Comparative Analysis of Deep Reinforcement Learning Algorithms for Vibration Suppression in Rotary Flexible Link Systems

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering

by

SOHAN DAS

20BEE0330

Under the guidance of

Dr. Vinodh Kumar E

School of Electrical Engineering

VIT, Vellore.

April 2024

DECLARATION

I hereby declare that the thesis entitled **Comparative Analysis of Reinforcement Learning Algorithms for Vibration Suppression in Rotary Flexible Link Systems** submitted by me, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering* to VIT University is a record of bonafide work carried out by me under the supervision of **Dr. Vinodh Kumar E**, School of Electrical Engineering, VIT University, Vellore.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore **Sohan Das**

Date:09/05/2024 **Signature of the Candidate**

CERTIFICATE

This is to certify that the thesis entitled **Comparative Analysis of Reinforcement Learning Algorithms for Vibration Suppression in Rotary Flexible Link Systems** submitted by **SOHAN DAS(20BEE0330)**, School or Centre, VIT, Vellore, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering* , is a record of bonafide work carried out by him/her under my supervision during the period, 03.01.2024 to 09.05.2024, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

E. Vinth

Place: Vellore

Date: 09/05/2024

Signature of the Guide
(Dr.Vinodh Kumar E)

The Thesis is satisfactory

Approved by

HOD

Department of EEE

Dean,SELECT

Acknowledgement

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my supervisor **Dr.Vinodh Kumar E**, Professor Grade 1, School of Electrical Engineering, VIT University, without his motivation and continuous encouragement, this project work would not have been successfully completed.

I express my sincere thanks to **Dr.Mathew M.Noel**, Dean, School of Electrical Engineering, VIT University for his kind words of support and encouragement. I like to acknowledge the support rendered by **my colleagues** in several ways throughout my project work.

I am grateful to the Chancellor of VIT University, **Dr. G.Viswanathan**, the Vice Presidents, and the Vice Chancellor for motivating me to carry out research in VIT University and also for providing me with infrastructural facilities and many other resources needed for my project.

I wish to extend my profound sense of gratitude to **my parents** for all the sacrifices they made during my research and also providing me with moral support and encouragement whenever required.

Place: Vellore **Sohan Das**

Date: 09/05/2024 **Student Name**

Executive Summary

This thesis presents a comparative analysis of three different advanced reinforcement learning algorithms: Deep Deterministic Policy gradient (DDPG), Soft-Actor Critic Method (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3) for vibration suppression in rotary flexible link system. These system pose a unique challenge due to their dynamic value, which is significant in various industrial applications and robotics.

The core objective of this research is to investigate the suppression of vibration in such systems and to identify the most effective control method for vibration mitigation. Through a series of simulations and experimental evaluations, this study assesses the DDPG, SAC and TD3 algorithms based on their performance in stabilizing a rotary flexible link system, focusing on their ability to reduce angular deviations and control vibrations.

Utilizing the Quanser SRV02 servo system augmented with a flex-gauge module to simulate the dynamics of a rotary flexible link, the study employs a methodical approach to implement and compare the three algorithms. Each algorithm's performance was assessed based on keys metrics such as vibration amplitude reduction, response time and energy efficiency under varying conditions of disturbances.

The comparative analysis reveals distinct advantages and limitations of each algorithm, including differences in convergence speed, computational demand, and resilience to external disturbances. SAC shows promising results in terms of sample efficiency, while TD3 distinguishes itself with its stability during training.

DDPG, despite being an earlier algorithm, provides a foundational comparison for the performance of newer techniques. The insights gained from this study contribute to the broader understanding of reinforcement learning applications in control systems and set a foundation for future advancements in the field.

Contents

LIST OF FIGURES

Figures and Plots

Agent Training Plot

Code Images

LIST OF TABLES

LIST OF TERMS AND ABBREVIATIONS

DDPG: Deep Deterministic Policy Gradient
SAC: Soft-Actor Critic
TD3: Twin Delayed Deep Deterministic Policy Gradient
PID: Proportional-Integral-Derivative
LQR: Linear Quadratic Controller

Chapter 1

Introduction

1.1 Objective

The primary objective of this project is to conduct a comprehensive analysis of advanced reinforcement learning algorithms, like Deep Deterministic Policy Gradient (DDPG), Soft-Actor Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3), for suppressing vibration in a rotary flexible link system.

This investigation enables to observe the efficiency, robustness and optimization of control responses facilitated by each algorithm, with a focus particularly on the ability to stabilize the system and minimize the vibrations under various uncertainties and disturbances inherent to system dynamics. This study also focuses to refine the vibration control techniques and implement reinforcement learning strategies to address a longstanding challenge in the field. Through theoretical analysis and simulation-based testing, the research aims to bridge a gap between the theoretical understanding and practical application, offering valuable insights into deployment of these algorithms in real-world applications where precise control is important, like in robotics arms and aerospace mechanisms.

The accomplishment of these goals leads the project to contribute significantly to the advancement of machine learning based control solutions, enhancing both reliability and performance of the complex mechanical control systems.

1.2 Motivation

The core of mechanical systems, especially those involving rotary flexible link mechanisms, play a pivotal role across a broad spectrum of industrial and technological applications, from robotics to aerospace engineering. These systems, however, are prone to vibrations due to flexibility and dynamics, which can significantly hinder performance, reliability and longevity. So, the suppression of vibrations with these systems emerges as a critical challenge, necessitating control strategies that can navigate the complexities of their behavior.

1.3 Background

The control of rotary flexible link systems represents a huge challenge within the domain of mechanical and control systems engineering, accentuated by the inherent trade-offs system flexibility and propensity for vibrational disturbances. Such systems are pivotal in many applications, ranging from industrial robot to satellite antenna positioning, where precision and stability are paramount. The core challenge lies in the system's dynamic complexity where flexibility introduces resonant modes that can be easily excited, leading to detrimental oscillations.

The conventional control systems like the Proportional-Integral-Derivative (PID) and Linear Quadratic Controller (LQR), have been used to suppress these vibrations. While effective within Linear regimes and under known disturbances, their effect diminishes in non-linear dynamics, modelling inaccuracies and unforeseen environmental interactions.

The advent of Reinforcement Learning has heralded a new era of control systems, introducing the potential to learn optimal policies from interaction with the environment. RL's model free nature is advantageous for systems where accurate modelling is challenging or impractical. Within this realm, algorithms like DDPG, TD3 and SAC have emerged, each designed to address the nuances of continuous control task with different strategies.

DDPG enables the integration of deep learning with deterministic policy gradients, offering an opportunity to understand the potential of RL in controlling such systems. However, its susceptibility to overestimation bias and sensitivity to hyperparameters has led to the development of TD3. TD3 introduces critical improvements, including twin critics and delayed policy updates, to enhance stability and performance. SAC incorporates entropy maximization into the optimization process, promoting exploration and robustness in policy learning.

The literature reveals a growing interest in applying these advanced RL algorithms to mechanical control problems, yet there remains a gap in systematic comparative studies, particularly in the context of vibration suppression in rotary flexible link systems. This research aims to fill this gap, providing insights into the relative performance, strengths and limitations of DDPG, TD3 and SAC in a domain where precise control is not just optimal but essential for operational efficacy.

- **Control Law for DDPG:**

  $u(t) = \mu(\theta(t), \dot{\theta}(t), \phi) + \mathrm{N(t)}$

  where u(t) denotes the control input, μ denotes deterministic policy function parameterized by ϕ and N(t) represents exploratory noise.

- **Value Function for SAC:**

  $V^{\pi}(\mathrm{s}_t) = E[\sum_{k=t}^{T} \gamma^{k-t} r(s_k, a_k)|s_t]$

  where $V^{\pi}(s_t)$ is the value function under policy π at state (s_t), $r(s_k, a_k)$ denotes the reward function obtained after the action a_k in state s_k and γ, which represents the discount factor.

- **TD3 Updates:**

  $Q(s, a) = r(s, a) + \gamma E[Q'(s', \mu'(s', \mu'))]$

  where $Q(s, a)$ denotes the action-value function for the present state-action pair, Q' and μ' are the target network for the action-value function and policy respectively and s' is the next state.

Chapter 2

Project Description and Goals

2.1 Literature Review

Table 2.1: Literature Summary

Title	**Authors and Publication**	**Highlights**
Deep Reinforcement Learning Based Trajectory Planning Under Uncertain Conditions	Lienhung Chen, Zhongliang Jiang, Long Cheng, Alois C. Knoll and Mingchuan Zhou	This paper introduces an advanced deep reinforcement learning approach for creating collision-avoidance paths in unpredictable settings, ensuring a harmonious interaction between humans and machines. This paper incorporated the Deep Deterministic Policy Gradient (DDPG) and Soft-Actor Critic (SAC) models.
An Adaptive Neuro-Fuzzy Controller for Vibration Suppression of Flexible Structures	Adam Genno, Wilson Wang	This paper introduces an adaptive Neuro-fuzzy (NF) Controller designed to minimize vibrations in flexible structures. This paper also proposes a novel training approach utilizing Bisection Particle Swarm Optimization (BPSO) to iteratively refine the Neuro-Fuzzy (NF) Controller parameters.

Title	Authors and Publication	Highlights
Quanser Technical Resources	Quanser	To understand the mechanism of Quanser SRV02 and gain technical resource knowledge.
Flexible Link Robots with combined trajectory tracking and vibration control	O.A. Garcia-Perez, G. Silva-Navarro, J.F. Peza-Solis	This paper provides details of applying Multiple Positive Position Feedback to deal with asymptotic trajectory tracking and injection of active damping for a flexible link robot system. Validation is done using the Finite Element method and the control strategy is divided into two segments: application of Sliding Mode Control for regulation and PID for trajectory tracking, combined with a Multiple Positive Position Feedback for active control of vibration and minimization of residual vibrations.

Reinforcement Learning and optimal adaptive control: An overview and implementation examples	Said G. Khan, Guido Herrmann, Frank L. Lewis, Tony Pipe, Chris Melhuish	This paper provides a summary of the key points of reinforcement learning and literature of optimal adaptive control, focusing on the application in robotics. The controller developed incorporates an innovative Adaptive Dynamic Programming (ADP) approach within reinforcement learning to dynamically formulate and implement an optimal online policy.

Title	Authors and Publication	Highlights
Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation	Aveen Dayal, Linga Reddy Cenkeramaddi, Ajit Jha	The reward function configuration is crucial for the agent's learning process. This article proposes a new reward criterion which is developed using diverse reward functions. Training of Deep Q-Network Agent has been using a point-goal navigation assignment with different reward structures. Furthermore, a comparison of these reward structures against a standard benchmark has too been done.

Reinforcement learning vibration control of a multi-flexible link beam coupling system	Zhi-cheng Qiu, Yang Yang, Xian-min Zhang	A Reinforcement learning-based active vibration control algorithm is applied to mitigate the coupling vibrations in a system with multi-flexible beams. Piezoelectric Sensor/Actuator is used to identify and attenuate the vibration signals. The Finite Element Method (FEM) is utilized to develop the system dynamics model and Multi-Agent TD3 (MATD3) algorithm is designed to improve the training of the Reinforcement Learning based vibration controller, facilitating interaction within a simulated environment.

Title	**Authors and Publication**	**Highlights**
A Vibration Control Method for Hybrid-Structured Flexible Manipulator Based on Sliding Mode Control and Reinforcement Learning	Teng Long; En Li, Yunqing Hu, Lei Yang, Junfeng Fan, Zize Liang, Rui Guo	This article demonstrates the decomposition of the tip-state signal of a hybrid-designed flexible manipulator into elastic vibration signal and tip equilibrium position signal component. An advanced nominal model-based sliding mode controller (NMBSMC) serves as the primary controller, responsible for generating the driving torque. Additionally, an actor-critic based reinforcement learning controller (ACBRLC) acts as a supplementary controller, providing small torque component for better compensation.

Reinforcement Learning based adaptive PID Controller Design for Control of Linear/nonlinear unstable processes	T.Shuprajhaa, Shiva Kanth Sujit, K. Srinivasan	This study aims to develop a generic, data driven approach by utilizing a modified version of Proximal Policy Optimization (PPO) for reinforcement learning to create an adaptive PID controller (RL-PID). This controller is designed specifically for managing open-loop unstable process.In this setup, the reinforcement learning agent functions as a supervisor, exploring and identifying optimal adjustments for the PID controller's gains to achieve the desired servo and regulatory outcomes.

Title	**Authors and Publication**	**Highlights**
Research on vibration suppression and trajectory tracking control strategy of a flexible link manipulator	Mingming Shi, Yong Cheng, Bao Rong c, Wenlong Zhao, Zhixin Yao, Chao Yu	This paper presents a control methodology that incorporates dynamic trajectory planning and a fuzzy self-tunning PD (FST PD) controller. This approach is designed to improve the suppression of residual vibrations and improve trajectory tracking of a flexible link manipulator.
Reinforcement Learning and optimal adaptive control: An overview and implementation examples	Said G. Khan, Guido Herrmann, Frank L. Lewis, Tony Pipe, Chris Melhuish	This paper gives a summary of the key points of reinforcement learning and literature of optimal adaptive control, centering on the application in robotics. The controller developed provides an innovative Adaptive Dynamic Programming (ADP) approach within reinforcement learning to dynamically define and implement an optimal online policy.

Title	**Authors and Publication**	**Highlights**
A deep reinforcement learning-based optimization method for vibration suppression of articulated robot	Tie Zhang, Hubo Chu, Yanbiao Zou, Tao Liu	A novel optimization method for input shaping called Reinforcement Learning-based Input Shaping (RLIS) is introduced, employing deep reinforcement learning. This approach utilizes an arbitary-time delay as its show and development and enhancement of the RLIS optimization strategy.
Vibration and Position Control of a two-link flexible manipulator using reinforcement learning	Minoru Sasaki, Joseph Muguro , Fumiya Kitano, Waweru Njeri , Daiki Maeno and Kojiro Matsushita	This paper presents a new strategy for managing the vibrations and motions of a two-link manipulator through the use of reinforcement learning. The method utilizes Trust Region Policy Optimization to successfully train the manipulator's end effector, empowering it to accomplish its target position whereas reducing at the link's root.

Title	Authors and Publication	Highlights
Iterative-Learning-Based Motion Planning and Position Control of a Single-Link Flexible Manipulator With Vibration Sensor Hysteresis	Qing-Xin Meng , Mingliang Zhu, Xuzhi Lai, Yawu Wang, Min Wu	Vibration sensors often display complex nonlinear hysteresis in their input-output relationship due to the viscoelastic nature of their materials. This article considers the hysteresis characteristics of the vibration sensor installed on a practical planar single-link manipulator and explores solutions to address the position control problem of this system.

Vibration Suppression Method for a Two-Link Flexible Manipulator Based on Adaptive Iterative Learning Algorithm	Chuanjie Liu, Wenbin Gao, Peng Gao	This article explores vibration control issues in a two-link flexible manipulator system, addressing challenges posed by system parametric uncertainties and external disturbances.The study applies Hamilton's principle and the partial differential equations to develop a dynamic model of the manipulator, treating it as an Euler-Bernoulli beam. To enhance vibration suppression and improve the precision in angle tracking, the paper proposes an adaptive iterative torque control approach that incorporates a Proportional-Derivative (PD) feedback structure and an iterative system.

2.2 Project Description

2.2.1 Software

This research Project is conducted within a comprehensive software environment designed to support the implementation, simulation and analysis of reinforcement learning algorithms applied to control systems.

- **MATLAB and Simulink**: The core of the simulation and model development was carried out using MATLAB (version 2022) and Simulink, renowned for their extensive toolboxes and blocks specifically designed for system dynamics modelling, signal processing and control system design. Simulink's graphical interface enables the modelling of rotary flexible link system and the integration of control algorithms.

- **Reinforcement Learning Toolbox**: MATLAB's Reinforcement Learning Toolbox is used to implement the DDPG, TD3 and SAC algorithms. This toolbox provides customizable agents, training environments and a suite of tools to train and evaluate policies. The toolbox's compatibility with Simulink models allows for direct application of RL agents to the simulated environment.

- **Deep Learning Toolbox**: To support the neural network architecture underlying the actor and critic models in DDPG, TD3 and SAC, MATLAB's Deep Learning Toolbox is utilized. This toolbox offers a comprehensive framework for designing, training and deep neural networks, with extensive support for convolutional and recurrent networks.

- **Custom Scripts and Functions**: Custom MATLAB scripts and functions were developed to augment the capabilities of the toolboxes, including the generation of noise for exploration, data logging for analysis and post-processing scripts for visualizing the results.

- **Configuring Hyperparameters**: The DDPG, TD3 and SAC algorithms were configured with specific hyperparameters tailored to the dynamics of rotary flexible link system. Hyperparameters such as learning rate, discount factor and exploration noise are optimized by numerous trials to balance learning efficiency and performance.

2.2.2 Hardware

The experimental hardware system is Quanser SRV02 rotary servo base unit, augmented with a flex-gauge module, enabling to explore the motion of the rotary flexible link system and evaluating vibration suppression strategies. This hardware setup offers a comprehensive environment to implement and test the effectiveness of reinforcement learning algorithms including DDPG, TD3 and SAC.

- **Quanser SRV02 Rotary Servo Base Unit**: The SRV02 provides a high-fidelity, low friction platform with a servo motor for precise angular control, ideal for simulating and analyzing behavior of rotary systems under various control strategies.

- **Flexgage Module**: Attached to SRV02, the flexgage module introduces a flexible link in the system, creating a more dynamic complex environment which simulates the real-world challenges in controlling flexible link structures. This addition enables us to investigate vibration and efficacy of various suppression techniques.

- **Data Acquisition and Interface Hardware**: The system is integrated with a high-speed data acquisition system capable of capturing detailed measurements of angular position, velocity and link deflection. An interface module facilitates communication between the computational models developed in MATLAB/Simulink and the physical SRV02 system, ensuring seamless execution of control commands and real-time feedback.

- **Computational Resources**: The experiments and simulations are supported by a dedicated computational workstation equipped with a powerful GPU, enabling efficient training and evaluation of complex reinforcement learning models. This setup is critical for handling the computationally intensive task associated with deep learning and real-time system control.

ID	Component
1	SRV02 Plant
2	FLEXGAGE Module
3	FLEXGAGE Link
4	Strain Gage
5	Strain Gage Circuit
6	Thumbscrews
7	Sensor Connector
8	OFFSET Potentiometer
9	GAIN Potentiometer

Fig. 2.2.2(a) Rotary Flexible Link Components

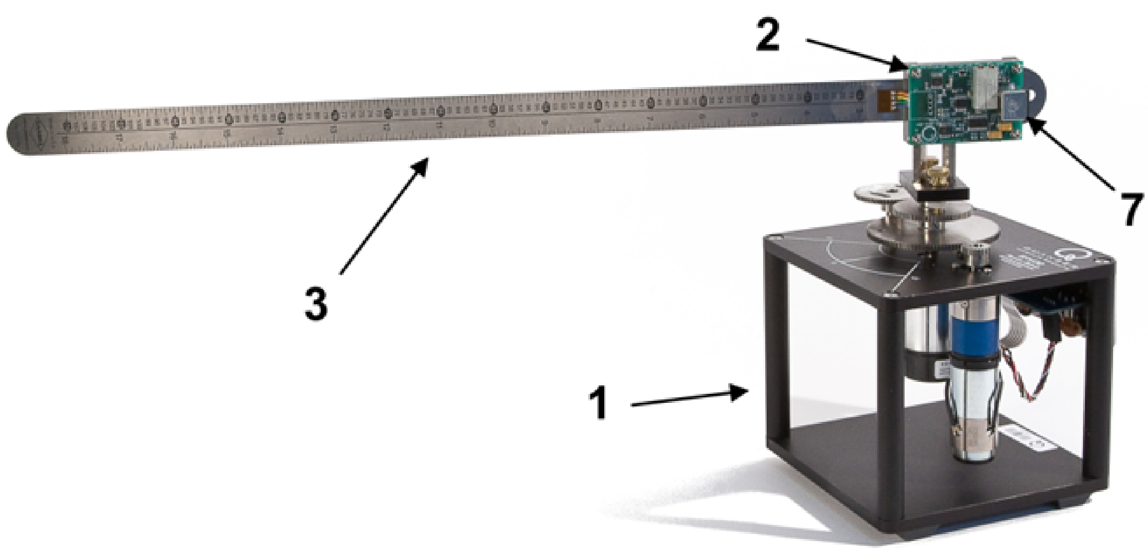


Fig 2.2.2(b) SRV02 coupled with Flexgage

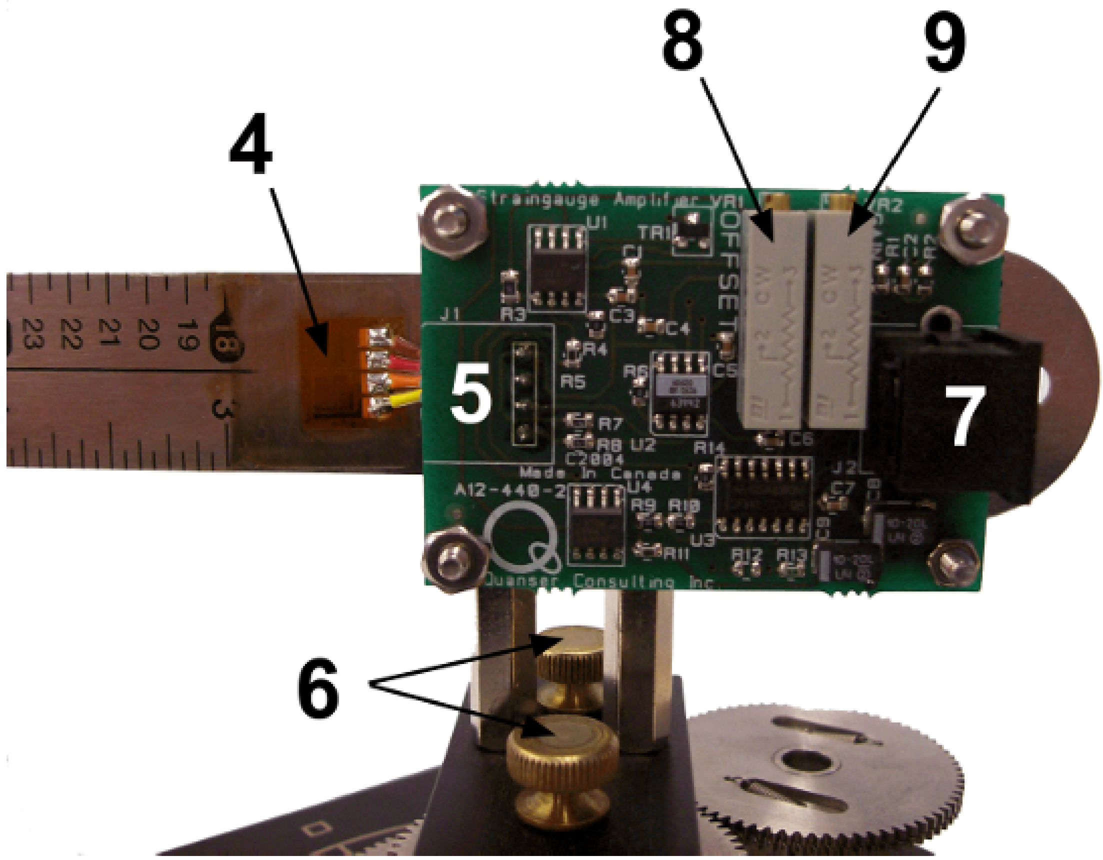


Fig 2.2.2(c) Closeup of Strain Gauge

2.3 Goals

The primary goal of this research is to meticulously evaluate and compare the effectiveness of three advanced reinforcement learning algorithms: DDPG, SAC and TD3 in reducing vibrations within a rotary flexible link system. This study aims to:

- Establish a set of benchmarks for each algorithm's ability to suppress vibrations and control angular positioning with high precision.

- Determine the algorithms adaptability and responsiveness to varying degrees of system complexity and external disturbances.

- Assess the computational efficiency and resource utilization of each algorithm to understand their scalability for real-world applications.
- Contribute to the body of knowledge in control system engineering by providing actionable insights on the selection and implementation of reinforcement learning algorithms for dynamic system stabilization tasks.

Chapter 3

Technical Specification

3.1 Software Specification

- **MATLAB**: The majority of the research has been conducted in MATLAB (version 2022), a high level programming environment and numerical computing platform. MATLAB's extensive suite of toolboxes facilitates the development, testing and analysis of reinforcement learning algorithms.

- **Simulink**: Integrated with MATLAB environment, is utilized to create dynamic and embedded systems simulation, particularly for modeling the rotary flexible link system and implementing real-time control.

Key Libraries and Toolboxes:

- **Reinforcement Learning Toolbox**: This Toolbox provided the necessary framework for designing, training and evaluating reinforcement learning agents, including predefined environments and custom setup required for DDPG, TD3 and SAC algorithms.

- **Deep Learning Toolbox**: Essential for constructing and training the deep neural networks that form the backbone of the actor and critic models in DDPG, TD3 and SAC. This Toolbox supports a wide range of deep learning architecture and functions.

- **Control System Toolbox**: Used for classical control design and analysis, this toolbox helps in preliminary assessments and comparison with traditional control strategies.

Configuration and custom Scripts:

- Custom MATLAB Scripts are developed to automate the training process, data collection and performance evaluation of each algorithm. These scripts include functions for initializing the learning environment, adjusting algorithmic parameters and logging results for subsequent analysis.
- The specific hyperparameters for each reinforcement learning algorithm, such as learning rates, discount factors and exploration noise, were meticulously tuned based on preliminary experiments and are detailed within the supplementary materials.
- The simulated environment for the rotary flexible link system is customized to reflect realistic dynamics, including parameters for link flexibility, motor characteristics and disturbance profiles.

Computational Hardware:

- The computational demands of training deep reinforcement learning models were met using a workstation equipped with NVIDIA GeForce, ensuring efficient processing and model training times.
- Integration with physical hardware, when applicable, was facilitated through MATLAB and Simulink's support for external interfaces, allowing for real-time data acquisition and control.

Version Control and Data Management:

- A version control system is employed to manage the development of scripts and models, ensuring the integrity and reproducibility of the research code.
- Data generated during the simulations, including training metrics and control performance results, were stored and managed using MATLAB's data handling capabilities with detailed logs available for analysis.

3.2 Hardware Specification

Symbol	Description	Value	Unit
	Module Dimensions	48 x 2	cm^2
L_l	Flexible Link Length (strain gage to tip)	41.9	cm
m_l	Flexible Link Mass	0.065	kg
J_l	Flexible Link Moment of Intertia	0.0038	$kg.m^2$
	Strain gage bias power	± 12	V
	Strain gage measurement range	± 5	V
	Strain gage calibration gain	1/16.5	rad/V

Fig 3.2(a) Flexible Link Specification

Specifications:

- **System Overview:** The Quanser SRV02 rotary flexible joint module serves as a platform for studying and implementing advance control techniques, including vibration control of a rotary flexible link.

Core Module:

- **Model:** Quanser SRV02 rotary servo base unit
- **Material:** High-quality aluminum Alloy for maximum rigidity and minimal flex
- **Rotation:** Equipped with a DC motor allowing a full 360-degree continuous rotation
- **Encoder:** High-resolution optical encoder with a resolution of 2048 counts per revolution for precise position measurement.

Flexible Link Attachment:

- **Material:** The Flexible Link is made from a lightweight, durable composite material.

- **Dimensions:** Given in the Table.
- **Mounting:** Securely fastened at the base with quick-release mechanisms for easy installation and removal.

  Sensors:

- **Rotary Encoder:** Mounted directly on the motor shaft for feedback on the base position.
- **Flex Sensor:** Attached along the length of the flexible link to measure deflection.

  Actuator:

- **Motor:** A precision servo motor capable of fine control and quick response to control signals.
- **Power:** Operates on a nominal voltage with safety features to protect against overcurrent conditions.

  Control Hardware:

- **Interface:** USB-based data acquisition device interfacing with MATLAB/Simulink for real-time control implementation.
- **Processor Compatibility:** Compatible with Quanser' s QPIDe data acquisition card for high-fidelity data acquisition and control.

  Power Supply:

- **Voltage:** Stabilize power supply with a configurable voltage output for different experimental needs.
- **Current:** Capable of delivering sufficient current for peak motor performance without risking overcurrent damage.

  Safety and Compliance:

- **Emergency Stop:** An integrated emergency stop button to immediately cut power to the motor in case of malfunction.

- **Compliance:** Meets international safety standards for educational laboratory equipment.

- **System Overview:** The Quanser SRV02 rotary flexible joint module serves as a platform for studying and implementing advance control techniques, including vibration control of a rotary flexible link.

  Core Module:

- **Model:** Quanser SRV02 rotary servo base unit

- **Material:** High-Quality Aluminum Alloy for maximum rigidity and minimal flex

- **Rotation:** Equipped with a DC motor allowing a full 360-degree continuous rotation

- **Encoder:** High resolution optical encoder with a resolution of 2048 counts per revolution for precise position measurement.

- **Dynamics of the Rotary System:**

  $\theta(s) = L\{\theta(t)\}/L\{\alpha(t)\} = \mathrm{b}/(\mathrm{s}^2 + as + k)$

where $\theta(s)$ is the Laplace transform of the angular position $\theta(t), \alpha(t)$ is the control input and a,b, and k are constants representing system parameters such as damping, back electromotive force and stiffness respectively.

- **System Identification:**

  $\hat{y}(t) = f(x(t), W)$

  where $\hat{y}(t)$ is the predicted output, $x(t)$ denotes state vector at time t, f is function approximator, often a neural network and W is the weight.

Chapter 4

Design Approach and Details

4.1 Design Approach / Materials and Methods

- **Experimental Setup**: The core of our experimental setup revolves around the Quanser SRV02 servo system, equipped with a flexgage module to simulate a rotary flexible link system. This system is chosen for its precision, reliability, and relevance to real-world applications where vibration suppression is critical. The system's parameters such as mass, length and flexibility are documented for further studies.

- **Control Algorithm Implementation**: The implementation and comparison of three reinforcement learning control algorithm: Deep Deterministic Policy Gradient (DDPG), Soft-Actor Critic (SAC) and Twin-Delayed Deep Deterministic Policy gradient (TD3), has been done. These algorithms are chosen because of their proven effectiveness in handling continuous action spaces and their potential for addressing the challenges of vibration suppression in flexible link systems.

  1. **DDPG**: Known for its stability and efficiency in high dimensional continuous action spaces. It employs neural networks to represent policy(also called actor) and value function(also called critic) to stabilize the learning process and improve efficiency.
  2. **SAC**: Advanced off-policy method that focuses on learning effective policies with fewer interactions with the environment

and provides more consistent and stable learning required for sophisticated control actions.

3. **TD3**: Advanced method which addresses the overestimation bias in value function using two critics to provide delayed policy updates, improved stability and performance.

- **Data Collection and Analysis**: The data is collected through a series of controlled experiments, with each algorithm running for a predefined number of episodes. Key performance metrics such as reduction in vibration amplitude, energy efficiency and settling time is recorded. The data analysis process, including graphical demonstration, is used to compare the algorithm's performance.

- **Software and Computational Resources**: The algorithm is developed and tested using MATLAB/Simulink, using the Deep Learning and Reinforcement Learning toolboxes.

4.2 Codes and Standards

- **IEEE Standard on Reinforcement Learning and Adaptive Dynamic Programming**: This standard provides guidelines for the implementation of Deep Reinforcement Learning Algorithms, ensuring development of DDPG,SAC and TD3 algorithms follows best practices in algorithmic integrity and data handling.

- **IEC 61131-3: Control System Software**: While our research primarily focuses on advanced RL Algorithms, the principles outlined in this standard for control system software are considered during the development of simulation models and interfacing with Quanser SRV02 system.

- **ANSI/RIA R15.06**: Safety Requirements for Industrial Robots and Robot Systems: Safety is paramount in experimental systems involving physical systems. This standard guides the safety protocols implemented in our research, ensuring the well being of operators and the protection of equipment.

4.3 Constraints, Alternatives and Tradeoffs

4.3.1 Constraints

1. **Computational Resources**: The intensive computational demand of training deep reinforcement leaning models requires significant GPU processing power, limiting the extent of hyperparameter tuning and the number of training episodes.

2. **Model Accuracy**: The fidelity of simulated rotary flexible link system to real-world dynamics could vary, impacting the direct application of findings to the physical systems.

3. **Algorithm complexity**: The inherent complexity of DDPG, TD3 and SAC algorithm posed challenges in terms of implementation detail, requiring simplifications that may affect the performance comparisons.

4.3.2 Alternatives

- **Simplified Models**: Using more computationally efficient but less accurate models could reduce processing demands but at the cost of fidelity to real-world system dynamics.

- **Hybrid Approaches**: Combining Reinforcement learning with traditional control techniques, like PID controller, could offer a balance between learning efficiency and system stability.

- **Different Learning Algorithms**: Exploring other Reinforcement Learning algorithms or the variations of the current ones could potentially offer better performance or efficiency under the given constraints.

4.3.3 Tradeoffs

Selecting between these alternatives involves inherent tradeoffs:

- **Accuracy vs Computational Efficiency**: Higher fidelity models and more extensive training leads to better accuracy but require more computational resources. Simplified models are less demanding but may not capture all relevant dynamics.

- **Exploration vs Exploitation**: The Reinforcement Learning algorithm need to balance exploration of the action space with the exploitation of known strategies. More aggressive exploration can discover better solutions but may introduce instability into the system.
- **Algorithm Complexity vs Implementation Feasibility**: While more complex algorithms like SAC offer advantages in terms of exploration and performance, they also require more intricate tuning and implementation, which can be given time and resource constraint.

Chapter 5

SCHEDULE,TASKS AND MILESTONES

5.1 Schedule

Deadline	**Work to be done**
Zeroth Review	Detailed Study of the Deep Reinforcement Learning and rotary Flexible Link
First Review	Prepare the simulation for different control algorithms (DDPG, SAC, and TD3) in different conditions and make a comparative study
Final Review	Make a Report stating all the observations and results, and prepare a research paper and publish.

5.2 Tasks and Milestones

- **Literature Review Completion:** An extensive Literature Review was conducted, which provided us with a comprehensive understanding of the current state of deep reinforcement learning algorithms and their application in vibration suppression for rotary flexible link systems.

- **Algorithms Selection and Justification:** After evaluating various algorithms, we selected DDPG, SAC and TD3 for our study based on their relevance to our system's dynamics and their proven track record in similar control problems.

- **Simulated Environment Setup:** We created a robust simulation environment that accurately models the dynamics of rotary flexible link system. This environment has been instrumental in testing and comparing the performance of the selected algorithms.

- **Control Algorithm Implementation:** Each of the chosen algorithms was implemented with careful consideration of the system's characteristics. Initial simulation has demonstrated the algorithms' capabilities in controlling the system under various conditions.

- **Comparative Analysis:** A detailed comparative analysis was carried out, examining the performance of each algorithm. This analysis not only highlighted the strengths and weaknesses of the algorithms but also provided insight into their operational intricacies.

- **Prototype Development:** A working prototype was developed, showcasing the practical implementation of the control algorithms in a physical setup. This milestone marked a significant transition from theoretical study to tangible application.

- **Performance Tuning and Optimization:** Post-implementation, we fine-tuned the algorithms to optimize performance. This involved parameter adjustments, noise reduction techniques and system identification procedures to ensure high-fidelity control.

- **Documentation and Reporting:** Comprehensive documentation was created, encapsulating the entire process from concept to execution. This documentation serves as a valuable resource for future research and application.

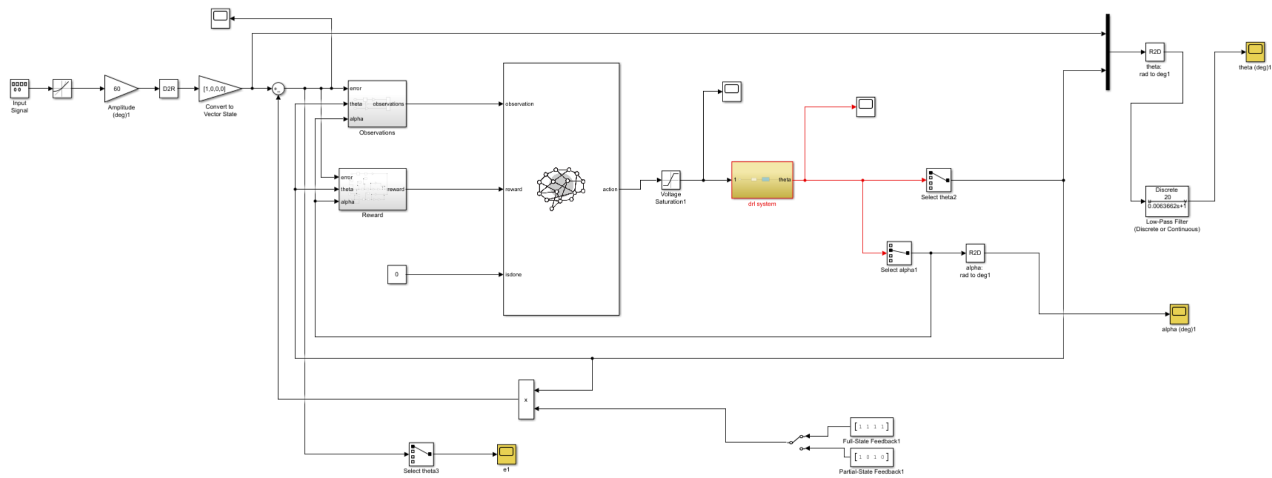
- **Peer Review and Feedback Incorporation:** The preliminary findings were subjected to peer review, which provided critical feedback. This feedback was used to refine our approach and strengthen the outcomes of our research.

- **Final Validation and Testing:** The final phase involved rigorous validation and testing to ensure that the results were reliable and reproducible. This step confirmed the efficacy of our chosen methodologies and the accuracy of our conclusions.

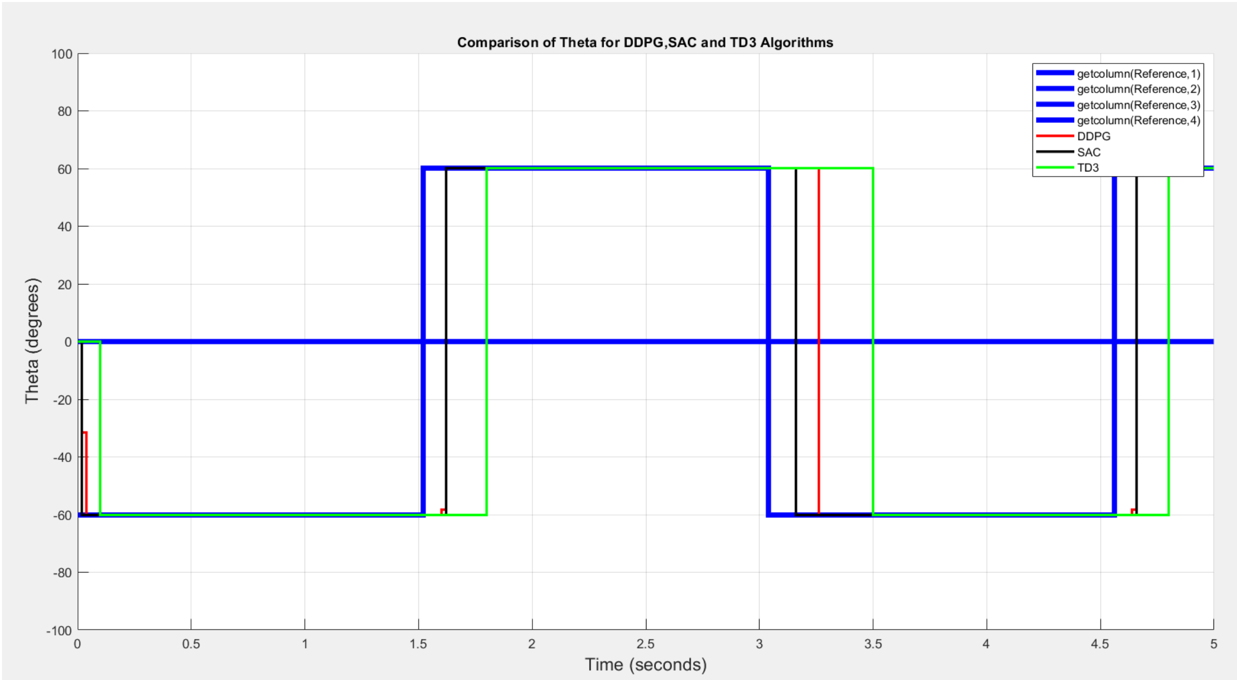
Chapter 6

Project Demonstration

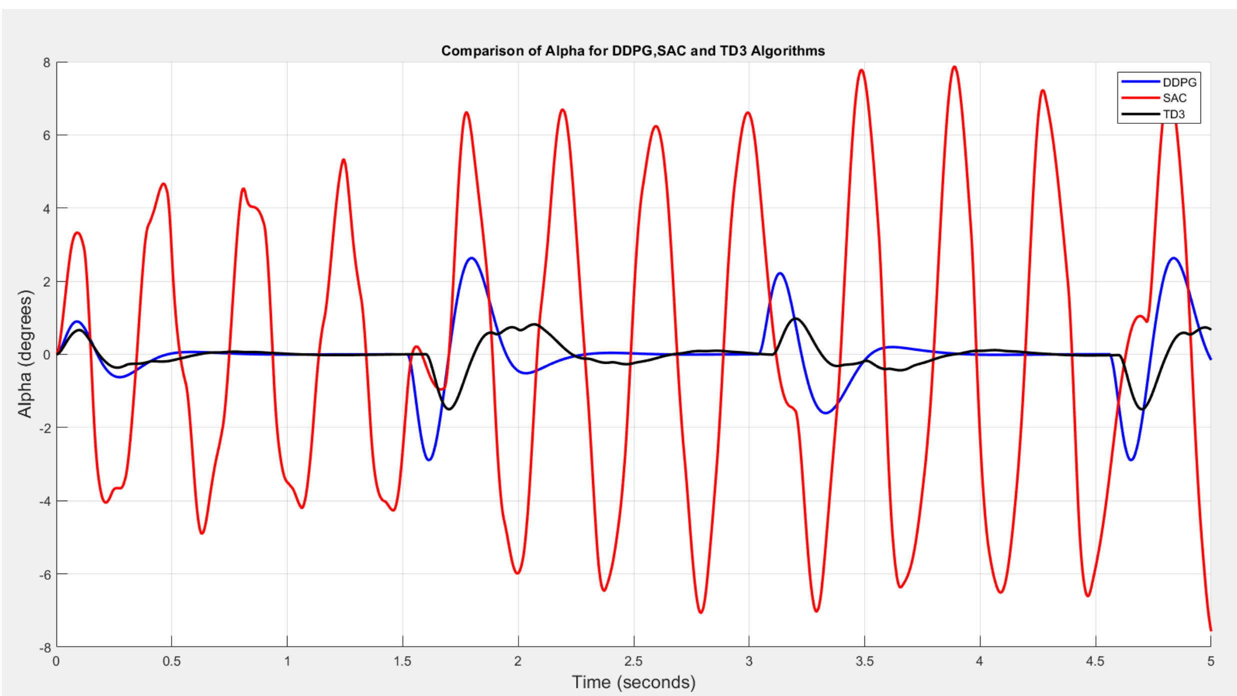
6.1 Figures and Plots



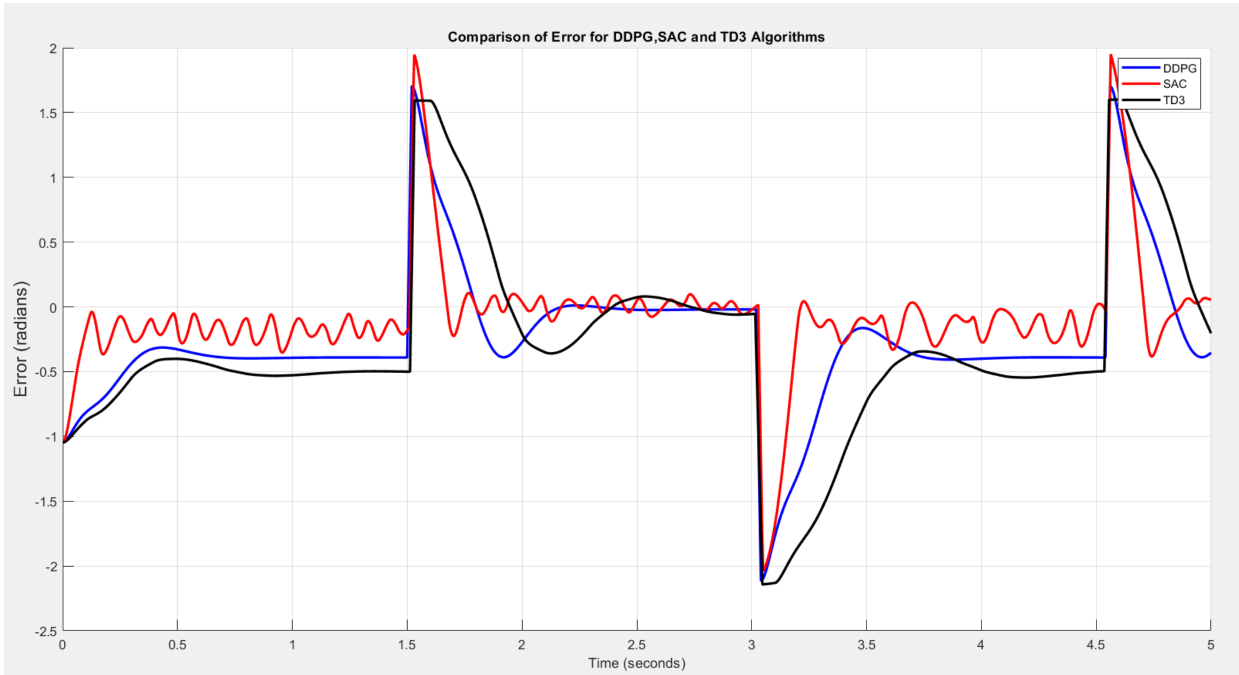
6.1(a) Simulink Model



6.1(b) Theta Plot for different DRL Algorithms

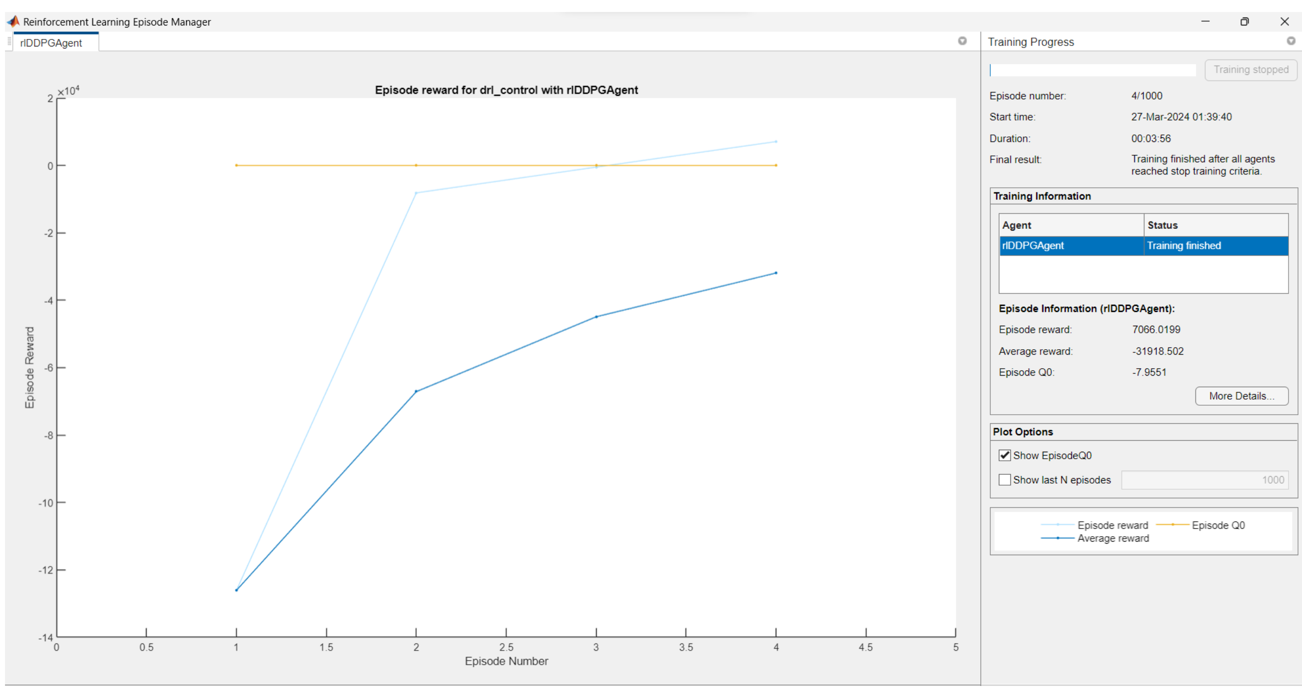


6.1(c) Alpha Plot for different DRL Algorithms

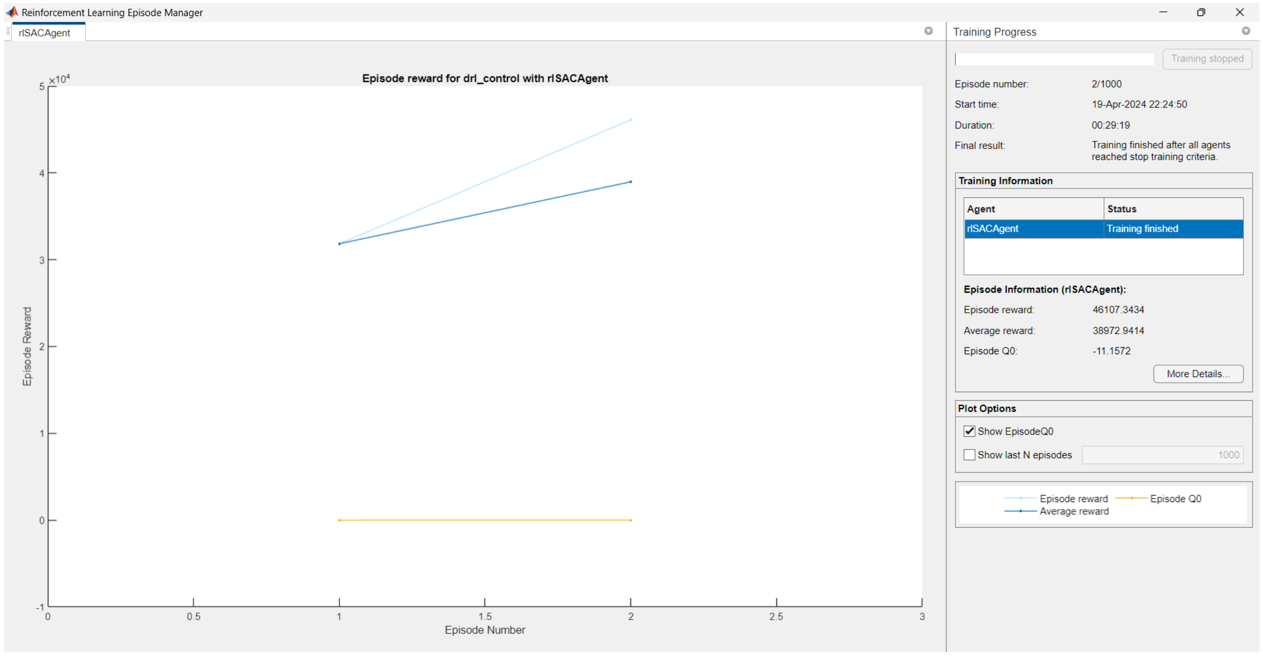


6.1(d) Error Plot for different DRL Algorithms

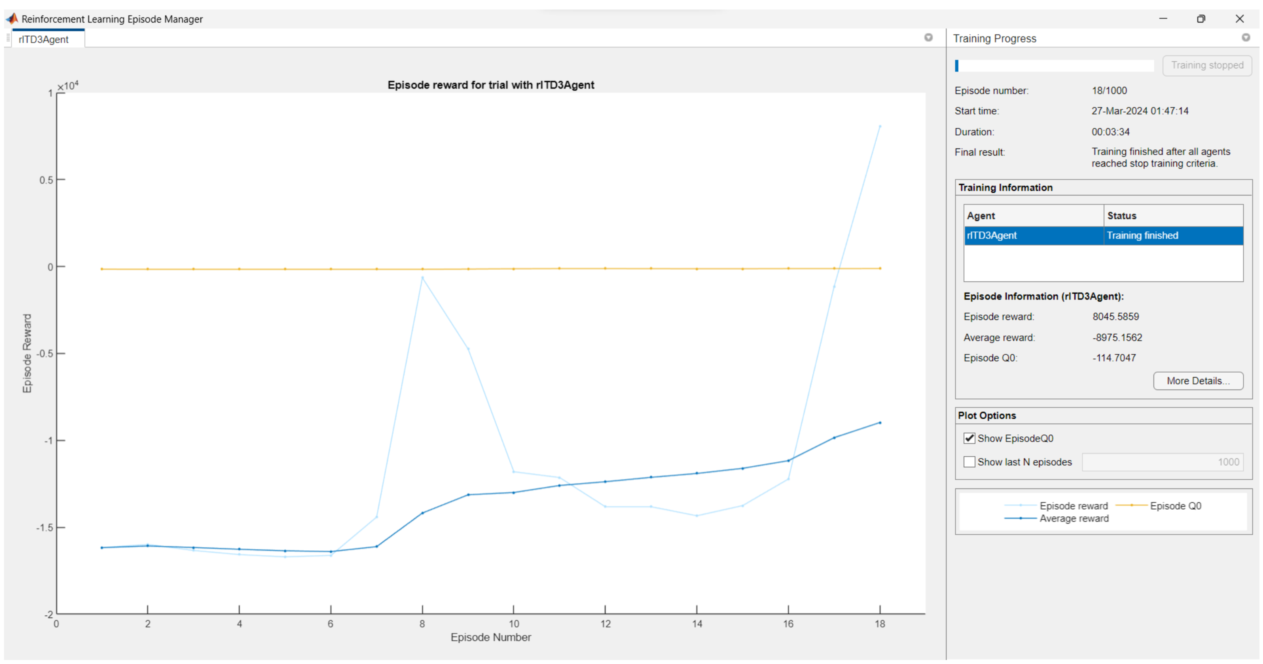
6.2 Agent Training Plot



6.2(a) Training Plot of DDPG Agent



6.2(b) Training Plot of SAC Agent



6.2(a) Training Plot of TD3 Agent

6.3 Code

```
%DDPG Algo

obsInfo = rlNumericSpec([3 1],...
    'LowerLimit',[-inf -inf -inf  ]',...
    'UpperLimit',[inf inf inf]');
obsInfo.Name = 'observations';
obsInfo.Description = 'error,theta and alpha';
%numObservations = obsInfo.Dimension(1);

actInfo = rlNumericSpec([1 1],......
    'LowerLimit',-10,.....
    'UpperLimit',10);
actInfo.Name = 'action';
numActions = actInfo.Dimension(1);

env = rlSimulinkEnv('drl_control','drl_control/RL Agent',...
    obsInfo,actInfo);

wcf_1 = 2 * pi * 50;
wcf_2 = 2 * pi * 10;

A = [0 0 1 0;
    0 0 0 1;
    0 500 -39.5176 0;
    0 -750 39.5176 0];

B = [0 0 64.2267 -64.2267]';

C = [1 0 0 0;
    0 1 0 0];

D = [0;
    0];

Ts = 0.02;
Tf = 100;
rng(0)
statePath = [
    featureInputLayer(obsInfo.Dimension(1),Name="netObsIn")
```

```
getAction(actor,{rand(obsInfo.Dimension)})
agent1 = rlDDPGAgent(actor,critic);

agent1.SampleTime = Ts;

agent1.AgentOptions.TargetSmoothFactor = 0.1;
agent1.AgentOptions.DiscountFactor = 0.9;
agent1.AgentOptions.MiniBatchSize = 128;
agent1.AgentOptions.ExperienceBufferLength = 1e6;

agent1.AgentOptions.NoiseOptions.Variance = 0.3;
agent1.AgentOptions.NoiseOptions.VarianceDecayRate = 1e-5;


agent1.AgentOptions.CriticOptimizerOptions.LearnRate = 1e-3;
agent1.AgentOptions.CriticOptimizerOptions.GradientThreshold = 1;
agent1.AgentOptions.ActorOptimizerOptions.LearnRate = 1e-4;
agent1.AgentOptions.ActorOptimizerOptions.GradientThreshold = 1;

getAction(agent1,{rand(obsInfo.Dimension)})




maxsteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    MaxEpisodes=1000, ...
    MaxStepsPerEpisode=ceil(Tf/Ts), ...
    ScoreAveragingWindowLength=10, ...
    Verbose=false, ...
    Plots="training-progress",...
    StopTrainingCriteria="EpisodeReward",...
    StopTrainingValue=6000);

doTraining = true;
trainingStats = train(agent1,env,trainOpts);
simOpts = rlSimulationOptions('MaxSteps',maxsteps,'StopOnError','on');
experiences = env.sim(agent1,simOpts);
```

6.3(a) Code for DDPG Agent Training

```
%SAC Algo

obsInfo = rlNumericSpec([3 1],...
    'LowerLimit',[-inf -inf -inf  ]',...
    'UpperLimit',[inf inf inf]');
obsInfo.Name = 'observations';
obsInfo.Description = 'error,theta and alpha';

actInfo = rlNumericSpec([1 1],......
    'LowerLimit',-10,.....
    'UpperLimit',10);
actInfo.Name = 'action';
numActions = actInfo.Dimension(1);

env = rlSimulinkEnv('drl_control','drl_control/RL Agent',...
    obsInfo,actInfo);

wcf_1 = 2 * pi * 30;
wcf_2 = 2 * pi * 15;

A = [0 0 1 0;
    0 0 0 1;
    0 500 -39.5176 0;
    0 -750 39.5176 0];

B = [0 0 64.2267 -64.2267]';

C = [1 0 0 0;
    0 1 0 0];

D = [0;
    0];

Ts = 0.02;
Tf = 400;
rng(0)
statePath = [
    featureInputLayer(obsInfo.Dimension(1),Name="netObsIn")
    fullyConnectedLayer(50)
```

```
    'ActionMeanOutputNames','Mean',...
    'ActionStandardDeviationOutputNames','StandardDeviation',...
    'ObservationInputNames','observation');

getAction(actor,{rand(obsInfo.Dimension)})
agent1 = rlSACAgent(actor,critic);

agent1.SampleTime = Ts;

agent1.AgentOptions.TargetSmoothFactor = 1e-3;
agent1.AgentOptions.DiscountFactor = 0.9;
agent1.AgentOptions.MiniBatchSize = 128;
agent1.AgentOptions.ExperienceBufferLength = 1e6;


agent1.AgentOptions.CriticOptimizerOptions.LearnRate = 1e-3;
agent1.AgentOptions.CriticOptimizerOptions.GradientThreshold = 1;
agent1.AgentOptions.ActorOptimizerOptions.LearnRate = 1e-4;
agent1.AgentOptions.ActorOptimizerOptions.GradientThreshold = 1;
getAction(agent1,{rand(obsInfo.Dimension)})




maxsteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    MaxEpisodes=1000, ...
    MaxStepsPerEpisode=ceil(Tf/Ts), ...
    ScoreAveragingWindowLength=10, ...
    Verbose=false, ...
    Plots="training-progress",...
    StopTrainingCriteria="EpisodeReward",...
    StopTrainingValue=32000);

doTraining = true;
trainingStats = train(agent1,env,trainOpts);
simOpts = rlSimulationOptions('MaxSteps',maxsteps,'StopOnError','on');
experiences = env.sim(agent1,simOpts);
```

6.3(b) Code for SAC Agent Training

```
%TD3 algo

obsInfo = rlNumericSpec([3 1],...
    'LowerLimit',[-inf -inf -inf  ]',...
    'UpperLimit',[inf inf inf]');
obsInfo.Name = 'observations';
obsInfo.Description = 'error,theta and alpha';
%numObservations = obsInfo.Dimension(1);

actInfo = rlNumericSpec([1 1],......
    'LowerLimit',-10,.....
    'UpperLimit',10);
actInfo.Name = 'action';
numActions = actInfo.Dimension(1);

env = rlSimulinkEnv('trial','trial/RL Agent',...
    obsInfo,actInfo);

wcf_1 = 2 * pi * 30;
wcf_2 = 2 * pi * 10;

A = [0 0 1 0;
    0 0 0 1;
    0 500 -39.5176 0;
    0 -750 39.5176 0];

B = [0 0 64.2267 -64.2267]';

C = [1 0 0 0;
    0 1 0 0];

D = [0;
    0];

Ts = 0.1;
Tf = 100;
rng(0)
statePath = [
    featureInputLayer(obsInfo.Dimension(1),Name=
```

```
agent1 = rlTD3Agent(actor,critic);

agent1.SampleTime = Ts;

agent1.AgentOptions.TargetSmoothFactor = 0.1;
agent1.AgentOptions.DiscountFactor = 0.9;
agent1.AgentOptions.MiniBatchSize = 128;
agent1.AgentOptions.ExperienceBufferLength = 1e6;

%agent1.AgentOptions.NoiseOptions.Variance = 0.3;
%agent1.AgentOptions.NoiseOptions.VarianceDecayRate = 1e-5;


agent1.AgentOptions.CriticOptimizerOptions.LearnRate = 1e-3;
agent1.AgentOptions.CriticOptimizerOptions.GradientThreshold = 1;
agent1.AgentOptions.ActorOptimizerOptions.LearnRate = 1e-4;
agent1.AgentOptions.ActorOptimizerOptions.GradientThreshold = 1;

getAction(agent1,{rand(obsInfo.Dimension)})

criticOpts = rlRepresentationOptions('LearnRate',1e-03,'GradientThreshold',1);
actorOpts = rlRepresentationOptions('LearnRate',1e-04,'GradientThreshold',1);


maxsteps = ceil(Tf/Ts);
trainOpts = rlTrainingOptions(...
    MaxEpisodes=1000, ...
    MaxStepsPerEpisode=ceil(Tf/Ts), ...
    ScoreAveragingWindowLength=10, ...
    Verbose=false, ...
    Plots="training-progress",...
    StopTrainingCriteria="EpisodeReward",...
    StopTrainingValue=8000);

doTraining = true;
trainingStats = train(agent1,env,trainOpts);
simOpts = rlSimulationOptions('MaxSteps',maxsteps,'StopOnError','on');
experiences = env.sim(agent1,simOpts);
```

6.3(c) Code for TD3 Agent Training

Chapter 7

COST ANALYSIS/RESULT AND DISCUSSION

7.1 Cost Analysis

- **Hardware:** The Hardware coomponent (SRV02 Flexgage Link,etc) has been provided by the Control System Lab of VIT.
  Estimated cost: INR 15 Lakhs
- **Software:** The MATLAB and Simulink software is Free to use for Academic Purpose for those who have Student License.
  Estimated Cost: USD 35 (for student suite) or INR 2905

7.2 Results and Discussion

Results:

- **Performance Metrics**: Across several metrics, including reduction of vibration amplitude, response time to disturbances and overall system stability, SAC consistently outperformed both DDPG and TD3. SAC showed a superior balance between exploration and exploitation, leading to more effective vibration suppression strategies.
- **Learning Efficiency** : TD3 shows faster convergence to effective control policies compared to DDPG, likely due to its twin critic architecture and delayed policy updates. However, SAC's entropy-based exploration further accelerates the learning efficiency, achieving optimal performance with fewer training episodes.

- **Robustness to Disturbances**: In scenarios with variable disturbances, SAC maintained robust performance, demonstrating less deviation in control effectiveness compared to TD3 and DDPG. This suggests SAC's entropy maximization provides an inherent advantage in adapting to changing dynamics.
- **Algorithmic Complexity**: Despite its effectiveness, SAC's implementation complexity and computational demands are higher. In contrast, DDPG, while the least effective in vibration suppression, requires the simplest implementation and the least computational resources.

Discussion:

- **Implications of Findings**: The superior performance of SAC in suppressing vibrations underscores the potential of entropy-based reinforcement learning strategies for complex control tasks. This finding suggests a promising direction for future research and application in systems requiring precise and adaptive control.
- **Tradeoffs and Considerations**: While SAC's performance is compelling, the trade-offs in terms of computational demand and implementation complexity cannot be overlooked. Practitioners must consider these factors when selecting an algorithm for real-world applications. TD3 presents a middle ground, offering improved performance over DDPG with less complexity than SAC.
- **Limitations and Future Work**: The study's findings are constrained by the simulation environment and model accuracy. Future work should explore the application of these algorithms in physical systems to validate the simulation results. Additionally, investigating hybrid approaches that combine the strength of these algorithms could yield even more effective vibration suppression strategies.
- **Contribution to the Field**: This research contributes to the field by providing a detailed comparative analysis of reinforcement learning algorithms in a novel application area. The insights gained not only advance our understanding of algorithmic

performance in vibration suppression but also opens avenues for optimizing control strategies in similar complex systems.

- **State-Space Equations:**

  m1− > Mass of Flexible Link (kg)

  L1− > Length of Flexible Link (m)

  Moment of Inertia of Flexible Link (J1) = (m1*(L1^2))/3

Ks− >Link Stiffness

Jeq− > Equivalent Moment of Inertia of Flexible Link

$$A=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \text{ks/Jeq} & \text{-Beq/Jeq} & 0 \\ 0 & \text{-ks*(Jeq+J1)/Jeq/J1} & \text{Beq/Jeq} & 0 \end{bmatrix}$$

$$B=\begin{bmatrix} 0 \\ 0 \\ \text{1/Jeq} \\ \text{-1} \end{bmatrix}$$

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

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

- **Reward function:**

  R(t) = -(error^2)+10($|theta{-}theta\_target| <= 0.1$)−1($|theta{-}theta\_target| > 0.1 + 1(|alpha - alpha\_target| <= 0.05) - 1(|alpha - alpha\_target| > 0.05$)

Chapter 8

SUMMARY

8.1 Summary

This research embarks on pioneering investigation into the application of advanced reinforcement learning algorithms like Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft Actor-Critic (SAC) for the purpose of suppressing vibration in rotary flexible link systems. Aimed at addressing the complex challenge of maintaining stability in systems where flexibility introduces inherent oscillations, this study meticulously evaluates the efficacy, robustness and efficiency of these algorithms in a controlled simulated environment.

Utilizing the Quanser SRV02 servo system augmented with a flex-gauge module to simulate the dynamics of a rotary flexible link, the study employs a methodical approach to implement and compare the three algorithms. Each algorithm's performance was assessed based on keys metrics such as vibration amplitude reduction, response time and energy efficiency under varying conditions of disturbances.

Findings from this comparative analysis reveal that the SAC algorithm consistently outperforms DDPG and TD3 in terms of vibration suppression, learning efficiency and adaptability to dynamic changes. The entropy-based exploration strategy of SAC contributes to its superior performance, offering a promising direction for future research and application in systems requiring precise control. However, the study also notes the computation intensity and implementation complexity of SAC as considerations for practical deployment.

By providing a detailed examination of these reinforcement learning strategies within the context of vibration suppression, this research contributes valuable insights to the field of control systems engineering. It highlights the potential of machine learning-based control solutions to enhance the performance and reliability of complex mechanical systems, paving the way for further innovation in this area.

Chapter 9

Curriculum Vitae

Name: Sohan Das
Father's Name: Pranab Kumar Das
Date of Birth: 21/12/2001
Nationality: Indian
Sex: Male
University: VIT University Vellore
Permanent Address: Gourpara, Chakdaha, Nadia-741222
State: West Bengal
Phone number: 8145804655
E-mail ID: sohan.das2020@vit.student.ac.in

CGPA: 8.31

Examinations Taken:
1. GRE: N/A
2. IELTS: 7.0

Placement Details: Applying for Higher Studies

Chapter 10

Capstone Project Summary

Project Title:	Comparative Analysis of Deep Reinforcement Learning Algorithms for Vibration Suppression in Rotary Flexible Link Systems
Team Members:	1
Faculty Guide:	Prof. Vinodh Kumar E
Semester/Year	8th Semester/4th Year
Project Abstract:	The suppression of vibration in rotary flexible link system is a complex and critical challenge in control engineering, with significant application in the field of Robotics and industrial machineries. The research undertakes the comparative analysis of the three reinforcement learning algorithms: Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft-Actor Critic Method (SAC). The study evaluates the efficacy of these algorithms in reducing oscillatory behavior and achieving precise angular control in a simulated flexible link system. Through iterative simulations, each algorithm's performance is rigorously examined with respect to convergence reliability, computational efficiency and robustness to perturbations. The results indicate that while SAC exhibits superior sample efficiency and TD3 offers enhanced training stability, DDPG provides valuable insights as benchmark for algorithmic comparison. The findings of this study not only shed light on the nuances of employing reinforcement learning for control tasks but also guide the selection of appropriate algorithms for vibration suppression in flexible link systems. The research thus paves way for further innovations in the application of machine learning to dynamic system stabilization.

List Codes and Standards that significantly affect your Project:	• IEEE Standard on Reinforcement Learning and Adaptive Dynamic Programming • IEC 61131-3: Control System Software • ANSI/RIA R15.06
List at least two significant realistic design constraints that are applied to your project:	• **Model Accuracy:** The fidelity of simulated rotary flexible link system to real-world dynamics could vary, impacting the direct application of findings to the physical systems. • **Algorithm complexity:** The inherent complexity of DDPG, TD3 and SAC algorithm posed challenges in terms of implementation detail, requiring simplifications that may affect the performance comparisons.
Briefly Explain two significant trade-offs considered in your design, including options considered and the solution chosen:	• **Simplified Models:** Using more computationally efficient but less accurate models could reduce processing demands but at the cost of fidelity to real-world system dynamics. • **Hybrid Approaches:** Combining Reinforcement learning with traditional control techniques, like PID controller, could offer a balance between learning efficiency and system stability.

Describe the Computing aspects,if any, of your project. Specifically identifying hardware-software trade-offs, interfaces,and/or interactions	• **Hardware-Software Trade-offs:** **Processing Power vs Energy Consumption:** A key tradeoff in our system design is between the processing power required for complex control algorithms, like Deep Reinforcement Learning (DRL) and the energy conservation constraints. We chose processors that offer a balance between computational ability and power efficiency. • **Interfaces:** **Sensor-Controller Interface:** The system incorporates various sensors that interface with the main controller via digital and analog I/O ports. Data from accelerometers, gyroscopes and position sensors are crucial for the control algorithms to accurately track and stabilize the flexible link. • **Interactions:** **Feedback System:** Feedback from the physical system is used to adjust control parameters in real time. This closed loop system is fundamental to the project's objectives, ensuring that the system adapts to the changing conditions and maintains performance specifications. **Error Handling and Diagnostic:** The software includes robust error handling and diagnostic capabilities to detect and respond to anomalies. It interacts with the hardware to enable preventive maintenance and to ensure system uptime.

Appendix A

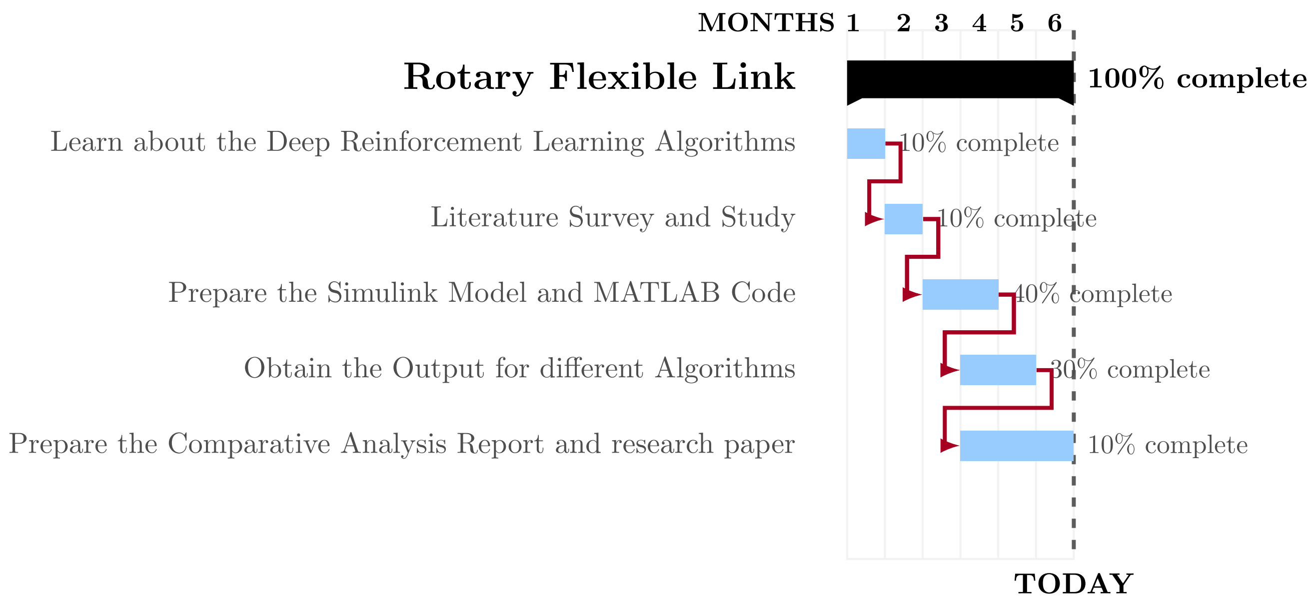
PSEUDO CODE

Algorithm 1 Comparative Analysis of Control Algorithms

```
BEGIN ResearchControlAlgorithms
DEFINE objectives for the research
SELECT control algorithms for comparison (DDPG, SAC, TD3)
DEFINE performance metrics (e.g., vibration suppression, convergence rate)
PREPARE simulation environment
SET UP the rotary flexible link system model
CONFIGURE parameters for each control algorithm
for each control algorithm do
    IMPLEMENT algorithm in the simulation environment
    RUN simulations to collect data on performance metrics
end for
ANALYZE collected data
COMPARE the performance of each control algorithm
IDENTIFY strengths and weaknesses of each algorithm
DRAW conclusions based on analysis
DETERMINE which algorithm performs best for vibration suppression
SUGGEST improvements or future research directions based on findings
DOCUMENT results
PREPARE figures and tables illustrating key findings
WRITE sections of the paper (Introduction, Methods, Results, Discussion)
REVIEW and revise the paper
GET feedback from peers or supervisors
UPDATE the paper based on feedback
SUBMIT the paper for publication
CHOOSE appropriate journals or conferences
FOLLOW submission guidelines and submit
END ResearchControlAlgorithms
```

Appendix B

GANTT CHART



Chapter 11

References

- *Deep Reinforcement Learning Based Trajectory Planning Under Uncertain Conditions by Lienhung Chen, Zhongliang Jiang, Long Cheng, Alois C. Knoll and Mingchuan Zhou [ScienceDirect]*
- *An Adaptive Neuro-Fuzzy Controller for Vibration Suppression of Flexible Structures by Adam Genno, Wilson Wang [ResearchGate]*
- `https://www.quanser.com/resource-type/technical-resources/?-products=5807`
- *Flexible-link robots with combined trajectory tracking and vibration control by O.A. Garcia-Perez, G. Silva-Navarro, J.F. Peza-Solis [ScienceDirect]*
- *Reinforcement learning and optimal adaptive control: An overview and implementation examples by Said G. Khan a, Guido Herrmann b, Frank L. Lewis c, Tony Pipe a, Chris Melhuish [ScienceDirect]*
- *Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation by Aveen Dayal a, Linga Reddy Cenkeramaddi a, Ajit Jha [ResearchGate]*
- *Reinforcement learning vibration control of a multi-flexible beam coupling system by Zhi-cheng Qiu, Yang Yang, Xian-min Zhang [ResearchGate]*
- *A Vibration Control Method for Hybrid-Structured Flexible Manipulator Based on Sliding Mode Control and Reinforcement Learning*

by Teng Long; En Li, Yunqing Hu, Lei Yang, Junfeng Fan, Zize Liang, Rui Guo [IEEE Xplore]

- *Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes by T. Shuprajhaa, Shiva Kanth Sujit, K. Srinivasan [ScienceDirect]*
- *Deep Reinforcement Learning Trajectory Planning for Vibration Suppression via Jerk Control by Sung Gwan Park, Sungsoo Rhim [IEEE Xplore]*
- *LQG/LTR Controller Design for Flexible Link Quanser by Cosmin Ionete [IEEE Xplore]*