REGULATION AND TRACKING USING ACTOR-CRITIC METHODS

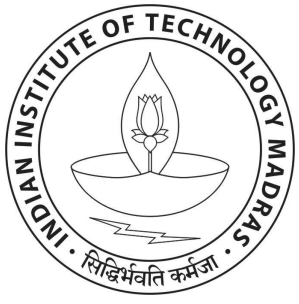
A Project Report

submitted by

ANJALI RAMESH

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

BACHELOR OF TECHNOLOGY
AND
MASTER OF TECHNOLOGY
(DUAL DEGREE)



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2016

THESIS CERTIFICATE

This is to certify that the thesis titled **REGULATION AND TRACKING USING ACTOR-CRITIC METHODS**, submitted by **Anjali Ramesh**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide 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
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai
Date: May, 2016

ACKNOWLEDGEMENTS

All that is gold does not glitter,
Not all those who wander are lost;
The old that is strong does not wither,
Deep roots are not reached by the frost.

I am eternally grateful to my parents for not restricting my freedom to wander and for their unshakable support during all my adventures.

This project would not be possible without the help and support of several people. Firstly, I would like to thank my guide Dr. Ramkrishna Pasumarthy for providing me with the opportunity to work on this project. I am highly grateful for his guidance and support during the entire duration of the project. I am also grateful to Dr. Arun Mahindrakar for his inputs and guidance.

I would also like to thank the students and scholars of Dynamic and Control Lab of Electrical Engineering Department for their support. Special thanks to Mr. Krishna Chaitanya Kosaraju for his guidance and involvement.

Last but not the least, I would like to thank all my friends for their support and friendship which made my stay in the institute memorable.

ABSTRACT

In this thesis, we aim to solve regulation and tracking problems for mechanical systems with Actor-Critic methods in Reinforcement Learning. These algorithms are nested in Interconnection and Damping Assignment Passivity Based Control for solving regulation problems and in feedforward proportional derivative control for regulation problems without having to solve the complex partial differential equations generated by the above techniques. The algorithm parametrizes these control techniques to learn the control laws using update policies. The simulations results for the regulation problem with ball on a beam and 2D spidercrane, and tracking problem with double gimbal are presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

PDE	Partial Differential Equation
ODE	Ordinary Differential Equation
RL	Reinforcement Learning
TD	Temporal Difference
IDA-PBC	Interconnection and Damping Assignment Passivity Based Control
PBC	Passivity Based Control
PH	Port Hamiltonian
MDP	Markov Decision Process

NOTATION

$\mathbb{R}^n$	n-dimensional Real numbers space
$\mathbb{R}_+$	positive Real numbers
$\mathbb{S}^1$	Circle
$\mathbb{S}^2$	Sphere
$\mathbb{R}^1 \times \mathbb{S}^1$	Cylinder
$\mathbb{S}^1 \times \mathbb{S}^1, \mathbb{T}^2$	Torus
I_n	Identity matrix of dimension $n \times n$
M^{-1}	Inverse of the matrix M
$M^\top$	Transpose of the matrix M
$\nabla_p H$	Gradient of H w.r.t p
∇	Connection
$\mathbb{G}$	Riemannian Metric
$\overset{\mathbb{G}}{\nabla}$	Levi-Civita connection associated with the Riemannian metric $\mathbb{G}$
$\overset{\mathbb{G}}{\nabla}_T$	Covariant derivative of transport map
$(M, \mathbb{G})$	A Riemannian manifold $\mathbb{M}$ with the metric $\mathbb{G}$
(TM)	Tangent manifold of M
$(T_x M)$	Tangent manifold of M at a point $x \in M$
$(T^* M)$	Dual of the Tangent manifold TM
(T^*)	Dual of the matrix T
$\|\|x\|\|_P$	Norm of x w.r.t matrix P
$\mathbb{G}(X, Y)$	Inner product of X, Y w.r.t G
$d_1\psi$	Partial differential of ψ w.r.t first argument
$d_2\psi$	Partial differential of ψ w.r.t second argument

CHAPTER 1

INTRODUCTION

1.1 Motivation

Most energy shaping control methodologies require solving Partial Differential Equations(PDEs) to arrive at a stabilizing control policy (Ortega *et al.*, 2001). These PDEs usually require deep subject knowledge for solving them. The control policy generated don't account for non-linearities like control saturation. Reinforcement Learning(RL) is capable of overcoming these issues. Here we will discuss how Actor-Critic methods can be used for solving regulation and tracking problems. We use the example of ball on a beam system and 2D spidercrane system to illustrate how regulation problems, and double gimbal to illustrate how tracking problems can be solved using Actor-Critic Methods.

In the regulation problems, Interconnection and Damping Assignment Passivity Based Control(IDA-PBC) is used to achieve stability by rendering the system passive with respect to a desired energy function (Ortega *et al.*, 2001). It is a Passivity Based Control(PBC) (Van der Schaft, 2012) technique used to stabilize under-actuated systems in Port-Hamiltonian(PH) form by shaping closed loop potential and kinetic energy. This formulation reduces to solving complex PDEs because of the structural properties of the PH systems.

The trajectory tracking problem is solved using feedforward proportional derivative controller (Lewis and Bullo, 2005). To achieve the control objective, we need to find two functions called tracking error function and transport map that are used to measure error in position and velocity respectively. For stability analysis these functions have to satisfy a set of PDEs called compatibility condition.

While both controls achieve stability by solving PDEs, which have the information of the system, many control techniques achieve near optimal performance (with respect to the reward function) with little or no knowledge of the system. RL is one

such technique. It can solve optimal control problems without the need of an explicit model. It is a stochastic, semi-supervised, model free learning technique which is used to solve control problems by maximizing the reward function, which is a function of states of the system and possibly the control action (Sutton and Barto, 1998). In large state spaces the learning is slow and monotonous. However, providing the agent with some knowledge of the system can increase the learning rate significantly. As the spaces are continuous in the real-time problems for regulation and tracking, we focus on Actor-Critic Methods.

1.2 Literature Survey

1.2.1 Regulation

In this project, we solve regulation problems using IDA-PBC for underactuated systems in Port-Hamiltonian form. Due to the structural properties of PH systems, IDA-PBC fomulation results in PDEs, one for kinetic energy and the other for potential energy. To solve PDEs, we try to rewrite them in a form that can be solved using standard PDE or ODE techniques, such as the method of the characteristics (Arnol'd, 2012). All the results in simplification of PDEs are restricted to mechanical systems. The mechanical systems with the degree of underactuation as one have received special attention. In such systems the kinetic energy PDE can be written as an ODE under certain assumptions (Gómez-Estern *et al.*, 2001). It has also been shown in Acosta *et al.* (2005) that the resulting ODE can be solved explicitly under additional assumptions. The explicit solutions can be found by parameterising the interconnection structure. This was extended in Gómez-Estern and Van der Schaft (2004) to include natural damping in the open-loop system. We also look at the possibility of change of coordinates to simplify the matching conditions. In Fujimoto and Sugie (2001), it is shown that the class of PH systems is invariant under change of coordinates of the state space. This allows us to consider a coordinate-free description of port-Hamiltonian systems. Simplification of the matching equation via a change of coordinates for mechanical systems with degree one of underactuation has been studied in Viola *et al.* (2007) and Viola (2008). The study shows that the forcing term in the kinetic energy PDE can be eliminated by an appropriate choice of coordinates, which generates a homogeneous linear PDE. Hence, a

change of coordinates could also be beneficial if we consider port-Hamiltonian systems other than mechanical control systems.

1.2.2 Tracking

In this project, we focus on trajectory tracking for fully actuated systems like the Double Gimbal system. The geometry in mechanical systems can be used to an advantage to give stronger control algorithms when compared to the generic non-linear control algorithms. The control objective of this project is to successfully track a trajectory without having to solve the PDEs which arise in the formulation.

Tracking of robot manipulators have received a lot of attention in the literature. Examples are Takegaki and Arimoto (1981), Wen and Bayard (1988) and Slotine and Li (1989), where non linear analysis is used for asymptotic and exponential tracking. These results have now become standard and are found in books on control like (Nijmeijer and Van der Schaft, 2013) and robotics (Murray *et al.*, 1994). Since then, similar techniques have been applied to the attitude control problem for satellites (Wen and Kreutz-Delgado, 1991), and likewise to the attitude and position control for underwater vehicles [Fossen (1994), Section 4.5.4].

1.3 Organisation of the Thesis

The rest of the discussion is organised as follows. Chapter 2 discusses RL and the standard Actor-Critic algorithm which is used to solve regulation and tracking problems. In Chapter 3, IDA-PBC is introduced as one of the ways to regulate underactuated systems. We use Actor-Critic methods to learn to solve the regulation problem by parametrizing IDA-PBC. In Chapter 4 and 5, the algorithm was simulated and the results are shown for ball on a beam system and 2D SpiderCrane respectively. In Section 6, a feedforward proportional-derivative controller is learned to solve the tracking problem using Actor-Critic Algorithm. Simulation results are shown for this methodology using double gimbal mechanism as an example in Chapter 7. Chapter 8 concludes the paper.

CHAPTER 2

REINFORCEMENT LEARNING

Reinforcement Learning is an area of machine learning with numerous applications in various fields, such as game theory, operations research, control theory etc.

The full reinforcement learning (RL) problem is a way of modeling sequential decision making problems. These problems consist of two entities: an environment and an agent. At various time-steps during what is called an episode, the environment is found to be at different states as the agent performs one action after the other in each time-step. The execution of an action results in two kinds of feedback from the environment. First, the agent receives a reward. Second, the environment makes a transition to another state. Both the above effects are dependent only on the state the agent was in when it took the action, and the action itself - but not any event that happened further back in time. The above system of the agent, the environment, states, actions, transitions and rewards is encapsulated as a Markov Decision Process. As an example, a robot navigating through a maze is a sequential decision making problem: at various instances, the robot makes a move in a specific direction (the action), finds itself in various positions in the maze (the state), and receives a reward (or a punishment) from the maze if it reaches its goal (or hits the wall). The 'problem' here is that the agent is required to maximize some form of cumulative reward called the return. When an agent is thus introduced to an unknown environment it has to learn through experience what is the 'best' action to be taken at every state. The quality of the action can be understood to correspond to the amount of return that is expected by taking that action. We will call this state-action mapping that is to be learned as the optimal policy. This process of learning will involve exploration - that enables to the agent to understand the environment better - and also exploitation that ensures that the agent makes best use of what it has learned. Thus, when an agent explores and receives a high reward, the reward is a means of reinforcement which encourages the agent to believe that the steps it took recently are good and can be exploited later. Algorithms that solve this problem are broadly of two kinds. One class of algorithms maintain a model of the world by approximately estimating the noisy feedback of the environment - these are

called model-based algorithms. The other class of algorithms are model-free in that they do not maintain any such model but only work with some sort of estimate of the 'value' or the goodness of the actions that can be taken at every 2 state.

The main advantage of RL is that it doesn't need any system information (Sutton and Barto, 1998). However, system information can help speed up the learning process. This is demonstrated through an example in the next section.

2.1 A Learning Control Problem: Pole Balancing

2.1.1 Cart-Pole System

Fig 2.1 shows the schematic of a Cart-Pole system. The cart can move only in a one dimensional track and the pole moves in the vertical plane of the track. The controller can apply an impulsive "left" or "right" force F of fixed magnitude to the cart at discrete time intervals (Barto *et al.*, 1983).

The cart pole model has four state variables:

- x position of the cart on the track
- θ angle of the pole with the vertical
- $\dot{x}$ cart velocity, and
- $\dot{\theta}$ rate of change of the angle

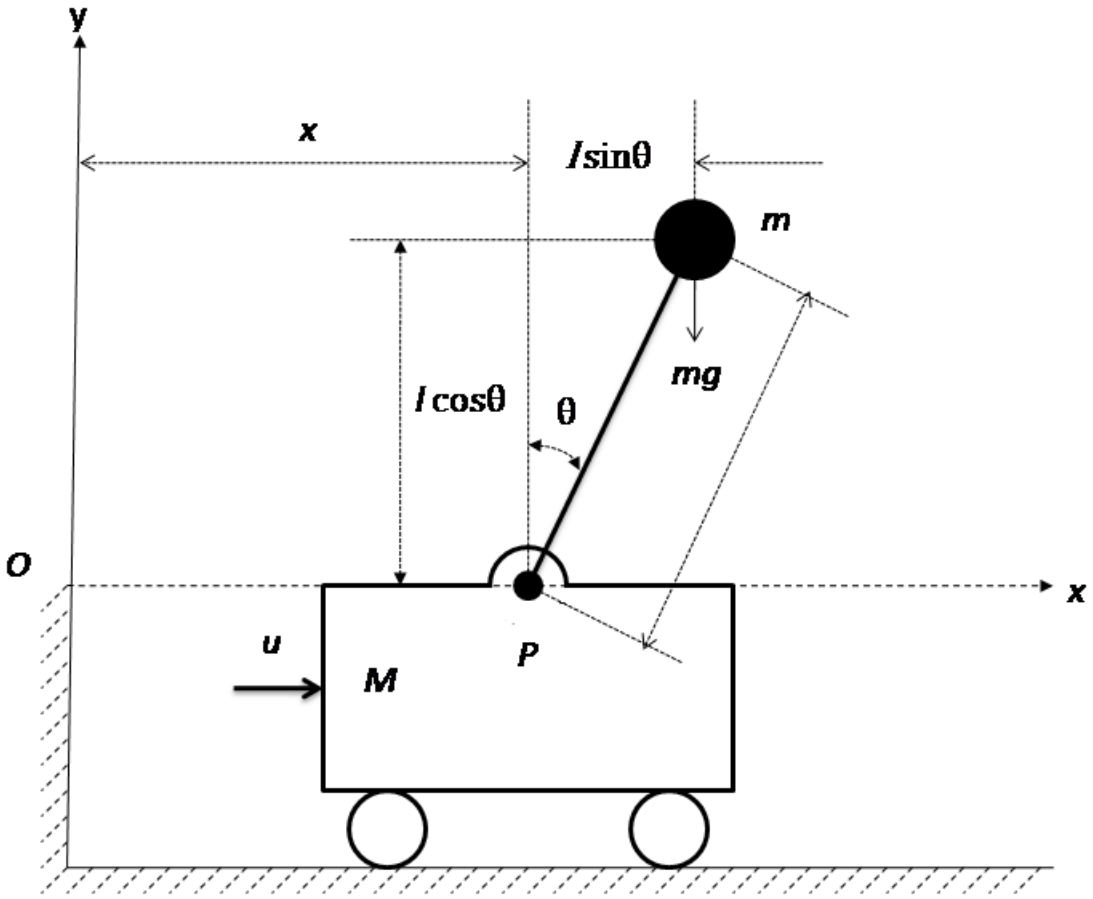


Figure 2.1: Cart-pole system

The parameters specify the pole length and mass, cart mass, coefficients of friction between the cart and the track and at the hinge between the pole and the cart, the impulsive control force magnitude, the force due to gravity, and the simulation time step.

2.1.2 Reinforcement Learning- SARSA Algorithm

We assume that the equations of motion of the cart-pole system are not known and that there is no pre-existing controller that can be imitated. At each step, the controller receives a vector giving the cart-pole system's state at that instant. If the pole falls or the cart hits the track boundary, the controller receives a failure signal, the cart-pole system (but not the controller's memory) is reset to its initial state, and another learning trial begins. The controller must attempt to generate controlling forces in order to avoid the failure signal for as long as possible. No evaluative feedback other than the failure signal is available. The SARSA Algorithm used for this problem is stated in Al-

gorithm 1 (Sutton and Barto, 1998). The SARSA algorithm is an On-Policy algorithm for Temporal difference(TD)-Learning.

Algorithm 1 SARSA Algorithm For Cart-Pole System

procedure SARSA ALGORITHM
    Initialise $Q(s, a)$ arbitrarily
    Repeat (for each episode)
    Initialise s
    Choose a from s using policy derived from Q(e.g., ϵ - greedy)
    Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q(e.g., ϵ- greedy)
    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') - Q(s, a)]$
    $s \leftarrow s'; a \leftarrow a'$
    until s is terminal
end procedure

where s is the current state, a is the action chosen which leads the system to state s' and returns a reward r which criticises the state-action pair. a' is the action taken in state s', $\alpha \in (0, 1)$ is the learning rate and $\gamma \in (0, 1)$ is the discount factor.

2.1.3 Simulation Results

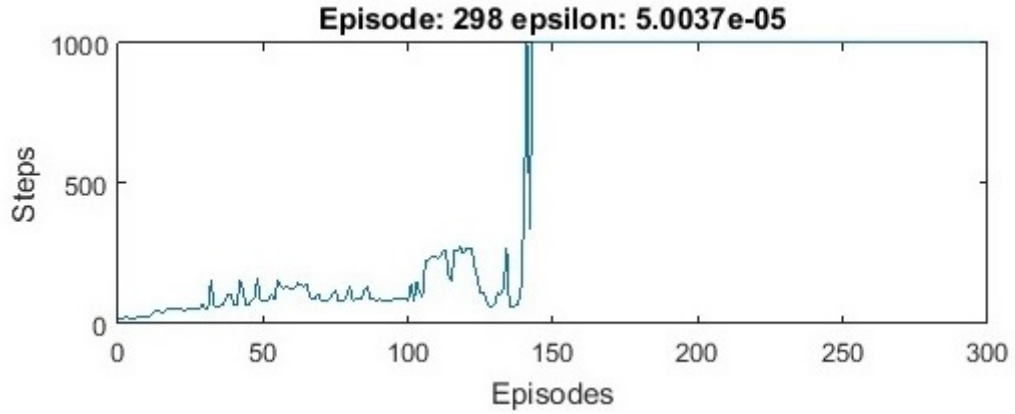


Figure 2.2: Simulation Results for Cart-Pole System

As seen in Figure 2.2, the system learns to balance the pole for 1000 steps after approximately 150 episodes of learning. The learning is fast as the system has discrete states. However, as all real systems have continuous spaces, SARSA algorithm cannot be implemented.

2.1.4 Reinforcement Learning in continuous spaces

Reinforcement Learning algorithms such as Q-Learning and SARSA operate only in discrete spaces as they are based on Bellman back-ups and discrete-space version of Bellman's equation (Sutton and Barto, 1998). However, most applications of reinforcement learning are in continuous spaces defined by continuous variables such as position, velocity etc. Usually the problem is tackled by discretizing the state space. However, this quickly leads to combinatorial explosion, also famously called the "curse of dimensionality".

Handling big spaces have been identified as the one of the most important research directions for reinforcement learning. The greatest impact of the "curse of dimensionality" is in robotic applications of reinforcement learning to high-dimensional perceptual spaces. Actor critic reinforcement learning methods are online approximations to policy iteration in which the value function parameters are estimated using temporal difference learning and the policy parameters are updated by stochastic gradient descent. Methods based on policy gradients in this way are of special interest because of their compatibility with function approximation methods, which are needed to handle large or infinite state spaces (Sutton and Barto, 1998). Hence we utilise Actor-Critic method for real-time problems such as regulation and tracking. We explain the Actor-Critic method in the next section.

2.2 Actor-Critic Method

In RL, the system is modelled as a Markov Decision Process (MDP) (Sutton and Barto, 1998) represented by the tuple $MDP(X, U, f, \rho, \gamma)$, where X is the state space, U is the control space, $f : X \times U \rightarrow X$ is the control policy and $\rho : X \times U \rightarrow \mathbb{R}$ is the reward function and the discount factor $\gamma \in (0, 1]$. The control policy f acts on the state ($x \in X$) at time instant t to lead to a state ($x' \in X$) at $t' > t$.

The reward function ρ acts on a state and possibly the control action $u \in U$ to give a reward r which criticizes the action.

The goal of Actor-Critic method is to optimize the control policy with respect to a cost function, which is a sum of the discounted instantaneous rewards from current

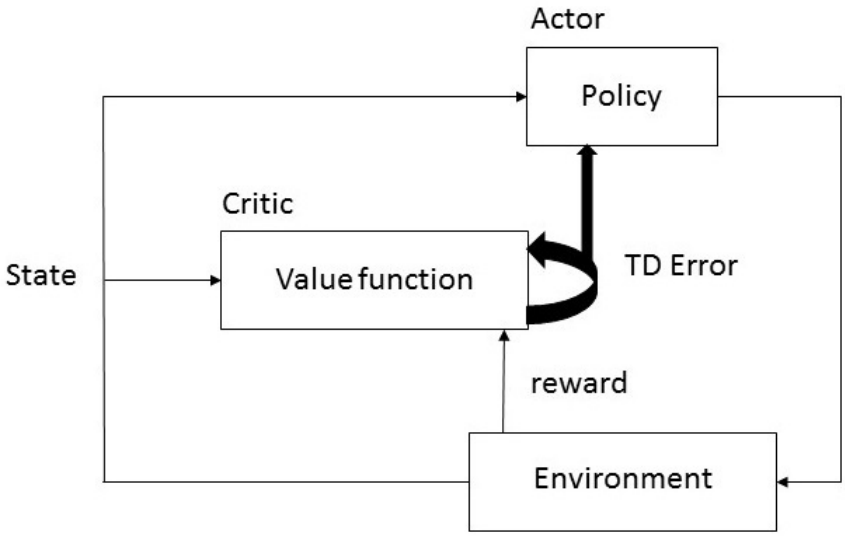


Figure 2.3: Actro Critic flow chart

time k to infinite time horizon, also called the Value function (Lewis and Vrabie, 2009) V_k.

$$V_k^{\pi}(x(k)) = \sum_{i=0}^{\infty} \gamma^i r_{k+i+1}^{\pi}(x(k), u(k))$$

is the value function $V^{\pi} : X \rightarrow \mathbb{R}$ under optimal control policy $\pi : X \rightarrow U$. In continuous spaces it becomes necessary to approximate the optimal policy $\hat{\pi}$ and the value function $\hat{V}$. The functions are approximated as,

$$\begin{aligned} \hat{V}_k(x(k)) &= \sum_{i=1}^{m} \theta_i \phi_i(x_k) \\ \hat{\pi}_k(x(k)) &= \sum_{j=1}^{n} \xi_j \Phi_i(x_k) \quad n, m \in \mathbb{Z}_+ \end{aligned}$$

by finite differentiable basis functionals (ϕ_i, Φ_j) so that gradient descent methods can be used to update the parameters θ_i, ξ_j. The Actor-Critic method updates both policy $\hat{\pi}$ and value function $\hat{V}$ iteratively to arrive at the best approximation $\hat{V}^{\hat{\pi}}$ (Sutton and Barto, 1998).

The actor generates the actions by the control policy and the critic is the estimated value function, which criticizes the policy through temporal difference error δ (as shown in Figure 2.3),

$$\delta_{k+1} = r_{k+1} + \gamma \hat{V}_{k+1} - \hat{V}_k$$

until the optimal policy and value function are learned. The critic parameters θ_i are updated by,

$$\begin{aligned} e_{k+1} &= \gamma\lambda e_k + \nabla_\theta \hat{V}(x_k, \theta_k) \\ \theta_{k+1} &= \theta_k + \alpha_c \delta_{k+1} e_{k+1} \end{aligned}$$

where $\alpha_c > 0$ is the learning rate for the critic parameter, e_k are the eligibility traces and $\lambda \in [0, 1)$ is the trace decay rate. The eligibility traces e_k store the knowledge of visited states and can be used to speed up the learning process (Sutton and Barto, 1998). The updated critic parameter is then used to update the policy. The algorithm learns from exploration by searching for states which might improve the reward function. The exploration Δu_k is drawn form a desired distribution (such as a Normal distribution) and added to the control policy (Sprangers *et al.*, 2015),

$$u_k = \hat{\pi}(x_k, \xi_k) + \Delta u_k.$$

The policy parameters ξ_i are updated towards (away) Δu_k if δ is positive (negative), with the update rule,

$$\xi_{k+1} = \xi_k + \alpha_a \delta_{k+1} \Delta u_k \nabla_\xi \hat{\pi}(x_k, \xi_k)$$

where $\alpha_a > 0$ is the learning rate of the policy parameter. The control saturation is incorporated by a generic control saturation function ζ. The above method is summarized in Algorithm 2.

Using actor critic methods without system knowledge can make the learning extremely slow and monotonous. Hence, providing information about the system speeds up the learning process significantly.

2.3 Summary

The Actor-Critic methods are used for RL in continuous spaces. Algorithm 2 takes into account non-linearities such as control saturation. We add a noise to control input for the purpose of exploration and to perturb the system in case it is stuck in a local minima.

Algorithm 2 Actor-critic Algorithm

procedure ACTOR CRITIC
    Input γ λ α_c α_a, for each actor
    Initialise $e_0(x)$=0 $\forall$ x
    Initialise x_0, θ_0 ξ_0
    $k \leftarrow 1$
    loop
    Execute:
    control action u_k=$\zeta(\hat{\pi}(x_k, \xi_k) + \Delta u_k)$
    where $\Delta u_k \sim N(0, \sigma^2)$
    $\Delta \bar{u}_k = u_k - \hat{\pi}(x_k, \xi_k)$
    Critic:
    Temporal difference: $\delta_{k+1} = r_{k+1}+$
    $\gamma \hat{V}(x_{k+1}, \theta_k) - \hat{V}(x_k, \theta_k)$
    Eligibility Trace: $e_{k+1} = \gamma \lambda e_k +$
    $\nabla_\theta \hat{V}(x_k, \theta_k)$
    Critic update: $\theta_{k+1} = \theta_k + \alpha_c \delta_{k+1} e_{k+1}$
    Actor:
    $\xi_{k+1} = \xi_k + \alpha_a \delta_{k+1} \Delta \hat{u}_k \nabla_\xi \hat{\pi}(x_k, \xi_k)$
end procedure

The next section discusses regulation problems and the classical methods in control theory used to solve them.

CHAPTER 3

REGULATION

3.1 Under-actuated Mechanical Systems

Under-actuated systems are those with fewer control inputs than the degrees of freedom. They arise in various fields, such as automobiles, airplanes, robotics and even animals. The Lagrangian dynamics of these systems may contain feedforward non-linearities, non-minimum phase zero dynamics, nonholonomic constraints, and other properties that place this class of systems at the forefront of research in nonlinear control (Spong, 1998).

3.1.1 Port Hamiltonian Representation

Under-actuated (Mechanical) systems assuming no natural damping can be written in Port Hamiltonian (PH) form,

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G(q) \end{bmatrix} u \tag{3.1}$$

where,

$$H(p,q) = \frac{1}{2} p^T M^{-1}(q) p + V(q) \tag{3.2}$$

represents the total energy of the system. $q \in \mathbb{R}^n$ and $p \in \mathbb{R}^n$ are the generalised positions and momenta respectively. $M(q) = M(q)^\top$ represents the inertia matrix and $V(q)$ the potential energy. The matrix $G \in \mathbb{R}^{n \times m}$ is determined by how the control input enters the systems and is invertible in the case of a fully actuated system, i.e., $m = n$. The main characteristic of PH system is that it models the system using its total energy, which can be used as a Lyapunov function in stability analysis.

3.2 Stabilisation using IDA-PBC

IDA-PBC is one of many PBC techniques and is used to regulate the position of under actuated systems. IDA-PBC achieves stability by making the system passive to a desired energy function. As shown in (Ortega *et al.*, 2002), IDA-PBC achieves this for an under-actuated system by shaping both the kinetic and potential energy of the system.

3.2.1 Target Dynamics

Motivated by (3.2) we propose the following form for the desired closed loop energy function,

$$H_d(p,q) = \frac{1}{2}p^T M_d^{-1}(q)p + V_d(q) \tag{3.3}$$

where $M_d = M_d^\top > 0$ and V_d represent the (to be defined) closed-loop inertia matrix and potential energy function, respectively. We will require that V_d have an isolated minimum at q_*, that is

$$q_* = argmin V_d(q) \tag{3.4}$$

In PBC, the control input is naturally decomposed into two terms

$$u = u_{es}(q,p) + u_{di}(q,p) \tag{3.5}$$

where the first term is designed to achieve the energy shaping and the second one injects the damping. The desired port-controlled Hamiltonian dynamics are taken of the form (Aeyels *et al.* (2008),Ortega *et al.* (2002), Van der Schaft (2012))

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \Big[J_d(q,p) - R_d(q,p)\Big]\begin{bmatrix}\nabla_q H_d\\ \nabla_p H_d\end{bmatrix} \tag{3.6}$$

where the terms

$$J_d = -J_d^\top = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_d M^{-1} & J_2(q,p) \end{bmatrix}$$

$$R_d = R_d^\top = \begin{bmatrix} 0 & 0 \\ 0 & GK_vG^\top \end{bmatrix} \geq 0$$

represent the desired interconnection and damping structures.

The following observations are made:

From (3.1) and (3.2), we have that $\dot{q} = M^{-1}p$. Since this is a nonactuated coordinate, this relationship should hold also in closed loop. Fixing (3.3) and (3.6) determines the (1,2)-block of J_d.

The matrix R_d is included to add damping into the system. This is achieved via negative feedback of the (new) passive output (also called L_gV control), which in the case is $G^\top \nabla_p H_d$. That is, we will select the second term of (3.5) as

$$u_{di} = -K_v G^\top \nabla_p H_d \tag{3.7}$$

where we take $K_v = K_v^\top > 0$. This explains the (2,2)-block of R_d.

We will show below that the skew-symmetric matrix J_2 (and some of the elements of M_d) can be used as free parameters in order to achieve the kinetic energy shaping. Providing these degrees of freedom is the essence of IDA-PBC.

3.2.2 Energy Shaping

To obtain the energy shaping term, u_{es}, of the controller we replace (3.5) and (3.7) in (3.1) and equate it with (3.6)

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

While the first row of the aforementioned equations is clearly satisfied, the second set of equations can be expressed as

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

Now, it is clear that if G is invertible, i.e., if the system is fully actuated, then we may uniquely solve for the control input u_{es} given any H_d and J_2. In the underactuated case, G is not invertible but only full column rank, and u_{es} can only influence the terms in the range space of G . This leads to the following set of constraint equations, which must be satisfied for any choice of u_{es}:

$$G^\top \left\{ \nabla_q H - M_d M^{-1} \nabla_q H_d + J_2 M_d^{-1} p \right\} = 0 \tag{3.8}$$

where $G^\top$ is a full rank left annihilator of G, i.e., $G^\top G = 0$. Equation (3.8), with H_d given by (3.3), is a set of nonlinear PDEs with unknowns M_d and V_d, with J_2 a free parameter, and p an independent coordinate. If a solution for this PDE is obtained, the resulting control law u_{es} is given as

$$u_{es} = \left(G^\top G\right)^{-1} G^\top \left(\nabla_q H - M_d M^{-1} \nabla_q H_d + J_2 M_d^{-1} p \right) \tag{3.9}$$

The PDEs (3.8) can be naturally separated into the terms that depend on p and terms which are independent of p, i.e., those corresponding to the kinetic and the potential energies, respectively. Thus, (3.8) can be equivalently written as

$$G^\top \left\{ \nabla_q \left(p^T M^{-1} p \right) - M_d M^{-1} \nabla_q \left(p^T M_d^{-1} p \right) + 2 J_2 M_d^{-1} p \right\} = 0 \tag{3.10}$$

$$G^\top \left\{ \nabla_q V - M_d M^{-1} \nabla_q V_d \right\} = 0 \tag{3.11}$$

The first equation is a nonlinear PDE that has to be solved for the unknown elements of the closed-loop inertia matrix M_d. Given M_d, (3.11) is a simple linear PDE, hence the main difficulty is in the solution of (3.10).

The following remarks are in order:

The derivations above characterize a class of under-actuated mechanical systems for which the newly developed IDA-PBC design methodology yields smooth stabilization. The class is given in terms of solvability of the nonlinear PDE (3.10), and the

linear PDE (3.11). Although it is well known that solving PDEs os generally hard, it is our contention that the added degree of freedom- the closed loop interconnection J_2 -simplifies this task.

There are two "extreme" particular cases of our procedure. first, if we do not modify the interconnection matrix then we recover the well-known potential energy shaping procedure of PBC. Indeed, if $M_d = M$ and $J_2 = 0$, then the controller equation (3.9) reduces to

$$u_{es} = (G^\top G)^{-1} G^\top (\nabla_q V - \nabla_q V_d)$$

which is th familiar potential energy shaping control.

3.3 Summary

We have seen that because of the structural properties of the PH form, the IDA-PBC formulation reduces to solving two PDEs, one for kinetic energy and the other for potential.

$$\begin{aligned} G^\top \left\{ \nabla_q \left(p^T M^{-1} p \right) - M_d M^{-1} \nabla_q \left(p^T M_d^{-1} p \right) + 2 J_2 M_d^{-1} p \right\} &= 0 \\ G^\top \left\{ \nabla_q V - M_d M^{-1} \nabla_q V_d \right\} &= 0 \end{aligned}$$

where J_2 is any skew symmetric matrix. Even with this added degree of freedom, it is still non-trivial to solve these PDEs and in some cases it takes significant effort to reduce them to Ordinary differential equations (ODEs), under some assumptions. We can use RL techniques to solve the complex PDEs for Ball on a beam system and 2D spidercrane as shown below.

In the next section, we use Actor-Critic methods nested in IDA-PBC formulation and present simulation results for ball on a beam system.

CHAPTER 4

BALL ON A BEAM SYSTEM

The Ball on a beam system is a system widely used by control engineers to validate new control strategies. It is a two degree-of-freedom system with a single actuator, which makes it an underactuated system with degree of under actuation equal to one. The traditional challenge offered by the ball and beam system is to stabilize the ball at the center of the centrally actuated beam which is an equilibrium point or to stabilize the ball at the center of the end actuated beam which is an operating point.

4.1 Problem Formulation

4.1.1 Model of Ball on a Beam

A schematic of a ball on an end actuated beam is shown in Figure 4.1 (Muralidharan *et al.*, 2010) . The ball is mounted on a beam of length L. The beam is end actuated and is attached to a gear of Radius R. The beam makes an angle α with the horizontal and the beam of the gear makes an angle θ with the horizontal.

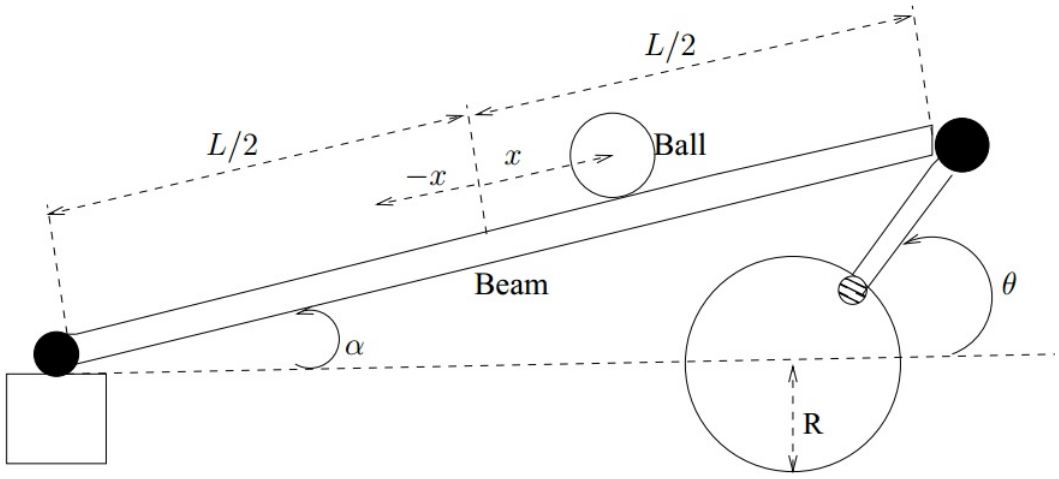


Figure 4.1: A schematic of Ball on a Beam system

As shown in Muralidharan *et al.* (2010), The state variables are q_1 and q_2 where,

$$\begin{aligned} q_1 &= x \\ q_2 &= \alpha \approx \frac{R\theta}{L} \end{aligned} \tag{4.1}$$

and q_1 represents the position of the ball and q_2 the inclination of the beam. The approximation, (4.1) is valid only for small α and small θ. The system's configuration space is $Q = \mathbb{R} \times \mathbb{S}^1$.

The mass matrix and the potential energy can be written as,

$$D(q_1) = diag\left(a_1, a_2 + a_3\left(q_1 + \frac{L}{2}\right)^2\right) \tag{4.2}$$

$$V(q) = b_1 \sin(q_2) + b_2(q_1 + L/2)\sin(q_2), \tag{4.3}$$

where the inertial parameters, defined in Table 4.1, are collected in constants

Table 4.1: System Parameters for Ball on Beam System

Parameter	Symbol
Length of the beam	L
Radius of the ball	r_b
Mass of the ball	M_b
Moment of Inertia of the ball	J_b
Mass of the beam	M_r
Radius of gear	R
Moment of inertia of the beam about its pivot	J_r
Moment of inertia of the gear	J_{fw}
Acceleration due to gravity	g

$$\begin{aligned} a_1 &= \frac{J_b}{r_b^2}M_b; \;\; a_2 = J_r\frac{R^2}{L^2} + J_{fw} \\ a_3 &= \frac{M_b R^2}{L^2}; \;\; b_1 = \frac{M_r g L}{2}; \;\; b_2 = M_b g. \end{aligned}$$

and the control input matrix is $G = [0, 1]^\top$. Unlike the ball on a centrally actuated beam system, the length of the beam L appears in the equations of motion. The control objective is to stabilize the system at $q^* = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top \in Q \times TQ$.

4.1.2 IDA-PBC Formulation

As shown in Muralidharan *et al.* (2010), $M_d = M_d^\top > 0$ is parametrised such that it solves the potential PDE

$$G^\top \{\nabla_q V - M_d D^{-1} \nabla_q V_d\} = 0$$

where $G^\top = e_1^\top$. In this direction, we retain the dependence of M_d on q_1, that is $M_d = M_d(q_1)$. With J_2 parametrised as $J_2 = p^\top M_d^{-1} A(q_1) W$, where $A(q_1) = [A_1 A_2] \in C^1$ is free and $W \in so(2) \in \mathbb{R}^2 \times \mathbb{R}^2$, the space of skew-symmetric matrices, then the kinetic energy PDE reduced to an ODE in M_d with respect to q_1 and is given by,

$$-M_d D^{-1} D' D^{-1} M_d + (G^\perp M_d D^{-1} e_1) M_d' + \begin{bmatrix} 0 & A_1(q_1) \\ A_1(q_1) & 2A_2(q_1) \end{bmatrix} = 0$$

Let us introduce the change of variables (Auckly and Kapitanski, 2002) given by

$$\lambda \triangleq M_d D^{-1}$$

The corresponding kinetic ODE's and the potential PDE in the new variable is termed the $\lambda-$equations. This change of variabes greatly aids in solving for the $M_d = M_d^T > 0$ and V_d. Let $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix}$, then the KE ODE can be written as,

$$-\lambda D' \lambda + \lambda_1 M_d' + \begin{bmatrix} 0 & A_1(q_1) \\ A_1(q_1) & 2A_2(q_1) \end{bmatrix} = 0$$

Where,

$$M_d \triangleq \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} a_1\lambda_1 & \lambda_2 \left(a_2 + a_3 \left(\frac{L}{2} + q_1\right)^2\right) \\ a_1\lambda_3 & \lambda_4 \left(a_2 + a_3 \left(\frac{L}{2} + q_1\right)^2\right) \end{bmatrix}$$

The ODE's for $m_{ij}'s$ in terms of $\lambda_i's$ are given by,

$$-2a_3(q_1 + L/2)\lambda_2^2 + a_1\lambda_1 \frac{d\lambda_1}{dq_1} = 0 \tag{4.4}$$

$$-2a_3(q_1 + L/2)\lambda_2\lambda_4 + \lambda_1 \frac{d}{dq_1}\lambda_2(a_2 + a_3(L/2 + q_1)^2) + A_1 = 0$$

$$-2a_3(q_1 + L/2)\lambda_4^2 + \lambda_1 \frac{d}{dq_1}\lambda_4(a_2 + a_3(L/2 + q_1)^2) + 2A_2 = 0$$

Form the non-trivial ODE (4.4), we get

$$\lambda_1(q_1) = k_1(q_1 + L/2) > 0, \;\; \forall q_1 \in (-L/2, L/2), \;\; \text{where } k_1 > 0 \tag{4.5}$$

$$\lambda_2 = k_2 = k_1\sqrt{\frac{a_1}{2a_3}} > 0 \tag{4.6}$$

From the closed loop mass matrix $M_d \geq 0 \; \forall q \in (-L/2, L/2)$ we get,

$$\lambda_3 = \frac{k_2}{q_1}(a_2 + a_3(L/2 + q_1)^2) \tag{4.7}$$

$$\begin{aligned} k_4 &> \underset{q_1 \in (\frac{-L}{2}+\epsilon, \frac{L}{2}-\epsilon)}{arg\ max} \frac{k_1(a_2 + a_3(L/2 + q_1)^2)^2}{2a_3(L/2 + q_1)} \\ &= \frac{k_1(a_2 + a_3(\epsilon)^2)^2}{2a_3\epsilon} \end{aligned} \tag{4.8}$$

where k_4 is $M_d(2,2)$ and ϵ is very small. The free functions in the gyroscopic term are extracted as,

$$A_1 = 2a_3\lambda_2(q_1 + L/2)(\lambda_4 - \lambda_1) \tag{4.9}$$

$$A_2 = a_3(q_1 + L/2)\lambda_4^2 \tag{4.10}$$

The closed loop potential energy V_d is approximated by

$$\hat{V}_d = \psi_1(1 - \cos q_2) + \psi_2 q_1^2 + \psi_3 q_2^2 + \psi_4 q_1 q_2$$

where ψ_1, ψ_2, ψ_3 and ψ_4 are learned using RL. The function V_d needs to have a minima at $q*$, From $\nabla V_d = 0$ we get,

$$\psi_2 < 0 \tag{4.11}$$

$$\psi_1 = \frac{b_2}{k_2} \tag{4.12}$$

$$\psi_3 = \frac{-k_1(q_1 + L/2)\psi_2}{2k_2} \tag{4.13}$$

From $\nabla^2 V_d \geq 0$ we get,

$$\psi_4 = \frac{-k_2\psi_2 q_1^2}{2k_1(q_1 + L/2)} \tag{4.14}$$

Finally from equations (4.5)-(4.14), the free parameters to be learned using Actor-Critic Algorithm 2 are k_1, k_4, ψ_2, k_v.

4.2 Implementation

4.2.1 Reinforcement Learning nested in IDA-PBC

The free parameters are learned using the update policy as given in Algorithm 3.

Algorithm 3 Actor update for Ball on a Beam system

procedure ACTOR UPDATE
    Actor:
    $\xi_k = (k_{1_k}, k_{4_k}, k_{v_k}, \psi_{2_k})$
    $k_{1_{k+1}} = k_{1_k} + \alpha_{k_1}\delta_{k+1}\Delta\bar{u}_k\nabla_{k_1}\zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{4_{k+1}} = k_{4_k} + \alpha_{k_4}\delta_{k+1}\Delta\bar{u}_k\nabla_{k_4}\zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{v_{k+1}} = k_{v_k} + \alpha_{k_v}\delta_{k+1}\Delta\bar{u}_k\nabla_{k_v}\zeta(\hat{\pi}(x_k, \xi_k))$
    $\psi_{2_{k+1}} = \psi_{2_k} + \alpha_{\psi_2}\delta_{k+1}\Delta\bar{u}_k\nabla_{\psi_2}\zeta(\hat{\pi}(x_k, \xi_k))$
end procedure

The value of system parameters used are $a_1 = 0.0896$,$a_2 = 0.002$, $a_3 = 0.000228$, $b_1 = 0.3130$, $b_2 = 0.6278$ and $L = 0.425$. The critic function (i^{th} basis for the k^{th} state) and the reward function for the k^{th} state used are,

$$\phi_c(q(k)) = i(cos(2iq_2(k)) - 1) - iq_1(k)^2$$

$$r(q(k)) = r_{q2}(cos(2iq_2(k)) - 1) - r_{q1}q_1(k)^2$$

$$-r_{p1}p_1(k)^2 - r_{p2}p_2(k)^2$$

The parameter values used in the algorithm are mentioned in Table 4.2.

Table 4.2: Parameter values for Ball on a Beam System

Parameter	Symbols	Values
Maximum input to the system	umax	6
Learning rate of k_1	α_{k1}	10^{-7}
Learning rate of k_4	α_{k4}	10^{-6}
Learning rate of k_v	α_{kv}	10^{-2}
Learning rate of ψ_2	α_{ψ_2}	10^{-6}
Learning rate of the critic	α_c	0.05
Discount factor	γ	0.99
Trace decay rate	λ	0.65
Reward function coefficient for q_1	r_{q1}	20000
Reward function coefficient for q_2	r_{q2}	20
Reward function coefficient for p_1	r_{p1}	1000
Reward function coefficient for p_2	r_{p2}	1

4.3 Simulation Results

The simulation was repeated 3000 times and the estimate of the average, minimum, maximum and confidence regions are plotted in figure 4.2.

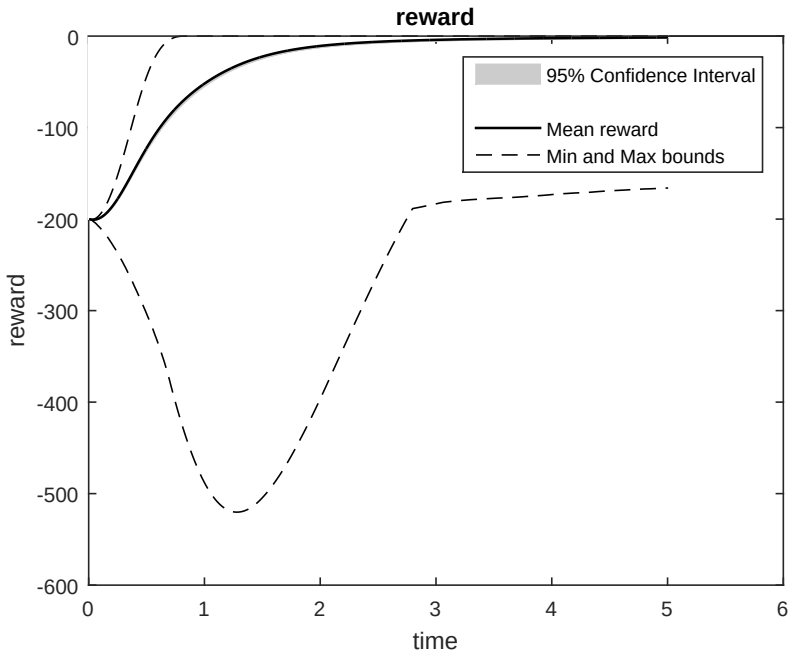


Figure 4.2: Reward Function

The system moves toward the minima of the Potential Energy as seen in the Figure 4.4.

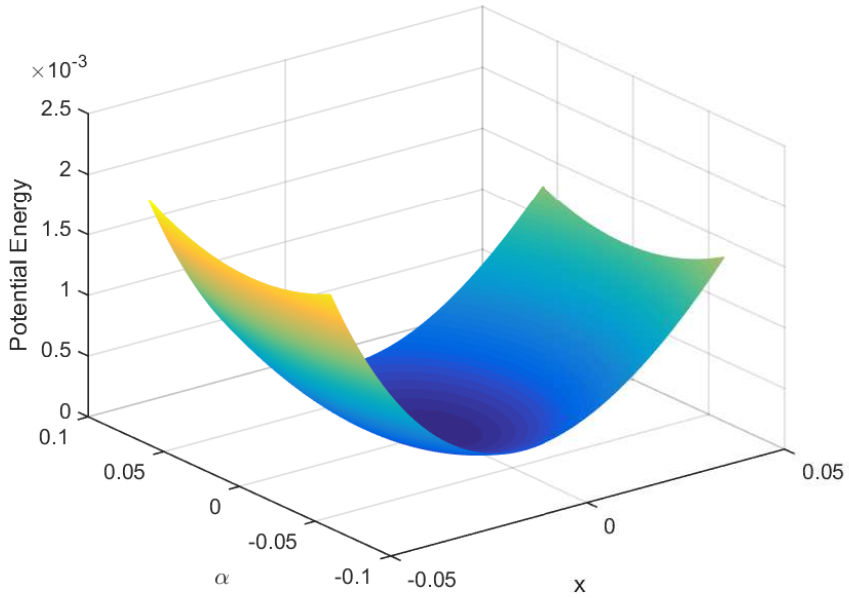


Figure 4.3: The learned Potential Energy

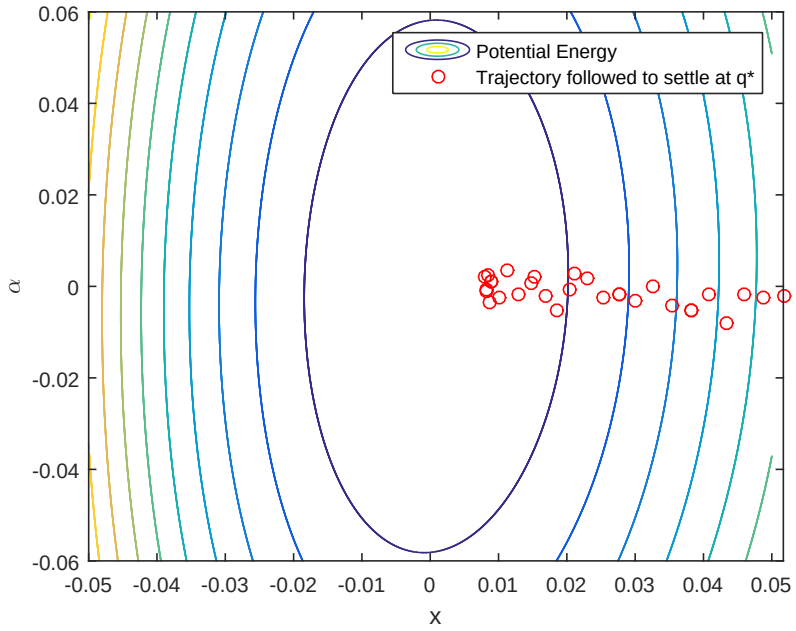


Figure 4.4: Trajectory Tracked on the potential energy contour

The system will not settle in any local minimas because of the exploration in the control policy. It has also been visualised in the configuration space $\mathbb{R} \times \mathbb{S}^1$ in figure 4.5.

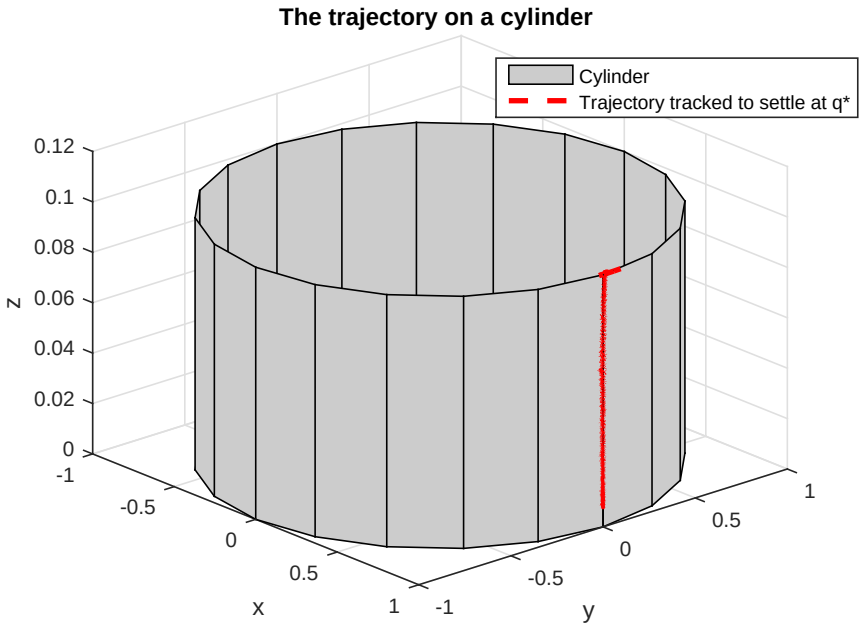


Figure 4.5: Trajectory Tracked on the cylinder

4.4 Summary

We have presented a method to parametrise IDA-PBC control laws which are robust to non-linearities such as control saturation. The free parameter values were calculated using Actor-Critic method. This gives us a way to numerically access the stability using passivity theory. By providing system knowledge, the convergence of the algorithm can be significantly improved, as the algorithm is computationally expensive.

In the next section, we use Actor-Critic methods nested in IDA-PBC formulation and present simulation results for 2D Spidercrane system.

CHAPTER 5

2D SPIDERCRANE

5.1 Problem Formulation

5.1.1 2D SpiderCrane Model

Consider the 2D SpiderCrane mechanism shown in 5.1 (Kazi *et al.*, 2008).

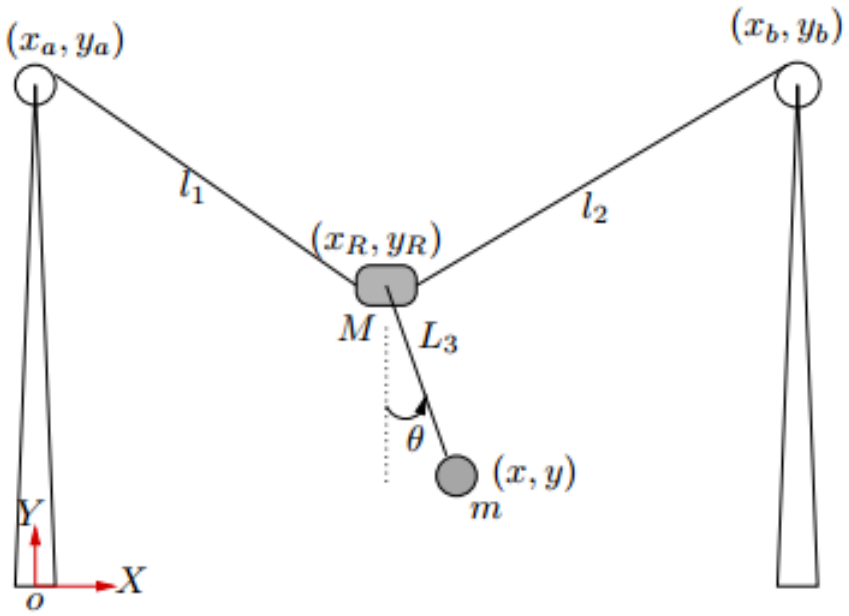


Figure 5.1: 2D SpiderCrane Mechanism

The load of mass m is at (x, y). The position of the load is varied by varying the lengths l_1 and l_2. The position of the two motors is (x_a, y_a) and (x_b, y_b) with rotatory inertia I_a and I_b respectively. The ring of mass M with position (x_R, y_R) is attached to the load using a cable of fixed length L_3. The following assumptions are made about the model,

- the cable is massless and inelastic
- Both pylons are assumed to be on the same height
- Dissipative forces on the cart and the winch are negligible

The model is an underactuated system with 2 holonomic constraints. It essentially captures all the control-theoretical perspectives of SpiderCrane discussed in Buccieri *et al.* (2005)

5.1.2 Dynamics of 2D Spider Crane

As shown in Kazi *et al.* (2008), the configuration variables are,

$$q = \begin{bmatrix} x_R & y_R & \theta & l_1 & l_2 \end{bmatrix}$$

where $\theta \in (0, 2\pi)$ represents the payload angle about the vertical axis. the ring position is specified with $x_R \in \mathbb{R}^1$ in the X axis and $y_R \in \mathbb{R}^1$ in the Y axis. l_1 and l_2 represent the cable lengths. The control force $u = [F_1, F_2]^\top$, where F_1 and F_2 are the control inputs acting on each cable. The control objective is to move the payload from any position $q_i = \begin{bmatrix} x_R i & y_R i & \theta_i & l_1 i & l_2 i \end{bmatrix}$ to a desired position $q_d = \begin{bmatrix} x_R d & y_R d & \theta_d & l_1 d & l_2 d \end{bmatrix}$ and the payload angle has to be at zero when at rest.

The holonomic constraints are,

$$C_1(q) = (x_R)^2 + (y_R - y_a)^2 - (l_1)^2 = 0 \tag{5.1}$$

$$C_2(q) = (x_R - x_b)^2 + (y_R - y_b)^2 - (l_2)^2 = 0 \tag{5.2}$$

The Lagrangian is,

$$\mathcal{L} = \frac{1}{2}\dot{q}^T M(q)\dot{q} - V(q) \tag{5.3}$$

where,

$$M(q) = \begin{bmatrix} M+m & 0 & mL_3\cos\theta & 0 & 0 \\ 0 & M+m & mL_3\sin\theta & 0 & 0 \\ mL_3\cos\theta & mL_3\sin\theta & mL_3^2 & 0 & 0 \\ 0 & 0 & 0 & I_a & 0 \\ 0 & 0 & 0 & 0 & I_b \end{bmatrix} \tag{5.4}$$

is the inertia matrix and

$$V(q) = (M+m)gy_R - mgL_3\cos\theta \tag{5.5}$$

is the potential energy function.

As the admissible system motions lie in th range of,

$$S(q) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{y_R - y_a}{l_1} & \frac{x_R}{l_1} \\ 0 & \frac{y_R - y_b}{l_2} & \frac{x_R - x_b}{l_2} \end{bmatrix} \tag{5.6}$$

We exclude the points where

- the ring mass is at the first pulley with $l_1 = 0$
- the ring mass is at the second pulley with $l_2 = 0$

from our discussions.

5.1.3 Decoupled SpiderCrane Model

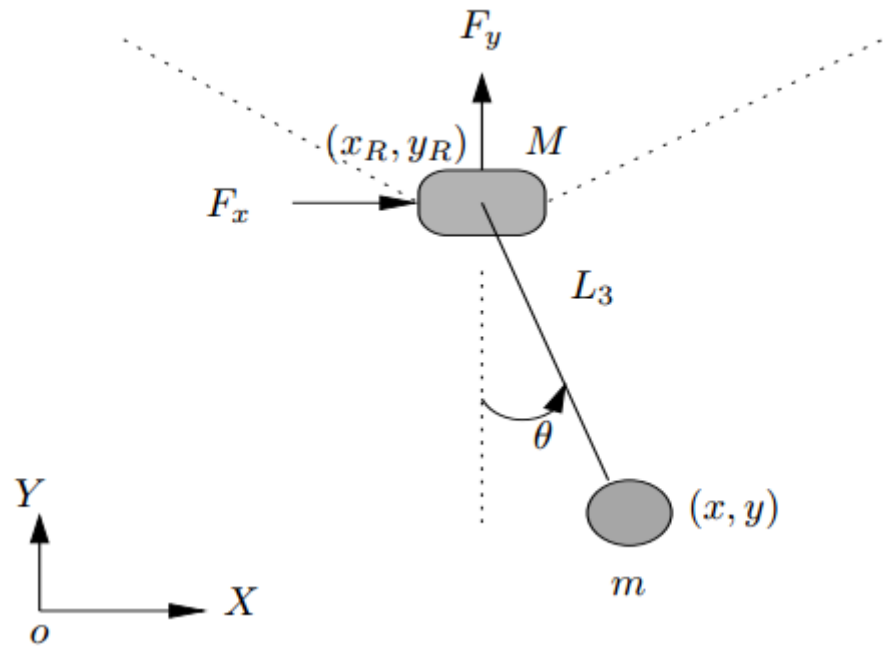


Figure 5.2: 2D SpiderCrane Gantry cart

We consider the 2D SpiderCrane as a decoupled system as shown in Fig. 5.2 (Kazi *et al.*, 2008). The configuration variables for the gantry mechanism are $q = (x_R, y_R, \theta)^T$ and the Lagrangian is

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} - V(q) \tag{5.7}$$

where,

$$M(q) = \begin{bmatrix} M+m & 0 & mL_3\cos\theta \\ 0 & M+m & mL_3\sin\theta \\ mL_3\cos\theta & mL_3\sin\theta & mL_3^2 \end{bmatrix} \tag{5.8}$$

and

$$V(q) = (M+m)gy_R - mgL_3\cos\theta \tag{5.9}$$

The resulting Euler Lagrangian equations are:

$$\begin{aligned} F_x &= (M+m)\ddot{x}_R + (mL_3\cos\theta)\ddot{\theta} - (mL_3\sin\theta)\dot{\theta}^2 \\ F_y &= (M+m)\ddot{y}_R + (mL_3\sin\theta)\ddot{\theta} + (mL_3\cos\theta)\dot{\theta}^2 + (M+m)g \\ 0 &= (mL_3\cos\theta)\ddot{x}_R + (mL_3\sin\theta)\ddot{y}_R + (mL_3^2)\ddot{\theta} + mgL_3\sin\theta \end{aligned}$$

The Hamiltonian of the system is

$$H(q, p) = \frac{1}{2}p^T M^{-1}(q)p + V(q) \tag{5.10}$$

where $q \in \mathbb{R}^n$ and $p \in \mathbb{R}^n$ are the generalised position and momenta, $M(q) = M^T(q) > 0$ is the inertia matrix, and $V(q)$ is the potential energy.

Assuming that the system has no natural damping it can be represented as,

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H \\ \nabla_p H \end{bmatrix} + \begin{bmatrix} 0 \\ G(q) \end{bmatrix} u \tag{5.11}$$

where for 2D spidercrane,

$$G(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{5.12}$$

This gave us $G^{\perp} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$.

5.1.4 Pulley Dynamics

A schematic of Pulley Dynamics is shown in Figure 5.3 (Kazi *et al.*, 2008).

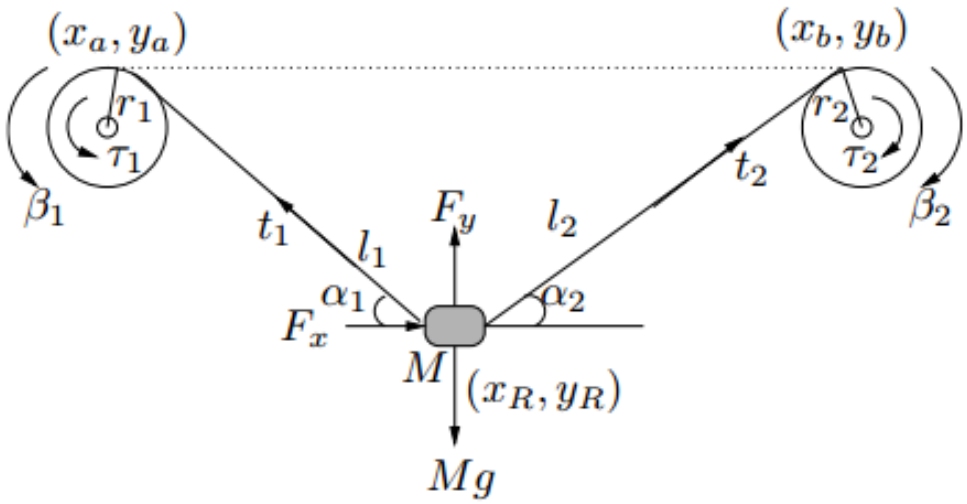


Figure 5.3: Pulley cable Schematic

where t_i is cable tension, τ_i and β_i are the motor torque and the pulley angle respectively, for the i_{th} pulley. Here $r_1 = r_2 = r$ is the pulley radius. Also, the no-slip constraint for the i_{th} pulley gives $r\dot{\beta}_i = \dot{l}_i$. The relation between the motor torques and F_x and F_y is as follows:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = r \begin{bmatrix} \cos\alpha_1 & -\cos\alpha_2 \\ -\sin\alpha_1 & -\sin\alpha_2 \end{bmatrix}^{-1} \begin{bmatrix} F_x \\ F_y - Mg \end{bmatrix} + r \begin{bmatrix} I_a & 0 \\ 0 & I_b \end{bmatrix} \begin{bmatrix} \ddot{l}_1 \\ \ddot{l}_2 \end{bmatrix} \tag{5.13}$$

$\alpha_i > 0, i = 1, 2$. The above equation is not valid for when the cables lie in a straight line.

5.1.5 IDA-PBC Formulation

Since the desired equilibrium of the system is the natural equilibrium , we only shape the potential energy of the system, keeping the kinetic energy unchanged. So $M_d = M$. To influence the underactuated coordinate θ we make the interconnection matrix J_2 skew symmetric and linear in p,

$$J_2 = k \begin{bmatrix} 0 & 0 & \dot{y}_R \\ 0 & 0 & -\dot{x}_R \\ -\dot{y}_R & \dot{x}_R & 0 \end{bmatrix} \tag{5.14}$$

This clearly satisfies the kinetic energy PDE,

$$G^\top \left\{ \nabla_q \left(p^T M^{-1} p \right) - M_d M^{-1} \nabla_q \left(p^T M_d^{-1} p \right) + 2 J_2 M_d^{-1} p \right\} = 0$$

We influence the swing by the tuning parameter k which influences the interconnection structure. The potential energy PDE,

$$G^\top \left\{ \nabla_q V - M_d M^{-1} \nabla_q V_d \right\} = 0$$

with $G^\perp = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, takes the form $\nabla_{q3} V - \nabla_{q3} V_d = 0$. Hence, we assume $V_d(q)$ to be of the form,

$$V_d = -mgL_3 \cos\theta + k_{px}(\exp(x_R - x_{Rd}) - x_R) + k_{py}(\exp(y_R - y_{Rd}) - y_R) \tag{5.15}$$

where $k_{px}, k_{py} > 0$, $q_{Rd} = (x_{Rd}, y_{Rd}, 0)$ and $\theta \in (-\pi/2, \pi/2)$ so as to satisfy the gradient and Hessian conditions for q_{Rd}= argmin$V_d(q)$. If we chose $V_d(q)$ to be a quadratic function we would recover a PD like control law as in Fang *et al.* (2003). Exponential function is chosen as they are steeper than quadratic function, implying that for a large deviation from the desired position the rate at which the system moves to the equilibrium is faster as compared to quadratic functions. From equation (5.14) and (5.15), we get the control law as,

$$u_{es} = \begin{bmatrix} -k_{px}(\exp(x_R - x_{Rd}) - 1) \\ (M+m)g - k_{py}(\exp(y_R - y_{Rd}) - 1) \end{bmatrix} + \begin{bmatrix} k\dot{y}_R\dot{\theta} \\ k\dot{x}_R\dot{\theta} \end{bmatrix} \tag{5.16}$$

And the damping injection term is designed as,

$$u_{di} = -K_v G^T \dot{q} = - \begin{bmatrix} k_a \dot{x}_R + k_b \dot{y}_R \\ k_b \dot{x}_R + k_c \dot{y}_R \end{bmatrix} \tag{5.17}$$

where K_v is a symmetric and positive definite matrix of the form $K_v = \begin{bmatrix} k_a & k_b \\ k_b & k_c \end{bmatrix}$. The gains $k_p x$ and $k_p y$ are like proportional gains acting on the error in the configuration variables giving us a proportion like term in the control-law. K_v acts on the derivative of the configurational variables and introduces damping into the system. The system has 5 free parameters, $k, k_{px}, k_{py}, k_a, k_b$ and k_c which are learnt using reinforcement learning

5.2 Implementation

5.2.1 Reinforcement Learning nested in IDA-PBC

The free parameters are learned using the update policy as given in Algorithm 4.

Algorithm 4 Actor update for 2D SpiderCrane

procedure ACTOR UPDATE
    Actor:
    $\xi_k = (k, k_{px}, k_{py}, k_a, k_b, k_c)$
    $k_{k+1} = k_k + \alpha_k \delta_{k+1} \Delta \bar{u}_k \nabla_k \zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{px_{k+1}} = k_{px_k} + \alpha_{k_{px}} \delta_{k+1} \Delta \bar{u}_k \nabla_{k_{px}} \zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{py_{k+1}} = k_{py_k} + \alpha_{k_{py}} \delta_{k+1} \Delta \bar{u}_k \nabla_{k_{py}} \zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{a_{k+1}} = k_{a_k} + \alpha_{k_a} \delta_{k+1} \Delta \bar{u}_k \nabla_{k_a} \zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{b_{k+1}} = k_{b_k} + \alpha_{k_b} \delta_{k+1} \Delta \bar{u}_k \nabla_{k_b} \zeta(\hat{\pi}(x_k, \xi_k))$
    $k_{c_{k+1}} = k_{c_k} + \alpha_{k_c} \delta_{k+1} \Delta \bar{u}_k \nabla_{k_c} \zeta(\hat{\pi}(x_k, \xi_k))$
end procedure

The value of system parameters used are $M = 0.5, m = 1$ and $l_3 = 0.5$. The desired position $q_{Rd} = (0.5, 1, 0)$ The critic function (i^{th} basis for the k^{th} state) and the

reward function for the k^{th} state used are,

$$\phi_c(q(k)) = i(cos(2iq_3(k)) - 1) - iq_1(k)^2 - iq_2(k)^2$$
$$r(q(k)) = r_{q3}(cos(2iq_3(k)) - 1) - r_{q1}(q_1(k) - q_{1_{Rd}})^2 - r_{q2}(q_2(k) - q_{2_{Rd}})^2$$
$$-r_{q4}q_4(k)^2 - r_{q5}q_5(k)^2 - r_{q6}q_6(k)^2$$

The parameter values used in the algorithm are mentioned in Table 5.1.

Table 5.1: Parameter values used for 2D spidercrane system

Parameter	Symbols	Values
Learning rate of k_a	α_{k_a}	10^{-7}
Learning rate of k_b	α_{k_b}	10^{-7}
Learning rate of k_c	α_{k_c}	10^{-7}
Learning rate of k	α_k	10^{-3}
Learning rate of k_{px}	$\alpha_{k_{px}}$	10^{-7}
Learning rate of k_{py}	$\alpha_{k_{py}}$	10^{-7}
Learning rate of the critic	α_c	0.05
Discount factor	γ	0.99
Trace decay rate	λ	0.65
Reward function coefficient for q_1	r_{q1}	8
Reward function coefficient for q_2	r_{q2}	8
Reward function coefficient for q_3	r_{q3}	1000
Reward function coefficient for q_4	r_{q4}	6
Reward function coefficient for q_5	r_{q5}	6
Reward function coefficient for q_6	r_{q6}	100

5.3 Simulation results

The reward function when the algorithm was run for 5 seconds is shown in Figure 5.4. The reward function reaches the maximum value of 0 after learning for approximately 3 seconds.

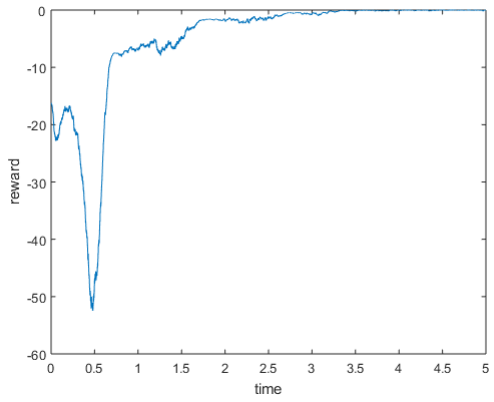


Figure 5.4: Reward Function

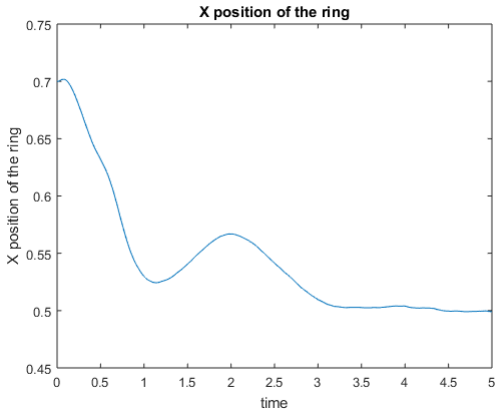


Figure 5.5: X position of the ring

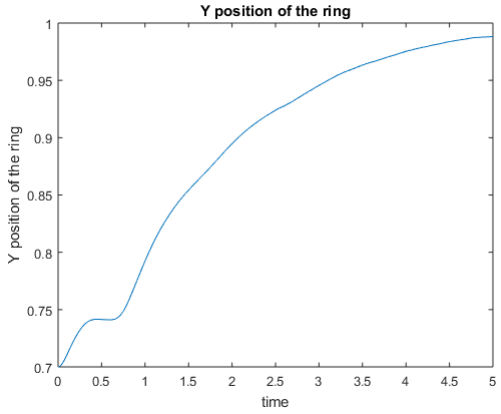


Figure 5.6: Y position of the ring

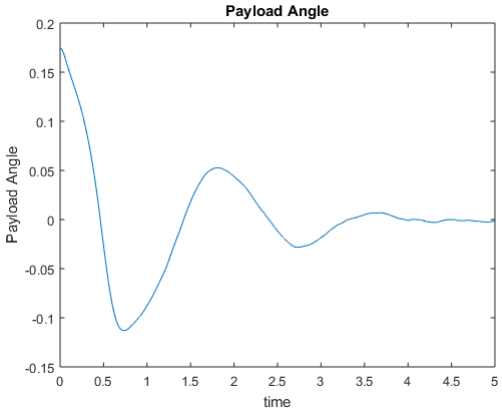


Figure 5.7: Payload Angle

The X position of the ring converges to the desired position $x_{Rd} = 0.5$. The Y position of the ring converges to the desired position $y_{Rd} = 1$. The payload angle reach $\theta = 0$ when the SpiderCrane is at rest.

5.4 Summary

The controller learnt by the actor-critic method stabilises the 2D spidercrane system at q_{Rd}. We have utilised the freedom in the interconnection matrix J_2 to influence the cable swing, the unactuated coordinate. The learnt control law, as observed performed well for point to point control with swing minimisation.

In the next section we formally introduce the concepts of feedforward proportional-derivative controller.

CHAPTER 6

TRACKING

Let us first define a simple mechanical system,

Definition 1 Simple mechanical system (Lewis and Bullo, 2005):
A simple mechanical system is a $4-tuple$ $(Q, \mathbb{G}, \overset{\mathbb{G}}{\nabla}, F)$*, where* Q *is configurational manifold,* $\mathbb{G}$ *is the Riemannian metric defined on* Q*,* $\overset{\mathbb{G}}{\nabla}$ *denotes the Levi-Civita connection defined using* $Q, \mathbb{G}$ *and* $F \in T^*Q$ *represents the control action. The Euler Lagrangian equations of the mechanical system are*

$$\mathbb{G} \overset{\mathbb{G}}{\nabla}_{\dot{\gamma}(t)} \dot{\gamma}(t) = F \tag{6.1}$$

where $\gamma : [0, \infty) \mapsto Q$ *represents a curve on* Q*.*

In this section we consider tracking problem of a fully actuated simple mechanical system. A system is fully actuated means its degree-of-freedom is equal to number of actuators. Therefore can calculate the approximate amount of control input to drive it from one state to another state, any error in position or velocity can be feedback to the system. This type of controller is called as feed forward proportional-derivative controller (Murray *et al.*, 1994).

In this controller the proportional term is generated by a linear error potential function say

$$\psi_l(x(t), r(t)) = (x(t) - r(t))^T k_p (x(t) - r(t)) \tag{6.2}$$

where $x(t)$ is the current position and $r(t)$ is the reference position on the configurational manifold, generally an Euclidian space $\mathbb{R}^n$ and $k_p \in \mathbb{R}^{n \times n}$. The gradient of the error potential function gives us the proportional control force

$$F_p = \frac{\partial \psi_l}{\partial x} = k_p (x(t) - r(t)) \tag{6.3}$$

required to correct the position error, which is indeed the position error in proportional derivative controller. This linear potential functions are suitable for system with configurational manifold $\mathbb{R}^n$, that is, translational joints. But when we have a rotational joints in the system, the configurational manifold will not be Euclidian and in these systems linear error potential functions generally fail in capturing the true error.

For example, consider a ball moving in a unit circle, in this case the configurational manifold is $Q = \mathbb{S}^1$. Consider two initial conditions 0 and 2π, which represents the same position, and let the desired position be $\frac{\pi}{2}$. The error generated by the error function $k_p(x(t) - r(t))$ corresponding to the same position 0 and 2π results in different values $-\frac{\pi}{2}$ and $\frac{3\pi}{2}$.

Now consider a nonlinear potential function

$$\psi_{nl}(x(t), r(t)) = k_p(1 - \cos(x(t) - r(t))), \tag{6.4}$$

the position error due to this scalar function ψ_{nl} is given by

$$\frac{\partial \psi_{nl}}{\partial x} = k_p \sin(x(t) - r(t)) \tag{6.5}$$

and when calculated at both the initial points $0, 2\pi$ the error is the same. This error potential function is defined using the properties of the manifold. But due to this nonlinear potential shaping, additional equilibrium points are introduced

$$\frac{\partial \psi_{nl}}{\partial x} = 0. \tag{6.6}$$

In the running example, in addition to the equilibrium point $x(t) = r(t)$, an additional equilibrium $x(t) = r(t) + \pi$ manifests and which can be shown to be unstable. So the control defined with this potential function is restricted to a domain defined by $D = \mathbb{S}^1 - \pi$. The formal definition of ψ is as follows.

Definition 2 Configurational error function

A smooth function $\psi : Q \to \mathbb{R}$ is configurational error function about $r \in Q$ if it is properly bounded below, and satisfies

(i) $\psi(r) = 0$

(ii) $d\psi(r) = 0$

(iii) $Hess\ \psi(r) > 0$.

Definition 3 Tracking error function:

A smooth and symmetric function $\psi : Q \times Q \mapsto \mathbb{R}$ *is a tracking error function if, for* $r \in Q$*, the function* $\psi_r : Q \mapsto \mathbb{R}$ *is a configurational error function about* r *and satisfies*

(i) $\psi(r, r) = 0$,

(ii) $d_1\psi(r, r) = 0$,

(iii) $Hess_1\psi(r, r) > 0$.

For $(q, r) \in (Q \times Q)$ we write the differential with respect to first argument as $d_1\psi = \frac{\partial\psi}{\partial q} \in T_q^*Q$ and the differential with respect to second argument as $d_2\psi = \frac{\partial\psi}{\partial r} \in T_r^*Q$.

6.1 Transport map

In general for a proportional-derivative control, the velocity error is calculated by $\dot{q}(t) - \dot{r}(t)$ where $q(t), r(t)$ are the current and reference trajectories respectively. For calculating this error we are actually comparing the vectors in two different tangent spaces $(q(t), r(t))$. Since there exists natural isomorphisms(in this case identity transformation) between any two tangent spaces, the notion of the error defined above is valid. But we can also find some other isomorphisms which can aid our control. This isomorphism we call it as transport map, formally defined below.

Definition 4 Transport map:

A transport map is a smooth vector bundle map

$$T : Q \times TQ \to TQ \times Q \tag{6.7}$$

over $id_{Q\times Q}$ *with the property that* $T(q, X_q) = X_q$ *for all* $q \in Q$ *and* $X_q \in T_qQ$

It maps vector field at reference position to vector field at current position. Let $(q, r) \in (Q \times Q)$ and (r, Y_r) is the vector field along the reference trajectory $(r(t))$,

then $(q, T(q,r).Y_r)$ is the transported vector field defined on controlled trajectory $q(t)$. Therefore we now formally define notion of velocity error.

$$\dot{e}(t) = \dot{q}(t) - T(q(t), r(t)).\dot{r}(t) \tag{6.8}$$

6.2 Compatibility of transport map and tracking error function

Let ψ and T be a tracking error function and transport map on the manifold Q, respectively. The pair (ψ, T) is compatible if, for all $(q, r) \in Q \times Q$,

$$d_2\psi(q,r) = -T(q,r)^* d_1\psi(q,r) \tag{6.9}$$

where $T(q,r)^* : T_q^*Q \to T_q^*Q$ is the dual of $T(q,r)$.

6.3 Control law design

Given a compatible pair (ψ, T), we define the function $V : TQ \times TQ \to \mathbb{R}$ by means of

$$V(q(t), r(t)) = \psi(q(t), r(t)) + \frac{1}{2}||\dot{e}(t)||^2_{\mathbb{G}} \tag{6.10}$$

Consider the C^∞ simple mechanical system (6.1). Let $q : \mathbb{R}_+ \to Q$, and $r : \mathbb{R}_+ \to Q$ be the controlled and reference trajectories respectively, let (ψ, T) be a compatible pair satisfying (6.9), then we define the control force $F = F_{FF} + F_{FB}$ as follows (Lewis and Bullo, 2005):

$$\begin{aligned} F_{FF}(t, \dot{q}(t)) &= -d_1\psi(q(t), r(t)) - k_v\dot{e}(t) \\ F_{FB}(t, \dot{q}(t)) &= \mathbb{G}(\overset{\mathbb{G}}{\nabla}_{\dot{q}} (T(q,r).\dot{r})) \end{aligned} \tag{6.11}$$

where $\overset{\mathbb{G}}{\nabla}_{\dot{q}} (T(q,r).\dot{r})$ denotes covariant derivative of $T(q,r).\dot{r}$ along the vector field $\dot{q}$. then the control law (6.11) asymptotically stabilizes the system (6.1) with (6.10) as

Lyapunov function.

In general for a given mechanical system, finding the tracking error function ψ and transport map T satisfying (6.9) will give a set of PDEs to solve that are not always trivial. In the next section we will show how reinforcement learning can be used to learn the control action F (6.11) by learning the error function ψ and transport map T.

6.4 Tracking using RL

We first parameterize tracking error function $\psi(q, r)$ and transport map $T(q, r)$ with finite number of basis functionals using Weierstrass higher order approximation Theorem

$$\begin{aligned} \hat{\psi}(q, r) &= \Sigma_1^n \psi_i \phi_i(q, r) && (6.12) \\ \hat{\tau}(q, r) &= \Sigma_1^m \tau_i \Phi_i(q, r) && (6.13) \end{aligned}$$

$n, m \in \mathbb{Z}_+$. Let $\Psi = [\psi_1 \cdots \psi_n]^\top$ and $T = [\tau_i \cdots \tau_m]^\top$. The approximated control law $\hat{\pi}(q, r, \dot{q}, \dot{r}, \Psi, T) = \hat{F}_{FF} + \hat{F}_{FB}$ now evaluates to

$$\begin{aligned} \hat{F}_{FF} &= -\Sigma_1^n \psi_i \frac{\partial \phi_i}{\partial q} - k_v \left(\dot{q} - \Sigma_1^m \tau_i \Phi_i \cdot \dot{r} \right) \\ \hat{F}_{FB} &= \mathbb{G} \Sigma_1^m \left(\tau_i \Phi_i \overset{\mathbb{G}}{\nabla}_{\dot{q}} \dot{r} + \dot{\Phi}_i \dot{r} \right) && (6.14) \end{aligned}$$

with $\hat{\psi}$, $\hat{\tau}$ satisfying Definition 3, Definition 4 respectively along with the compatibility condition (6.9).

Using this the update laws for Actor

$$u = \hat{\pi}(q, r, \dot{q}, \dot{r}, \Psi, T) + \Delta u. \quad (6.15)$$

in Actor-Critic Algorithm 2 can be can be rewritten as shown in Algorithm 5.

Algorithm 5 Actor critic for tracking

1: **procedure** ACTOR

2: **Actor**

3: $\Psi_{k+1} = \Psi_k + \alpha_a \delta_{k+1} \Delta \bar{u}_k \nabla_\Psi \zeta(\hat{\pi}(q_k, T_k, \Psi_k))$

4: $T_{k+1} = T_k + \alpha_a \delta_{k+1} \Delta \bar{u}_k \nabla_T \zeta(\hat{\pi}(q_k, T_k, \Psi_k))$

5: **Constraints:**

6: $\sum \tau_{i_{k+1}} = 1$ ($\dot{e} = 0$ for $x(t) = r(t)$)

7: $\frac{d\psi}{dr} + \tau \frac{d\psi}{dq} = 0$ Compatibility condition (6.9).

8: **end procedure**

6.5 Summary

In this section we formulated the tracking problem generically and nested Actor-Critic algorithm in feedforward proportional-derivative controller. The generic algorithm 5 is implemented for a Double Gimbal system in the next section and the results of the same are presented.

CHAPTER 7

DOUBLE GIMBAL

7.1 Problem Formulation

7.1.1 Double Gimbal Mechanism (DGM)

A gimbal is a pivoted support that allows the rotation of an object about a single axis (Hilkert (2008),Osborne *et al.* (2008)). DGM is used as an inertially stabilized platform to provide line-of-sight between platform payload sensor and target (Wang and Williams, 2008).

Consider the double gimbal (see Figure 7.1 (Kosaraju, 2013)) whose center of mass coincident with the geometric center. Let XYZ be defined as the earth inertial frame (E-frame) with the origin at O and $x_1y_1z_1$ be the body frame (B-frame) attached to the inner frame with the origin O_1 located at the intersection of the axes of rotation of the two gimbals. Both the coordinate frames follow the right-hand coordinate system. Denoting the rotation of the outer-frame about fixed Z-axis with angle θ (azimuth) and the rotation of the inner-frame about the body y-axis with an angle α (elevation).

Let J_1 denote the moment-of-inertia of the outer-gimbal about its axis of rotation and $I = diag(I_x, I_y, I_z) \in \mathbb{R}^{3\times3}$ is the inertia tensor of the inner-frame with respect to the B-frame.

7.2 Tracking with RL

The control objective is to track a given reference trajectory $r = (\theta_r(t), \alpha_r(t))$ using RL. The tracking error function $\psi(q, r)$ and transport map $T(q, r)$ are parameterized with finite number of basis functionals using Weierstrass higher order approximation

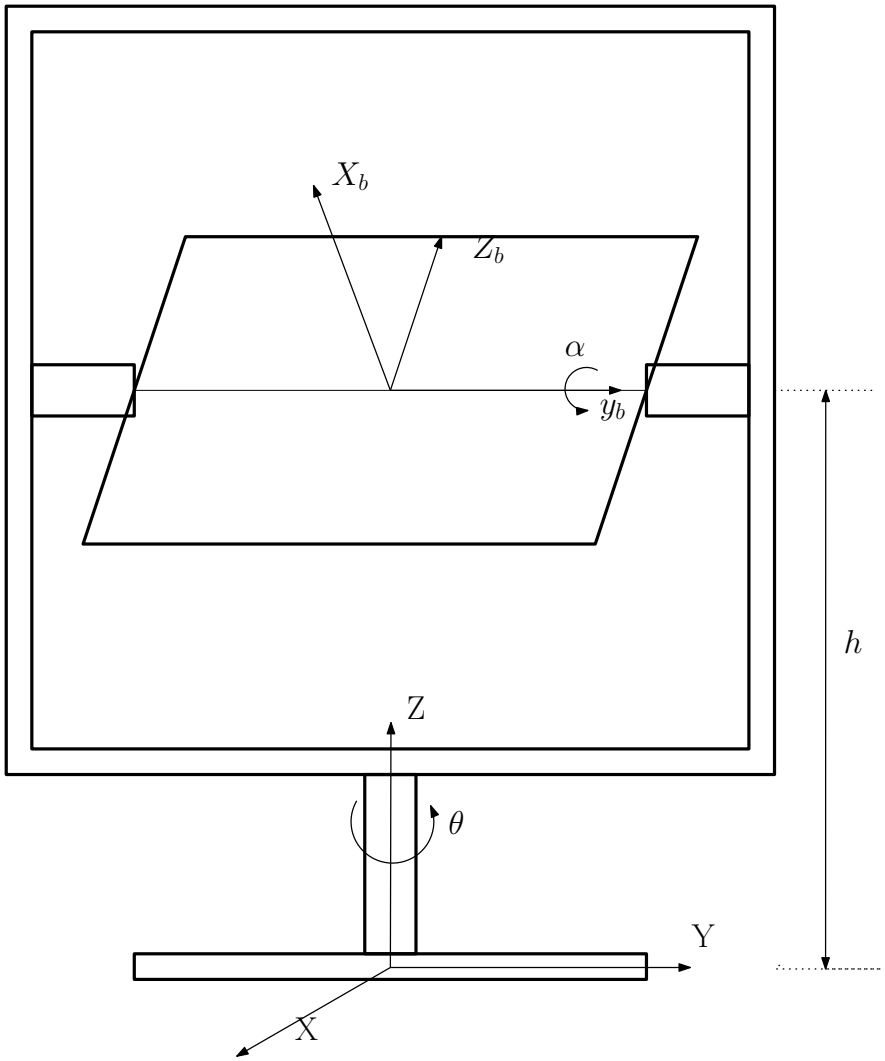


Figure 7.1: Two-axes double gimbal

Theorem

$$\hat{\psi}(q,r) = \Sigma_1^n \psi_i \phi_i(q,r) \tag{7.1}$$

$$\hat{\tau}(q,r) = \Sigma_1^m \tau_i \Phi_i(q,r) \tag{7.2}$$

$n, m \in \mathbb{Z}_+$. Let $\Psi = [\psi_1 \cdots \psi_n]^\top$ and $T = [\tau_i \cdots \tau_m]^\top$. The approximated control law $\hat{\pi}(q, r, \dot{q}, \dot{r}, \Psi, T) = \hat{F}_{FF} + \hat{F}_{FB}$ now evaluates to

$$\begin{aligned} \hat{F}_{FF} &= -\Sigma_1^n \psi_i \frac{\partial \phi_i}{\partial q} - k_v \left(\dot{q} - \Sigma_1^m \tau_i \Phi_i \cdot \dot{r}\right) \\ \hat{F}_{FB} &= \mathbb{G}\Sigma_1^m \left(\tau_i \Phi_i \overset{\mathbb{G}}{\nabla}_{\dot{q}} \dot{r} + \dot{\Phi}_i \dot{r}\right) \end{aligned} \tag{7.3}$$

with $\hat{\psi}$, $\hat{\tau}$ satisfying the compatibility condition (6.9).

Using this the update laws for Actor

$$u = \hat{\pi}(q, r, \dot{q}, \dot{r}, \Psi, T) + \Delta u. \tag{7.4}$$

in Actor-Critic Algorithm 2 can be can be rewritten as shown in Algorithm 6.

Algorithm 6 Actor critic for tracking

1: **procedure** ACTOR
2: **Actor**
3: $\Psi_{k+1} = \Psi_k + \alpha_a \delta_{k+1} \Delta \bar{u}_k \nabla_\Psi \zeta(\hat{\pi}(q_k, T_k, \Psi_k))$
4: $T_{k+1} = T_k + \alpha_a \delta_{k+1} \Delta \bar{u}_k \nabla_T \zeta(\hat{\pi}(q_k, T_k, \Psi_k))$
5: **Constraints:**
6: $\sum \tau_{i_{k+1}} = 1$ ($\dot{e} = 0$ for $x(t) = r(t)$)
7: $\frac{d\psi}{dr} + \tau \frac{d\psi}{dq} = 0$ Compatibility condition (6.9).
8: **end procedure**

To approximate critic $\hat{V}$ and the two actor functionals ψ and τ in Actor Critic Algorithm 6, we use fourier basis for function approximators

$$\begin{aligned} \hat{\tau} &= \tau_0 + \sum_{j=1}^{2} \sum_{i=1}^{3} \tau_{q(j)i} \cos[i(q(j) - r(j))] \\ \hat{\psi} &= \psi_0 + \sum_{j=1}^{2} \sum_{i=1}^{3} \psi_{q(j)i} \cos[i(q(j) - r(j))] \end{aligned}$$

where $q(1) = \theta$, $q(2) = \alpha$, $r(1) = \theta_r$ and $r(2) = \alpha_r$. The approximated critic function

Table 7.1: Parameter values used for Double Gimbal System

Parameter	Values
Learning rate of τ_0,ψ_0	10^{-1}
Learning rate of ψ_{θ_1}	10^{-3}
Learning rate of ψ_{α_1}	10^{-2}
Learning rate of $\tau_{\theta_1}, \tau_{\theta_2}, \tau_{\theta_3}$, $\tau_{\alpha_1} \cdots \tau_{\alpha_3}$, ψ_{θ_2}, ψ_{θ_3}, ψ_{α_2}, ψ_{α_3}	10^{-7}
Learning rate of k_v	1
Learning rate of k_p	1
Learning rate of ψ_2	10^{-6}
Learning rate of the critic	0.005
Discount factor	0.8
Trace decay rate	0.6

is chosen as

$$\hat{V} \quad = \quad \theta_0 + \sum_{i=1}^{20} \theta_i \cos[i(\theta - \theta_r)]$$

note that $(\theta_0, .., \theta_{20})$ are critic parameters and θ is the system's state variable (DGM's outer rotor angle).

The reward function is chosen as:

$$R(q, r) = cos(\theta - \theta_r) + cos(\alpha - \alpha_r) - 2$$

7.3 Simulation results

The reward function has four critical points. The system settles only at the maxima of the reward function (i.e. $q = r$) because of the exploration term in the control policy. The system parameters for simulation are taken as $J_1 = 10$ and $I = diag(2, 1, 7)$. In Actor critic Algorithm 6 the update parameters are given in Table 7.1. The control trajectory and desired trajectory are plotted on torus in Figure 7.2.

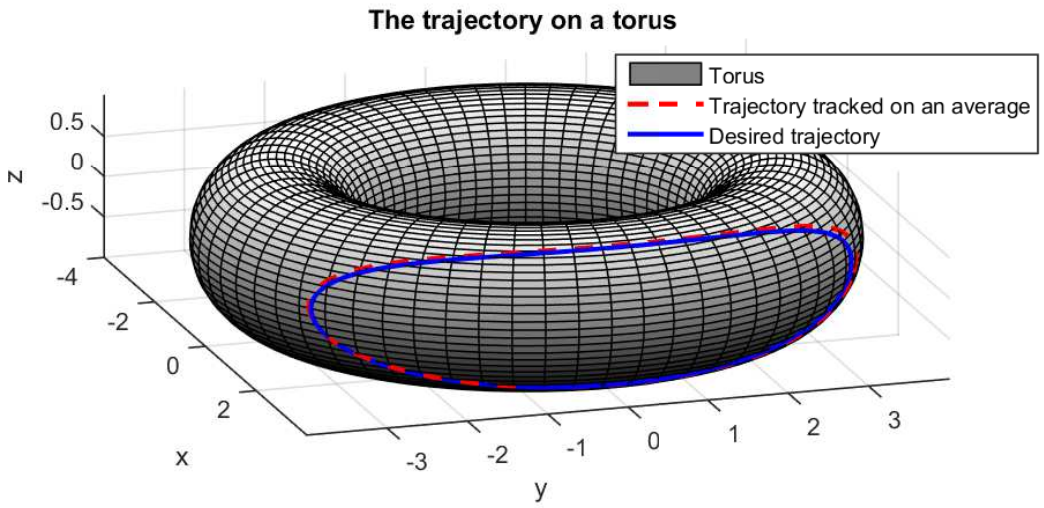


Figure 7.2: Controlled trajectory $q(t)$ (Red) and Desired Trajectory $r(t)$ (Blue) plotted on Torus $\mathbb{T}^2$ (configurational manifold)

The desired tracking error function (approximated) $\hat{\psi}$ learned is plotted in Figure 7.3 and its contours are plotted in Figure 7.4. Finally the reward function is plotted on Figure 7.5.

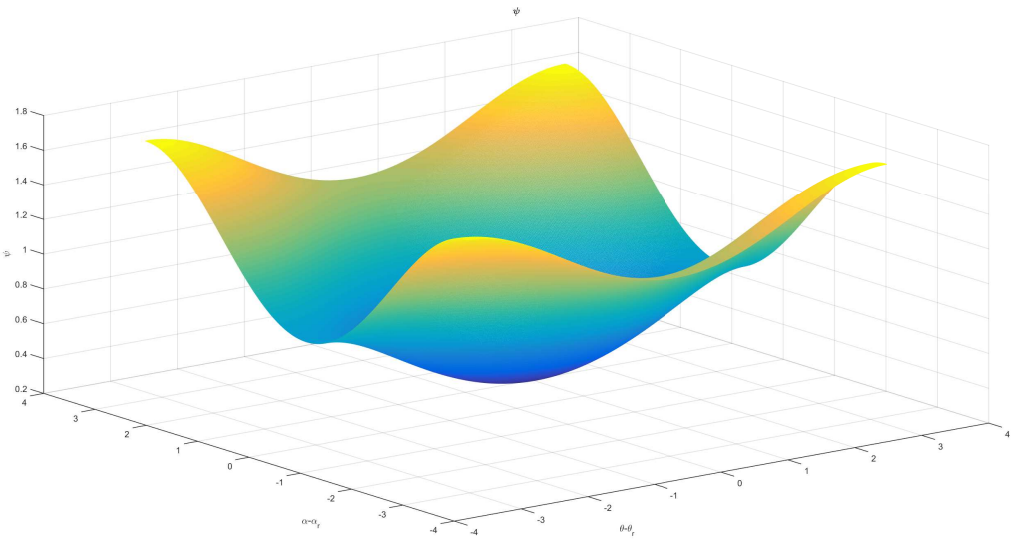


Figure 7.3: Approximated tracking error function $\hat{\psi}(q - r)$

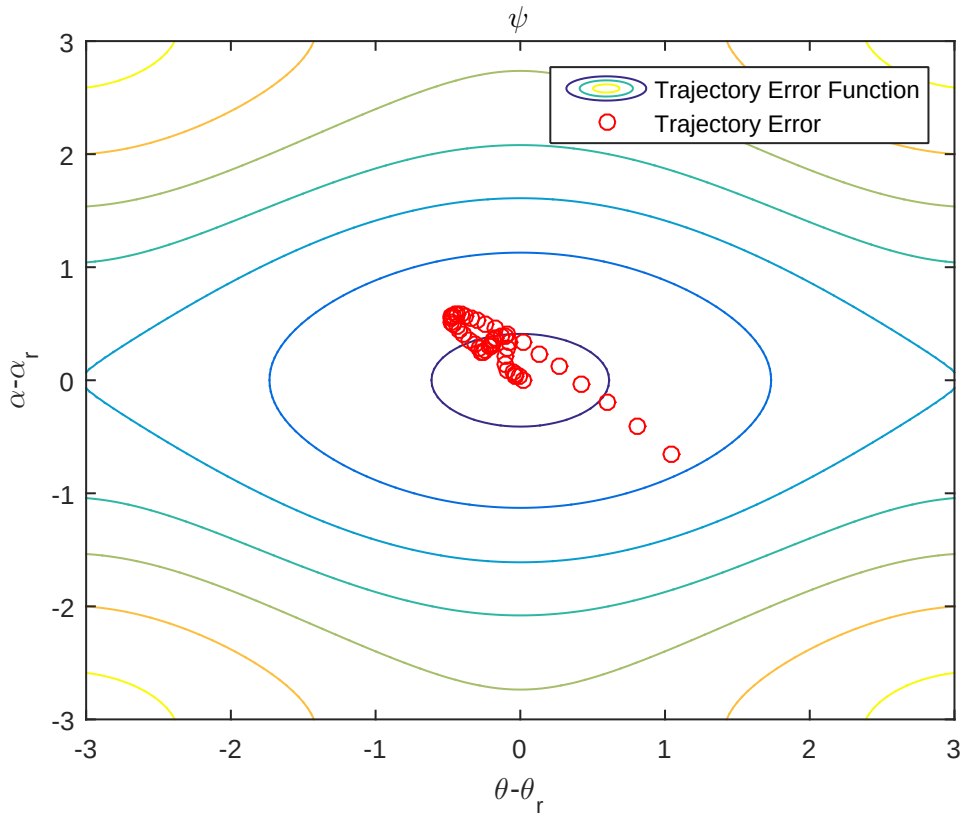


Figure 7.4: Plot of $q(t) - r(t)$ on the contour's of $\hat{\psi}(q - r)$

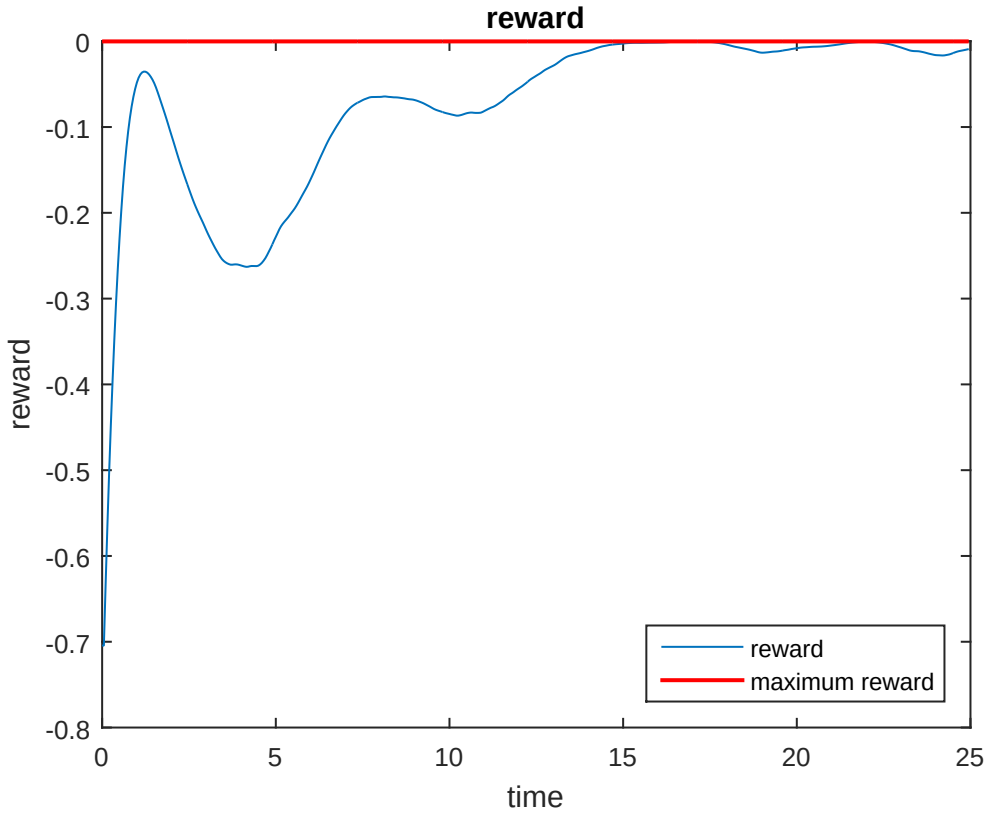


Figure 7.5: Reward function

7.4 Summary

We have presented a method to parametrise feedforward proportional-derivative control laws. The free parameter values were calculated using Actor-Critic method. By providing system knowledge, the convergence of the algorithm can be significantly improved, as the algorithm is computationally expensive.

CHAPTER 8

CONCLUSIONS

In this project, we have used Actor-Critic method to solve PDEs which arise in various control techniques. Actor -critic methods are used as the problems are in continuous spaces. This is shown through the example of Ball on a Beam system and 2D SpiderCrane for regulation problems, and through the example of Double Gimbal system for Tracking problems.

We have seen that for regulation problems, because of the structural properties of the PH form, the IDA-PBC formulation reduces to solving two PDEs, one for kinetic energy and the other for potential, and for tracking problems the PDEs arise form the compatibility condition. In most cases is non-trivial to solve these PDEs and in some cases it takes significant effort to reduce them to Ordinary differential equations (ODEs), under some assumptions. We have used Actor-Critic method to parametrise control techniques which are robust to non-linearities such as control saturation. For the purpose of exploration, noise is added. This also helps in pertubing the system incase it is stuck in a local minima. By providing system knowledge, the convergence of the algorithm can be significantly improved, as the algorithm is computationally expensive.

REFERENCES

1. **Acosta, J. A.**, **R. Ortega**, **A. Astolfi**, and **A. D. Mahindrakar** (2005). Interconnection and damping assignment passivity-based control of mechanical systems with underactuation degree one. *Automatic Control, IEEE Transactions on*, **50**(12), 1936–1955.

2. **Aeyels, D.**, **F. Lamnabhi-Lagarrigue**, and **A. van der Schaft**, *Stability and stabilization of nonlinear systems*, volume 246. Springer, 2008.

3. **Arnol'd, V. I.**, *Geometrical methods in the theory of ordinary differential equations*, volume 250. Springer Science & Business Media, 2012.

4. **Auckly, D.** and **L. Kapitanski** (2002). On the λ-equations for matching control laws. *SIAM Journal on Control and Optimization*, **41**(5), 1372–1388.

5. **Barto, A. G.**, **R. S. Sutton**, and **C. W. Anderson** (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *Systems, Man and Cybernetics, IEEE Transactions on*, (5), 834–846.

6. **Buccieri, D.**, **P. Mullhaupt**, and **D. Bonvin**, Spidercrane: Model and properties of a fast weight-handling equipment. *In IFAC World Congress*. 2005.

7. **Byrnes, C. I.**, **A. Isidori**, and **J. C. Willems** (1991). Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems. *Automatic Control, IEEE Transactions on*, **36**(11), 1228–1240.

8. **Fang, Y.**, **W. Dixon**, **D. Dawson**, and **E. Zergeroglu** (2003). Nonlinear coupling control laws for an underactuated overhead crane system. *Mechatronics, IEEE/ASME Transactions on*, **8**(3), 418–423.

9. **Fossen, T. I.**, *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.

10. **Fujimoto, K.** and **T. Sugie** (2001). Canonical transformation and stabilization of generalized hamiltonian systems. *Systems & Control Letters*, **42**(3), 217–227.

11. **Gómez-Estern, F.**, **R. Ortega**, **F. R. Rubio**, and **J. Aracil**, Stabilization of a class of underactuated mechanical systems via total energy shaping. *In Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2. IEEE, 2001.

12. **Gómez-Estern, F.** and **A. J. Van der Schaft** (2004). Physical damping in ida-pbc controlled underactuated mechanical systems. *European Journal of Control*, **10**(5), 451–468.

13. **Hilkert, J.** (2008). Inertially stabilized platform technology concepts and principles. *Control Systems, IEEE*, **28**(1), 26–46.

14. **Kazi, F.**, **R. N. Banavar**, **P. Mullhaupt**, and **D. Bonvin**, Stabilization of a 2d-spidercrane mechanism using damping assignment passivity-based control. *In Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea*. 2008.

15. **Kosaraju, K. C.** (2013). Stability and tracking of double gimbal system in geometric framework. *Master Thesis, IITM*.

16. **Lewis, A.** and **F. Bullo** (2005). Geometric control of mechanical systems.

17. **Lewis, F. L.** and **D. Vrabie** (2009). Reinforcement learning and adaptive dynamic programming for feedback control. *Circuits and Systems Magazine, IEEE*, **9**(3), 32–50.

18. **Muralidharan, V.**, **S. Anantharaman**, and **A. D. Mahindrakar** (2010). Asymptotic stabilisation of the ball and beam system: design of energy-based control law and experimental results. *International Journal of Control*, **83**(6), 1193–1198.

19. **Murray, R. M.**, **Z. Li**, **S. S. Sastry**, and **S. S. Sastry**, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

20. **Nijmeijer, H.** and **A. Van der Schaft**, *Nonlinear dynamical control systems*. Springer Science & Business Media, 2013.

21. **Ortega, R.**, **M. W. Spong**, **F. Gómez-Estern**, and **G. Blankenstein** (2002). Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *Automatic Control, IEEE Transactions on*, **47**(8), 1218–1233.

22. **Ortega, R.**, **A. J. Van der Schaft**, **I. Mareels**, and **B. Maschke** (2001). Putting energy back in control. *Control Systems, IEEE*, **21**(2), 18–33.

23. **Osborne, J.**, **G. Hicks**, and **R. Fuentes** (2008). Global analysis of the double-gimbal mechanism. *Control Systems, IEEE*, **28**(4), 44–64.

24. **Slotine, J.-J. E.** and **W. Li** (1989). Composite adaptive control of robot manipulators. *Automatica*, **25**(4), 509–519.

25. **Spong, M. W.**, Underactuated mechanical systems. *In Control problems in robotics and automation*. Springer, 1998, 135–150.

26. **Sprangers, O.**, **R. Babuska**, **S. P. Nageshrao**, and **G. A. Lopes** (2015). Reinforcement learning for port-hamiltonian systems. *Cybernetics, IEEE Transactions on*, **45**(5), 1017–1027.

27. **Sutton, R. S.** and **A. G. Barto**, *Reinforcement learning: An introduction*. MIT press, 1998.

28. **Takegaki, M.** and **S. Arimoto** (1981). A new feedback method for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, **103**(2), 119–125.

29. **Van der Schaft, A.**, *L2-gain and passivity techniques in nonlinear control*. Springer Science & Business Media, 2012.

30. **Viola, G.** (2008). *Control of underactuated mechanical systems via passivity-based and geometric techniques*. Ph.D. thesis, PhD thesis, Universita degli Studi di Roma âĂIJTor Vergata.

31. **Viola, G.**, **R. Ortega**, **R. Banavar**, **J. Á. Acosta**, and **A. Astolfi** (2007). Total energy shaping control of mechanical systems: simplifying the matching equations via coordinate changes. *Automatic Control, IEEE Transactions on*, **52**(6), 1093–1099.

32. **Wang, H. G.** and **T. C. Williams** (2008). Strategic inertial navigation systems-high-accuracy inertially stabilized platforms for hostile environments. *Control Systems, IEEE*, **28**(1), 65–85.

33. **Wen, J. T.** and **D. S. Bayard** (1988). New class of control laws for robotic manipulators part 1. non–adaptive case. *International Journal of Control*, **47**(5), 1361–1385.

34. **Wen, J.-Y.** and **K. Kreutz-Delgado** (1991). The attitude control problem. *Automatic Control, IEEE Transactions on*, **36**(10), 1148–1162.

LIST OF PAPERS BASED ON THESIS

SUBMITTED:

1. Anjali Ramesh, Krishna Chaitanya Kosaraju, Ramkrishna Pasumarthy and Arun D. Mahindrakar Regulation and Tracking using Actor-Critic Methods *In Proceedings of 55th IEEE Conference on Decision and Control*, Las Vegas, USA, 2016.