN	+0.29 m	+0.29 m
DAN	−0.50 m	−0.50 m
UDTN	+0.33 m	+0.33 m
DUTN	−0.77 m	−0.81 m

4.6 Conclusions:

This Chapter investigated the proposed hopping system with the rotary actuator. The spring charging is fixed during each hop. It is observed that this single DOF actuation

has strong maneuvering action on hopping model during flight phase. This actuator exert the torque permitting the model to reorient. The open-loop model is still unstable and pose a challenging problem to stabilize the hopper. Also, it is desired that the angular velocity of the actuator should not be unbounded. In that context, we proposed a simple state feedback stabilizing controller. The optimal state feedback controller utilize the non-linear approximated model. It has the basin of attraction more or less to that of classical SFB controller. The main advantage of the optimal controller is that it can be used when the control input is available in digital form. While, its disadvantage is more computational time.

Also, we presented two full SFB based DSMC strategies, which utilizes linear and uncertain linear augmented approximate models. The uncertainty is of unmatched type. In both DSMC approaches, control effort required is moderate than Classical SFB controller. Furthermore the present study shows that the basin of attraction of these FSB controllers is much larger than the linear actuated model. Now, the major concern is about the number of hops required to stabilize the model. It is also worthwhile persuing the study by extending the rotary actuated model with linear actuator. It may give the solution of our concern.

Chapter 5

State Feedback Control of SLOM Hopper with Two Actuators: Linear and Rotary

5.1 Introduction:

The under-actuated hopping system investigated previously in this thesis uses either a linear or a rotary actuator. It seems that there is possibility of improved performance of hopping system using these both actuators combined. It may result in larger basin of attraction as well as stable hopping motion with a few number of hops, as compared to using these actuators independently. In this Chapter, we focuses on the hopping system with such two actuators. There are numerous studies [4], [3] focused on the use of two actuators in symmetrical configurations. The major role of the linear actuator to control the hopping height, while the rotary actuator is used to steer the hopper to locomote in forward direction. This is a first attempt to investigate offset-mass one-legged hopping system with two actuators. Further, the proposed system is under-actuated.

Further discussion in this Chapter is organized as follows: In Section 5.2, the proposed under-actuated hopping model is described. The linear deviational model is derived to investigate the stability. In Section 5.3 we describe a classical FSB controller assuming that the all states are available for measurement. In Section 5.4, two DSMC based FSB controllers are suggested to stabilize the proposed model. We concluded the investigation presented in this Chapter in Section 5.5.

5.2 Description of the Proposed Model:

The hopping model with both linear and rotary actuator is shown in **Fig. 5.1**. The kinetic arrangement of proposed robot is the same as described in Section 2.1.

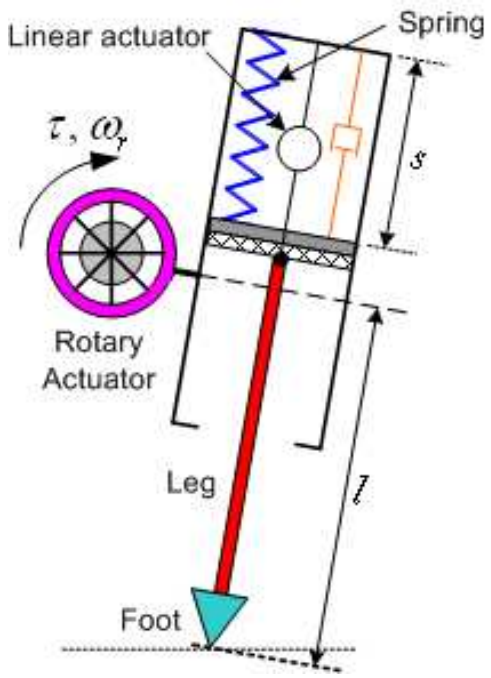

Figure 5.1: A schematic diagram of SLOM model with two actuators

The role of the linear actuator is to input the energy through spring retraction. In the present investigation, we selected the **step function type** torque trajectory for rotary actuation as discussed in Chapter 4. The rotary actuator mechanical specifications are the same as mentioned in the Table 4.2.

5.2.1 Augmented Linear Deviational model:

The procedure applied to formulate the linear deviational model is the same as described in the previous actuated model. Since, the hopping system uses two actuators, the linear deviational model in the present case has the form,

$$\triangle \mathbf{x}(k+1) = \mathbf{A}\triangle \mathbf{x}(k) + \mathbf{B}\triangle \mathbf{u}(k) \tag{5.1}$$

where, $\triangle \mathbf{x} = \mathbf{x} - \overline{\mathbf{x}}$ and $\triangle \mathbf{u} = \mathbf{u} - \overline{\mathbf{u}}$. Here, it is to be noted that, Also, $\mathbf{x} = [\ \theta \quad \dot{l} \quad \dot{\theta}\ ]^T$ and $\mathbf{u} = [u_1 \ \ u_2]^T$. u_1 is a spring retraction and u_2 is a torque applied during the partial flight phase.

The fixed point obtained will be same as earlier except the control inputs fixed

point value. The fixed point is,

$$\bar{\mathbf{x}} = \begin{bmatrix} 1.3992 \text{ rad} \\ 2.9194 \text{ m/s} \\ 0.1454 \text{ rad/s} \end{bmatrix}, \qquad \bar{\mathbf{u}} = \begin{bmatrix} 0.05 \text{ m} \\ 0.0 \text{ N-m} \end{bmatrix} \tag{5.2}$$

Note that, $T_a = t_2 - t_1$, is fixed during each hop. Since, A in does not depend upon control input, the same A in equation (2.20) is considered here. The vector B is obtained numerically considering the quintic spring retraction profile and step type torque profile. The $(\mathbf{A}, \mathbf{B})$ pair is,

$$\mathbf{A} = \begin{bmatrix} 1.7115 & -0.0325 & 0.9007 \\ 0.9110 & 0.7947 & 0.2425 \\ 6.2283 & -0.0661 & 3.0436 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0.3743 & 0.1509 \\ 11.6735 & 0.0312 \\ 3.0053 & 0.4813 \end{bmatrix} \tag{5.3}$$

By referring equation (4.4) and (4.8), and with the augmented state vector,

$$\mathbf{x}_e = \begin{bmatrix} \mathbf{x} \\ w_r \end{bmatrix}, \tag{5.4}$$

we can write,

$$\triangle \mathbf{x}_e(k+1) = \mathbf{A}_e \triangle \mathbf{x}_e(k) + \mathbf{B}_e \triangle \mathbf{u}(k) \tag{5.5}$$

where,

$$\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & 0_{(3\times 1)} \\ 0_{(1\times 3)} & 1 \end{bmatrix}, \qquad \mathbf{B}_e = \begin{bmatrix} \mathbf{B} \\ 0, \quad -(T_a/I_r) \end{bmatrix} \tag{5.6}$$

Using $I_r = 0.005$ kg-m^2 and $T_a = 0.26$ s, we have,

$$\mathbf{A}_e = \begin{bmatrix} 1.7115 & -0.0325 & 0.9007 & 0 \\ 0.9110 & 0.7947 & 0.2425 & 0 \\ 6.2283 & -0.0661 & 3.0436 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{B}_e = \begin{bmatrix} 0.3743 & 0.1509 \\ 11.6735 & 0.0312 \\ 3.0053 & 0.4813 \\ 0 & -52 \end{bmatrix} \tag{5.7}$$

The eigenvalues of this augmented matrix $\mathbf{A}_e$ are, $\lambda_1 = 4.8276$, $\lambda_2 = -0.0787$, $\lambda_3 = 0.8010$ and $\lambda_4 = 1.0000$. It is observed that the basin of attraction of the full SFB controller designed for $(\mathbf{A}_e, \mathbf{B}_e)$ model is very small (step size of order 0.05 m) in all cases. Hence, the SFB controller design presented in this study is for a linear deviational model, $(\mathbf{A}, \mathbf{B})$ model, instead of the augmented model. It is to be noted that in such SFB control system, it is expected that the angular velocity of rotary actuator does not come to rest at steady state.

5.3 Classical Full State Feedback Control System Design:

This Section focuses on the classical state feedback control system based on approximate linear deviational model in (5.1), and applied to nonlinear model (non-linear hybrid model). We assume that all state are measurable and available for feedback. It is observed that the linear model is completely state controllable i.e.

$$\text{Rank}\left(\begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A^2B}\end{bmatrix}\right) = 3.$$

The regulatory control law is,

$$\triangle\mathbf{u}(k) = -\mathbf{K_p}\triangle\mathbf{x}(k). \tag{5.8}$$

Controller design:

The linear model, the pair $(\mathbf{A}, \mathbf{B})$ has the original poles at,

$$\lambda_1 = 4.8276, \quad \lambda_2 = -0.0787, \quad \lambda_3 = 0.8010$$

We chosen the closed loop poles as, 0.5, 0.6, and 0.2. We obtained constant gain matrix, $\mathbf{K_p}$, as,

$$\mathbf{K_p} = \begin{bmatrix} 0.1560 & 0.0234 & -0.0073 \\ 6.8880 & -0.2222 & 6.0421 \end{bmatrix}. \tag{5.9}$$

Simulation results:

As we are looking for stabilizing the actual nonlinear hopping model, the control law derived based on linear model is applied to the actual model. Note that the fixed point is given in (5.2). In order to determine the basin of attraction of this controller, we simulated model considering different type of step disturbances. It is observed that the maximum step height (H_{step}) that can be introduced during motion is 0.31 m, -0.80 m, 0.33 m and -0.75 m in UAN, DAN, UDTN and DUTN type of step disturbance respectively. The hopper is initiated with fixed point.

The simulation results of the hopper subjected to these maximum step height is shown in the **Fig. 5.4**. Hopper state ($\mathbf{x}$) and control input ($\mathbf{u}$) converges to steady state value. As expected the actuator angular velocity (w_r) does not come to rest at steady state, but, remains within the operating limits.

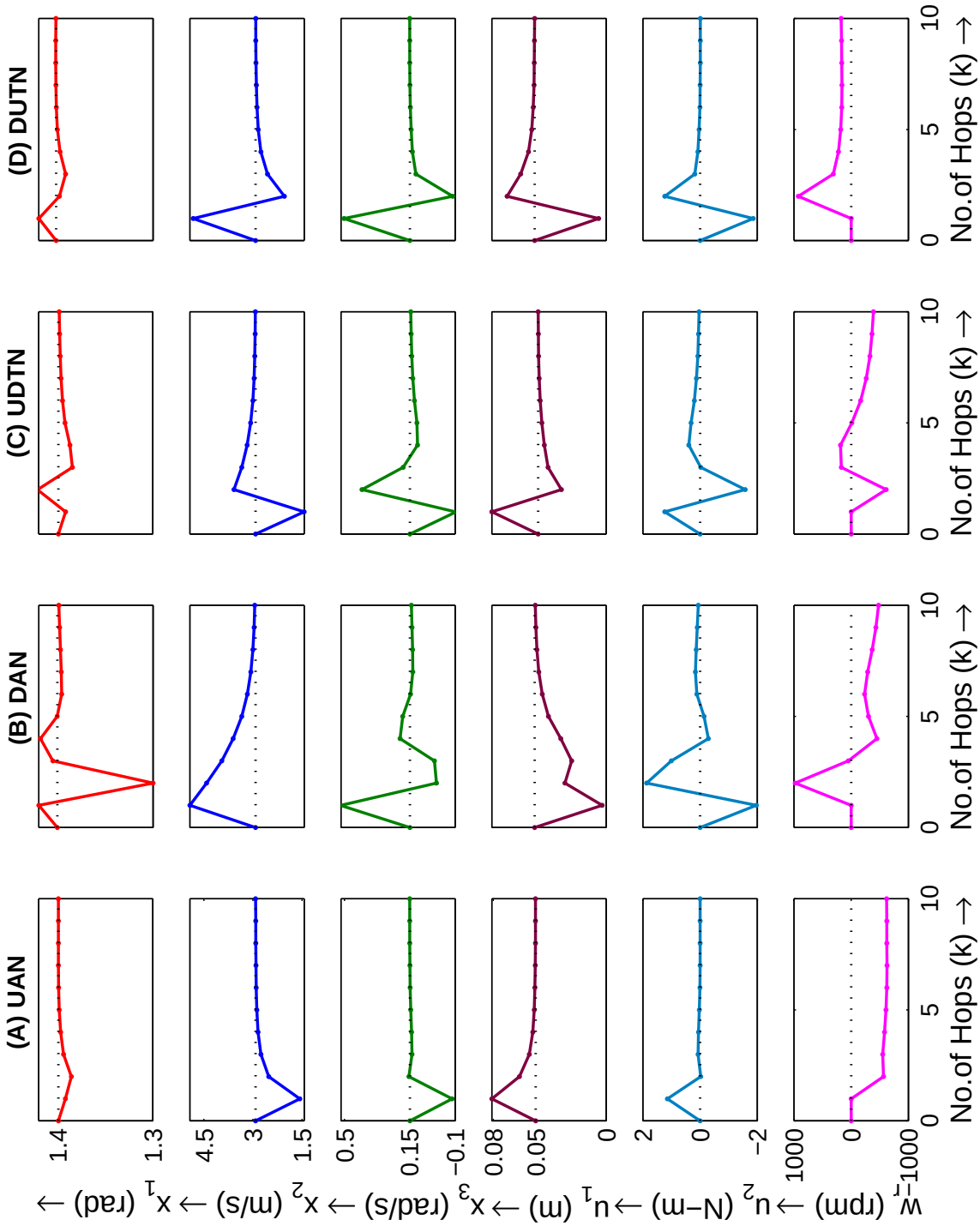

Figure 5.2: Full order classical state feedback controller applied to stabilise the hopping model, with two actuators and subjected to step disturbance [(A): UAN ($H_{step} = 0.31$ m), (B): DAN ($H_{step} = -0.80$ m), (C): UDTN ($H_{step} = 0.33$ m), (D): DUTN ($H_{step} = -0.75$ m) and Dotted line: Nominal values].

The nonlinear approximate model, $\left(\widehat{f}\right)$ (like in (3.19)) turns to be more complex in case of this multi-input system. So, in the present work, the optimal SFB control problem is not considered.

5.4 Discrete-time Sliding Mode Control:

In order to devise DSMC controllers, here we considered, two-input linear and uncertain linear discrete time model as,

$$\triangle \mathbf{x}(k+1) = \mathbf{A}\triangle \mathbf{x}(k) + \mathbf{B}\triangle \mathbf{u}(k) \tag{5.10}$$

$$\triangle \mathbf{x}(k+1) = \mathbf{A}\triangle \mathbf{x}(k) + \mathbf{B}\triangle \mathbf{u}(k) + \widehat{d}(k) \tag{5.11}$$

It is reported that the DSMC strategy applied for linear uncertain system with unmatched uncertainty [25] is applicable only for a single input system. In this thesis, we extended Misawa's approach for two-input hopping system. However, the analytical stability analysis is not carried out.

Switching vector design:

The present study deals with a two input system ($m = 2$). The nominal sliding mode system defined by equation (3.27) and (3.28) is of order 1 ($n - m$), with 2 control inputs. The control distribution matrix $A_{12} \in \mathbb{R}^{1\times 2}$ have rank 1. This enables to design a stable sliding surface by appropriate selection of F_1 in equation (3.31). However, it can be noted that rank of $A_{12} < m$. This interprets that the sliding system has redundant inputs which must be removed [36]. An appropriate method to design F_1 is 'robust eigenstructure assignment', where the sensitivity of assigned eigenvalues is bounded. Here, in both DSMC approaches, the switching vector G is determined using the above mentioned robust eigenstructure assignment technique(as described in [36], page no. 68). We selected a dominant pole out of the poles assigned for classical SFB controller design as reduced order poles, i.e.

$$P_{rd} = 0.6$$

The corresponding switching vector is,

$$G = \begin{bmatrix} 0.3708 & -0.9481 & -0.3765 \\ 6.7995 & 0.0747 & -1.1370 \end{bmatrix}.$$

DSMC Approach 1 (Using Gao's Reaching Law):

This approach use the linear model (5.10). The sliding manifold s is defined as:

$$\Psi = \Big\{ \triangle \mathbf{x}_e(k) || s(k) = G \triangle \mathbf{x}_e(k) = 0 \Big\}$$

The Gao's reaching law [24] and the control law is:

$$s(k+1) - s(k) = -\epsilon\tau \text{sgn}(s(k)) - q\tau s(k)$$

$$\triangle \mathbf{u}(k) = -(G\mathbf{B}_e)^{-1} \left[G\mathbf{A}_e \triangle \mathbf{x}_e(k) - G \triangle \mathbf{x}_e(k) + q\tau s(k) + \epsilon\tau \text{sgn}(s(k)) \right]$$

We selected, $\epsilon = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.001 \end{bmatrix}$, $q = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.7 \end{bmatrix}$, $\tau = 1.$

DSMC Approach 2 (Using Saturation Type Reaching Law):

This approach use the linear model (5.11). The sliding manifold s is defined as the boundary layer,

$$\Psi = \Big\{ \triangle \mathbf{x}_e(k) \| s(k) = |G \triangle \mathbf{x}_e(k)| \leq \phi \Big\}$$

The approach use saturation type reaching law and control law as:

$$s(k+1) - s(k) = -K \text{sat} \left(\frac{s(k)}{\phi} \right)$$

$$\triangle \mathbf{u}(k) = -(G\mathbf{B}_e)^{-1} \left[G\mathbf{A}_e \triangle \mathbf{x}_e(k) - Gs(k) + K \text{sat} \left(\frac{s(k)}{\phi} \right) \right]$$

As mentioned in Section 3.5.3, $\gamma = 0.31$. Finally, we selected remaining design parameters as, $\phi = \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}$, $K = \begin{bmatrix} 0.3120 & 0 \\ 0 & 0.3120 \end{bmatrix}$, $\Delta t = 1$, $\varepsilon = 0.001$.

Simulation results:

Two numerical simulations are presented in this Section. The simulation result shown in **Fig. 5.3** is the performance of DSMC approach 1. In this case, the hopper is subjected to different type of step disturbance (H_{step}). The simulation result in **Fig. 5.4** is for the DSMC approach 2. From the simulation results, it can be observed that the two-input uncertain system can be controlled by extending Misawa's approach [25]. Here possible maximum step size is introduced during motion. It can be observed here also that the DSMC approach 2 utilizes less control effort than approach 1. Also it is

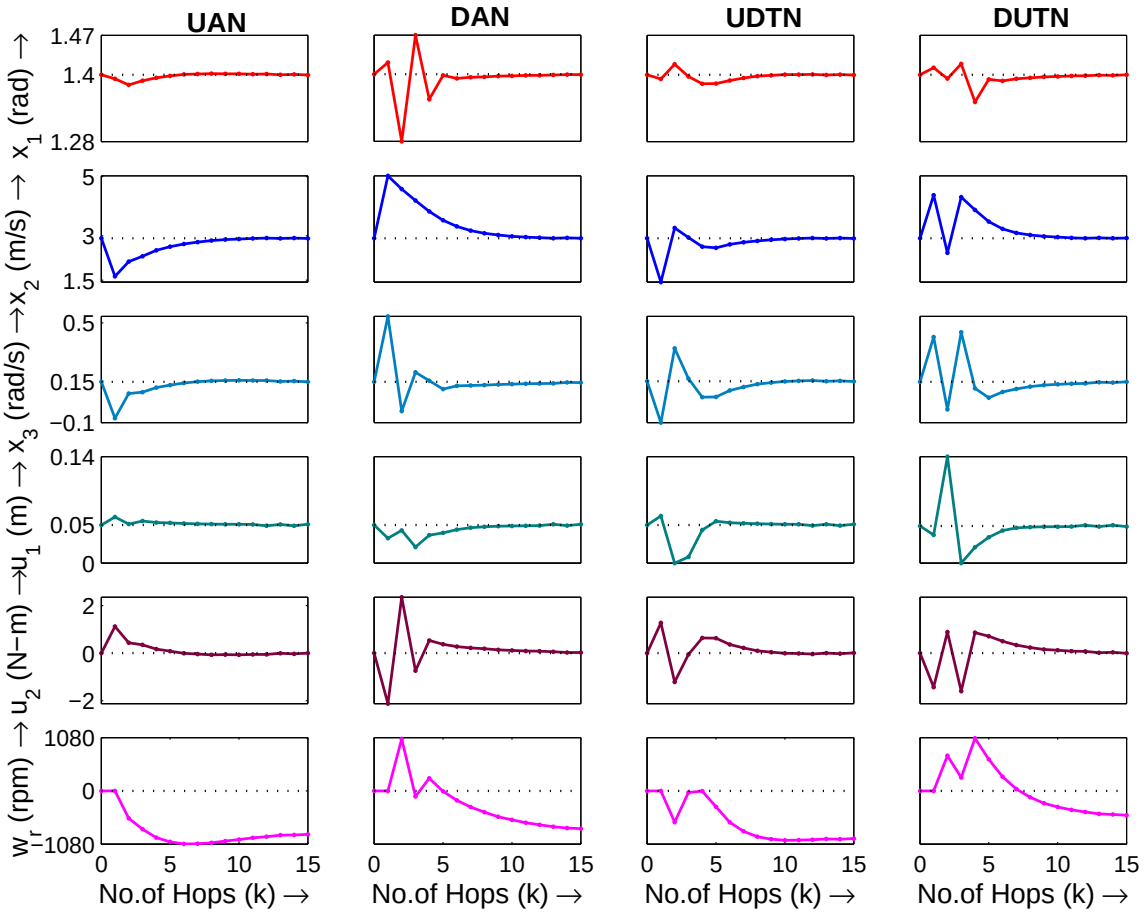

Figure 5.3: FSB DSMC approach 1 applied to the hopping model, with two actuators and subjected to step disturbance [(A): UAN (H_{step} = 0.30 m), (B): DAN (H_{step} = −0.85 m), (C): UDTN (H_{step} = 0.33 m), (D): DUTN (H_{step} = −0.54 m) and Dotted line: Nominal values].

Table 5.1: Basin of attraction of DSMC SFB controller with both actuators

Type of Step Disturbance	DSMC approach 1	DSMC approach 2
UAN	+0.30 m	+0.29 m
DAN	−0.85 m	−0.81 m
UDTN	+0.33 m	+0.29 m
DUTN	−0.54 m	−0.50 m

to be noted that in all cases the actuator angular velocity is within bounds . The basin of attraction of these two DSMC controllers is reported in the Table 5.1.

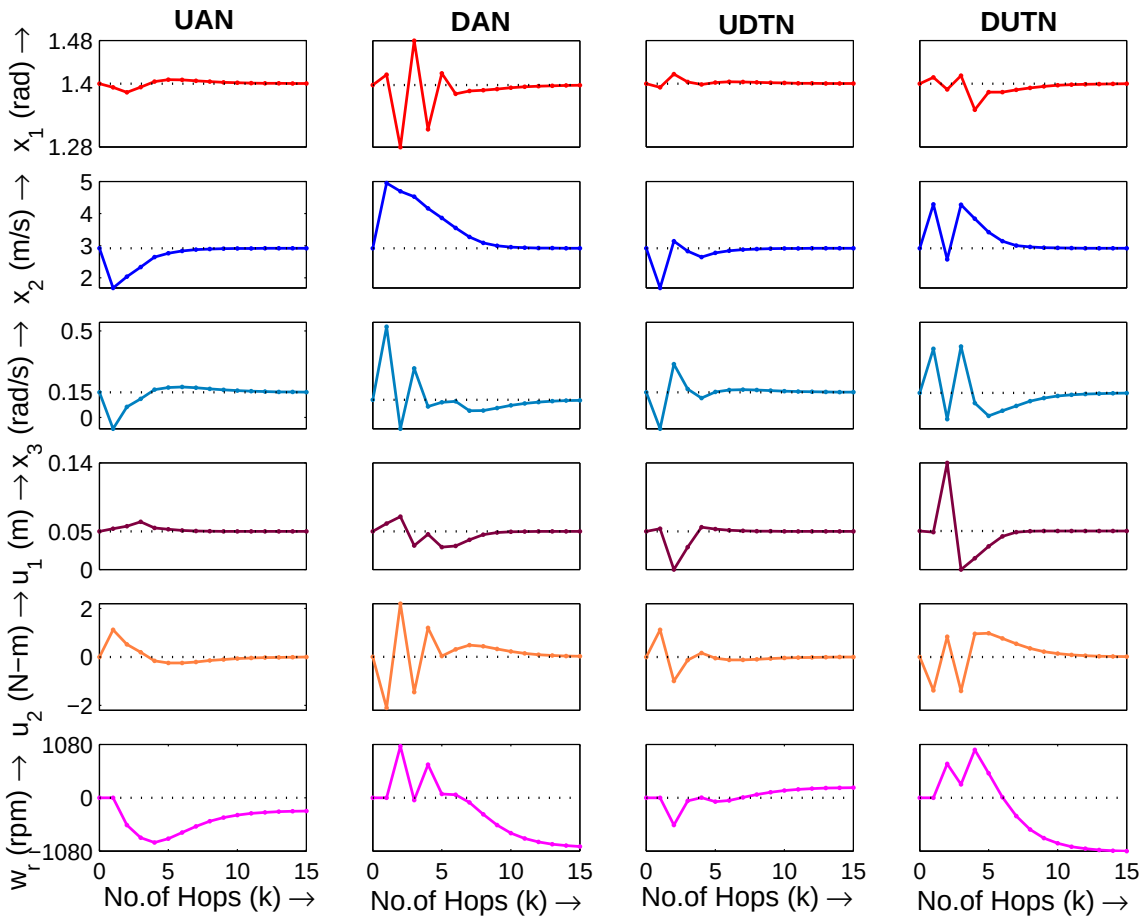

Figure 5.4: FSB based DSMC approach 2 applied to the hopping model, with two actuators and subjected to step disturbance [(A): UAN ($H_{step} = 0.29$ m), (B): DAN ($H_{step} = -0.81$ m), (C): UDTN ($H_{step} = 0.29$ m), (D): DUTN ($H_{step} = -0.50$ m) and Dotted line: Nominal values].

5.5 Conclusions:

This Chapter investigated the proposed hopping model with two actuators. The linear actuator retracts the spring to input the energy during each hop. Also, the rotary actuator help to leg orientation. The linear deviational model is unstable and pose a challenging problem to stabilize the nonlinear hopper. We presented the full SFB based classical and two DSMC strategies, which utilize linear and uncertain linear augmented approximate models. The uncertainty is of unmatched type. Also, it can be observed that the basin of attraction of these strategies using two actuators is more or less same to the strategies using only the rotary actuator. However, it can be observed that the number of hops required to stabilize the hopper is less in present under-actuated system than the previously discussed under-actuated systems.

Chapter 6

Conclusions and Recommendations

In this chapter the conclusions are presented.

6.1 Conclusions:

In an attempt to understand the basic operating principles and dynamic balancing for locomotion of single legged hopping robots, we invsetigated a simple conceptual planar single-legged hopping robot. We proposed an asymmetrical single -legged hopping robot, referred as "Springy Legged Offset-Mass" hopper. This is a special type of configuration which poses the ability for compensation of forward pitching. It consisted of a body and a leg, which are in contact with a sufficiently wide horizontal ground surface. The linear spring is incorporated between leg and body. The major conclusions from this thesis are summarized as below:

- We presented a generic kinetic model of the asymmetrical one-legged hopping robot. It is different from Shanmuganathan's geometrical model [10].
- Based on constrained Euler-Lagrangian methodology, a mathematical model of SLOM hooping robot is presented for passive as well as active configurations. We defined a 3-dimensional Poincaré map in order to study the dynamic behaviour of proposed model. we validated the model based on basic principles of physics.
- We determined periodic motions by solution of equations for fixed point of a Poincaré map and local stability of periodic motion. The study is focused on the stabilization of hopper to the fixed point.
- Three different actuated models are investigated;, utilizing: (i) the simple one linear actuator, (ii) the reaction wheel and (iii) both linear and rotary actuators,

to realize the continuous hopping motion.

- We developed linear state feedback controller as well as different state based observers and studied stability using simulations.

- An approximated non-linear model is formulated, which is an extension of the approach followed for formulating linear model using linearized Poincaré map. This study uses a Taylor series approximation technique. This model is used to solve the optimal state feedback control problem.

- We developed a discrete-time sliding mode full state feedback controller based on both linear and non-linear models. Moreover, to our knowledge, this is the first attempt to apply DSMC for single-legged hopping robot. Also, the Misawa's approach is extended for two-input system for stabilizing the hopper.

- In an attempt to set the basis for determining the basin of attractions of proposed controller, we suggested a simple approach to introduce the structured disturbance (fixed sized step) during motion.

The focus of research presented in the thesis is on a hopping system that can be minimally controlled, i.e., which utilize the natural passive dynamics for locomotion and require smaller control effort for energy loss compensation and stabilization. This proposed system is under-actuated and may become practically realizable if the power pack requirements are small enough. Since the research in practical systems would also have to deal with real issues of dealing with the environmental factors like uneven ground or obstructions and would also have to deal with issues of sensor's input and fusion. The step disturbance considered in the stability analysis is capturing the sufficient unevenness of the ground surface. The major conclusion of this study is that a very simple control techniques can control and stabilize the nonlinear hopping model, which is highly nonlinear and hybrid dynamical system.

6.2 Recommendations and scope for future work:

The following list gives a idea about the future scope of research work that can be carried out in continuation of SLOM's investigation:

1. It is well known that the present SLOM configuration is unable to perform in-place hopping. It is recommended to modify the current configuration in such a

way that it is possible to perform in-place hopping, forward hopping and backward hopping. This can be achieved by adding one more degree of freedom to the current SLOM configuration. Either prismatic joint or rotary tail placed at hip position can achieve this objective.

2. It is worthwhile to investigate the 3-dimensional motion of the proposed model.
3. Real hopping systems have to perform on the field where there is dirt between their toes. In modelling the hopper, it is practical to focus on a ground model (considering friction, damping and unevenness of ground surface).
4. In the present study, it is assumed that the control input is a continuous signal except in optimal state feedback controller design. It is desired to devise the stabilization strategy with availability of digital control input. For example, suppose in case of linear actuated hopper, the spring retraction is digital signal with a 1 mm quantization.
5. It is practical to design the control strategies to stabilize the proposed hopping system with a saturated input.
6. It is observed that the hopping gait of the proposed hopper is governed by the mechanical parameters. In order to attain better stability of the hopping motion, it is important to design these mechanical parameters appropriately. For example, Kuswadi et.al. [11], determined these parameters based on a numerically obtained map, referred as "transition map". Kuswadi's approach is dealing with a single DOF state space model. It is recommended to evaluate these parameters using a nonlinear parameter optimization technique which deal with proposed hopper model.
7. C. K. Reddy and R. Pratap reported a global stability analysis of a two-mass passive hopper using the non-dimensional variables in their dynamic model [45]. It is recommended to perform non-dimensional analysis of the proposed hopper. It will give insight of an influence of the possible normalized quantities, i.e. $\frac{m_b}{m_l}$, $\frac{d}{h}$, $\frac{I_b+I_l}{m_l d^2}$, $\frac{m_l g}{K_s d}$, $\frac{L_0-h}{d}$, etc, on the hopping gait.
8. The extension of the techniques for the emerging field of sliding mode observer or output feedback sliding mode based control is also avenue worth pursuing.
9. The area of passivity based control in proposed hopping system also offers interesting possibility.
10. Neural network, fuzzy-logic and genetic algorithm based control algorithms are gaining acceptance in the recent past. So, it is recommended to evaluate the functionality of these controllers for attaining hopping gait stability.

Appendix A

Kinematic and Dynamic Equations

In this **Appendix**, we formulate the dynamic model of unactuated (passive) system as shown in the **Fig. A.1**.The core objective of modelling is to derive a mathematical model (set of equations) in order to simulate the dynamical behaviour of the robot.

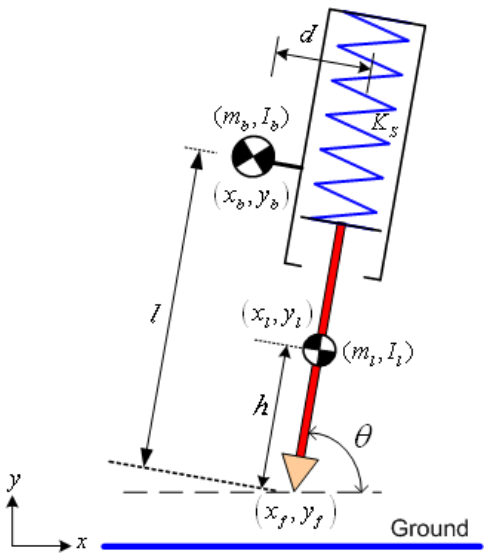

Figure A.1: A posture of *SLOM* hopper considered for deriving different equations

It is possible to derive different sets of equations of motion for *flight* and *stance* phases independently. However, we have derived a single set of dynamic equations using *Euler-Lagrangian* (EL) approach assuming four DOF: x_b, y_b, l and θ. The behaviour of robot during different regimes is based on the constraints imposed during that state, which is well illustrated in the last Section.

A.1 Kinematic Equations:

A.1.1 Position:

$$x_b = x_f + l\cos\theta - d\sin\theta \tag{A.1}$$

$$y_b = y_f + l\sin\theta + d\cos\theta \tag{A.2}$$

$$x_b = x_l + (l-h)\cos\theta - d\sin\theta \tag{A.3}$$

$$y_b = y_l + (l-h)\sin\theta + d\cos\theta \tag{A.4}$$

$$x_l = x_b - (l-h)\cos\theta + d\sin\theta \tag{A.5}$$

$$y_l = y_b - (l-h)\sin\theta - d\cos\theta \tag{A.6}$$

A.1.2 Velocity:

$$\dot{x_b} = \dot{x_f} + \dot{l}\cos\theta - l\sin\theta\dot{\theta} - d\cos\theta\dot{\theta} \tag{A.7}$$

$$\dot{y_b} = \dot{y_f} + \dot{l}\sin\theta + l\cos\theta\dot{\theta} - d\sin\theta\dot{\theta} \tag{A.8}$$

$$\dot{x_l} = \dot{x_b} - \dot{l}\cos\theta + (l-h)\sin\theta\dot{\theta} + d\cos\theta\dot{\theta} \tag{A.9}$$

$$\dot{y_l} = \dot{y_b} - \dot{l}\sin\theta - (l-h)\cos\theta\dot{\theta} + d\sin\theta\dot{\theta} \tag{A.10}$$

Let

$$A(l,\theta) = (l-h)\sin\theta + d\cos\theta \tag{A.11}$$

$$B(l,\theta) = (l-h)\cos\theta - d\sin\theta \tag{A.12}$$

Hence,

$$\dot{x_l} = \dot{x_b} - \dot{l}\cos\theta + A(l,\theta)\dot{\theta} \tag{A.13}$$

$$\dot{y_l} = \dot{y_b} - \dot{l}\sin\theta - B(l,\theta)\dot{\theta} \tag{A.14}$$

$$y_l = y_b - A(l,\theta) \tag{A.15}$$

A.2 Hopper Energy Equations:

In general we can write the following energy equations for our *SLOM* hopper,

$$\begin{aligned} Total\ Energy &= Kinetic\ Energy + Potential\ Energy \\ T.E. &= K.E. + P.E. \end{aligned} \tag{A.16}$$

Kinetic Energy:

$$T = \frac{1}{2}I_b\dot{\theta}^2 + \frac{1}{2}m_b(\dot{x_b}^2 + \dot{y_b}^2) + \frac{1}{2}I_l\dot{\theta}^2 + \frac{1}{2}m_l(\dot{x_l}^2 + \dot{y_l}^2) \tag{A.17}$$

Potential Energy:

$$V = \frac{1}{2}K_s(l - L_0)^2 + m_b g y + m_l g y_l \tag{A.18}$$

Total Energy:

$$\begin{aligned} E &= \frac{1}{2}(I_b + I_l)\dot{\theta}^2 + \frac{1}{2}m_b(\dot{x_b}^2 + \dot{y_b}^2) \\ &+ \frac{1}{2}m_l(\dot{x_l}^2 + \dot{y_l}^2) + \frac{1}{2}K_s(l - L_0)^2 + m_b g y_b + m_l g y_l \end{aligned} \tag{A.19}$$

A.3 Motion Equations Using EL Formulation:

The equations of dynamic motion of hopping robot in EL form are given as,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q_i}}\right) - \frac{\partial L}{\partial q_i} = \tau \tag{A.20}$$

where, q_i, the generalized co-ordinate, and τ, the generalized external force applied to a system. $L = T - V$, Lagrangian. For passive dynamics we have, $\tau = 0$. Hence,

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q_i}}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = 0 \tag{A.21}$$

In general, we have the four generalized co-ordinate as,

$$q_i = [\ x_b \quad y_b \quad l \quad \theta\ ]^T \tag{A.22}$$

Considering (A.17)-(A.18) and (A.13)-(A.15), we have,

$$\begin{aligned} T(q,\dot{q}) &= \frac{1}{2}(I_b + I_l)\dot{\theta}^2 + \frac{1}{2}m_b({\dot{x_b}}^2 + {\dot{y_b}}^2) \\ &\quad + \frac{1}{2}m_l(\dot{x_b} - \dot{l}\cos\theta + A(l,\theta)\dot{\theta})^2 \\ &\quad + \frac{1}{2}m_l(\dot{y_b} - \dot{l}\sin\theta - B(l,\theta)\dot{\theta})^2 \end{aligned} \tag{A.23}$$

$$V = \frac{1}{2}K_s(l - L_0)^2 + (m_b + m_l)gy_b - m_l g A(l,\theta) \tag{A.24}$$

Also, it may be noted that, the following relations will be useful to derive the Lagrangian equations,

$$\dot{A}(l,\theta) = \frac{d}{dt}(A(l,\theta)) = \dot{l}\sin\theta + B(l,\theta)\dot{\theta} \tag{A.25}$$

$$\dot{B}(l,\theta) = \frac{d}{dt}(B(l,\theta)) = \dot{l}\cos\theta - A(l,\theta)\dot{\theta} \tag{A.26}$$

$$\frac{\partial}{\partial\theta}(A(l,\theta)) = B(l,\theta) \tag{A.27}$$

$$\frac{\partial}{\partial\theta}(B(l,\theta)) = -A(l,\theta) \tag{A.28}$$

$$A(l,\theta)\cos\theta - B(l,\theta)\sin\theta = d \tag{A.29}$$

$$A(l,\theta)\sin\theta + B(l,\theta)\cos\theta = (l - h) \tag{A.30}$$

$$A^2(l,\theta) + B^2(l,\theta) = (l-h)^2 + d^2 \tag{A.31}$$

$$A(l,\theta)\frac{\partial}{\partial\theta}\Big(A(l,\theta)\Big) + B(l,\theta)\frac{\partial}{\partial\theta}\Big(B(l,\theta)\Big) = 0 \tag{A.32}$$

★ For $\underline{\underline{q_1 = x_b}}$,

$$\frac{\partial T}{\partial \dot{x_b}} = m_b\dot{x_b} + m_l\Big(\dot{x_b} - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)$$

$$\frac{\partial T}{\partial \dot{x_b}} = (m_b + m_l)\dot{x_b} - m_l\dot{l}\cos\theta + m_l A(l,\theta)\dot{\theta}$$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x_b}}\right) = (m_b + m_l)\ddot{x_b} - m_l\ddot{l}\cos\theta + m_l\dot{l}\sin\theta\dot{\theta} &+ m_l A(l,\theta)\ddot{\theta} \\ &+ m_l\Big(\dot{l}\sin\theta + B(l,\theta)\dot{\theta}\Big)\dot{\theta} \end{aligned}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x_b}}\right) = (m_b + m_l)\ddot{x_b} - m_l\ddot{l}\cos\theta + m_l A(l,\theta)\ddot{\theta} + 2m_l\dot{l}\sin\theta\dot{\theta} + m_l B(l,\theta)\dot{\theta}^2$$

$$\frac{\partial T}{\partial x_b} = 0, \qquad \frac{\partial V}{\partial x_b} = 0$$

$$(m_b + m_l)\ddot{x}_b - m_l\ddot{l}\cos\theta + m_l A(l,\theta)\ddot{\theta} + 2m_l\dot{l}\sin\theta\dot{\theta} + m_l B(l,\theta)\dot{\theta}^2 = 0 \tag{A.33}$$

★ For $\underline{\underline{q_2 = y_b}}$,

$$\frac{\partial T}{\partial \dot{y}_b} = m_b\dot{y}_b + m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)$$

$$\frac{\partial T}{\partial \dot{y}_b} = (m_b + m_l)\dot{y}_b - m_l\dot{l}\sin\theta - m_l B(l,\theta)\dot{\theta}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{y}_b}\right) = (m_b + m_l)\ddot{y}_b - m_l\ddot{l}\sin\theta - m_l\dot{l}\cos\theta\dot{\theta} &- m_l B(l,\theta)\ddot{\theta} \\ &- m_l\Big(\dot{l}\cos\theta - A(l,\theta)\dot{\theta}\Big)\dot{\theta}\end{aligned}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{y}_b}\right) = (m_b + m_l)\ddot{y}_b - m_l\ddot{l}\sin\theta - m_l B(l,\theta)\ddot{\theta} - 2m_l\dot{l}\cos\theta\dot{\theta} + m_l A(l,\theta)\dot{\theta}^2$$

$$\frac{\partial T}{\partial y_b} = 0, \qquad \frac{\partial V}{\partial y_b} = (m_b + m_l)g$$

$$(m_b+m_l)\ddot{y}_b - m_l\ddot{l}\sin\theta - m_l B(l,\theta)\ddot{\theta} - 2m_l\dot{l}\cos\theta\dot{\theta} + m_l A(l,\theta)\dot{\theta}^2 + (m_b+m_l)g = 0 \tag{A.34}$$

★ For $\underline{\underline{q_3 = l}}$,

$$\frac{\partial T}{\partial \dot{l}} = m_l\Big(\dot{x}_b - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)(-\cos\theta) + m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)(-\sin\theta)$$

$$\frac{\partial T}{\partial \dot{l}} = m_l\begin{pmatrix} -\dot{x}_b\cos\theta - \dot{y}_b\sin\theta + \dot{l}\cos^2\theta + \dot{l}\sin^2\theta \\ -\Big(A(l,\theta)\cos\theta - B(l,\theta)\sin\theta\Big)\dot{\theta} \end{pmatrix}$$

$$\frac{\partial T}{\partial \dot{l}} = m_l(-\dot{x}_b\cos\theta - \dot{y}_b\sin\theta + \dot{l} - d\dot{\theta})$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{l}}\right) = m_l(-\ddot{x}_b\cos\theta + \dot{x}_b\sin\theta\dot{\theta} - \ddot{y}_b\sin\theta - \dot{y}_b\cos\theta\dot{\theta} + \ddot{l} - d\ddot{\theta})$$

$$\frac{\partial T}{\partial l} = m_l\Big(\dot{x}_b - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)(\sin\theta\dot{\theta}) + m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)(-\cos\theta\dot{\theta})$$

$$\frac{\partial T}{\partial l} = m_l\begin{pmatrix} \dot{x}_b\sin\theta\dot{\theta} - \dot{l}\sin\theta\cos\theta\dot{\theta} + A(l,\theta)\sin\theta\dot{\theta}^2 \\ -\dot{y}_b\cos\theta\dot{\theta} + \dot{l}\sin\theta\cos\theta\dot{\theta} + B(l,\theta)\cos\theta\dot{\theta}^2 \end{pmatrix}$$

$$\frac{\partial T}{\partial l} = m_l\Big(\dot{x}_b\sin\theta\dot{\theta} - \dot{y}_b\cos\theta\dot{\theta} + \Big(A(l,\theta)\sin\theta + B(l,\theta)\cos\theta\Big)\dot{\theta}^2\Big)$$

$$\frac{\partial T}{\partial l} = m_l\Big(\dot{x}_b\sin\theta\dot{\theta} - \dot{y}_b\cos\theta\dot{\theta} + (l-h)\dot{\theta}^2\Big)$$

$$\frac{\partial V}{\partial l} = -m_l g\sin\theta + K_s(l - L_0)$$

$$-m_l\ddot{x}_b\cos\theta - m_l\ddot{y}_b\sin\theta + m_l\ddot{l} - dm_l\ddot{\theta} - m_l(l-h)\dot{\theta}^2 - m_l g\sin\theta + K_s(l-L_0) = 0 \quad \text{(A.35)}$$

★ For $\underline{\underline{q_4 = \theta}}$,

$$\begin{aligned}\frac{\partial T}{\partial \dot{\theta}} = (I_b + I_l)\dot{\theta} &+ m_l\Big(\dot{x}_b - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)\Big(A(l,\theta)\Big) \\ &+ m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)\Big(-B(l,\theta)\Big)\end{aligned}$$

$$\begin{aligned}\frac{\partial T}{\partial \dot{\theta}} = (I_b + I_l)\dot{\theta} &+ m_l\Big(\dot{x}_b A(l,\theta) - \dot{y}_b B(l,\theta)\Big) \\ &- m_l\dot{l}\Big(A(l,\theta)\cos\theta - B(l,\theta)\sin\theta\Big) \\ &+ m_l\dot{\theta}\Big(A^2(l,\theta) + B^2(l,\theta)\Big)\end{aligned}$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) &= m_l A(l,\theta)\ddot{x}_b - m_l B(l,\theta)\ddot{y}_b \\ &+ m_l\dot{x}_b\Big(\dot{l}\sin\theta + B(l,\theta)\dot{\theta}\Big) - m_l\dot{y}_b\Big(\dot{l}\cos\theta - A(l,\theta)\dot{\theta}\Big) \\ &- dm_l\ddot{l} + \Big(I_b + I_l + m_l\Big((l-h)^2 + d^2\Big)\Big)\ddot{\theta} + 2m_l(l-h)\dot{l}\dot{\theta}\end{aligned}$$

$$\begin{aligned}\frac{\partial T}{\partial \theta} &= m_l\Big(\dot{x}_b - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)\frac{\partial}{\partial\theta}\Big(-\dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big) \\ & \quad m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)\frac{\partial}{\partial\theta}\Big(-\dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)\end{aligned}$$

$$\begin{aligned}\frac{\partial T}{\partial \theta} &= m_l\Big(\dot{x}_b - \dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)\Big(\dot{l}\sin\theta + B(l,\theta)\dot{\theta}\Big) \\ & \quad m_l\Big(\dot{y}_b - \dot{l}\sin\theta - B(l,\theta)\dot{\theta}\Big)\Big(-\dot{l}\cos\theta + A(l,\theta)\dot{\theta}\Big)\end{aligned}$$

Using equation (A.25) and (A.26),

$$\frac{\partial T}{\partial \theta} = m_l\Big(\dot{x}_b - \dot{B}(l,\theta)\Big)\Big(\dot{A}(l,\theta)\Big) + m_l\Big(\dot{y}_b - A(l,\theta)\Big)\Big(-B(l,\theta)\Big)$$

$$\frac{\partial T}{\partial \theta} = m_l\dot{x}_b\Big(\dot{l}\sin\theta + B(l,\theta)\dot{\theta}\Big) - m_l\dot{y}_b\Big(\dot{l}\cos\theta - A(l,\theta)\dot{\theta}\Big)$$

$$\frac{\partial V}{\partial \theta} = -m_l g B(l,\theta)$$

$$\begin{aligned}m_l A(l,\theta)\ddot{x}_b - m_l B(l,\theta)\ddot{y}_b - dm_l\ddot{l} &+ \Big(I_b + I_l + m_l\Big((l-h)^2 + d^2\Big)\Big)\ddot{\theta} \\ &+ 2m_l(l-h)\dot{l}\dot{\theta} - m_l g B(l,\theta) = 0 \qquad \text{(A.36)}\end{aligned}$$

Let,

$$\begin{aligned}I &= I_b + I_l + m_l\Big((l-h)^2 + d^2\Big) \\ M &= m_b + m_l \qquad \text{(A.37)}\end{aligned}$$

So, we have Euler-Lagrangian dynamic motion equations for a passive system as,

$$M\ddot{x}_b - m_l\ddot{l}\cos\theta + m_l A(l,\theta)\ddot{\theta} + 2m_l\dot{l}\sin\theta\dot{\theta} + m_l B(l,\theta)\dot{\theta}^2 = 0$$

$$M\ddot{y}_b - m_l\ddot{l}\sin\theta - m_l B(l,\theta)\ddot{\theta} - 2m_l\dot{l}\cos\theta\dot{\theta} + m_l A(l,\theta)\dot{\theta}^2 + Mg = 0$$

$$-m_l\ddot{x}_b\cos\theta - m_l\ddot{y}_b\sin\theta + m_l\ddot{l} - dm_l\ddot{\theta} - m_l(l-h)\dot{\theta}^2 - m_l g\sin\theta + K_s(l-L_0) = 0$$

$$m_l A(l,\theta)\ddot{x}_b - m_l B(l,\theta)\ddot{y}_b - dm_l\ddot{l} + I\ddot{\theta} + 2m_l(l-h)\dot{l}\dot{\theta} - m_l g B(l,\theta) = 0 \qquad \text{(A.38)}$$

Standard Representation of SLOM Dynamics:

$$D(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau \tag{A.39}$$

$$\begin{bmatrix} M & 0 & -m_l\cos\theta & m_lA(l,\theta) \\ 0 & M & -m_l\sin\theta & -m_lB(l,\theta) \\ -m_lcos\theta & -m_l\sin\theta & m_l & -dm_l \\ m_lA(l,\theta) & -m_lB(l,\theta) & -dm_l & I \end{bmatrix} \begin{bmatrix} \ddot{x_b} \\ \ddot{y_b} \\ \ddot{l} \\ \ddot{\theta} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & 0 & m_l\sin\theta\dot{\theta} & m_l\dot{l}\sin\theta + m_lB(l,\theta)\dot{\theta} \\ 0 & 0 & -m_l\cos\theta\dot{\theta} & -m_l\dot{l}\cos\theta + m_lA(l,\theta)\dot{\theta} \\ 0 & 0 & 0 & m_l(l-h)\dot{\theta} \\ 0 & 0 & m_l(l-h)\dot{\theta} & m_l(l-h)\dot{l} \end{bmatrix} \begin{bmatrix} \dot{x_b} \\ \dot{y_b} \\ \dot{l} \\ \dot{\theta} \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ Mg \\ -m_lgsin\theta + K_s(l-L_0) \\ -m_lB(l,\theta) \end{bmatrix} = \tau \tag{A.40}$$

We have verified the following three general properties of this EL system [46]:

* **P.1:** The kinetic energy of the system is given by $T(q,\dot{q}) = \frac{1}{2}\dot{q}^T D(q)\dot{q}$.

* **P.2:** The matrix $D(q)$ is symmetric and positive definite for each $q \in R^n$.

* **P.3:** The matrix $W(q,\dot{q}) = \dot{D}(q) - 2C(q,\dot{q})$ is skew-symmetric.

A.4 Constrained EL Equations of Motion:

Generally, the constraint described during particular phase or events are holonomic in nature and have the form,

$$\psi_i(q) = 0, i = 1, \ldots, m \tag{A.41}$$

where, m is the number of constraints. In such constrained system, the EL equation of motion differs from the form as shown in equation (A.39) to the following form,

$$\begin{aligned} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) &= \tau - J^T(q)\lambda_r \\ J(q)\dot{q} &= 0 \end{aligned} \tag{A.42}$$

where,

$$J(q) = \frac{\partial}{\partial q}\psi(q) \in R^{m\times n} \tag{A.43}$$

is the Jacobian matrix of the constrained robotic system and λ_r is the generalized constraint force. From (A.42),

$$\ddot{q} = D^{-1}(q)\{\tau - C(q,\dot{q})\dot{q} + G(q)\} - D^{-1}(q)J^T(q)\lambda_r \tag{A.44}$$

Multiply (A.44) by $J(q)$ to both sides,

$$J(q)\ddot{q} = J(q)D^{-1}(q)\{\tau - C(q,\dot{q})\dot{q} + G(q)\} - J(q)D^{-1}(q)J^T(q)\lambda_r \tag{A.45}$$

Also, we have,

$$\begin{aligned} J(q)\dot{q} &= 0 \\ J(q)\ddot{q} + \dot{J}(q)\dot{q} &= 0 \\ J(q)\ddot{q} &= -\dot{J}(q)\dot{q} \end{aligned}$$

Hence, equation (A.45) becomes,

$$-\dot{J}(q)\dot{q} = J(q)D^{-1}(q)\{\tau - C(q,\dot{q})\dot{q} + G(q)\} - J(q)D^{-1}(q)J^T(q)\lambda_r$$

$$J(q)D^{-1}(q)J^T(q)\lambda_r = J(q)D^{-1}(q)\{\tau - C(q,\dot{q})\dot{q} + G(q)\} + \dot{J}(q)\dot{q}$$

Let, $\phi_r = J(q)D^{-1}(q)J^T(q)$ and so we can explicitly express λ_r as,

$$\lambda_r = {\phi_r}^{-1}\left\{J(q)D^{-1}(q)\Big[\tau - C(q,\dot{q})\dot{q} + G(q)\Big] + \dot{J}(q)\dot{q}\right\} \tag{A.46}$$

Equation of Impact:

Due to impact there is sudden change in the generalized velocities ($\dot{q}$). The change of ($\dot{q}$) can be determined by substituting $\ddot{q} = \dot{q}_+ - \dot{q}_-$ in equation (A.42).

$$\begin{aligned} D(q)(\dot{q}_+ - \dot{q}_-) &= -J^T(q)\lambda_I \\ J(q)\dot{q}_+ &= 0 \end{aligned} \tag{A.47}$$

From (A.47),

$$\begin{aligned} \dot{q}_+ - \dot{q}_- &= -D^{-1}(q)J^T(q)\lambda_I \\ \dot{q}_+ &= \dot{q}_- - D^{-1}(q)J^T(q)\lambda_I \\ J(q)\dot{q}_+ &= J(q)\dot{q}_- - J(q)D^{-1}(q)J^T(q)\lambda_I \\ 0 &= J(q)\dot{q}_- - \phi_I\lambda_I \end{aligned}$$

where, $\phi_I = J(q)D(q)^{-1}J^T(q)$ and so we can express λ_I as,

$$\lambda_I = {\phi_I}^{-1}(J(q)\dot{q}_-) \tag{A.48}$$

A.5 Constraints and Jacobian:

Flight phase: The constraint during flight phase is only on the length of the leg, i.e. during flight the length of leg l should be equal to the L_{to}. L_{to} is length l when the body hits the leg. So, we have,

$$\psi(q) = l - L_{to} = 0 \tag{A.49}$$

$$J(q) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.50}$$

$$J(q)\dot{q} = \dot{l} = 0 \tag{A.51}$$

$$\dot{J}(q) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.52}$$

Stance phase: The constraint during stance phase is only on the coordinate of the foot position (the foot of robot should remain on the ground). Hence,

$$\psi(q) = \begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} x_b - l\cos\theta + d\sin\theta \\ y_b - l\sin\theta - d\cos\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{A.53}$$

$$J(q) = \begin{bmatrix} 1 & 0 & -\cos\theta & l\sin\theta + d\cos\theta \\ 0 & 1 & -\sin\theta & -(l\cos\theta - d\sin\theta) \end{bmatrix} \tag{A.54}$$

$$J(q)\dot{q} = \begin{bmatrix} \dot{x}_b - \dot{l}\cos\theta + (l\sin\theta + d\cos\theta)\dot{\theta} \\ \dot{y}_b - \dot{l}\sin\theta - (l\cos\theta - d\sin\theta)\dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{x}_f \\ \dot{y}_f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{A.55}$$

$$\dot{J}(q) = \begin{bmatrix} 0 & 0 & \sin\theta\dot{\theta} & \dot{l}\sin\theta + (l\cos\theta - d\sin\theta)\dot{\theta} \\ 0 & 0 & -\cos\theta\dot{\theta} & -\dot{l}\cos\theta + (l\sin\theta + d\cos\theta)\dot{\theta} \end{bmatrix} \tag{A.56}$$

Take-off Event: The constraint during take-off event is only on the length of the leg (as discussed for the flight phase), i.e. the length of leg should be equal to the leg length L_{to} (due to mechanical stop restriction and assumed condition for take-off). Hence constraint equation and Jacobian are same as applicable to the flight phase, given in equations (A.49)-(A.52).

Touchdown Event: The constraint during impulsive landing event is on the coordinate of the foot position, i.e. during landing the foot of robot should remain on the ground. There are no other constraints. Hence, constraint equation and Jacobian are same as applicable to the stance phase, given in equations (A.53)-(A.56).

Appendix B

Poincaré Map

It is well known that the one-legged hopping system under study is highly nonlinear periodic dynamical system. *Poincaré map* or *first return map* is one of the classical tool to study the existence and stability of the dynamical system which repeats periodically. The purpose of this **Appendix** is to introduce the reader the basic concepts and terminology used to analyze the system's motion. Since it is useful to present the properties of such systems in geometrical images, we give some definitions from nonlinear systems theory. We will define here two of the basic geometrical objects associated with dynamical systems, the *solution curve* and the *orbit*. Consider the set of the nonlinear differential equations of an *autonomous* system,

$$\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} = f(\mathbf{x}), \tag{B.1}$$

where $\mathbf{x} = \mathbf{x}(t) = [x_1, x_2, \ldots, x_n]$ is a vector function of an independent variable, time and $f : U \rightarrow R^n$ is a smooth vector function defined on some subset $U \subseteq R^n$. The vector field f generates a *flow* $\phi_t : U \rightarrow R^n$, where $\phi_t(\mathbf{x}) = \phi(t, \mathbf{x})$ is a smooth function defined for $\mathbf{x} \in R^n$ and $t \in I = [t_1, t_2] \subseteq R$, and ϕ satisfies equation (B.1). Let an initial condition is defined as, $\mathbf{x}_{in} = \mathbf{x}(0) \in U$ and corresponding solution is $\phi(\mathbf{x}_{in}, t)$ such that $\phi(\mathbf{x}_{in}, 0) = \mathbf{x}_{in}$. Sometimes such a solution is written as $\mathbf{x} = \mathbf{x}(\mathbf{x}_{in}, t)$ or simply $\mathbf{x} = \mathbf{x}(t)$. For each $\mathbf{x}_{in} \in U$ the solution defines two geometrical objects as shown in **Fig. B.1**,

- A *solution curve*

$$C_r(\mathbf{x}_{in}) = \Big\{(t, \mathbf{x}) \in ([t_1, t_2] \subset R) \times R^n | \mathbf{x} = \mathbf{x}(t, \mathbf{x}_{in})\Big\}. \tag{B.2}$$

- A *orbit* or *trajectory*, which is the projection of $C_r(\mathbf{x}_{in})$ onto the state space

$$O_r(\mathbf{x}_{in}) = \Big\{\mathbf{x} \in R^n | \mathbf{x} = \mathbf{x}(t, \mathbf{x}_{in}), t \in [t_1, t_2] \subset R)\Big\} \subset R^n. \tag{B.3}$$

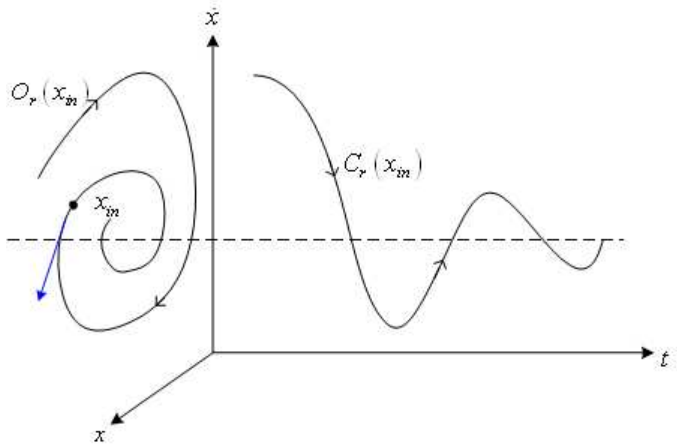

Figure B.1: Solution curve and orbit of a two dimensional system

Often we seek a class of solutions of differential equations called the *fixed points*, $\overline{\mathbf{x}}$. Fixed points are defined as,

$$f(\overline{\mathbf{x}}) = 0. \tag{B.4}$$

Further these fixed points are categorized as,

- Fixed point $\overline{\mathbf{x}}$ is said to be *stable* if a solution that starts in a neighbourhood of $\overline{\mathbf{x}}$ remains close to $\overline{\mathbf{x}}$ for all time.
- Fixed point is called *asymptotically stable* if in addition to above-mentioned condition, the solution converges to $\overline{\mathbf{x}}$.
- Fixed point is called *unstable* if it is not stable.

Note that since the above definitions concern the behaviour of solutions near the fixed point $\overline{\mathbf{x}}$ they are *local* in nature. More rigorous definitions can be found in classical books on nonlinear systems and dynamical systems, such as [26] and [31].

In context with the fixed points, it is important to predict their existence and characterize their stability properties. As was mentioned above, a very useful tool for analyzing periodic motions is the Poincaré map. The Poincaré map replaces an n^{th} order continuous time autonomous system by an $(n-1)^{th}$ order discrete time system. Note that discrete Poincaré maps are particularly suited for the analysis of intermittent dynamical systems like dynamically stable one-legged robots. This is because not only they reduce the order of the system, but also they let us examine the periodicity and stability of the motion with respect to a particular event in the locomotion cycle, called *Poincaré section*.

Suppose that γ is a closed orbit of some flow ϕ_t in R^n in arising from the nonlinear vector field $f(\mathbf{x})$ of the system (B.1), shown in the **Fig. B.2**. It is often possible to find a *local cross section* about γ, which is of dimension $n-1$. Consider a point $\mathbf{p}$ on the orbit γ and let Σ be a $(n-1)$-dimensional hypersurface (local cross section). It must be mentioned here that Σ need not be planar, however, it must be *transverse* to the flow. All the orbits in the neighborhood of γ must pass through $\Sigma \in R^n$. The hypersurface can be defined as the zero-level set of a smooth scalar function $g : R^n \to R, g(\mathbf{p}) = 0$ so that,

$$\Sigma = \Big\{ \mathbf{x} \in R^n | g(\mathbf{x}) = 0 \Big\}. \tag{B.5}$$

The hypersurface Σ is transversal to γ at $\mathbf{p}$ means the gradient

$$\nabla g(x) = \left[\begin{array}{cccc} \frac{\partial g(\mathbf{X})}{\partial x_1}) & \frac{\partial g(\mathbf{X})}{\partial x_2}) & \dots & \frac{\partial g(\mathbf{X})}{\partial x_n}) \end{array} \right]^T \tag{B.6}$$

is not orthogonal to the flow at $\mathbf{p}$, i.e.

$$\nabla g(\mathbf{p})^T f(\mathbf{p}) \neq 0. \tag{B.7}$$

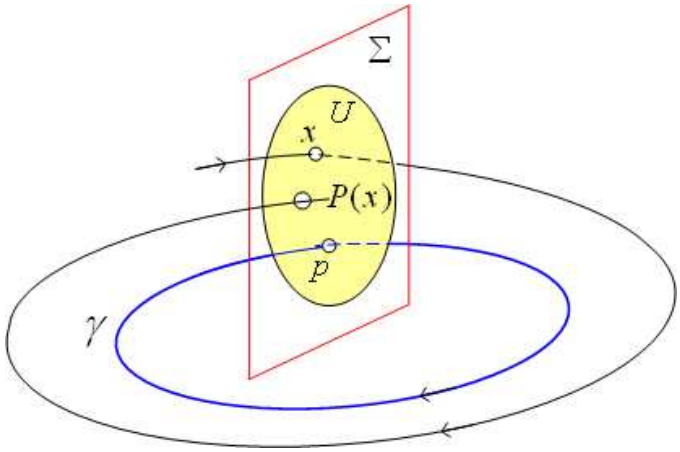

Figure B.2: The definition of a Poincaré map: Hyperplane Σ and map $\mathbf{P}$

The simplest choice of Σ is a hyperplane orthogonal to γ at $\mathbf{p}$. A hyperplane is a surface in the state space, which is defined by point $\mathbf{p}$ and its normal vector $\mathbf{n} \in R^n$, i.e.

$$\Sigma = \Big\{ \mathbf{x} \in R^n | \mathbf{n}^T (\mathbf{x} - \mathbf{p}) = 0 \Big\}. \tag{B.8}$$

In that case, the transversality condition can be written as

$$\mathbf{n}^T f(\mathbf{p}) \neq 0. \tag{B.9}$$

If the orbit γ is periodic of period $T = T(\mathbf{p})$, the trajectory starting from $\mathbf{p}$ will hit Σ at exactly the same point $\mathbf{p}$ after T s. A periodic orbit that returns directly to itself is a fixed point of the Poincaré map. Moreover, an orbit starting on Σ from a sufficiently small neighbourhood of $\mathbf{p}$ will have a return time close to T. Let $U \subset \Sigma$ be a neighbourhood of $\mathbf{p}$ such that γ intersects U only once at $\mathbf{p}$. The Poincaré map $\mathbf{P} : U \to \Sigma$ is defined for a point $\mathbf{x} \in U$ by

$$\mathbf{P}(\mathbf{x}) = \phi_\tau(\mathbf{x}), \tag{B.10}$$

where $\phi_\tau(\mathbf{x})$ is the flow of the system. Note that the return time, τ, generally depends on the point $\mathbf{x}$ and need not be equal to T. However, as $\mathbf{x} \to \mathbf{p}$ then $\tau \to T$. It must be mentioned here that the Poincaré map need not be defined for all $\mathbf{x} \in \Sigma$, however henceforth in discussion we assume that the Poincaré map is defined in U. Starting with $\mathbf{x}^{(0)} \in U$ then $\mathbf{x}^{(1)} = \mathbf{P}\left(\mathbf{x}^{(0)}\right)$ and if $\mathbf{x}^{(1)} \in U$, so that the Poincaré map is defined at $\mathbf{x}^{(1)}$, then $\mathbf{x}^{(2)} = \mathbf{P}\left(\mathbf{x}^{(1)}\right)$. Sequentially, for $\mathbf{x}^{(k)} \in U$ the Poincaré map is defined as the solution of a discrete system,

$$\mathbf{x}^{(k+1)} = \mathbf{P}\left(\mathbf{x}^{(k)}\right). \tag{B.11}$$

As mentioned above, when the trajectory, which starts at $\mathbf{p}$ will hit Σ at the same point, then $\mathbf{p}$ will be an fixed point (equilibrium point) of the discrete system described by equation (B.11),

$$\mathbf{p} = \mathbf{P}(\mathbf{p}). \tag{B.12}$$

It is well known that the stability of the periodic orbit γ is equivalent to the stability of the fixed point $\mathbf{p}$ of the Poincaré map, equation (B.11). Therefore, γ is stable if all the eigenvalues of $(n-1) \times (n-1)$ the Jacobian matrix of $\mathbf{P}$ calculated at $\mathbf{p}$,

$$\mathbf{J} = \frac{d}{d\mathbf{x}}\mathbf{P}(\mathbf{x})\bigg|_{\mathbf{x}=\mathbf{p}} \tag{B.13}$$

are located inside the unit circle (stability of discrete systems). For a formal proof of this fact the interested reader is referred to [26], Theorem 7.3, p. 306. It must be mentioned here that the eigenvalues do not depend on the selection of the point $\mathbf{p}$ on the orbit γ, the cross section Σ or its representation. A proof of this can be found in [31], Lemma 1.2, p. 25. The same discussion can also be found in [5] where the passive dynamics of quadrupedal robot is studied using the Poincaré map.

Appendix C

Taylor Series Approximation

Let $f(x) = f(x_1, x_2, x_3, x_4)$ is given non-linear function of x and let $\overline{x}$ is the equilibrium. Let's define a new variable that simply measure how far we are from $\overline{x}$, call the variable,

$$\triangle x = x - \overline{x}. \tag{C.1}$$

Then, using Taylor series formula,

$$\begin{aligned} f(x) &= f(x_1, x_2, x_3, x_4) \\ &= f(\overline{x} + \triangle x) \\ &= \sum_{m=0}^{n} \frac{1}{m!}\Big(\triangle x_1 \frac{\partial}{\partial x_1} + \triangle x_2 \frac{\partial}{\partial x_2} + \triangle x_3 \frac{\partial}{\partial x_3} + \triangle x_4 \frac{\partial}{\partial x_4}\Big)^m f(\overline{x}) \end{aligned} \tag{C.2}$$

where, n is the order of approximation. We chosen $n = 3$.

Let,

$$f(x) = A_0 + A_1 + A_2 + A_3 \tag{C.3}$$

Where, A_0, A_1, A_2, A_3 are the terms evaluated for equation (C.2) considering $m = 0, 1, 2, 3$ respectively.

★ **First term** (A_0):

$$\begin{aligned} A_0 &= \frac{1}{0!}\Big(\triangle x_1 \frac{\partial}{\partial x_1} + \triangle x_2 \frac{\partial}{\partial x_2} + \triangle x_3 \frac{\partial}{\partial x_3} + \triangle x_4 \frac{\partial}{\partial x_4}\Big)^0 f(\overline{x}) \\ A_0 &= f(\overline{x}) \end{aligned} \tag{C.4}$$

★ **Second term** (A_1):

$$\begin{aligned} A_1 &= \frac{1}{1!}\Big(\triangle x_1 \frac{\partial}{\partial x_1} + \triangle x_2 \frac{\partial}{\partial x_2} + \triangle x_3 \frac{\partial}{\partial x_3} + \triangle x_4 \frac{\partial}{\partial x_4}\Big)^1 f(\overline{x}) \\ &= \triangle x_1 \frac{\partial f(\overline{x})}{\partial x_1} + \triangle x_2 \frac{\partial f(\overline{x})}{\partial x_2} + \triangle x_3 \frac{\partial f(\overline{x})}{\partial x_3} + \triangle x_4 \frac{\partial f(\overline{x})}{\partial x_4} \end{aligned}$$

$$= \begin{bmatrix} \triangle x_1 & \triangle x_2 & \triangle x_3 & \triangle x_4 \end{bmatrix} \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \frac{\partial f(x)}{\partial x_3} \\ \frac{\partial f(x)}{\partial x_4} \end{bmatrix}_{x=\overline{x}}$$

$$A_1 = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial x_2} & \frac{\partial f(x)}{\partial x_3} & \frac{\partial f(x)}{\partial x_4} \end{bmatrix}_{x=\overline{x}} \begin{bmatrix} \triangle x_1 \\ \triangle x_2 \\ \triangle x_3 \\ \triangle x_4 \end{bmatrix} \tag{C.5}$$

★ **Third term** (A_2):

$$\begin{aligned} A_2 &= \frac{1}{2!}\Big(\triangle x_1\frac{\partial}{\partial x_1} + \triangle x_2\frac{\partial}{\partial x_2} + \triangle x_3\frac{\partial}{\partial x_3} + \triangle x_4\frac{\partial}{\partial x_4}\Big)^2 f(\overline{x}) \\ &= \frac{1}{2}\Big(\triangle x_1\frac{\partial f(\overline{x})}{\partial x_1} + \triangle x_2\frac{\partial f(\overline{x})}{\partial x_2} + \triangle x_3\frac{\partial f(\overline{x})}{\partial x_3} + \triangle x_4\frac{\partial f(\overline{x})}{\partial x_4}\Big)^2 \\ &= \begin{bmatrix} \triangle x_1 & \triangle x_2 & \triangle x_3 & \triangle x_4 \end{bmatrix} \begin{bmatrix} \frac{\partial^2 f(x)}{\partial {x_1}^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_3} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_4} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial {x_2}^2} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_3} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_4} \\ \frac{\partial^2 f(x)}{\partial x_3 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_3 \partial x_2} & \frac{\partial^2 f(x)}{\partial {x_3}^2} & \frac{\partial^2 f(x)}{\partial x_3 \partial x_4} \\ \frac{\partial^2 f(x)}{\partial x_4 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_4 \partial x_2} & \frac{\partial^2 f(x)}{\partial x_4 \partial x_3} & \frac{\partial^2 f(x)}{\partial {x_4}^2} \end{bmatrix}_{x=\overline{x}} \begin{bmatrix} \triangle x_1 \\ \triangle x_2 \\ \triangle x_3 \\ \triangle x_4 \end{bmatrix} \\ &= \frac{1}{2}(\triangle x)^T \nabla^2 f(\overline{x})(\triangle x) \\ &= \frac{1}{2}\begin{Bmatrix} \triangle {x_1}^2\frac{\partial^2 f(x)}{\partial {x_1}^2} + \triangle {x_2}^2\frac{\partial^2 f(x)}{\partial {x_2}^2} + \triangle {x_3}^2\frac{\partial^2 f(x)}{\partial {x_3}^2} + \triangle {x_4}^2\frac{\partial^2 f(x)}{\partial {x_4}^2} \\ 2\triangle x_1\triangle x_2\frac{\partial^2 f(x)}{\partial x_1 \partial x_2} + 2\triangle x_1\triangle x_3\frac{\partial^2 f(x)}{\partial x_1 \partial x_3} + 2\triangle x_1\triangle x_4\frac{\partial^2 f(x)}{\partial x_1 \partial x_4} \\ 2\triangle x_2\triangle x_3\frac{\partial^2 f(x)}{\partial x_2 \partial x_3} + 2\triangle x_2\triangle x_4\frac{\partial^2 f(x)}{\partial x_2 \partial x_4} + 2\triangle x_3\triangle x_4\frac{\partial^2 f(x)}{\partial x_3 \partial x_4} \end{Bmatrix} \\ A_2 &= \frac{1}{2}D_1 f_1(x) \end{aligned} \tag{C.6}$$

where, in our case, $D_1 \in R^{3\times 10}$ and $f_1 \in R^{10\times 1}$ as given below in simplified form,

$${D_1}^T = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial {x_1}^2} \\ \frac{\partial^2 f(x)}{\partial {x_2}^2} \\ \frac{\partial^2 f(x)}{\partial {x_3}^2} \\ \frac{\partial^2 f(x)}{\partial {x_4}^2} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_4} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_4} \\ \frac{\partial^2 f(x)}{\partial x_3 \partial x_4} \end{bmatrix}_{x=\overline{x}}, \qquad f_1(x) = \begin{bmatrix} \triangle {x_1}^2 \\ \triangle {x_2}^2 \\ \triangle {x_3}^2 \\ \triangle {x_4}^2 \\ 2\triangle x_1\triangle x_2 \\ 2\triangle x_1\triangle x_3 \\ 2\triangle x_1\triangle x_4 \\ 2\triangle x_2\triangle x_3 \\ 2\triangle x_2\triangle x_4 \\ 2\triangle x_3\triangle x_4 \end{bmatrix} \tag{C.7}$$

★ **Fourth term** (A_3):

Using the same procedure as done for second term A_2,

$$\begin{aligned} A_3 &= \frac{1}{3!}\Big(\triangle x_1\frac{\partial}{\partial x_1}+\triangle x_2\frac{\partial}{\partial x_2}+\triangle x_3\frac{\partial}{\partial x_3}+\triangle x_4\frac{\partial}{\partial x_4}\Big)^3 f(\overline{x}) \\ &= \frac{1}{6}\Big(\triangle x_1\frac{\partial f(\overline{x})}{\partial x_1}+\triangle x_2\frac{\partial f(\overline{x})}{\partial x_2}+\triangle x_3\frac{\partial f(\overline{x})}{\partial x_3}+\triangle x_4\frac{\partial f(\overline{x})}{\partial x_4}\Big)^3 \\ A_3 &= \frac{1}{6}D_2 f_2(x) \end{aligned} \tag{C.8}$$

where, in our case, $D_2 \in R^{3\times 20}$ and $f_2 \in R^{20\times 1}$ as given below in simplified form,

$$D_2{}^T = \begin{bmatrix} \frac{\partial^3 f(x)}{\partial x_1{}^3} \\ \frac{\partial^3 f(x)}{\partial x_2{}^3} \\ \frac{\partial^3 f(x)}{\partial x_3{}^3} \\ \frac{\partial^3 f(x)}{\partial x_4{}^3} \\ \frac{\partial^3 f(x)}{\partial x_1{}^2\partial x_2} \\ \frac{\partial^3 f(x)}{\partial x_1{}^2\partial x_3} \\ \frac{\partial^3 f(x)}{\partial x_1{}^2\partial x_4} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_2{}^2} \\ \frac{\partial^3 f(x)}{\partial x_2{}^2\partial x_3} \\ \frac{\partial^3 f(x)}{\partial x_2{}^2\partial x_4} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_3{}^2} \\ \frac{\partial^3 f(x)}{\partial x_2\partial x_3{}^2} \\ \frac{\partial^3 f(x)}{\partial x_3{}^2\partial x_4} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_4{}^2} \\ \frac{\partial^3 f(x)}{\partial x_2\partial x_4{}^2} \\ \frac{\partial^3 f(x)}{\partial x_3\partial x_4{}^2} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_2\partial x_3} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_2\partial x_4} \\ \frac{\partial^3 f(x)}{\partial x_1\partial x_3\partial x_4} \\ \frac{\partial^3 f(x)}{\partial x_2\partial x_3\partial x_4} \end{bmatrix}_{x=\overline{x}} , \qquad f_2(x) = \begin{bmatrix} \triangle x_1{}^3 \\ \triangle x_2{}^3 \\ \triangle x_3{}^3 \\ \triangle x_4{}^3 \\ 3\triangle x_1{}^2\triangle x_2 \\ 3\triangle x_1{}^2\triangle x_3 \\ 3\triangle x_1{}^2\triangle x_4 \\ 3\triangle x_1\triangle x_2{}^2 \\ 3\triangle x_2{}^2\triangle x_3 \\ 3\triangle x_2{}^2\triangle x_4 \\ 3\triangle x_1\triangle x_3{}^2 \\ 3\triangle x_2\triangle x_3{}^2 \\ 3\triangle x_3{}^2\triangle x_4 \\ 3\triangle x_1\triangle x_4{}^2 \\ 3\triangle x_2\triangle x_4{}^2 \\ 3\triangle x_3\triangle x_4{}^2 \\ 6\triangle x_1\triangle x_2\triangle x_3 \\ 6\triangle x_1\triangle x_2\triangle x_4 \\ 6\triangle x_1\triangle x_3\triangle x_4 \\ 6\triangle x_2\triangle x_3\triangle x_4 \end{bmatrix} \tag{C.9}$$

Finally, with the help of equation (C.3), we can write the nonlinear approximation model of the original mapped system as,

$$f(x) = f(\overline{x}) + \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial x_2} & \frac{\partial f(x)}{\partial x_3} & \frac{\partial f(x)}{\partial x_4} \end{bmatrix}_{x=\overline{x}} \begin{bmatrix} \triangle x_1 \\ \triangle x_2 \\ \triangle x_3 \\ \triangle x_4 \end{bmatrix} + \frac{1}{2}D_1 f_1(x) + \frac{1}{6}D_2 f_2(x) \tag{C.10}$$

Bibliography

[1] M. H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.

[2] F. Hardarson. Locomotion for difficult terrain. Technical Report TRITA-MMK 1998:3, Dept. of Machine Design, Royal Institute of Technology, Sweden, April 1998.

[3] A. Sayyad, B. Seth, and P. Seshu. Single-legged hopping robotics research- a review. *Robotica*, 25(05):587–613, 2007.

[4] M. H. Raibert. Running with symmetry. *Int. J. Robotics Research*, 5(4):45–61, 1986.

[5] I. Poulakakis. On the passive dynamics of quadraupedal running. Master's thesis, McGill University, Montreal, Canada, July 2003.

[6] A. Sato and M. Buehler. planar hopping robot with one actuator: design, simulation, and experimental results. In *Proc. of IROS*, pages 3540–3545, Sendai, Japan, September/ October 2004.

[7] K. Matsuoka. A model of repetitive hopping movements in man. In *Proc. of Fifth World Congress on Theory of Machines and Mechanisms*, 1979.

[8] M. H. Raibert. Hopping in legged systems-modeling and simulation for the two-dimensional one-legged case. *IEEE Trans. on Systems, Man and Cybernetics*, 14(3):451–463, 1984.

[9] H. B. Brown M. H. Raibert and M. Chepponis. Experiments in balance with a 3d one-legged hopping machine. *Int. J. Robotics Research*, 3(2):75–92, 1984.

[10] P. V. Shanmuganathan. *Dynamics and Stabilization of Under-Actuated Monopedal Hopping*. PhD thesis, Indian Institute of Technology, Bombay, India, July 2002.

[11] A. Ohnishi M. Sampei S. Kuswadi, A. Takahashi and S. Nakaura. A one linear actuator hopping robot: Modelling and control. *Advanced Robotics*, 17, 2003.

[12] R. D. Quinn H. Verma T.E. Wei, G. M. Nelson and S. L. Garverick. Design of a 5-cm monopod hopping robot. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, volume 3, San Francisco, CA, April 2000.

[13] Website. http://www.eng.nus.edu.sg/research/2003/2003C3_038.htm, 2003. Singapore robotic games,19-21 May 2003.

[14] H. Zou and J. P. Schmiedeler. The effect of asymmetrical body-mass distribution on the stability and dynamics of quadruped bounding. *IEEE Trans. on Robotics*, 22(04):711–723, 2006.

[15] M. Sampei T. Funato, S. Kuswadi and S. Nakaura. Continuous hopping motion experiment of one linear actuator robot with adaptive fuzzy control. In *Proc. of SICE Annual Conference*, pages 2506–2511, Fukui, Japan, August, 2003.

[16] S. Nakaura K. Takeuchi, S. Kuswadi and N. Sampei. Continuous hopping motion control experiment of one linear actuator robot. In *Proc. of* 41^{st} *SICE Annual Conference*, pages 232–237, Osaka, Japan, August 2002.

[17] Y. Morita and K. Ohnishi. Attitude control of hopping robot using angular momentum. In *IEEE Int. Conf. on Industrial Technology*, pages 173–178, Maribor, Slovenia, December, 2003.

[18] J. W. Burdick R. T. M'Closkey and A. F. Vakakis. On the periodic motions of simple hopping robots. In *Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 771–777, 1990.

[19] A. F. Vakakis and J. W. Burdick. Chaotic motions in the dynamics of a hopping robot. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1464–1469, 1990.

[20] F. U. Rehman and H. Michalska. Geometric approach to feedback stabilization of a hopping robot in the flight phase. In *Proc. of* 8^{th} *International Conference on Advanced Robotics*, pages 551–556, 1997.

[21] F. U. Rehman and H. Michalska. Steering control of a hopping robot model during the flight phase. In *IEE Proc. of Control Theory and Applications*, pages 645–653, 2005.

[22] K. D. Young, V. I. Utkin, and U. Ozguner. A control engineer's guide to sliding mode control. *IEEE Trans. Contr. Syst.*, 7:328–342, 1999.

[23] B. Bandyopadhyay and S. Janardhanan. *Discrete-time Sliding Mode Control using Multirate Output Feedback*. Springer-Verlag, Berlin, 2006.

[24] W. Gao, Y. Wang, and A. Homaifa. Discrete-time variable structure control systems. *IEEE Trans. Ind. Electron*, 42:117–122, 1995.

[25] E. A. Misawa. Discrete-time sliding mode control: The linear case. *ASME J. Dyn. Sys., Meas, and Control*, 119:819–821, 1997.

[26] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, second edition, 1986.

[27] Koditschek D. and Buehler M. Analysis of a simplified hopping robot. *Int. J. of Robotics Research*, 10, 1991.

[28] J. G. Cham and M. R. Cutkosky. Dynamic stability of open-loop hopping. *submitted to ASME Journal of Dynamic Systems, Measurement and Control*, 2004.

[29] J. P. Ostrowski and J. W. Burdick. Designing feedback algorithms for controlling the periodic motion of legged robots. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 260–266, 1993.

[30] Li Z. and He J. An energy perturbation approach to limit cycle analysis in legged locomotion systems. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 989–994, 1990.

[31] Y. Kuznetsov. *Elements of Applied Bifurcation Theory*. Springer-Verlag, second edition, 1998.

[32] C.S.Hsu. A theory of cell-to-cell mapping dynamical systems. *ASME J. of Applied Mechanics*, 47, 1980.

[33] C.S.Hsu and R.S. Guttalu. An unravelling algorithm for global analysis of dynamical systems: Applications of cell-to-cell mapping. *ASME J. of Applied Mechanics*, 47, 1986.

[34] K. Ogata. *Discrete Time Control Systems*. Singapore, second edition, 2004.

[35] K. Furuta. Sliding mode control of a discrete system. *System Control Letters*, 14:145–152, 1990.

[36] C. Edwards and S. K. Spurgeon. *Sliding Mode Control, Theory and Applications*, volume 7 of *Systems and Control Book Series*. Taylor and Francis Ltd., 1998.

[37] Y. Dote and R. G. Hoft. Microprocessor based sliding mode controller for dc motor drives. presented at the Ind. Applicat. Soc. Annu. Meeting, Clincinnati, OH.

[38] D. Milosavljevic. General conditions for the existance of quasi-sliding mode on the switching hyperplane in discrete variable structure systems. *Automa Remote Contr.*, 46:307–314, 1985.

[39] S. Z. Sarpturk, Y. Istefanopulos, and O. Kayank. On the stability of discrete-time sliding mode control systems. *IEEE Trans. Automa Contr.*, 32(10):930–932, 1987.

[40] A. Bartoszewicz. Discrete-time quasi-sliding-mode control strategies. *IEEE Trans. Ind. Electron*, 45:633–637, 1998.

[41] E. A. Misawa. Discrete-time sliding mode control for nonlinear systems with unmatched uncertainties and uncertain control vector. *ASME J. Dyn. Sys., Meas, and Control*, 119:503–512, 1997.

[42] H. Richter. Hyperplane design in observer-based discrete-time sliding mode control. Master's thesis, Oklahoma State University, Oklahoma, USA, May 1997.

[43] H. Johansson F. Roos and J. Wikander. Optimal selection of motor and gearhead in mechatronic applications. *Mechatronics*, 16:63–72, 2006.

[44] T. Kailath. *Linear Systems*. Prentice-Hall, second edition, 1980.

[45] C. K. Reddy and R. Pratap. Multimodal hop and complex basin of attraction of a simple hopper. *Physical Review E*, 68, 2003.

[46] D. M. Dawson L. Lewis and C. T. Abdallah. *Robot Manipulator Control: Theory and Practice*. Control Engineering Series. Marcel Dekker, New York, second edition, 2004.

Research papers published during the research tenure

Refereed Journals:

1. **Ajij Sayyad**, B. Seth and P. Seshu, "A Single Legged Hopping Robotics Research - A Review", *Robotica*, Vol. 25, Issue 05, pp. 587-613, September 2007.

2. **Ajij Sayyad**, K. K. Issac, and B. Seth, "An Asymmetrical One-legged Hopping Robot Control: Discrete-time Sliding Mode Approach", accepted to the special volume of Revue *Annals of the University of Craiova - Series: Automation, Computers, Electronics and Mechatronics*, printed by UNIVERSITARIA Craiova, Romania, ISSN: 1841-0626, April 2008.

Conference Proceedings:

1. **Ajij Sayyad**, B. Seth and K. K. Issac, "Dynamics and Control of a 2-D one-legged *SLOM* Hopping Robot", *Proc. of the IFToMM World Congress 2007 (IFToMM '07)*, France, pp. 1 - 6, June 17-21, 2007.

2. **Ajij Sayyad**, B. Seth and K. K. Issac, "Discrete-time sliding mode control of one-legged hopping robot", *Proc. of the Inter. Symposium SINTES-13*, Craiova, Romania, 18-20, October 2007.

Acknowledgements

'***Success***' is like reaping the fruit of a seed you planted long ago. I am not sure to mention the date when this seed (motivation towards research) was planted, but have been able to fulfill my vision successfully through continuous efforts and hardwork. This is only possible when you will get surrounded by good society, like the IIT Bombay. As per a say, "A society is as good as its people", my journey become easier when I took admission here in IIT Bomaby. I thank the almighty for bestowing me this opportunity and showering his blessings on me to come out successful against all odds. I bow my head with gratitude to salute this supreme power. In this voyage of endless research and perseverance, there were many people who have sailed with me till the date, may be for a long period or for a brief period. It is my duty and pleasure to reminisce and acknowledge all of them who have helped me directly or indirectly for all sorts of assistance that they have offered to me during the course of my research in this prestigious institute.

First of all, I would like to place on record my hearty thanks and sincere gratitude to my supervisors, Prof. Bhartendu Seth and Prof. K. Kurien Issac for their valuable guidance, constant encouragement, inspiration, motivation, suggestions from time to time. I am thankful to them for helping me in understanding the finer points of the topic with keen interest and cooperation in the hour of difficulties. I also thank them for guiding and supporting me *off the field*, even in matters no way connected to research topic. Working with them was a pleasure for me and that has aspired me to put perfection in the work.

Next, I would like to thank my progress evaluating committee members Prof. P. Seshu of Mechanical Engg. Dept., Prof. B. Bandyopadhyay and Prof. P. S. V. Nataraj and Prof. Ravi N. Banavar of Systems and Control Engg. Dept., Prof. Madhu Belur of Electrical Engg. Dept., IIT Bombay for their timely help and suggestions during the course of the research work.

I would like to express my gratitude towards my parents, Jinnatbi and Dildar. I am not sure to find words fit enough to express my feelings in this respect. It is said that actions speak louder than words. Perhaps, a lifetime of action is a good option to start with. A million thanks for keeping patience and faith in me, to my beloved wife, Naseem (Fatima). Naseem and my children, Mahvish and Aaryan who allowed me to slip away into the research for prolonged periods of time especially during the nights. Many a times I have missed them, when they needed me the most, especially during their teens. Words cannot substitute the sacrifices made by my wife during all these years **!** I will also take this opportunity to thank my parents-in-law and specially, Advocate Pathan Yunus, my brother-in-law, for providing me the encouragement, monetary help and moral backing to complete the research.

The stretching long working hours in Syscon Lab with my friends and other fellow researchers in Systems and Control and Mechanical Engg. Dept. of IIT Bombay interlaced with tea/coffee breaks, discussions was worth remembering. I express my deep sense of gratitude to all my teachers till date who have made me capable enough to come to this level and raise my head and look to the future.

I express my special thanks to IIT Bombay society, for driving me to this level through offering all necessary facilities. I can not forget their financial support (Institute scholarship) and family accommodation for about 4 years. I also thank for getting financial support for short duration jointly by IRCC and Systems and Control Engineering Department. It is not possible to list all peoples who supported me during my stay here directly or indirectly. I am grateful to their help.

Place : IIT Bombay, Mumbai-76.

Date : 11 May 2008 **Ajij D. Sayyad**