Reinforcement Learning for Stabilization and Performance of Mechanical Systems

A Project Report

submitted by

S BHUVANESWARI

in partial fulfillment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

May 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Reinforcement Learning for Stabilization and Performance of Mechanical Systems**, submitted by **S Bhuvaneswari**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Ramkrishna Pasumarthy
MTech Project guide
Assistant Professor
Dept. of Electrical Engineering
IIT Madras, Chennai - 600036

Dr. Balaraman Ravindran
MTech Project guide
Associate Professor
Dept. of Computer Science and Engineering
IIT Madras, Chennai - 600036

ABSTRACT

Reinforcement Learning is a branch of machine learning that tries to mathematically formulate and emulate learning as it happens in animals and human beings. This thesis investigates the applicability of reinforcement learning to the control and stabilization of two mechanical systems - the Twin-Rotor MIMO system and the Pendulum-on-a-cart system. First, reinforcement learning is employed in a model-free setting and stabilization is achieved without providing any knowledge of the system model to the RL controller. The control law is chosen to be bang-bang and the RL controller learns the parameters of this control law. This method is successfully applied on the Pendulum on a Cart system, both in simulation and real-time experiments.

Next, RL is used in the context of a control theoretic stabilization algorithm called IDA-PBC. In this method, the control takes the form of the IDA-PBC law and the RL controller learns its parameters, namely the Mass Matrix, the Potential energy and a damping coefficient term. Introducing RL to learn these parameters guarantees optimality in the control while still having the interpretability of the IDA-PBC. The learning experiments are performed and successfully applied on both the systems in simulation. IDA-PBC is an energy-based algorithm and is not very considerate about the system response characteristics measured in terms of the maximum overshoot, settling time etc. Using RL, learned a state-modulated damping control term which is added to the IDA-PBC control law to provide appropriate damping and improve the system response.

ACKNOWLEDGEMENTS

I am ever-grateful to my project guides and teachers, Dr. Ravindran and Dr. Ramkrishna for their immense support through out all the conversations I have had with them. Under their guidance, learning happened over a variety of domains from technical to work ethics to effective communication of thoughts. They have been patient enough with my shortcomings and facilitated me to embrace and learn from my mistakes. I thank them again for giving me the opportunity to work with them.

I would also like to thank Arun sir for his support. The technical discussions I had with him helped steer my work in the right direction at crucial junctures in the course of the project. I am also thankful to Anup for the useful discussions on TRMS. Thanks to Chaitanya for helping me out during the initial days of IDA-PBC. Thanks to Jayadev for readily listening to some technical problems and giving his suggestions. Thanks to all my lab-mates for making my stay in lab memorable.

IIT-Madras, as an institute cum campus, goes every way to help students enrich their life during their stay. With 24 hours security in departments, I could keep working in the department labs till late hours. The Sanskrit classes, the yoga classes in LTAP, the Manohar C.Watsa stadium and the campus deers and monkeys were real stress-busters amidst the academic schedule. Professors who taught me here will remain inspirations for my lifetime. Thank you IIT Madras for providing this experience.

Lastly, I thank my family and my mother who has been my pillar of strength throughout.

TABLE OF CONTENTS

List of Tables

List of Figures

ABBREVIATIONS

PDE	Partial Differential Equation
RL	Reinforcement Learning
IDAPBC	Interconnection & Damping Assignment Passivity Based Control
PH	Port Hamiltonian
SISO	Single Input Single Output
MIMO	Multiple Input Multiple Output
TRMS	Twin Rotor MIMO System

Chapter 1

Introduction

> "Reinforcement learning is one of the major neural-network approaches to learning control. How should it be viewed from a control systems perspective? Control problems can be divided into two classes: 1) regulation and tracking problems, in which the objective is to follow a reference trajectory, and 2) optimal control problems, in which the objective is to extremize a functional of the controlled system's behavior that is not necessarily defined in terms of a reference trajectory. Adaptive methods for problems of the first kind are well known, and include self-tuning regulators and model-reference methods, whereas adaptive methods for optimal-control problems have received relatively little attention. Moreover, the adaptive optimal-control methods that have been studied are almost all indirect methods, in which controls are recomputed from an estimated system model at each step. This computation is inherently complex, making adaptive methods in which the optimal controls are estimated directly more attractive. We view reinforcement learning methods as a computationally simple, direct approach to the adaptive optimal control of nonlinear systems"

The above paragraph quoted from [12] strongly motivates reinforcement learning as a way of doing adaptive-optimal control. This thesis applies reinforcement learning to control theory problems of stabilization of two mechanical control systems - the Twin-Rotor MIMO system and the Pendulum on a Cart system. Learning is introduced in two variants. In the first variant, the RL controller learns the parameters of an arbitrary control law(in this thesis, a bang-bang control law is used) to stabilize the system using a

suitable rewarding mechanism without any knowledge of the system model. In the second variant, the RL controller is used to learn the parameters of an IDA-PBC control law, parametrized by M_d(closed loop mass matrix), V_d(closed loop potential energy function) and K_v(damping coefficient). This approach is inspired from [8] which learns an IDA-PBC control law for a simple, fully-actuated SISO inverted pendulum system. In this thesis, this procedure is extended to more complex 2-input, 2-output system like the Twin-rotor system and the under-actuated Pendulum on a Cart system.

The second variant of learning requires the knowledge of the system model. However, compared to the model-free learning of the first variant, it buys a lot more advantages in that cost. For instance, in control theory, the IDA-PBC control law is typically obtained by mathematically solving a set of complex partial differential equations for the Mass matrix and Potential energy and by choosing an arbitrary positive damping term. Instead, employing RL to learn these parameters guarantees optimality in the IDA-PBC control. More importantly, the resultant IDA-PBC control law applied on the system is still interpretable in terms of energy shaping part and damping injection part. Compare this to the first approach where the bang-bang control law is learned with no way of relating which component of the control law to be responsible for energy shaping or damping etc. Thus, the resulting control is both optimal and interpretable.

The IDA-PBC algorithm typically has system response characteristics with multiple overshoots and longer settling and rise times. It is possible to add an extra state-modulated damping term to the IDA-PBC control law to provide better system response curves while still maintaining the passivity and the port-Hamiltonian structure of the overall closed loop system as in [4]. This uses a pre-designed functional form for the state-modulated damping term. However, when RL is used to learn this function, the design process is automated and at the end of learning, a reward-optimal damping function that minimizes both overshoot and settling time is learned.

What are the other motivations for employing learning to control? Control theory in general, assumes perfect knowledge about the system of interest and the objective is to find the best control law to be applied to the system in order to achieve a control task. However, due to the presence of noise or other factors whose interactions with the system are difficult to model, it is not always possible to come up with a perfect system model. In this viewpoint, reinforcement learning methods are on-line algorithms which are continually responsive to system changes and hence do not require a perfect system model. Another point to note is the ability of RL to learn a control law respecting the

physical system's constraints like Control input saturation[1] in addition to handling the noise factors of the real-time system.

On the contrary, it is also apparent that RL cannot learn a good controller without failing in the task initially. It would be disastrous to employ RL to tasks where failures matter more than the successes[2]. These initial failures in the task and the time taken to learn a control law from the interactions with the system are some of the compromises of RL, so to say.

1.1 Organization of the thesis

Chapter 1 of this thesis puts forth the problem statement and motivates the application of RL to the control theory problem of stabilization of mechanical systems. Chapter 2 provides the system specifications of the mechanical control systems used, namely the Pendulum on a Cart system and the Twin-rotor MIMO system. Chapter 3 explains the idea of IDA-PBC algorithm applied on port-Hamiltonian systems, and also shows how the complex PDEs can be simplified to system-specific ODEs for the two mechanical systems under consideration. These ODEs are later used in Chapter 5 to set up the RL controller. State-modulated damping is introduced to improve the system response characteristics of IDA-PBC. Chapter 4 introduces basic terminology used in Reinforcement Learning and also discusses the Actor-Critic learning algorithm which is used for the learning experiments in Chapter 5. Chapter 5 discusses the implementation of the actor-critic algorithm in two different settings:-model-free setting and in a model-based setting to learn the parameters of the IDAPBC algorithm. Results obtained in simulation/real-time are also included. Chapter 6 concludes the thesis.

[1]Many physical systems allow only control values within a limited range, say [-2.5 V,2.5 V]

[2]Safe Reinforcement Learning is an active area of research to address the safety concerns during the learning process of RL[7]

Chapter 2

Mechanical Systems of interest

2.1 Pendulum on a Cart

The pendulum on a cart is a SIMO(Single Input Multiple Output) plant. The two outputs from the plant are the cart position and the pendulum angle readings from the sensors. The control input can directly control only the position of the cart on the track. Therefore the system is under-actuated. Figure 2.1 shows the system setup used for experiments and 2.2 shows the flow of signals in the control system. Summing the forces acting on the

Figure 2.1: Pendulum on a Cart setup used for experiments[3]

system, we obtain the following nonlinear system dynamics equations:

$$\begin{aligned}(m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta &= F \\ (I+ml^2)\ddot{\theta} - mgl\sin\theta + ml\ddot{x}\cos\theta + d\dot{\theta} &= 0\end{aligned} \tag{2.1}$$

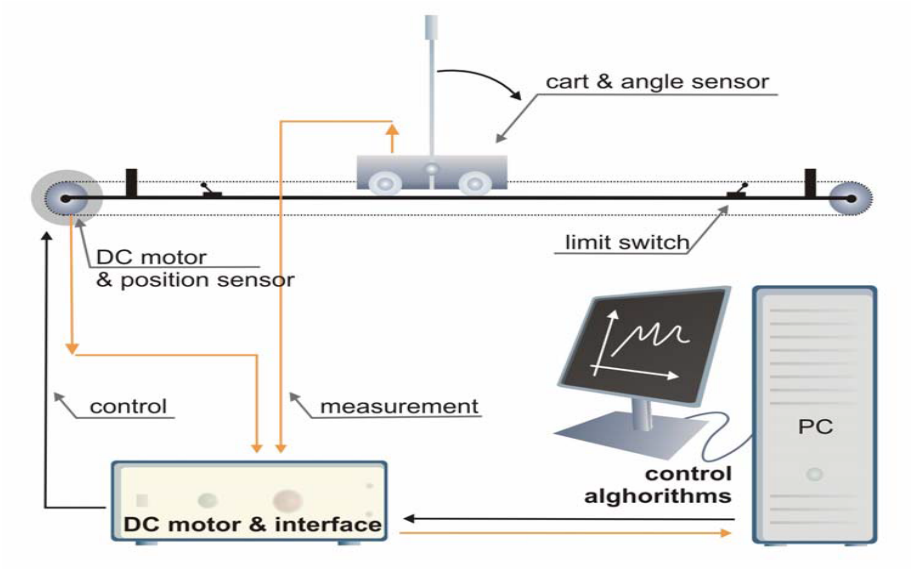

Figure 2.2: Pendulum Control System[3]

with the following system parameters

Parameter	Value
g - gravity	9.81 m/s^2
l - pole length	0.36 m
M - cart mass	2.4 kg
m - pole mass	0.23 kg
I - moment of inertia of the pole	0.099 kgm^2
b - cart friction coefficient	0.05 Ns/m
d - pendulum damping coefficient	negligible, 0.005 Nms/rad

Table 2.1: Pendulum on a Cart system parameters

x and θ are the position of the cart on the track and the pendulum angle respectively. The system has two equilibrium points at $\theta = 0$(inverted pendulum) and $\theta = \pi$(freely hanging pendulum). Hence, stabilization is possible only in one of these two positions. The control task in this system is to stabilize the system in the inverted, unstable equilibrium position $\theta = 0$. Both the cart position and the control signal are bounded for this system. The bound for the control signal is the interval [-2.5V...2.5V] and the cart position is limited to [-0.4m...0.4m].

Figure 2.3: Twin Rotor MIMO System[13]

2.2 Twin Rotor MIMO System (TRMS)

The Twin Rotor System demonstrates the principles of a non-linear multi-input, multi-output(MIMO) system. Its dynamics is similar to that of a helicopter though the underlying mechanism of creating thrust forces on the rotor blades is different. Unlike the helicopter, the angle of attack of the rotors is fixed and the pitch/yaw thrust forces are varied by adjusting the speeds of the motors controlled by the two input voltages. The setup has two perpendicular rotors/propellers driven by DC motors at the ends of a beam, which is pivoted on a stationary base. The larger pitch rotor moves the system in the pitch direction of angular motion. Similarly, the smaller yaw rotor moves the system in the yaw direction. The beam can rotate in the pitch and yaw directions of angular motion as can be seen from Figure (2.3). Significant cross-coupling is observed in the control inputs of the two rotors, with the input control signal to the pitch/yaw rotor affecting both the Pitch and Yaw angular positions. Also it is clear from figure(2.3) that the system can neither fly nor be controlled in the roll direction of angular motion. Control algorithms are tested for regulating the system to desired pitch and yaw angles. Unlike the Pendulum on a Cart system, the equilibrium point to stabilise the system could be any pitch or yaw angular position admissible in the physical system. The system dynamics of the TRMS is more detailed than that of the Pendulum on a Cart due to the angular speeds of the rotors also entering into the system dynamics[1]. Additionally, the pitch thrust/yaw thrust forces and the corresponding control inputs to the pitch and yaw rotors are also not linearly related. The readers are referred to [5] for a detailed derivation of the full system dynamics and to [13] for the setup in figure(2.3).

[1]In addition to the pitch angle(α_v), yaw angle(α_h), $\dot{\alpha_v}$ and $\dot{\alpha_h}$, the rotor speeds(ω_v and ω_h) also affect the system dynamics

Chapter 3

Interconnection and Damping Assignment Passivity Based Control

3.1 Introduction to IDA-PBC

Passivity-based control (PBC) is an energy-based method to achieve stabilization of a system by passivation of the closed-loop dynamics. The objective here, is to render the closed-loop system passive with a stored energy function that has a minima at the desired equilibrium state.

Consider a system with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$ and output $y \in \mathbb{R}^m$. The system map from $u \rightarrow$y is passive if there exists a state-dependent function $H(x)$, bounded from below, and a non-negative function $d(t) \geq 0$ such that

$$\underbrace{\int_0^t u^T(s)y(s)ds}_{\text{energy supplied to the system}} = \underbrace{H(x(t)) - H(x(0))}_{\text{stored energy}} + \underbrace{d(t)}_{\text{dissipated energy}} \quad (d(t) \geq 0) \tag{3.1}$$

The system's stored energy $H(x(t)) - H(x(0))$ is also called the Hamiltonian function of the system. The above definition of a passive system implies that for a bounded energy supplied by the inputs, the system has a bounded stored energy for $t \geq 0$. Thus a passive system is a stable system. Also, if this passive system is left at rest, the system stabilizes at the position of lowest energy configuration of the system.

Passivity Based Control (PBC) applies control to make the closed loop system passive at the desired position configuration. The control action is split into two parts:

1. Energy Shaping

2. Damping Injection

Given an unstable mechanical system, the energy shaping part transforms the system via feedback to a stable mechanical system whose stored energy function gets a minimum value at the equilibrium point of interest. The damping injection part injects a dissipative feedback force to obtain asymptotic stability at this equilibrium point.

Interconnection and Damping Assignment Passivity Based Control (IDA-PBC), in addition to following the same steps of PBC also preserves the closed-loop structure of the system by allowing to choose for a desired interconnection and damping structure for the closed loop system. An IDA-PBC controller can only be applied to systems that can be represented in a port-Hamiltonian(PH) structure. Given below is the general form of system dynamics of a port-Hamiltonian system

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \Big[J(x) - R(x) \Big] \begin{bmatrix} \nabla_q H(x) \\ \nabla_p H(x) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \tag{3.2}$$

Here $q \in \mathbb{R}^{n\times 1}$, $p \in \mathbb{R}^{n\times 1}$ are the generalized position and momentum vectors of the system. n is the number of degrees of freedom in the system. The variable x encompasses the state of the system, ie. $x = [q, p]$. J $\in \mathbb{R}^{n\times n}$ and R $\in \mathbb{R}^{n\times n}$are the natural Interconnection and Damping matrices of the system respectively. In the case of mechanical systems, if we assume no natural damping in the system, R can be neglected and equation 3.2 can be re-written as

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \tag{3.3}$$

H is the total energy or the Hamiltonian of the open loop system given by

$$H(q, p) = \frac{1}{2} p^T M^{-1}(q) p + V(q) \tag{3.4}$$

where M is the Mass matrix($M = M^T > 0$) and V is the potential energy of the open loop system. The matrix G captures the actuated degrees of freedom in the system. For instance, G $\in \mathbb{R}^{2\times 1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ for a Pendulum-on-a-Cart system with under-actuation degree one. For the fully actuated Twin Rotor system, G $\in \mathbb{R}^{2\times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Since IDAPBC preserves the closed loop structure, a port-Hamiltonian system in feedback with an IDAPBC controller also yields a port-Hamiltonian system in closed loop. The

closed loop system dynamics can be taken to be of the form

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} J_d - R_d \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} \tag{3.5}$$

$$J_d = -J_d^T = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(q,p) \end{bmatrix}; J_2 \text{is skew-symmetric} \tag{3.6}$$

$$R_d = R_d^T = \begin{bmatrix} 0 & 0 \\ 0 & GK_vG^T \end{bmatrix}; K_v > 0 \tag{3.7}$$

where R_d and J_d represent the desired interconnection and damping structures for the system. The total energy function can similarly be written as

$$H_d(q,p) = \frac{1}{2}p^T M_d^{-1}(q)p + V_d(p,q) \tag{3.8}$$

where $M_d{=}M_d^T > 0$ and V_d represent the closed loop mass matrix and potential energy function respectively. To get the minima of the stored energy of the closed loop system at the desired position q^*, it is also required that

$$q^* = \text{argmin } H_d(q,p) = \text{argmin } V_d(q) \tag{3.9}$$

The second equality follows in equation (3.9) since the task is that of stabilisation where the final system configuration is at rest with zero kinetic energy. In addition to the port-Hamiltonian structure, the system also retains its passivity in closed loop. We can mathematically check the passivity of the closed loop system using the differential form of equation(3.1) as follows

$$\begin{aligned} \dot{H}_d(x) &= (\nabla_x H_d(x))^T \dot{x} \\ &= (\nabla_x H_d(x))^T (J_d - R_d)(\nabla_x H_d(x)) \\ &= (\nabla_x H_d(x))^T J_d (\nabla_x H_d(x)) - (\nabla_x H_d(x))^T R_d (\nabla_x H_d(x)) \\ &= -(\nabla_x H_d(x))^T R_d (\nabla_x H_d(x)) \text{ Since } J_d \text{ is skew-symmetric} \end{aligned} \tag{3.10}$$

This implies that the closed loop system is passive as long as the damping matrix R_d is positive-definite.

Following the energy shaping and damping injection steps of IDAPBC, the control law can be split as

$$u = u_{es}(q,p) + u_{di}(q,p) \tag{3.11}$$

The following can be observed from the above equations:

1. From the first component of the vector equation (3.3) and the vector equation (3.5), the following result is valid in both open-loop and closed-loop,

$$\dot{q} = M^{-1}p \tag{3.12}$$

2. The matrix R_d is included to add damping into the system. This is achieved via damping injection step which gives a negative feedback of the passive output[9], $G^T\nabla_p H_d$. Hence

$$u_{di} = -K_v G^T \nabla_p H_d \tag{3.13}$$

Substituting this expression for u_{di} in the matching equation (3.14) explains the choice of the chosen for R_d in equation (3.7).

3. Usually, the mass matrix M is totally determined by the position coordinate, q. For a fully-actuated system, $M(q)$ is a fixed function of q for the system. Hence energy shaping can only shape the potential energy of the system satisfying the minima criterion of equation (3.9). However, for an under-actuated system, the closed loop mass matrix M_d can be different from the open-loop mass matrix M of the system. Energy shaping can shape both the kinetic and potential energy terms through the Kinetic PDE and the potential PDE constraints as discussed later in equations (3.19) and (3.20).

4. The skew-symmetric matrix J_2 and some of the elements of M_d can be used as free parameters in order to achieve kinetic energy shaping.

Equations (3.3) and (3.5) are dynamics of the same system in closed loop and hence can be equated. The damping terms cancel out in this equation. We obtain

$$\begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u_{es} = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(q,p) \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix} \tag{3.14}$$

This equation is also called the **matching equation** in the context of IDAPBC. The success of IDAPBC relies upon the ease of coming up with a solution for the closed loop variable H_d, and in turn M_d and V_d from this matching equation. The first row component of the vector equation (3.14) is clearly satisfied. The second row of equations can be expressed as

$$Gu_{es} = \nabla_q H - M_d M^{-1}\nabla_q H_d + J_2 M_d^{-1} p \tag{3.15}$$

This gives us the following expression for u_{es}

$$u_{es} = (G^T G)^{-1} G^T (\nabla_q H - M_d M^{-1}\nabla_q H_d + J_2 M_d^{-1} p) \tag{3.16}$$

subject to the corresponding constraint in the null space of G,

$$G^{\perp}\{\nabla_q H - M_d M^{-1}\nabla_q H_d + J_2 M_d^{-1} p\} = 0 \tag{3.17}$$

where $G^{\perp}G = 0$. Substituting equations(3.16) and (3.13) into equation (3.11) gives the IDAPBC control law as

$$u = (G^T G)^{-1} G^T (\nabla_q H - M_d M^{-1}\nabla_q H_d + J_2 M_d^{-1} p) - K_v G^T \nabla_p H_d \tag{3.18}$$

Grouping the terms in the constraint PDE (3.17) by powers of p, the PDEs can be split into the terms corresponding to the kinetic and potential energies, respectively. This leads to

$$\textbf{KE-PDE :}\ G^{\perp}\{\frac{1}{2}[\nabla_q (p^T M^{-1} p) - M_d M^{-1}\nabla_q (p^T M_d^{-1} p)] + J_2 M_d^{-1} p\} = 0 \tag{3.19}$$

$$\textbf{PE-PDE :}\ G^{\perp}\{\nabla_q V - M_d M^{-1}\nabla_q V_d\} = 0 \tag{3.20}$$

The idea is to choose the free parameter J_2 in such a way that the PDE (3.19) admits, for all p, a solution with M_d symmetric and positive definite. This matrix M_d is then replaced into equation (3.20), which is a PDE involving only q, and solved for a V_d which satisfies equation (3.9). The next section gives the illustration of IDAPBC on the two systems of interest - TRMS and the Pendulum on a Cart. It is to be noted that the KEPDE is trivially satisfied for a fully-actuated system and the solving process is easy. However, for an under-actuated Pendulum on a Cart system, the constraint equations do not vanish and the solution process is not straightforward. To get an illustration of this entire solving process, the readers are encouraged to go through [1] which has methodically solved the IDAPBC problem for the Pendulum-on-a-cart system. One way of getting ready to solve these PDE constraints is to simplify them into system-specific ODE constraints, if possible, as shown in [9] and then come up solutions for these ODEs. We will follow the simplification approach in the following sections.

3.2 IDAPBC for Pendulum on a Cart system

The Pendulum on a Cart system has two degrees of freedom - Position of the cart on the track(q_1) and Pendulum angle(q_2), out of which only q_1 is actuated as explained in figure(3.1). On the lines of equation(3.2), assuming no natural damping in the system,

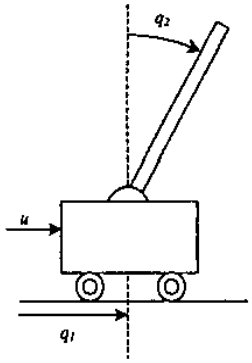

Figure 3.1: Position coordinates in Pendulum on Cart system [9]

the port-Hamiltonian model for the Pendulum on a Cart system can be written as

$$\begin{aligned} \dot{x} &= \Big[J - R\Big] \nabla_x H + \begin{bmatrix} 0 \\ G \end{bmatrix} u \\ \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} &= \begin{bmatrix} 0 & I_2 \\ -I_2 & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \qquad q, p \in \mathbb{R}^2, u \in \mathbb{R} \end{aligned} \tag{3.21}$$

$q =< q_1, q_2 >$ are the position of the cart on the track and the pendulum angle respectively. $p =< p_1, p_2 >$ are the corresponding momenta associated with the motion in q-coordinates. The system interconnection matrix(J), the damping matrix(R) and the G matrix being

$$J = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \; ; R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \; ; G = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

and the total energy of the system

$$H = \frac{1}{2} p^T M^{-1} p + V(q) \tag{3.22}$$

with

$$M(q) = \begin{bmatrix} \mathrm{a} & \mathrm{c}\cos q_2 \\ \mathrm{c}\cos q_2 & \mathrm{b} \end{bmatrix} \text{ and } V(q) = -\mathrm{mgl}\cos q_2 \tag{3.23}$$

a = ml^2, b = m + M and c = ml. m denotes the mass of the pendulum, M the mass of the cart and l, the length of the pendulum. *Note that for the pendulum on a cart system, IDA-PBC can only stabilize the system if* $q_2 \in (\frac{-\pi}{2}, \frac{\pi}{2})$. Refer to [9] or [1] for details on such a restriction.

As already stated, in this thesis, we try to simplify the PDEs to system-specific ODEs and then find the solutions of the ODEs. For simplifying the KE-PDE (3.19) for the Pendulum on a Cart system, the following results from [9] are used.

- The closed loop mass matrix M_d is a function of the coordinate q_2 alone. It is also clear that M is also a function of q_2 only from equation(3.23).

- $$\nabla_q M = e_2 \nabla_{q_2} M \text{ and } \nabla_q M_d = e_2 \nabla_{q_2} M_d \tag{3.24}$$

- $$\nabla_{q_2} M^{-1} = -M^{-1}(\nabla_{q_2} M)M^{-1} \text{ and } \nabla_{q_2} M_d^{-1} = -M_d^{-1}(\nabla_{q_2} M_d)M_d^{-1} \tag{3.25}$$

Additionally, the following form for J_2 is taken from [1]

$$J_2 = p^T M_d^{-1} \alpha W \text{ where } \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \text{ and } W = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{3.26}$$

Also since $G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $G^\perp$ takes the form $G^\perp = e_2^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$. This makes $G^\perp = e_2^T$. Also, it is to be recalled that if $x^T A x = 0 \ \forall$ non-zero x, then A can be a zero matrix or a skew-symmetric matrix.

Starting from the KE-PDE,

$$\begin{aligned}
G^\perp\{\nabla_q(p^T M^{-1} p) - M_d M^{-1} \nabla_q (p^T M_d^{-1} p) + 2 J_2 M_d^{-1} p\} = 0 \\
G^\perp e_2 \nabla_{q_2}(p^T M^{-1} p) - G^\perp M_d M^{-1} e_2 \nabla_{q_2}(p^T M_d^{-1} p) + 2(p^T M_d^{-1} \alpha) G^\perp W M_d^{-1} p = 0 \\
\Rightarrow p^T [G^\perp e_2 \nabla_{q_2} M^{-1} - G^\perp M_d M^{-1} e_2 \nabla_{q2} M_d^{-1} + 2 M_d^{-1} \alpha G^\perp W M_d^{-1}] p = 0 \\
\Rightarrow p^T [e_2^T e_2 \nabla_{q_2} M^{-1} - e_2^T M_d M^{-1} e_2 \nabla_{q2} M_d^{-1} + 2 M_d^{-1} \alpha e_2^T W M_d^{-1}] p = 0 \\
\Rightarrow p^T [\nabla_{q_2} M^{-1} - [M_d M^{-1}]_{22} \nabla_{q2} M_d^{-1} + 2 M_d^{-1} \alpha e_2^T W M_d^{-1}] p = 0
\end{aligned} \tag{3.27}$$

Denoting $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix} = M_d M^{-1}$, $[M_d M^{-1}]_{22}$ can be replaced with λ_4. In the last step, the term in the square brackets factorized by $p^T[\ ]p$ is either a zero or a skew-symmetric matrix. Since the first and second matrix terms inside these brackets are symmetric, the sum of these three matrix terms cannot be skew-symmetric. Hence

$$[\nabla_{q_2} M^{-1} - \lambda_4 \nabla_{q2} M_d^{-1} + 2 M_d^{-1} \alpha e_2^T W M_d^{-1}] = 0$$

This can be further simplified as follows

$$[-M^{-1} \nabla_{q_2}(M) M^{-1} + \lambda_4 M_d^{-1} \nabla_{q2} M_d M_d^{-1} + 2 M_d^{-1} \alpha e_2^T W M_d^{-1}] = 0$$

Pre-multiplying by $-M_d$ and Post-multiplying by M_d,

$$[M_d M^{-1} \nabla_{q_2}(M) M^{-1} M_d - \lambda_4 \nabla_{q2} M_d - 2\alpha e_2^T W] = 0$$

Or,

$$[-\lambda\nabla_{q_2}(M)\lambda^T + \lambda_4\nabla_{q2}(M_d) + 2\alpha e_2^T W] = 0 \tag{3.28}$$

The third term in equation(3.28) can be split into symmetric and skew-symmetric part as

$$2\alpha e_2^T W = 2\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

$$2\alpha e_2^T W = -2\begin{bmatrix}\alpha_1 & 0\\ \alpha_2 & 0\end{bmatrix} = \begin{bmatrix}-2\alpha_1 & -\alpha_2\\ -\alpha_2 & 0\end{bmatrix} + \begin{bmatrix}0 & \alpha_2\\ -\alpha_2 & 0\end{bmatrix}$$

Since the skew-symmetric component of $2\alpha e_2^T W$ cannot contribute to equation (3.28), we have

$$-\lambda\nabla_{q_2}(M)\lambda^T + \lambda_4\nabla_{q2}(M_d) + \begin{bmatrix}-2\alpha_1 & -\alpha_2\\ -\alpha_2 & 0\end{bmatrix} = 0$$

Substituting for M from equation (3.23), the above matrix equation can be broken down into 3 element-wise equations as follows,

$$\begin{gathered}2\lambda_1\lambda_2 c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_1 a + \lambda_2 c\cos(q_2)) - 2\alpha_1 = 0\\ (\lambda_1\lambda_4 + \lambda_2\lambda_3)c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_1 c\cos(q_2) + \lambda_2 b) - \alpha_2 = 0\\ 2\lambda_3\lambda_4 c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_3 c\cos(q_2) + \lambda_4 b) = 0\end{gathered} \tag{3.29}$$

The PE-PDE (3.20) can be simplified using $V = \text{mgl}\cos(q_2)$ as follows

$$\begin{gathered}G^\perp\nabla_q V = G^\perp M_d M^{-1}\nabla_q V_d\\ \Rightarrow -\text{mgl}\sin(q_2) = \lambda_3\frac{\partial V_d}{\partial q_1} + \lambda_4\frac{\partial V_d}{\partial q_2}\end{gathered} \tag{3.30}$$

This section stops with this simplification process and does not attempt to solve the M_d and V_d satisfying the simplified PDEs in equation(3.29) and equation(3.30) and the equation(3.9) since the process is complex. These equations are later used in Chapter 5 where it is shown how RL can be used to learn the solutions of these equations.

3.3 IDAPBC for Twin Rotor MIMO system

The Twin Rotor MIMO System is a fully-actuated system with two degrees of freedom - Pitch angle(q_v) and Yaw angle(q_h). The control objective in this system is to regulate

the angular positions to a desired point, q^*. The corresponding control inputs u_v and u_h do not directly provide the thrust force in the pitch and yaw direction. The control inputs affect the speeds of the dc motors mounted on each rotor, which then provide the required thrust in the pitch and yaw direction. This mechanism of producing the thrust forces necessarily includes the rotor speeds - Speed of the main/yaw rotor(ω_m) and Speed of the tail rotor/pitch rotor(ω_t) as the state variables of the system. The resulting system has 6 state variables $< q_v, q_h, \dot{q_v}, \dot{q_h}, \omega_m, \omega_t >$,[1] out of which ω_m and ω_t are not generalized state variables. IDA-PBC does not hold valid for systems which do not have generalized state variables. To observe the effect of IDAPBC on this system, one reasonable assumption to make is that the control inputs are directly the thrust forces in the pitch and yaw direction. This assumption truncates the original system dynamics of ω_m and ω_t and gives a system on which IDA-PBC can be applied.

Following this assumption, the system dynamics of the TRMS is written on the lines of equation (3.2) as

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \Big[J - R\Big] \nabla_x H(x) + Gu \tag{3.31}$$

$q =< q_v, q_h >$ are the pitch and the yaw angles. $p =< p_v, p_h >$ are the corresponding momenta with the system interconnection matrix, damping matrix and the G matrix being

$$J = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \; ; \; R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & b_v & 0 \\ 0 & 0 & 0 & b_h \end{bmatrix} \; ; \; G = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.32}$$

b_v and b_h are the damping coefficients with b_v,$b_h > 0$. The Hamiltonian function H for the system is defined

$$\begin{gathered} H(q,p) = V(q_v) + \frac{1}{2} p^T M^{-1}(q_v) p \text{ where} \\ V(q_v) = \delta_v \cos(q_v) + \delta_h \sin(q_v) \text{ and} \\ M = \begin{bmatrix} a_1 & a_2 \cos(q_v) - a_3 \sin(q_v) \\ a_2 \cos(q_v) - a_3 \sin(q_v) & a_5 + a_4 \cos^2(q_v) \end{bmatrix} \end{gathered} \tag{3.33}$$

The a_i's and δ_i's are system constants dependent on the mass and the geometry of the system as defined in Chapter 2 of [5]. Since the system is fully-actuated, the closed-loop mass matrix(M_d) has to be same as the open-loop mass matrix(M) and only the

[1] Refer to [5] for detailed derivation of the complete system dynamics

closed-loop potential energy(V_d) can be shaped. Hence we have $J_d = J$, $R_d = R$ and

$$H_d(q,p) = (\frac{1}{2}\gamma_v \widetilde{q_v}^2 + \frac{1}{2}\gamma_h \widetilde{q_h}^2) + \frac{1}{2}p^T M^{-1}(q_v)p \tag{3.34}$$

such that $\gamma_v,\ \gamma_h > 0$ and $\widetilde{q_v} = q_v - q_v^*$ and $\widetilde{q_h} = q_h - q_h^*$. It is to be noted that the parametrization for V_d in equation3.34 satisfies the minima condition in equation(3.9) for positive γ_v,γ_h. Solving the IDA-PBC control law from the matching equation for the system gives the following expression

$$\begin{aligned} u &= (G^T G)^{-1} G^T [(J_d - R_d)\frac{\partial H_d}{\partial x} - (J - R)\frac{\partial H}{\partial x}] \\ u &= \begin{bmatrix} \gamma_h \widetilde{q_h} \\ -\gamma_v \widetilde{q_v} - \delta_v \sin(q_v) + \delta_h \cos(q_v) \end{bmatrix} \end{aligned} \tag{3.35}$$

Note that the KE-PDE constraints that came up for the under-actuated, Pendulum on a Cart system in the previous section are trivially satisfied for this fully-actuated system and the control law is easily solved in equation(3.35). Also, the IDA-PBC control law works for any $\gamma_v > 0$ and $\gamma_h > 0$. Instead of making this choice arbitrary, this result is used in Chapter 5 to learn the optimal values of γ_v and γ_h.

3.4 State-modulated damping with IDA-PBC

State-modulated damping provides a damping control as a function of the state. Intuitively, the idea of using this state-dependent damping control is to provide higher damping when the system is closer to the desired state and lesser damping when the system is farther from the desired state. A circuit element called memristor is used to provide state-modulated damping as in [4].

As a circuit element[2], a memristor can be seen as a variable resistance dependent on the charge q, which is the state of the system. The current i and the voltage across its terminals v are the output and input of the system respectively.

$$v = r_M(q)i \tag{3.36}$$

Outside Circuit theory, in a more general setup, a memristor can be given by a similar expression using corresponding input, output and state variables. It is to be noted that a memristor is a passive element and its dynamics can be written in a port-Hamiltonian

[2]A memristor is a hypothetical circuit element that is modeled only in simulations

form as follows [3]

$$\begin{aligned} \dot{x}_M &= u_M \\ y_M &= \widetilde{M}(x_M)u_M \end{aligned} \tag{3.37}$$

where $\widetilde{M}(x_M)$ denotes the memristor as a function of the state. Notice the similarity between equations (3.37) and (3.36). Consider a passive, port-Hamiltonian system denoted by $\sum$. Connecting an IDA-PBC controller in feedback to this system gives a passive, port-Hamiltonian system $\sum_c$ in closed loop. If $\sum_c$ is again connected in feedback to a passive, port-Hamiltonian memristor element, it gives an other port-Hamiltonian, passive system $\sum_M$. Connecting a memristor in this way adds an extra state-modulated damping term to the original IDA-PBC control law. This damping can clip the unnecessary overshoots or reduce the rising time in the system response of a simple IDA-PBC control. The overall control law takes the following form

$$u = (G^T G)^{-1} G^T (J_d - R_d - G\widetilde{M}G^T)\frac{\partial H_d}{\partial x} - (J - R)\frac{\partial H}{\partial x} \tag{3.38}$$

Equation (3.38) has an extra damping term compared to the original IDA-PBC control given in equation (3.35). If the IDA-PBC control could be called u_c, then the above equation can be simplified to

$$\begin{aligned} u &= u_c - \widetilde{M}G^T\frac{\partial H_d}{\partial x} \\ u &= u_c - \widetilde{M}M^{-1}p \\ u &= u_c - \widetilde{M}\dot{q} \end{aligned} \tag{3.39}$$

In particular, for the TRMS with two actuated states, the desired memristance $\widetilde{M}$ takes the form of a 2×2 matrix

$$\widetilde{M} = \begin{bmatrix} f_v(\widetilde{q_v}) & 0 \\ 0 & f_h(\widetilde{q_h}) \end{bmatrix} \tag{3.40}$$

However, choosing the functional form for the elements of $\widetilde{M}$ is a design question since different functions produce different system responses. [4] uses pre-defined bell-shaped curves for $\widetilde{M}$ which have to be carefully chosen for the system and the design requirements at hand. Instead of using these pre-defined functional forms, it makes the design more easier by learning these functions online in an RL framework. Accordingly, the memristor is connected in feedback with a closed-loop IDA-PBC system and the functional form for the elements of the $\widetilde{M}$ matrix, f_v and f_h, are learned. The details of this experiment conducted for the TRMS are in Chapter 5.

[3] Refer [4] for a more general expression

Chapter 4

Reinforcement Learning

The basic idea in reinforcement learning is to build a controller to solve a sequential decision making problem. The controller takes a series of actions in order to maximise the total evaluative feedback at a future time step. The evaluative feedback, also called as the reward, is designed to be indicative of what task is to be achieved. For a racer bot, it could be a 'pat on the back' after winning the race or a hard punch on losing it. The RL controller is called the agent and the system to be controlled is called the environment. RL tries to fit the decision making task into a framework of an agent interacting with an environment to achieve a goal, under the guidance of a reward which is given to the agent by the environment. Therefore, the implicit assumption is that the rewards are provided to the agent in such a way that maximizing them achieves the goal for us. RL also assumes that the system/environment with which the learning agent interacts follows markovian dynamics. Most tasks encountered in real world follow this property. In particular, the system dynamics of mechanical control systems discussed in this thesis are markov.

The agent and the environment interact at each of a discrete sequence of time steps t=0,1,2,3.. At time t, the agent sees the environment at state S_t and selects action A_t. The environment responds to the action A_t and transitions to a new state S_{t+1}. In addition to the next state information, the agent also gets a numerical reward signal R_{t+1} from the environment as a consequence of taking action A_t in state S_t and reaching state S_{t+1}. Here, the state S_t belongs to the set of possible states $\mathbb{S}$, the action A_t belongs to the set of possible actions $\mathbb{A}$ and the reward R_{t+1} belongs to the set of real numbers $\mathbb{R}$.

To take an action at each time step t, the agent maintains a mapping from the states in $\mathbb{S}$ to the probabilities of selecting each possible action in $\mathbb{A}$. This mapping is called the

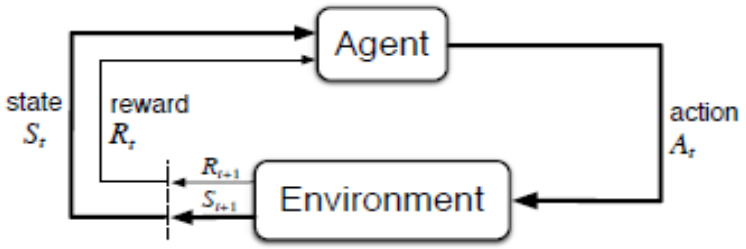

Figure 4.1: Reinforcement Learning happens in the course of a series of interactions of the agent with the system under the guidance of an evaluative signal, called reward [11].

agent's *policy* $\pi_t(a|s)$[1]. This policy is the probability of taking action a in state s.

One of the main challenges of solving this sequential decision task under the guidance of a scalar reward signal obtained at each step is the temporal credit assignment problem. In the racer bot example, the agent takes actions (here the actions could be the changes in the speed of running or the adjustments in the stride length) at every state of the system in a trial-and-error fashion and gradually learns the mapping between the actions that could be taken and the state of the system using the reward. But the reward signal could be delayed. The bot gets the pat or the punch only at the end of the race. Because of this, it is important to find out which of the actions taken at what instant during the race actually led to winning or losing the race. And the good news is RL algorithms solve it. Another main challenge is the exploration-exploitation dilemma faced by the agent. The readers are encouraged to follow the online NPTEL course on Reinforcement Learning or read the book [11] for deeper technical understanding.

4.1 Basic definitions in RL

The agent's goal, as already said, is to maximise the cumulative reward it receives over the long run. To formally define how rewards can be cumulated, we have the notion of *return* in an RL task. Generally tasks in RL fall into two main categories - Episodic and Continuing. Episodic tasks are where the agent-environment interaction breaks naturally into intervals of finite time steps, such as the plays of a game or the trips through a maze. In these cases, the agent momentarily stops learning in finite time steps when the game is won/lost or when the agent successfully comes out of the maze for example. Learning

[1]The policy π is synonymous with the control law u in a feedback control system

is resumed by placing the agent in some starting state and a new episode starts. The cumulative reward or the return at time t can be calculated as

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... + R_T \tag{4.1}$$

On the other hand, tasks where the agent-environment interaction does not break naturally into identifiable episodes, but goes on continually without limit are called continuing tasks. This would be the natural way to formulate the return for a continual process control task, or an application to a robot with a long life span or for maintenance tasks like stabilising an inverted pendulum at the inverted upright position. Ideally, we would want such tasks to extend their learning episodes forever without failing, making $T \to \infty$. The continuing return is

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \tag{4.2}$$

where the parameter γ is such that $0 \leq \gamma < 1$ is called the discounting factor. The task of the RL agent is thus to take actions that maximizes the return.

The return can be more compactly written as

$$G_t = \sum_{k=0}^{T} \gamma^k R_{t+k+1} \tag{4.3}$$

including the possibility of T being ∞ or γ being 1 but not both.

In its endeavor to maximize the return, the agent keeps the following functions in record and updates them at each time step t and bases its actions on these functions

- Policy $\pi_t(a|s)$, which is the agent's control behaviour. It is a mapping from the set of states to the set of actions.
- Value function or Action-value function, denoted by $V(s)$ or $Q(s, a)$ respectively. This is the agent's internal evaluation of the goodness of each state or the goodness of taking action 'a' in state 's'.
- The agent can also optionally have a model, which is the agent's representation of the environment's behaviour.

Value functions, as already mentioned are a measure of goodness of the state s or the state-action pair (s,a). The notion of 'how good' is defined in terms of future rewards that can be expected, or the expected return. Since the rewards received by the agent depend

on the actions it will take, the value functions are defined with respect to the policy being followed by the agent and denoted by $v_\pi(s)$. $v_\pi(s)$ can be formally defined as

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] \tag{4.4}$$

where $\mathbb{E}_\pi$ denotes the expected value of a random variable given that the agent follows policy π. This can also be written as

$$v_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \tag{4.5}$$

The first step in a learning algorithm usually involves evaluating the value function for the current policy of the agent, also called the prediction/evaluation step. For convenience, we will denote $v_\pi(s)$ simply as $v(s)$. One way of obtaining an estimate of $v(s)$ in terms of the value of other states is to expand the right-hand side equation (4.5) as

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(s_{t+1})|S_t = s] \tag{4.6}$$

and take the sample return from this equation as an estimate for $v(s)$. In fact, the quantity $R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$, also called the temporal difference error δ_t can be seen as a measure of the error in the current value function $v_(S_t)$. This temporal difference error can be used to update the value function. In actor-critic learning algorithm it also plays a role in policy updation as discussed in the next section.

4.2 Actor-Critic RL

In Actor-Critic learning, the agent maintains a policy $\pi(a|s, \theta)$, also known as an actor as well as a value function $v(s, w)$, known as the critic. Here θ and w are the parameters of the policy and value function respectively. The critic is so called because it criticizes the actions of the actor through the temporal difference error. The diagram in figure (4.2) illustrates the steps in an actor-critic algorithm. At time t+1, given the state s_t, the action a_t, the reward r_{t+1} and the state s_{t+1}, the algorithm performs the following steps

1. **Temporal Difference error :**

$$\delta_t = r_{t+1} + \gamma v(s_{t+1}, w_t) - v(s_t, w_t) \tag{4.7}$$

2. **Critic updation :** This step evaluates the current policy by updating the critic or the value function by gradient descent with the loss function taking the form of the squared TD error and w as the parameter that is being updated.

$$w_{t+1} \leftarrow w_t + \beta \delta_t \nabla_w v(s_t, w_t) \tag{4.8}$$

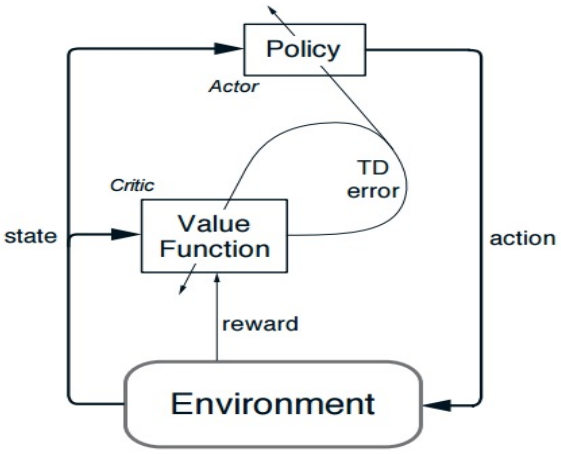

Figure 4.2: Actor-Critic architecture [11]

3. **Actor updation :** With the expected return as the performance measure, the actor's parameter θ is updated in the direction of performance improvement by gradient ascent.

$$\theta_{t+1} \leftarrow \theta_t + \alpha\delta_t \frac{\nabla_\theta \pi(a_t|s_t,\theta_t)}{\pi(s_t|a_t,\theta_t)} \tag{4.9}$$

4. **Control :** The RL agent takes the action sampled from $\pi(a|s_{t+1},\theta_{t+1})$.

γ is the discount factor, α is the actor learning rate and β is the critic learning rate.

Chapter 5

Experiments

5.1 Model-free Actor Critic

In this part, the Pendulum on a Cart has been used to demonstrate learning. The learning algorithm is plain actor-critic with a suitable rewarding mechanism and a valid policy parametrization. For the task of stabilizing the pendulum on a cart system at the inverted position, the learning is restricted to the linearized region around the inverted position as shown in figure (5.1) The experiment is started from inverted position of the pendulum

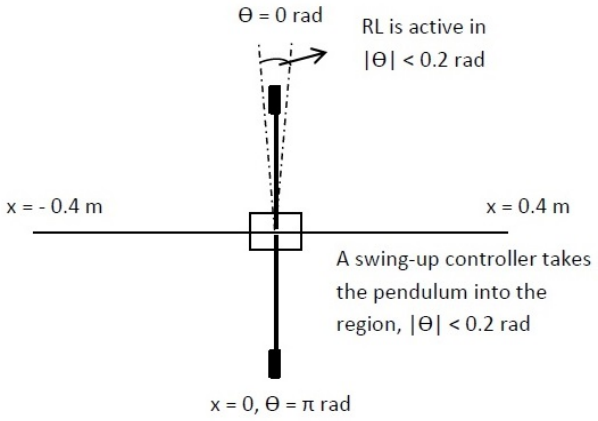

Figure 5.1: The state space for the Pendulum on a Cart

and left to learn to stabilize. Every time the pendulum falls out of the linearized region, a failure happens and a negative reward is given to the RL agent. The goal is to keep the pendulum inside the linearized region as long as possible.

Choosing the state space, the action space and the rewarding mechanism is the starting

point of solving an RL task. If x is the position of the cart on the track and θ is the pendulum angle,

1. State Space : The state is described as a tuple by $s =< x, \dot{x}, \theta, \dot{\theta} >$ such that $-0.4m < x < 0.4m$ and $-0.2rad < \theta < 0.2rad$.

2. Actions : +1 N force or -1 N force (+1 N or -1 N force is equivalent to +M volts or -M volts in the experimental Digital Pendulum control system shown in Chapter 2. M is the proportionality constant between the control voltage and the force.)

3. Reward : -1 , upon failure ie when $x \notin (-0.4m, 0.4m)$;$\theta \notin (-0.2rad, 0.2rad)$ and 0 otherwise.

Also every time a failure occurs, the system is taken back to $\theta = 0$ position and left to stabilize. This is easy to implement in simulation. However, in the real-time implementation, a separate swing-up controller has been used to bring the system back to the linearized region so that the RL agent can take up the control from there.

The state is discretized in the $< x, \dot{x}, \theta, \dot{\theta} >$ space. This means that the continuous state space is boxed into finite number of discrete states. This enables the value function and the policy parameter to be stored in finite-sized tables. The policy is the probability of taking action +1. This probability is parametrized by

$$\pi(a = +1/s, \theta_p) = \frac{1}{1 + \exp(-\max(-50, \min(\theta_p(s), 50)))}$$

as suggested in [10]. This parametrization for the policy is a morphed sigmoid that is a sigmoid in the (-50,50) interval and are continuous constant lines outside this interval. Also, a major part of the implementation code in this section is based on [10].

Simulation Results for Model-free Actor-Critic on Pendulum-on-a-Cart

The results are shown in Figure (5.2) and Figure(5.3). These are graphs of the state of the system showing $< x, \theta >$ in units of metre and radian respectively vs. the simulation time in seconds.

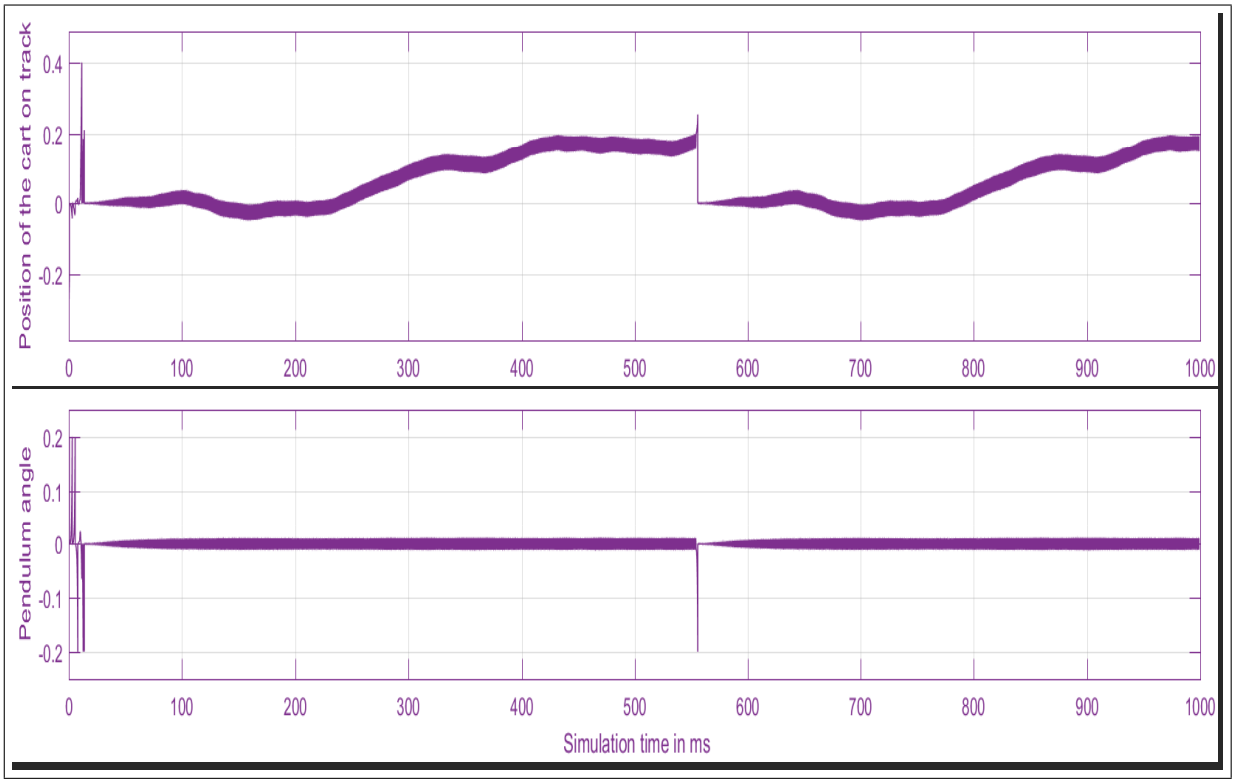

Figure 5.2: Simulation results of Model-free Actor-Critic on a Pendulum on a Cart system

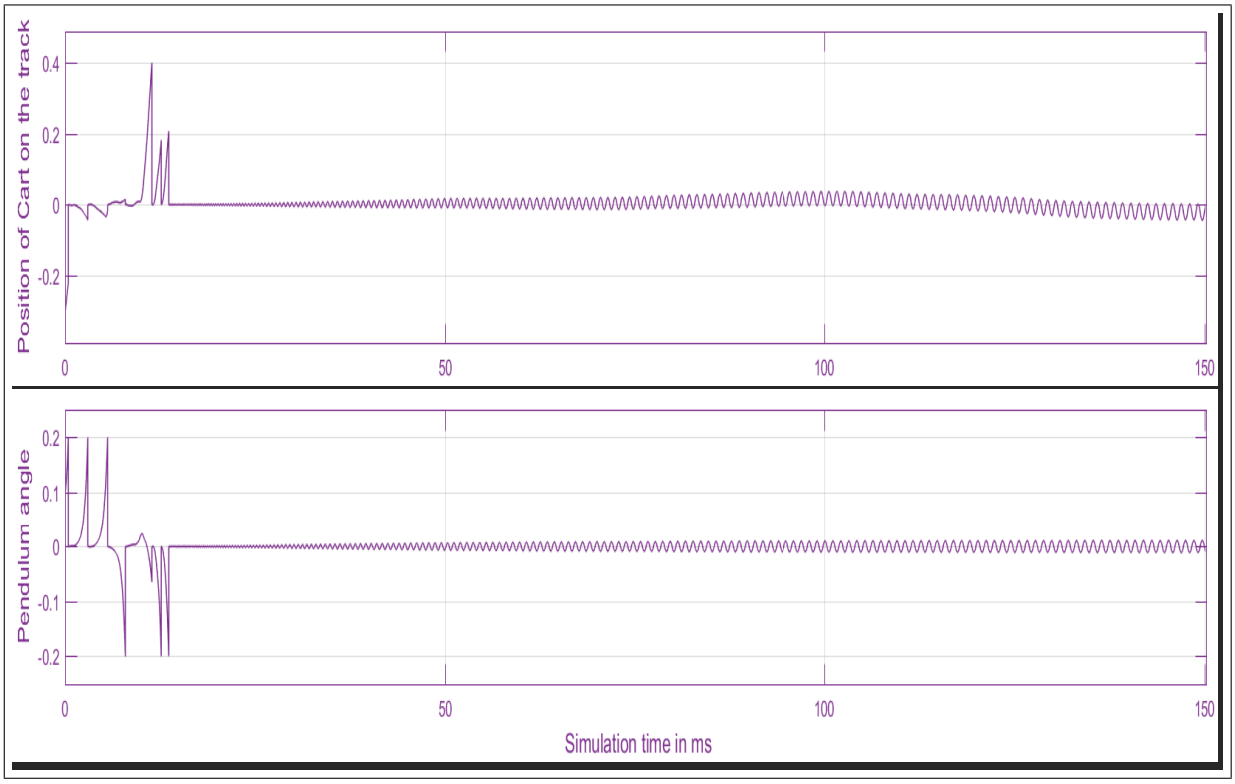

Figure 5.3: Zoomed version of the Figure 5.2 to show the initial failures clearly

5.2 Actor Critic for IDA-PBC

In this part, the Actor-Critic algorithm(1) learns the parameters of the IDA-PBC control law. These parameters are the closed loop mass matrix M_d, the closed loop potential energy function V_d and the damping coefficient matrix K_v.

Learning IDA-PBC parameters for Pendulum on a Cart

The notation for the position of the cart and the pendulum angle are q_1 and q_2 as in 3.1 of Chapter 3. Relabelling the simplified PDE constraints from equations (3.29) and (3.30) of Chapter 3,

$$\textbf{C1 :}\ 2\lambda_1\lambda_2 c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_1 a + \lambda_2 c\cos(q_2)) - 2\alpha_1 = 0$$

$$\textbf{C2 :}\ (\lambda_1\lambda_4 + \lambda_2\lambda_3)c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_1 c\cos(q_2) + \lambda_2 b) - \alpha_2 = 0$$

$$\textbf{C3 :}\ 2\lambda_3\lambda_4 c\sin(q_2) + \lambda_4\frac{d}{dq_2}(\lambda_3 c\cos(q_2) + \lambda_4 b) = 0$$

$$\textbf{C4 :}\ \Rightarrow -\text{mgl}\sin(q_2) = \lambda_3\frac{\partial V_d}{\partial q_1} + \lambda_4\frac{\partial V_d}{\partial q_2}$$

where $\lambda = M_d M^{-1}$ and a, b, c are system constants. The constraint equations labelled C1, C2, C3, C4 have to be satisfied along with the minima condition of V_d at the desired position q^* and M_d being symmetric and positive-definite.

The following points lead the way to solving this problem using RL

- Since M_d has to be symmetric, positive definite, it is taken to be of the form $\begin{bmatrix} a_1^2 + a_2^2 & a_2 a_3 \\ a_2 a_3 & a_3^2 \end{bmatrix}$ where a_2 and a_3 are non-zero numbers. Also $M^{-1} = \begin{bmatrix} b & -c\cos(q_2) \\ -c\cos(q_2) & a \end{bmatrix}$ is obtained from the expression for M in Chapter 3.
- Substituting for λ in terms of M_d and M^{-1} in both C3 and C4 gives new equations in terms of unknown variables a_2, a_3 and V_d.
- The equations C1 and C2 can be trivially satisfied using the free parameters α_1 and α_2. Substitute for $\lambda = M_d M^{-1}$ in C1 and C2 to find α_1 and α_2.
- The closed loop potential function V_d is necessarily a function of both q_1 and q_2 if the final position of stabilization is $q_1^* = 0$ and $q_2^* = 0$. In such a case, equation

C4 is still a difficult PDE to solve. However, if we relax our requirements to only stabilize the pendulum in the inverted position anywhere within the track, ie $q_2^* = 0$ and $q_1^* \in (-0.4, 0.4)$, V_d could be a simple function of q_2 alone, say

$$V_d = \frac{K}{\cos^2(q_2)} \tag{5.1}$$

This function approximation for V_d also satisfies the minima condition for V_d at $q_2 = 0$ (The domain of stabilization for IDA-PBC is the upper half plane containing $q_2 = 0$. $q_2 = \pi$ is also a minima for 5.1 but is outside its stabilization domain)

- C4 could now be simplified easily. And C4 and C3 are two ordinary differential equations with two unknowns a_2 and a_3 which could be deterministically solved at every time step in the learning process.

- a_1 is an unknown quantity and could take any real value. Another unknown variable in the control law is the damping coefficient K_v which should take a positive value. From 5.1, V_d also has to be learnt. The unknown variables V_d, K_v and a_1 are then learnt using actor-critic RL as given in algorithm 1.

The state space, action space and the rewarding mechanism are chosen as follows. If q_1 is the position of the cart on the track and q_2 is the pendulum angle,

1. State Space : The state is described as a tuple by $s =< q_1, \dot{q_1}, q_2, \dot{q_2} >$ such that $-0.4m < q_1 < 0.4m$ and $-1rad < q_2 < 1rad$.

2. Actions : Since the control law is IDA-PBC, the actions are continuous, real valued capped by control input saturation beyond 2.5V or below -2.5V.

3. Reward :

$$\begin{aligned} r &= 2(\cos(q_2) - 1) - 100\dot{q_1}^2 - 100\dot{q_2}^2 - 50, \text{ upon failure} \\ &= 2(\cos(q_2) - 1) - 100\dot{q_1}^2 - 100\dot{q_2}^2, \text{ else} \end{aligned}$$

Failure occurs when either $q_1 \notin (-0.4, 0.4)$ or $q_2 \notin (-1, 1)$radians. Also every time a failure occurs, the system is taken back to $q_2 = 0$ position and left to stabilize. It is to be remembered IDA-PBC is a nonlinear control algorithm and its domain of stabilization for the Pendulum on a Cart is the half interval $(-\frac{\pi}{2}, \frac{\pi}{2})$ around the inverted position.

Input: PH model of the system in open loop, λ, γ, α_a for each actor, α_c for the critic. Initialise critic weights, critic eligibility traces and actor weights to 0.

while *1* **do**

Draw $\Delta u_k \sim N(0,1)$. Calculate action $u_k = \zeta(\pi(x_k, (\theta_{a1})_k, (\theta_{vd})_k, (\theta_{kv})_k) + \Delta u_k)$.

$\Delta \bar{u}_k = u_k - \pi(x_k, (\theta_{a1})_k, (\theta_{vd})_k, (\theta_{kv})_k)$.

Observe next state x_{k+1} and calculate reward $r_{k+1} = \rho(x_{k+1}, u_k)$

Critic: Temporal Difference $\delta_{k+1} = r_{k+1} + \gamma V(x_{k+1}, \theta_k) - V(x_k, \theta_k)$

Eligibility trace: $e_{k+1} = \gamma\lambda e_k + \nabla_\theta V(x_k, \theta_k)$

Critic Update: $(\theta_c)_{k+1} = (\theta_c)_k + \alpha_c \delta_{k+1} e_{k+1}$

Actor Updates:

$$M_d(x, \theta_{a1}) : (\theta_{a1})_{k+1} = (\theta_{a1})_k + \alpha_{a1}\delta_{k+1}\Delta\bar{u}_k \nabla_{\theta_{a1}} \zeta(\pi(x_k, (\theta_{a1})_k, (\theta_{vd})_k, (\theta_{kv})_k) + \Delta u_k)$$

$$V_d(x, \theta_{vd}) : (\theta_{vd})_{k+1} = (\theta_{vd})_k + \alpha_{vd}\delta_{k+1}\Delta\bar{u}_k \nabla_{\theta_{vd}} \zeta(\pi(x_k, (\theta_{a1})_k, (\theta_{vd})_k), (\theta_{kv})_k) + \Delta u_k)$$

$$K_v(x, \theta_{kv}) : (\theta_{kv})_{k+1} = (\theta_{kv})_k + \alpha_{kv}\delta_{k+1}\Delta\bar{u}_k \nabla_{\theta_{kv}} \zeta(\pi(x_k, (\theta_{a1})_k, (\theta_{vd})_k), (\theta_{kv})_k) + \Delta u_k)$$

end

Algorithm 1: Energy-based Actor-Critic for learning IDA-PBC parameters[8]. The ζ around the control action u_k incorporates control input saturation

Simulation results for IDA-PBC on Pendulum-on-a-Cart system

Figure(5.4) shows how the system learns to balance the pendulum at the inverted position by learning the parameters of IDA-PBC using the algorithm(1). Since we assumed V_d to be a function of the pendulum angle, q_2 alone, the system initially learns to balance the pendulum as if the track length is infinite. However, once it learns to stabilize at $q_2 = 0$, it is now exposed to the reality of limited track length. The IDA-PBC control has been learnt the moment q_2 stabilizes at 0 at around t=550s in the graph. After this, an additional control term dependent on the track position q_1 is learnt. This control term tries to avoid the track edges. This learning is also evident from the result since the cart slowly learns to take U-turns to avoid the edges.

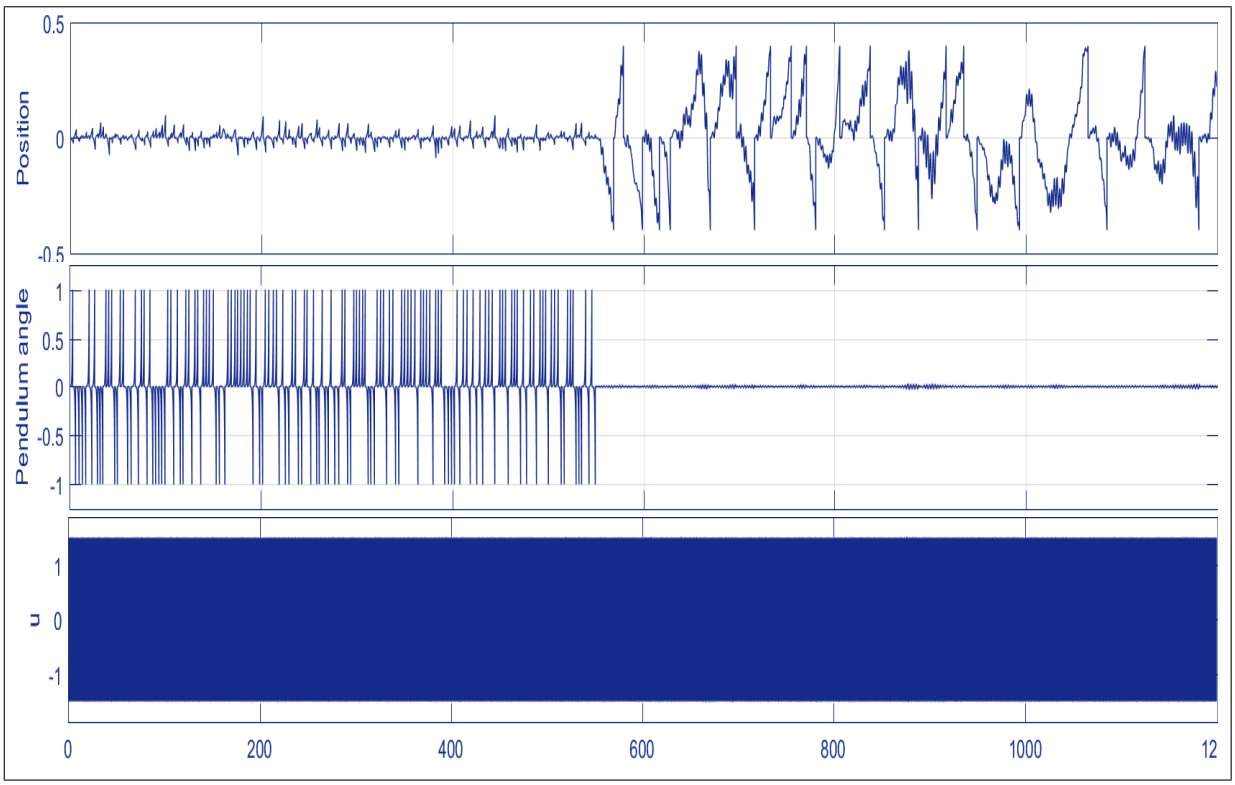

Figure 5.4: Simulation results of learning IDA-PBC parameters for Pendulum-on-a-Cart system: 1-Position of the Cart on the track, 2-Pendulum angle and 3-control applied

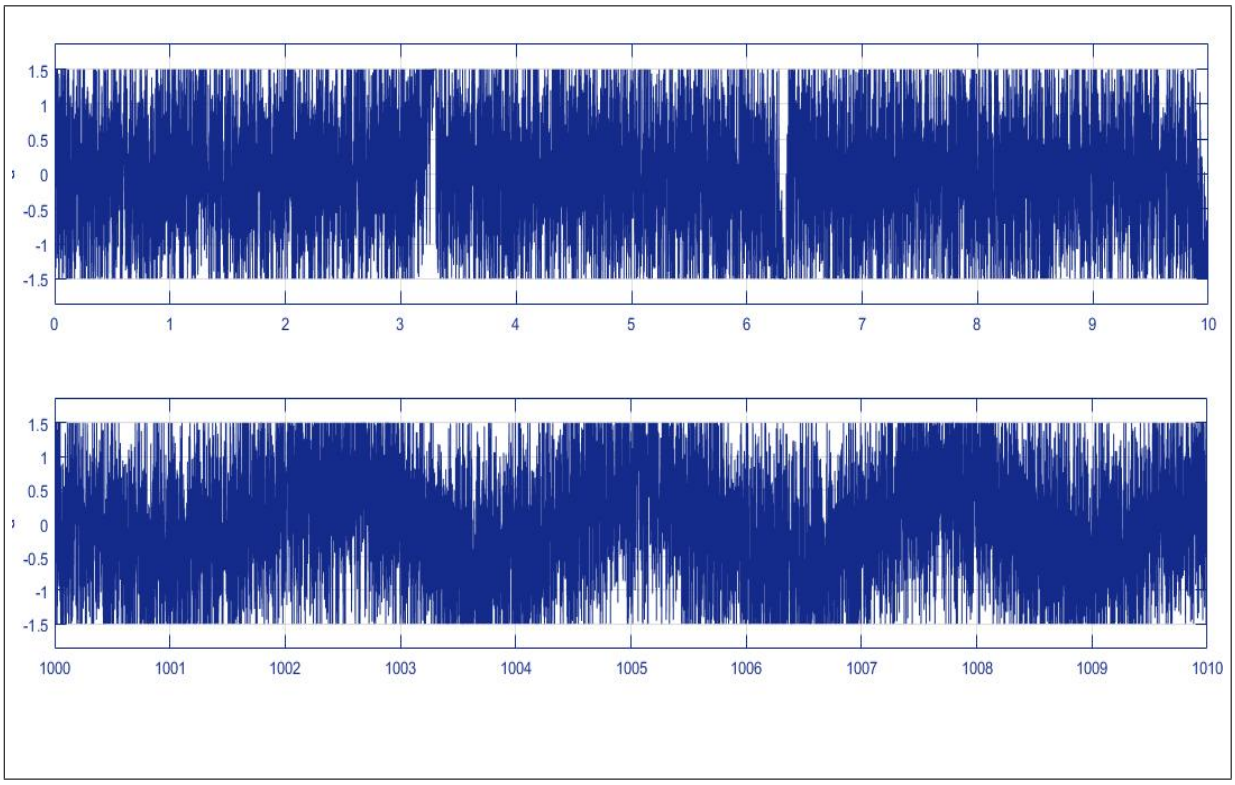

Figure 5.5: The graph of u in figure(5.4) magnified at the interval 1. t = 0 - 10s, 2. t = 1000 to 1010s

Learning IDA-PBC parameters for Twin-Rotor MIMO system

The Twin-Rotor MIMO System (TRMS) is a fully-actuated system. Hence, the closed loop Mass matrix is the same as the open loop mass matrix, $Md = M$. Also, the damping coefficients b_v and b_h appearing in the damping matrix R_d do not affect the control output, u in equation (3.35). Only the potential energy function V_d needs to be learnt. As discussed in equation(3.34) of Chapter 3, V_d is of the form

$$V_d = \frac{1}{2}\gamma_v \widetilde{q_v}^2 + \frac{1}{2}\gamma_h \widetilde{q_h}^2 \tag{5.2}$$

With this function approximation for V_d, the coefficients γ_v and γ_h are learnt using the algorithm(1).

Simulation results for IDA-PBC on Twin-Rotor MIMO System

The desired final position of the system is $< q_v, q_h >=< 0, 0 >$

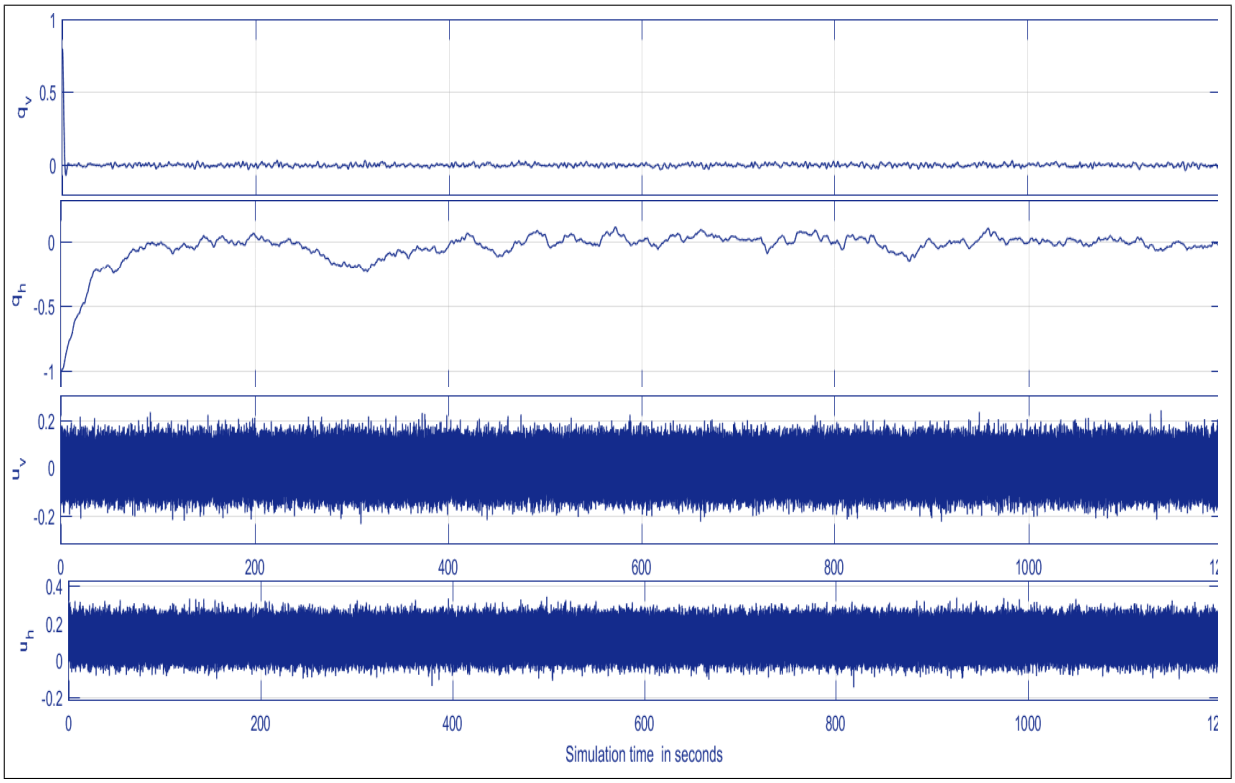

Figure 5.6: Simulation results of learning IDA-PBC parameters for Twin-Rotor MIMO System: 1 - Pitch angle(q_v), 2-Yaw angle(q_h), 3-Pitch control(u_v) and 4-Yaw control(u_h)

5.3 Actor-Critic for State-modulated damping with IDA-PBC

For this part, the Twin-Rotor MIMO System is the system of interest. The control action of IDA-PBC typically gives multiple overshoots in the system response before reaching the desired position $< q_v^*, q_h^* >$. This can be taken care of by adding a state-modulated damping term to the IDA-PBC control law. For this, the elements $f(\widetilde{q_v})$ and $f(\widetilde{q_h})$ of the memristor matrix $\widetilde{M}$ needs to be learnt as in equation (3.40). Once, the IDA-PBC parameters are learnt, they are kept fixed and the matrix $\widetilde{M}$ is learnt using the same algorithm(1).

Simulation results for State-modulated damping with IDA-PBC on Twin-Rotor MIMO System

Figure(5.7 shows the results of having learnt a state-modulated damping term over and above the IDA-PBC control. Learning is explicitly divided into episodes of length 50 seconds. At the beginning of each episode, the system starts at a fixed starting position $< q_v, q_h > = < 0.8, -1.0 >$. In the initial few episodes, the system overshoots once or many times before reaching $< q_v^*, q_h^* > = < 0, 0 >$. As the damping term is learnt over the episodes, the system smoothly transitions into the final desired state without any overshoots. This can be seen in Figure(5.7).

The learnt functions f_v and f_h, which are the elements of the memristor matrix$\widetilde{M}$ are shown in Figure(5.8). The function approximations used for f_v and f_h are fourier series of 3rd order with f_v being dependent only on q_v and f_h being dependent only on q_h.

$$\begin{aligned} f_v(\widetilde{q_v}) &= v_0 + v_1\cos(\widetilde{q_v}) + v_2\cos(2\widetilde{q_v}) + v_3\cos(3\widetilde{q_v}) \\ f_h(\widetilde{q_h}) &= h_0 + h_1\cos(\widetilde{q_h}) + h_2\cos(2\widetilde{q_h}) + h_3\cos(3\widetilde{q_h}) \end{aligned} \tag{5.3}$$

$\widetilde{q_v} = q_v - q_v^*$, $\widetilde{q_h} = q_h - q_h^*$ and v_i and h_i are actor coefficients that are updated while learning. The results in figure(5.8) are in tune with the intuition that the damping coefficient is higher near the desired state and lower in the states farther away.

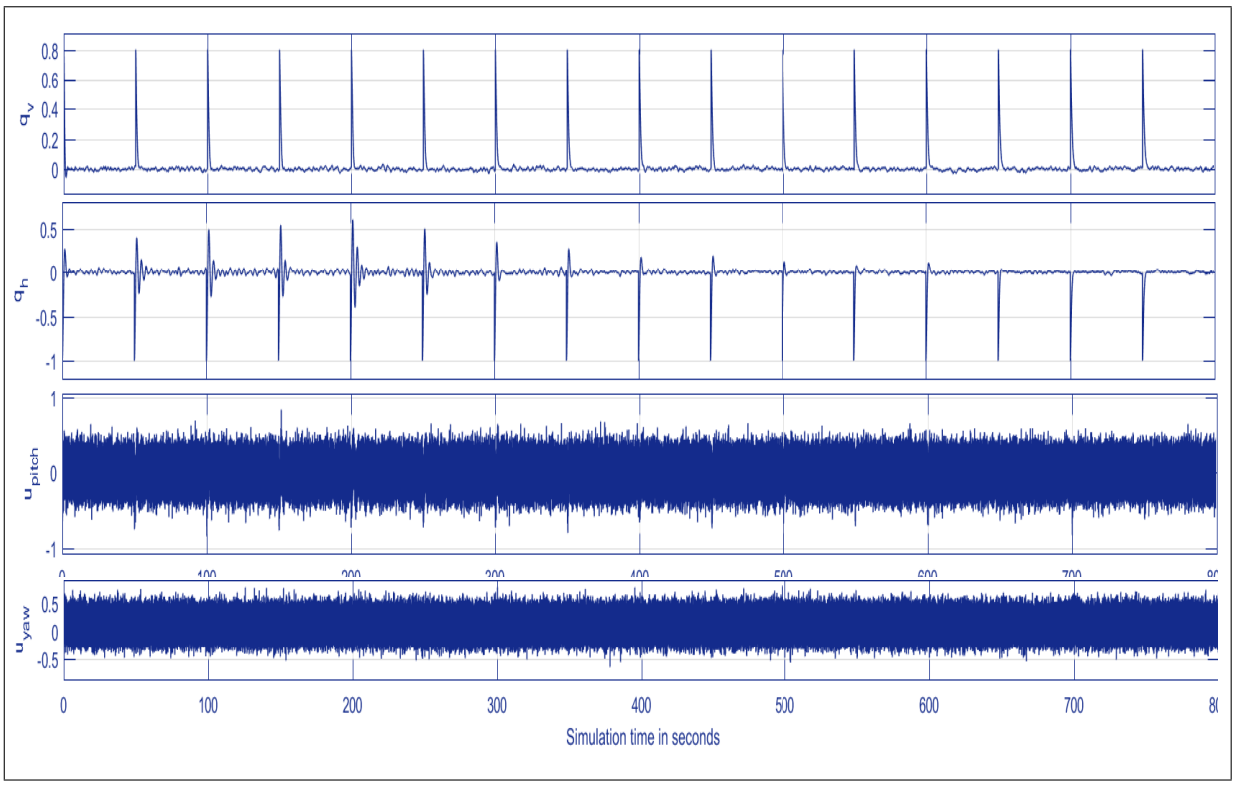

Figure 5.7: Simulation results of learning State-modulated damping term for Twin-Rotor MIMO System: 1 - Pitch angle, 2 - Yaw angle, 3 - Pitch control, 4 - Yaw control

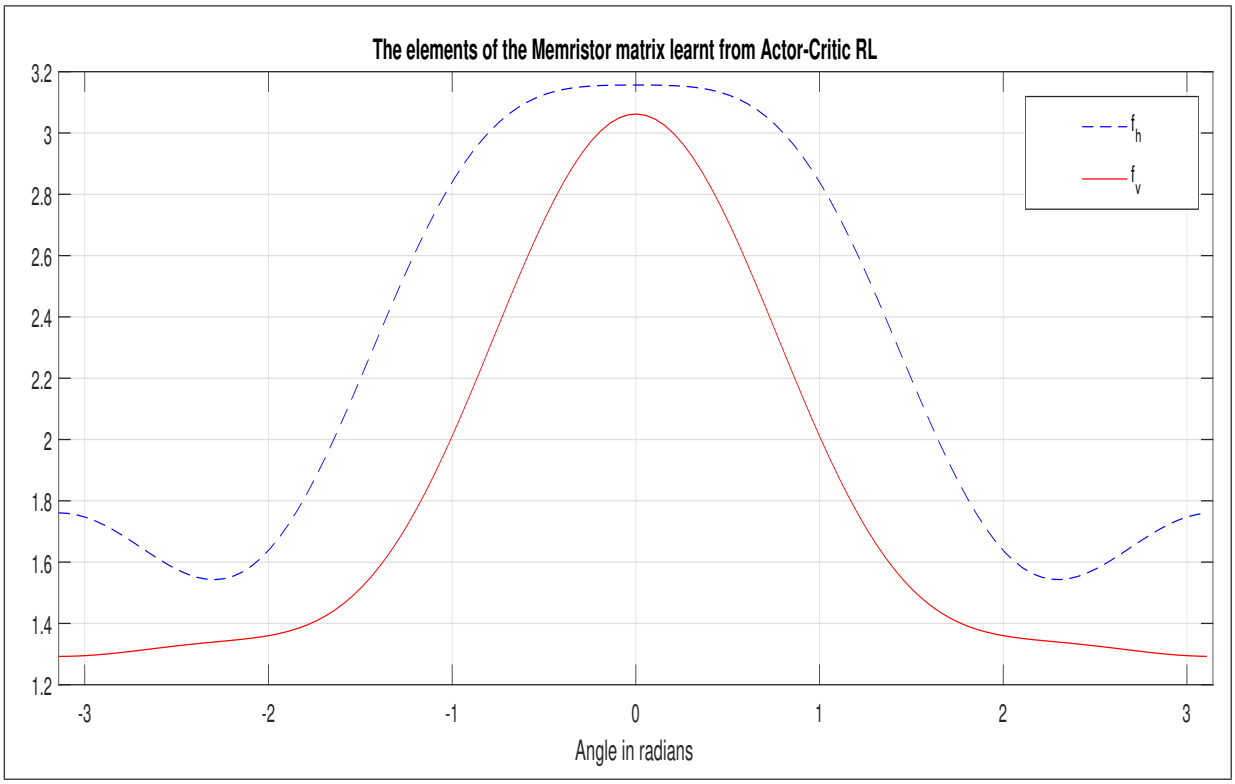

Figure 5.8: The learnt functions for the elements of $\widetilde{M}$ - f_v and f_h.

Chapter 6

Conclusion

This thesis has explored the application of reinforcement learning for stabilization tasks which are usually handled by hard-core control theory. Firstly, a direct RL controller was learnt to stabilize the Pendulum-on-a-Cart system in the inverted position. Following this, an IDA-PBC controller was designed by learning the parameters of the IDA-PBC control law using RL. This learnt IDA-PBC controller was implemented in simulation experiments on the fully-actuated Twin-Rotor and the more difficult under-actuated Pendulum-on-a-Cart system. Since IDA-PBC on under-actuated systems requires solving partial differential equations for the closed loop mass matrix subject to it being symmetric and positive definite, the mass matrix was by default given a symmetric, positive-definite form. The parameters of this form of the matrix are then learnt. A more challenging and generalized way of handling these constraints on the mass matrix would be to design an actor-critic update that respects these constraints during learning. Finally, using the concept of state-modulated damping control to improve system response in terms of overshoot, settling time etc., an additional damping control term was learnt for the closed-loop IDA-PBC system.

Bibliography

[1] Jose Angel Acosta Romeo Ortega Alessandro Astolfi and Arun D. Mahindrakar. "Interconnection and Damping Assignment Passivity-Based Control of Mechanical Systems with Underactuation Degree One". In: *IEEE transactions on Automatic Control* (2005).

[2] Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. "Neuronlike adaptive elements that can solve difficult learning control problems". In: *IEEE Transactions on Systems, Man, and Cybernetics* 13.5 (Sept. 1983), pp. 835–846. URL: http://www.cs.ualberta.ca/~sutton/papers/barto-sutton-anderson-83.pdf.

[3] "Digital Pendulum Control Experiments 33-936S". In: Feedback Instruments.

[4] A. Doria-Cerezo, L. Van der Heijden, and J.M.A. Scherpen. "Memristive port-Hamiltonian control : Path-dependent damping injection in control of mechanical systems". In: *European Journal of Control* (2013).

[5] Anup K. Ekbote. "Sliding Mode Control of a Twin Rotor Multiple Input Multiple Output System". In: *M.Tech thesis IIT Madras* (2010).

[6] George Konidaris and Sarah Osentoski. "Value Function Approximation in Reinforcement Learning using the Fourier Basis". In: (2008).

[7] Theodore J. Perkins and Andrew G. Barto. "Lyapunov design for Safe Reinforcement Learning". In: *Journal of Machine Learning* (2002).

[8] Olivier Sprangers, Gabriel A. D. Lopes, and Robert Babuska. "Reinforcement Learning for Port-Hamiltonian Systems". In: *arXiv:1212.5524v2* (22 August 2013).

[9] "Stabilization of a class of Underactuated mechanical systems via Total Energy Shaping". In: *Proceedings of the 40th IEEE Conference of Decision and Control* (2001).

[10] Richard S. Sutton. *Cart-Pole balancing using Reinforcement Learning*. URL: https://webdocs.cs.ualberta.ca/~sutton/book/code/pole.c.

[11] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. 2016, in progress.

[12] Richard S. Sutton, Andrew G. Barto, and Ronald J. Williams. "Reinforcement Learning is Direct Adaptive Optimal Control". In: *IEEE Control Systems* (April1992).

[13] "Twin-Rotor MIMO System Control Experiments 33-949S". In: Feedback Instruments.