VIBRATION CONTROL OF THIN STRUCTURES USING A REINFORCEMENT LEARNING APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SANDRA NAFUNA WANYONYI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

SEPTEMBER 2023

Approval of the thesis:

VIBRATION CONTROL OF THIN STRUCTURES USING A REINFORCEMENT LEARNING APPROACH

submitted by **SANDRA NAFUNA WANYONYI** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, **Graduate School of Natural and Applied Sciences** _____

Prof. Dr. Serkan Özgen
Head of the Department, **Aerospace Engineering** _____

Prof. Dr. Dilek Funda Kurtuluş
Supervisor, **Aerospace Engineering, METU** _____

Dr. Ipar Ferhat
Co-Supervisor, **Aerospace Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Ozan Tekinalp
Aerospace Engineering, METU _____

Prof. Dr. Dilek Funda Kurtuluş
Aerospace Engineering, METU _____

Assoc. Prof. Dr. Halil Ersin Söken
Aerospace Engineering, METU _____

Assoc. Prof. Dr. Nilay Sezer Uzol
Aerospace Engineering, METU _____

Asst. Prof. Dr. Emre Kara
Aerospace Engineering, GAÜN _____

Date: 04.09.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name :

Signature :

ABSTRACT

VIBRATION CONTROL OF THIN STRUCTURES USING A REINFORCEMENT LEARNING APPROACH

Sandra Nafuna, Wanyonyi
Master of Science, Aerospace Engineering
Supervisor: Prof. Dr. Dilek Funda Kurtuluş
Co-Supervisor: Dr. Ipar Ferhat

September 2023, 121 pages

Vibration control in thin structures is an important research topic with numerous practical applications in engineering and robotics. With the continued use of thin structures for space applications like solar sails, space reflectors, and satellite antennas, there is an ever-growing interest in effective and robust vibration suppression methods. This thesis presents a comprehensive study into the application of Reinforcement Learning (RL) as a control approach to suppress the vibration in thin beams with pinned-pinned boundary conditions. The primary focus of this research is to contribute a novel insight into the field of vibration control for thin structures. With this goal in mind, a novel approach for vibration control in thin beams with pinned supports is introduced. A robust RL controller is developed to effectively handle parameter uncertainties and varying external disturbances. In this case, they are modeled as varying natural frequency and initial displacements of the the beam-actuator environment, respectively. By incorporating these uncertainties, the research offers one of the earliest illustrations of the impact of parameter uncertainty on the performance of an RL controller for vibration control in thin

beams. The learning and control performance of off-policy and on-policy RL algorithms for vibration control in thin pinned-pinned beams is evaluated, reviewed, and discussed with an aim to identify the most suitable RL approach for effectively suppressing vibrations in such structures. A detailed comparative study is provided on the reward-shaping process of the problem by two different reward function schemes being implemented, and the results obtained from both are compared and discussed. The controllers developed from the different reward schemes illustrate a significant difference in their performance and reinforce the importance of the process in the RL controller development process. Despite the simple uncertainty model, the results indicate that the developed RL controller can cope with changing system parameters and varying initial excitations. The controlled response of the beam system vibration for agents that were trained in environments with different dynamics also indicates the success of the RL controller application.

Keywords: Reinforcement Learning, Vibration Control, Active Control, Thin Structures

ÖZ

İNCE YAPILARIN GÜÇLENDİRİLMİŞ ÖĞRENME YAKLAŞIMIYLA TİTREŞİM KONTROLÜ

Sandra Nafuna, Wanyonyi
Yüksek Lisans, Havacılık ve Uzay Mühendisliği
Tez Yöneticisi: Prof. Dr. Dilek Funda Kurtuluş
Ortak Tez Yöneticisi: Dr. Ipar Ferhat

Eylül 2023, 121 sayfa

İnce yapılarda titreşim kontrolü, mühendislik ve robotik alanlarında çok sayıda pratik uygulaması olan önemli bir araştırma konusudur. İnce yapıların, uzay araçlarının alt sistemleri olan güneş yelkenleri, uzay yansıtıcıları ve uydu antenleri gibi bileşenlerde kullanılmaya devam edilmesiyle birlikte, etkili ve gürbüz titreşim kontrol yöntemlerine olan ilgi giderek artmaktadır. Bu tez, basit-basit sınır koşullarına sahip ince kirişlerdeki titreşimin sönümlenmesi için, bir kontrol türü olan Pekiştirmeli Öğrenme (PÖ) yönteminin uygulanmasına ilişkin kapsamlı bir çalışma sunmaktadır. Bu araştırmanın öncelikli odak noktası, ince yapıların titreşim kontrolü alanına yeni bir bakış açısıyle katkıda bulunmaktı. Bu amaçla, basit-basit sınır koşullarına sahip ince kirişlerde titreşim kontrolünü sağlamak için yeni bir yaklaşım tanıtılmıştır. Parametre belirsizliklerini ve değişken dış etkenleri güçlü bir şekilde kontol edebilmek için gürbüz PÖ kontolörü geliştirildi. Bu çalışmada belirsiz ve değişken değerler, sistemin doğal frekansının değişmesi ve ortamdaki başlangıç yerdeğiştirme değerleri olarak modellendi. Bu tür belirsizliklerin dahil edilmesiyle, bu araştırma, parametre belirsizliliklerinin ince kirişlerin titreşim kontrolü için geliştirilen PÖ kontrolörünün performansı üzerindeki etkisini gösteren ilk

çalışmalardan biri olmaktadır. Bu tarz yapılar için en uygun PÖ yaklaaşımını belirlemek amacıyla, sabit-sabit sınır koşullu ince bir kirişin titreşimini sönümlemek için geliştirilen politika-içi ve politika-dışı PÖ algoritmaların öğrenme ve kontrol performansları gözden geçirilmiş, değerlendirilmiş ve tartışılmıştır. İki farklı ödüllendirme fonksiyon şeması uygulanarak detaylı bir karşılaştırmalı çalışma yapılmış ve her ikisinden elde edilen sonuçlar karşılaştırılmış ve tartışılmıştır. İki farklı ödül şemasına göre geliştirilen kontrolörler performans açısından ciddi farklar göstermekte ve PÖ kontrolör geliştirme sürecinin önemini pekiştirmektedir.Belirsizlik modelinin basit olmasına rağmen, sonuçlar geliştirilen PÖ kontrolörünün değişen sistem parametreleri ve çeşitli başlangıç uyarımlarının üstesinden gelebildiğini göstermektedir. Farklı dinamiklere sahip ortamlarda eğitilmiş ajanlar için kiriş sistemi titreşiminin kontrollü tepkisi de PÖ kontrolörü kullanımının başarısını göstermektedir.

Anahtar Kelimeler: Pekiştirmeli Öğrenme, Titreşim Kontrolü, Aktif Kontrol, İnce Yapılar

To my family

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisors Prof. Dr. Funda Kurtuluş and Dr. İpar Ferhat for their continued help and guidance throughout my master's program. I am grateful for their advice, constructive criticisms, support, and insight throughout my research journey. I am extremely grateful to have had great advisors who believed in this work and my effort.

I would also like to thank my sisters, Shirley, Nicole, Maureen, and Linda for giving me strength and believing in me throughout my master's program. For their continued support, solace, and comfort in the toughest times provided a source of strength that I could use to power through. I would like to thank my parents, Charles and Carol for their provision, belief, and support as well. Last but not least, I would like to thank some of my closest friends James and Gwyron for providing inspiration and motivation, in their own unique ways, all through my thesis journey.

I would also like to extend my gratitude to the various colleagues who worked alongside me in the department and the instructors who provided courses that aided my understanding of this work. I am grateful for their effort, participation, and collaboration. And to anyone else, who provided a word of encouragement, comfort, and understanding in this journey, I am sincerely grateful.

This work is funded by the Scientific and Technological Research Council of Turkey (TUBITAK) with grant number 118C286 under TUBITAK BIDEB 2232 Program.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

SYMBOLS

a_t	Reinforcement learning action at time t
$\hat{A}_t$	Advantage function estimator
b_a	Width of the actuator
d_{33}	Electric charge constant
e	Exponential
E	Young's Modulus
E_k	Electric field tensor
I	Moment of inertia
Q	State action value function
r_t	Reinforcement learning reward at time t
s_t	Reinforcement learning state at time t
S_{ij}	Mechanical strain tensor
s_{ijkl}^E	Compliance tensor
t_b	Thickness of the beam
t_a	Thickness of the actuator
V	State value function
V_a	Voltage applied to the actuator
$\bar{W}$	Neural network weights
γ	Reinforcement learning discount factor

ε_{ik}^{T}	Permittivity tensor
ρ	Density
υ	Poisson's ratio
π	Reinforcement learning policy function
ω_n	Natural frequency
θ, ϕ, ψ	Neural network parameters

CHAPTER 1

INTRODUCTION

This work focuses on the vibration control of a thin pinned-pinned beam using a reinforcement control approach. This chapter presents an extensive literature review of the vibration control of thin structures, the application of reinforcement learning control in engineering, and the implementation of reinforcement learning for vibration suppression. Later in the chapter, the motivation behind the research, the contribution towards existing literature on vibration control of thin structures, and an outline of the rest of the thesis document are provided.

1.1 Literature Review

This research aims to apply RL for the active vibration control of thin structures. In order to add to the available knowledge and contribute to current literature and research, it is vital to understand how much work has already been done in this field. This chapter presents a summary of some of the existing literature on vibration control of (thin) structures. A summary of the implementation of RL as a control method in aerospace is also given to demonstrate the applicability of the method in the field. The few existing sources of literature focused on vibration control of structures using RL and the related literature from the Middle East Technical University theses catalog are outlined.

1.1.1 System Modeling

Different approaches have been taken for modeling and model order reduction (MOR) of thin structures with piezoelectric actuators. Lee and Alandoli [1], establish that the finite element method (FEM) is widely used for the mathematical modeling of distributed parameter systems. Given this fact, most of the literature highlighted in this subsection implements the FEM for mathematical modeling of various different structures. Liu et al. [2], investigate the optimal placement of piezoelectric actuators on a membrane structure for vibration control. To do this, they establish a dynamic model of a membrane structure and piezoelectric actuator system using the Kirchoff plate theory and obtain the dynamic finite element equation from the Galerkin method. They use three-node triangular elements for discretizing the structure. They consider the effects of the additional mass and stiffness of the piezoelectric actuators, as this affects the optimal placement. Yipaer and Sultan [3], present research on optimal control of membrane structures with their focus placed on using the linear-time-invariant (LTI) second order vector form of system dynamics. They model the structure as a thin plate and generate the system's mass, stiffness and damping matrices using the weak-form FEM. Ferhat and Sultan [4, 5] model 2-dimensional a thin plate-piezoelectric actuator system using FEM and carry out system analysis for controllability and observability using vector second order form approaches in order to reduce the dimension of the discretized system while keeping the high fidelity finite element model. Gupta et al. [6] propose an active vibration control technique that is robust to temparature changes. They derive the finite element equation of motion (EOM) using Hamilton's variational principle for a cantilevered smart piezo plate using 4-noded plane finite elements. They use modal analysis to obtain the uncoupled form of the equation of motion. Liu et al. [7] design an integrated control system for both attitude and vibration control of a spacecraft system. They present a model of a flexible cantilever beam and derive the governing equation using Hamilton's principle. Deshpande and Sankar [8] illustrate the efficiency of three different finite element methods for passive control using a 10-

bay plane truss and the natural frequencies, mode shapes, and loss factor of a pin-connected truss containing several damped members. They implement truss finite element method, equivalent beam element method, and scaled beam element method.

With the use of FEM, the number of elements increases imposing a more complex model and increasing the computational limitations [1]. This presents the need for model order reduction, commonly implemented in literature for computational efficiency. Williams et al. [9] present reasearch on a practical solution to vibration control that can be applied to various structures with little disruption. They create an analytical model of a system link structure based on Euler–Bernoulli beam theory and validate it using a finite element model and experimental data. They assume that only the first mode of the system is excited and utilize the Galerkin decomposition method to reduce the order of the EOM. Zhang et al. [10] illustrate vibration control for a more complex mathematical model of a thin smart structure. They construct a finite element model based on the first-order shear deformation hypothesis, including strain components. They derive the smart structure's dynamic model from Hamilton's principle and reduce the model obtained by using a truncated modal matrix that only includes the first r modes. In this work, however, the model is obtained directly from the transfer function representing the dynamics of the system. Examples in literature of this approach include Le [11] who models a cantilever beam bonded with piezoelectric sensors and actuators using a transfer function summarizing the dynamics of the beam-sensor-actuator system. Goodwin et al. [12] illustrate an example of modeling a cantilever beam with one actuator patch using a system transfer function as well. Aksoy [13] also uses three separate transfer functions to model a plate that has three groups of actuators in a study that tries to determine an effective vibration control scheme after determining the placement of the actuators and sensors on the plate.

1.1.2 Control Approaches

To achieve vibration control of thin structures passive control and/or active control can be implemented. Passive control refers to the modification of the stiffness, mass and damping of the vibrating system to make it less responsive to its environment. On the other hand, active control implies the use of external assistance to drive devices that will act on the structure to generate a vibration that will cancel the initial one [14]. Extensive research is available on the use of different control methods for vibration suppression. Researchers have implemented passive and classical, optimal, robust and adaptive active control methods for vibration control of various structures. The following sub sections discuss the various control approaches implemented in existing literature.

1.1.2.1 Passive Control

Various studies have explored the possibility of improving vibration characteristics and have proposed a wide range of solutions to this problem. This is because passive methods provided and still do provide a solution to the vibration control problem that is effective and less complex than active methods. In one of the earliest examples in literature, Kerwin Jr. [15] presents a quantitative analysis of the damping effectiveness of a constrained viscoelastic layer, which was at that time used in aircraft. Pranoto et al. [16] present a linear damper as a passive control method for vibration suppression in large flexible structures like aircraft wings. Takács and Rohal'-Ilkiv suggest the possibility of utilizing an adaptive passive approach instead of actively controlling the vibration amplitudes, velocities or accelerations [17]. Sankar and Deshpande [8] derive the complex stiffness matrix and the mass matrix of a uniaxial bar subjected to constrained layer damping over its length, with different configurations of damping. They compare the results obtained from these system matrices for each case and also include the case without damping over the truss structure. Hagood and Crawley [18] present experiments they conducted for

two damping enhancement schemes for large/precision space structures. They implement tunable proof-mass dampers and resonant shunted piezoelectric damping concepts on a truss structure that is lightly damped.

1.1.2.2 Active Control

There is extensive research on the application of active, non-adaptive control methods for vibration supression of thin structures. To establish the history of active vibration control implementations, some notable works have been summarized in this chapter. Each highlights which active control method was applied and what specific contribution was unique to this field. Lynch and Banda [19] demonstrate the application of the LQG with Loop Transfer Recovery control design technique in arriving at a robust vibration control system for large space structures modeled as a two-bay truss. Ferhat and Sultan [20] implement an LQG controller in order to develop a robust controller and analyze the effect of varying system parameters on the stability of the system to compare the impact of the different structural parameters. Ruggiero and Inman [21], examine the possibility of integrating a lead zirconate titanate (PZT) bimorph near the boundary of a strip sample to eliminate detrimental vibration. They develop an LQR controller, simulate it and demonstrate the control over the dynamics of the membrane sample. In a later work [22], they again demonstrate their design of a control system, to eliminate any detrimental vibration of the membrane mirror, using distributed bimorph actuators. They implement a two-dimensional (2D) LQR controller for vibration suppression after providing a deeper insight into the modeling of a membrane with an attached PZT bimorph. Guo et al. [23] propose the combination of a phase compensation active control strategy with a traditional proportional integral (PI) regulator for stability control of a flexible solar array driving system. They demonstrate that the proposed phase compensation active control strategy has high phase margin, excellent stability and significantly improves the system dynamic performance. Ferrari and Amabili [24] demonstrate the active vibration control of a sandwich plate by non-collocated

arrangement of piezoelectric actuators using positive position feedback (PPF) control. They test the Single-Input Single-Output (SISO) PPF and tune the controller transfer function parameters according to the measured values of modal damping and determine the participation matrices for the Multi-Input Multi-Output (MIMO) control experimentally. Ferhat [25] develops MIMO controllers in the form of second order vectors using reduced Algebraic Riccati Equation and the Hamiltonian approach. Both approaches are similarly effective on vibration suppression, while they both have different special requirements in order to be implemented. An et al. [26] design a controller with time-delayed acceleration feedback. They implement the developed controller on a cantilever beam for different controller gain–delay combinations, and evaluate the control performance by comparing it to that of an acceleration feedback controller. He et al. [27] implement a boundary control approach to control a two-link rigid-flexible wing whose design is based on the principle of bionics to improve the mobility and the flexibility of aircraft. They develop a control strategy to restrain the vibrations in bending and twisting deflections of the flexible link of the wing and achieve the desired angular position of the wing. They prove that the wing system is stable using Lyapunov's direct method and demonstrate effectiveness of designed boundary controllers through numerical simulations.

In early research and more so recent literature there is an increasing focus in the application of adaptive active control methods for vibration control. Gustafon and Maybeck [28] present the development and performance of moving-bank multiple model adaptive control algorithm for quelling vibrations induced in the SPICE 2 space structure. The controller implements a parallel bank of Kalman filters and LQG controllers. Takács et al. [17] present a real-time application of explicit model predictive control (EMPC) to minimize the tip deflections of an aluminum cantilever beam using piezoceramic actuators. They design the EMPC algorithm such that it is running as a stand-alone and in real-time on a microcontroller, gaining its feedback from position measurements and supplying input to the control system via an operational amplifier. They also compare the model predictive controller

performance to an open-loop case without control and a PPF controller. Xu et al. [29], propose an online learning fuzzy control algorithm for the vibration control of smart truss structures. The utilized algorithm comprises a reward function, a Q learning algorithm, a rule base generator and a conventional fuzzy controller. They also demonstrate that the proposed control algorithm performs better than fuzzy control. Yang and Lee [30] implement neural networks for system identification and vibration suppression of a smart structure. They develop three neural networks, one for system identification, the second for on-line state estimation, and the third for vibration suppression. They demonstrate that the neural networks can identify, estimate, and suppress the vibration of a composite structure by the embedded piezoelectric sensor and actuator, through analysis and in experiment. Homaifar et al. [31] focus on methods for achieving active damping on plate structures by use of discrete point piezoelectric sensors and actuators. They build a digital control system to test the Fuzzy Type II control technique, using MATLAB-SIMULINK modeling software and demonstrate that the off-line simulation results of the control method are robust and efficient in the suppression of steady-state resonance vibrations.

1.1.2.3 Reinforcement Learning Control in Aerospace

RL is a branch of machine learning [32] that allows an intelligent agent to learn a behavior policy through interactions with its environment. Using RL for control allows strategies to be learned rather than designed. RL can act as the sole controller using neural networks as the control agent or can supplement another controller and calculate the control gains for adjustment. This section summarizes some noteworthy research detailing the application of reinforcement learning control in aerospace operations.

Bohn et al. [33] develop a RL controller capable of stabilizing the attitude of a fixed-wing unmanned air vehicle (UAV) to a given attitude reference, exploring the use of RL methods for low-level control. They implement a flight simulator tailored to the Skywalker X8 flying wing, in which the RL controller is tasked with controlling the

roll and pitch angles, and the airspeed of the aircraft. Barros and Colombini [34] employ a soft actor critic (SAC) algorithm to perform low-level control of a UAV in the go-to-target task. Hovell and Ulrich [35] introduce a guidance strategy for spacecraft proximity operations, which uses deep reinforcement learning. They present a proof-of-concept spacecraft pose tracking and docking scenario, in simulation and experiment, to test the feasibility of the proposed approach. They use control theory alongside deep reinforcement learning to lower the learning burden and facilitate the transfer of the learned behavior from simulation to reality. Federici et al. [36] investigate the use of machine learning techniques for real-time optimal spacecraft guidance during terminal rendezvous maneuvers. They compare the performance of two well-studied deep learning methods for control problems, Behavioral Cloning and RL, on a sample linear multi-impulsive rendezvous mission. Koch et al. [37] investigate the performance and accuracy of the inner control loop providing attitude control when using flight control systems trained with the RL algorithms, Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO). They develop an open-source, high-fidelity simulation environment to train a flight controller attitude control of a quadrotor through RL then use their environment to compare the performance of the RL controllers to that of a PID controller to identify if using RL is appropriate in high-precision, time-critical, flight control. Li et al. [38] combine the stability of conventional feedback PID controllers with the self-improving performance of model-free RL techniques, to develop a more practical application to UAV control. Wilson and Riccardi [39] address the problem of long training times and sensitivity in performance to hyperparameters that arise when RL is used for control in space applications. They present a 3-Degree-of-Freedom (DOF) powered descent problem with uncertainties in the initial conditions.

Guilherme et al. [40] show the feasibility of applying RL methods to optimize a stochastic control policy to perform the position control of a quadrotor. Gaudet et al. [41] develop a deep reinforcement learning based approach for six Degree-of-Freedom (DOF) planetary powered descent and landing. They propose a navigation

system capable of estimating the lander's state in real-time and a guidance and control system that can map the estimated lander state to a commanded thrust for each lander engine. Willis et al. [42] use neural RL to control a spacecraft around a small celestial body whose gravity field is unknown. Vedant et al. [43] investigate the ability of a general RL agent to find an optimal control strategy for spacecraft attitude control problems by presenting two attitude control systems (ACS). They consider a general ACS problem with full actuation, but with saturation constraints on the applied torques and an attitude control problem with reaction wheel based ACS.

1.1.2.4 Reinforcement Learning Vibration Control

The use of RL for control in aerospace applications is a field of research with a lot of interest currently. Even so, there is still limited literature available on RL for vibration control. Most of the existing literature is very recent and has been summarized in this section to provide the reader with a brief understanding of the progress of the research in this field.

Long et al. [44], combine RL and sliding mode control, to improve the tip positioning and tracking accuracy of the hybrid-structured flexible manipulator. They develop an actor-critic based reinforcement learning controller that selects a strategy to output the compensation torque to reduce tip amplitude, effectively suppressing vibration after interaction with the environment. Qiu et al. [45], analyze the vibration characteristics of coupled flexible beams and design effective controllers to suppress vibration. They propose a three-coupled flexible beam with multi-body coupling, close frequency and multi-modal vibration. They establish an initial model by FEM and then modify it by experiments according to the characteristics of close modes. They design a PPO-based RL controller to suppress residual vibration that learns off-line and adopts the theoretical finite element model based on system identification. They show that the PPO RL algorithm can compensate for nonlinearity and uncertainty of the investigated experimental system by adopting a nonlinear function

and that it illustrates a better vibration suppression effect. In another work, Qiu et al. [46], train modal controllers using an RL approach to improve the vibration suppression effect of a flexible hinged plate. They place an emphasis on developing a method to simultaneously control the bending and torsional vibration modes by using combined dual-channel piezoelectric actuators, with the aim of reducing the number of attached PZT patch actuators.

Finally, Wanyonyi et al. [47] implement RL algorithms for vibration control and perform a comparative study on the effect of action space definition on the efficiency of the agent learning and performance. They define the action space as discrete and continuous, present the training and simulation results of vibration suppression on a mass-spring-damper system, and discuss the effect observed from the action space definition.

1.1.2.5 Related Previous Research from METU

Some previous research work related to experimental piezoelectric actuation, vibration control, and RL has been done and published as notable works and dissertations by engineering students from Middle East Technical University. The works focusing on piezoelectric actuation include Comez et al. [48, 49] and Harputlu et al. [50]. Those that have a focus on active vibration control include Ekici [51] and Aksoy [13] while the work with a focus on RL control include Kopşa [52] and Uğurlu [53]. More literature on related topics can be found in [54, 55, 56, 57].

1.2 Research Motivation

From as early as 1996, research into the feasibility of deployable space systems was underway. To demonstrate whether the Inflatable Antenna Experiment (IAE), a deployable space antenna, could be packaged efficiently and deployed successfully in space, all while being developed at a low cost, NASA sent out the IAE in May 1996 for a flight experiment. The IAE was deployed successfully, though in an

uncontrollable manner, and demonstrated the most important thing at the time, that the lightweight antenna had high mechanical packaging efficiency and was able to deploy successfully and maintain structural performance [58]. In the same fashion, ongoing in 2023, NASA is planning to demonstrate the successful space deployment of solar arrays, antennas, drag sails, and solar sails from small satellites [59]. The deployable antennas, solar sails, and similar space structures are thin enough to enable such a demonstration and future applications in space. Thin structures have many attributes that make their use in aerospace structures common. They are foldable and flexible all while maintaining their structural performance [2]. During operations, vibrations may arise in the structure from the perturbations on onboard systems, thermal deformations, external disturbances, and the complex space environment or even from commanded maneuvers like slewing [60, 19]. The vibrations on the aerospace structures must be controlled for satisfactory mission performance and to prevent structural damage. This requires the reduction of the structural vibrations to zero, or to an acceptable level, within a given time [19]. To achieve vibration reduction or elimination, the techniques used are; passive control and active control. Passive control refers to the modification of the stiffness, mass, and damping of the vibrating system to make it less responsive to its environment. On the other hand, active control implies the use of external assistance to drive devices that will act on the structure to generate a vibration that will cancel the initial one [14].

1.3 Thesis Contribution

The objectives of this study are outlined in this section. They provide the contribution that this work has to research in the field of vibration control of thin structures. The said objectives are:

- To provide results of RL control for a thin beam with pinned (simply supported) boundary conditions. To the author's knowledge, this would be the first work focused on a thin beam with pinned supports.

- To develop a RL controller that can handle parameter and external disturbance uncertainties. A unique contribution here is including the effect of thermal loads in space structures while developing the controller. Because of thermal loading on space structures, a dimension change is observed and certain inertial beam constants are bound to change. This work provides one of the first, if not the first, illustrations of the effect of parameter uncertainty on the performance of a RL controller, for vibration control in thin beams.
- To provide a logical breakdown of the reward-shaping process for the RL controller. This research work aims to provide the reader with some insight on the process of define the reward function and the effect that the reward function has on the agent's ability to learn.
- To review the performance of off- and on-policy RL algorithms for the vibration control of a thin pinned-pinned beam.

1.4 Thesis Outline

In Chapter 2, the mathematical system model is presented. An analytical beam model and the respective mode shapes of the beam are derived. Then, piezoelectric actuators are analyzed and discussed and a transfer function representing the beam-actuator system is derived. Finally, the finite element analysis of the beam and actuator system is derived, the mode shapes are illustrated and the natural frequencies of the system are given.

Chapter 3 presents the system analysis and model uncertainty of the system. State space matrices are obtained by transforming the transfer function that represents the beam-actuator system. The observability and controllability of the obtained state space matrices are investigated and the results are highlighted. Thereafter, one of the core ideas behind this research, system uncertainty, is discussed and justified. A section discussing model uncertainty and deriving a parameter-uncertain model is presented.

Chapter 4 gives the basics of the RL framework. There, the elements of RL are introduced and reviewed. Thereafter, a branch of RL algorithms is introduced and the selected algorithms, soft actor-critic and proximal policy optimization, for the RL controller in this work are studied. Furthermore, the use of reinforcement learning as the control method is discussed and justified.

In Chapter 5, the details of the controller development and design are provided. The RL controller architecture that is designed in Simulink is illustrated and explained. Each subsystem in the block diagram is described with the purpose it serves. The RL agent neural network structure and the training options are also provided. Finally, a detailed description of the reward-shaping process is provided alongside an improved reward function that is used to obtain better performance.

Chapter 6 highlights all the results obtained from this work. The training results are shown and discussed first. Thereafter, the controller simulation results are presented and reviewed. Then, a brief comparative study of the reward function and its effect on the learning and simulation results of the agents is carried out and described.

Finally, a conclusion to the research provided in the thesis is given and suggestions for future work are provided in Chapter 7.

CHAPTER 2

MATHEMATICAL SYSTEM MODEL

The mathematical model of the beam and the derivation of the beam and piezoelectric actuator model are discussed in this section. First, the beam model is highlighted, and the natural frequencies and the corresponding modes are calculated and presented for a pinned-pinned beam. Then the derivation of the beam and piezoelectric actuator system dynamics is reviewed. Later on, details of the finite element method model of the beam with the piezoelectric actuators are provided.

2.1 Analytical Beam Model

A flexible, thin aluminum pinned-pinned beam with a pair of actuators was modeled with the Euler-Bernoulli Beam theory. The beam mode shapes are obtained from the closed form solution of the equation for the thin beam theory equation. Furthermore, a derivation of the transfer function of the system relating the elastic deflection of the beam to a force applied by an actuator is made and utilized for the creation of the reinforcement learning environments later on.

Note that the thin beam is merely a simplification of the larger 2-dimensional thin space structure by taking a 1-dimensional thin beam strip. The goal of this work is to illustrate whether the controller scheme can handle parameter uncertainty and varying initial displacements to the system. Therefore, the choice of a thin beam with pinned-pinned boundary conditions was made with this consideration.

2.1.1 Derivation of Flexible Beam Mode Shapes

In order to describe the dynamics of the pinned-pinned flexible beam, the Euler Bernoulli theory is used to derive the equation of motion. Some simple schematics of the beam in consideration for this work are shown in Figure 2.1 and Figure 2.2. Note that A and B represent the pinned ends of the beam while L_a and L_m represent the actuator length and beam length respectively. The aim of this research is to control the transverse vibration of this beam, i.e. the displacement in the z direction. This notation is only used for the schematic representation, and a different notation is utilised for the analytical calculations and to build the numerical model as well, as it will be described in later sections.

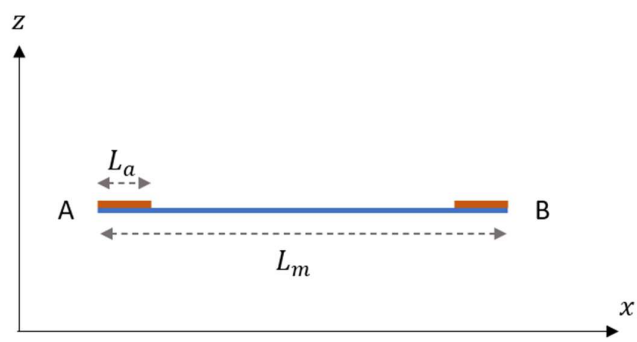


Figure 2.1: Simple schematic of the beam in consideration, side view

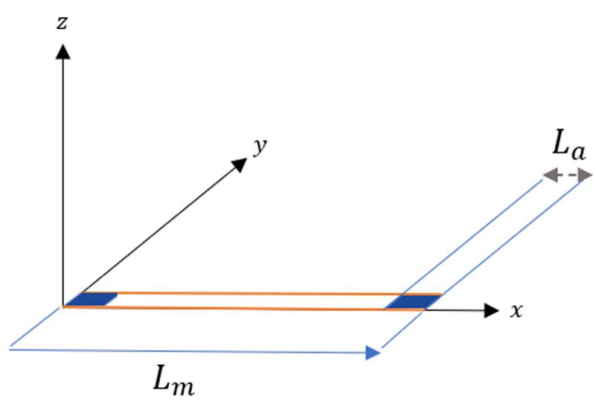


Figure 2.2: Simple schematic of the beam in consideration, isometric view

The partial differential equation describing the dynamic of the forced transverse vibration of a uniform beam is [61]:

$$EI\frac{\partial^4 w}{\partial x^4}(x,t) + \rho A\frac{\partial^2 w}{\partial t^2}(x,t) = f(x,t) \tag{2.1}$$

where, E is the elastic modulus of the beam, I is the moment of inertia of the beam, ρ is the uniform mass density of the beam, A is the cross-sectional area of the beam, w is the transverse displacement of the beam, $f(x,t)$ is any external force applied to the beam.

Using the separation of variables, the closed form solution of the homogeneous equation is obtained as [61]:

$$W(x) = C_1 cos\beta x + C_2 sin\beta x + C_3 cosh\beta x + C_4 sinh\beta x \tag{2.2}$$

where C_1, C_2, C_3, C_4 are all different constants and can be found from the boundary conditions. The boundary conditions at the pinned (simply-supported) end of a beam are zero deflection and no bending moment [62], i.e.; $w = 0,\ EI\frac{\partial^2 w}{\partial x^2} = 0$.

The natural frequencies of thin beams are calculated as [62]:

$$\omega_n = (\beta_n l)^2\sqrt{\frac{EI}{\rho A l^4}} \tag{2.3}$$

Where βl is calculated based on the type of boundary the beam has. For the pinned-pinned case, the value of βl is obtained from the equation [61]:

$$sin\beta l = 0 \tag{2.4}$$

Considering all this, the mode shapes for a pinned-pinned beam can be obtained from [62]:

$$W_n(x) = C_n[sin\beta_n x] \tag{2.5}$$

Ultimately, the transverse displacement is expressed in the form of an infinite series [62]:

$$w(x,t) = \sum_{i=1}^{\infty} W_i(x)\eta_i(t) \tag{2.6}$$

Where $W_i(x)$ represents the ith mode shape and $\eta_i(t)$ is the corresponding generalized displacement.

2.1.2 Piezoelectric Actuators

The direct piezoelectric effect is the property of a material to generate an electric charge on its surface in response to the application of external mechanical stress. When this happens the material changes its polarization. Conversely, the indirect piezoelectric effect is the property of a material to produce mechanical strain when an electric charge is applied to it [63]. To further illustrate this phenomenon, Figure 2.3 is provided.

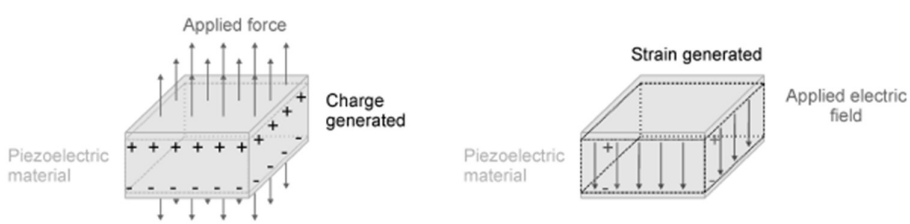


Figure 2.3: Schematic illustrating direct and indirect piezoelectric effect [64]

The vibration of the pinned-pinned beam structure is controlled by distributed piezoelectric actuators. Piezoelectric materials are commonly used to induce [50, 49] or suppress [9, 65, 13] vibration in structures. In this work, two uni-morph piezoelectric actuators are used to implement control over the system. Some naturally occurring piezoelectric materials are quartz crystals that are inherently polar within their molecular structure. Despite being naturally piezoelectric, crystals have a very low capacity to generate electric charges and require a very high voltage of electric current to obtain considerable mechanical strain to be used in most industrial applications. However, synthetic piezoelectric materials have a very high capacity and are easy to manufacture in bulk and various shapes. These materials are

not piezoelectric at first. They are inherently isotropic materials. However, under high electric applications, they become anisotropic and alter to piezoelectric materials permanently. The process of applying a required electric field to polarize a material is called poling [64]. This is part of the manufacturing process. It can be inversed by applying the opposite sign of the electric field, and this inversion process is called depoling. The ability to manufacture piezoelectric materials allows us to have the desired shape and characteristics of a piezoelectric actuator within certain limits. Today, there are a few different types of piezoelectric actuators commercially available for different uses. The most common ones are piezoelectric sheets which are plate-like structures that are considered two-dimensional. They can be stacked as multiple layers for different uses, and are called piezostack actuators. Each layer adds more capacity. They can be used as uni-morph or bimorph. Uni-morph attachment is when the piezoelectric sheet is on one surface of the structure while bi-morph attachment is described as when there is a piezoelectric sheet on both surfaces of the structure.

Up to a certain level of electric field and strain, piezoelectric materials behave linearly. One of the standard forms of piezoelectric constitutive equations is given in the equations below [63]:

$$S_{ij} = s_{ijkl}^{E} T_{kl} + d_{kij} E_k \tag{2.7}$$

$$D_i = d_{ikl} T_{kl} + \varepsilon_{ik}^{T} E_k \tag{2.8}$$

Where, S_{ij} is the mechanical strain tensor, s_{ijkl}^{E} is the compliance tensor, d_{kij} is the piezoelectric coefficient tensor, T_{kl} is the mechanical stress tensor, ε_{ik}^{T} is the permittivity tensor, E_k is the electric field tensor, and D_i is the electric displacement tensor. Note that the subscripts of the stress and strain tensors in Equations (2.7) and (2.8) are labeled based on Voigt's notation [66] and are therefore vectors of six components [63].

In matrix form, the same equations are represented as [63]:

$$\begin{bmatrix} S \\ D \end{bmatrix} = \begin{bmatrix} s^E & d^t \\ d & \varepsilon^T \end{bmatrix} \begin{bmatrix} T \\ E \end{bmatrix} \tag{2.9}$$

Where the superscripts E and T indicate that those constants are evaluated at constant electric field and stress, respectively. Note that the piezoelectric coefficient tensor is transposed for this calculation.

For this work, the actuator used to help suppress the beam vibration is a macro fiber composite (MFC) piezoelectric actuator. MFC actuators (and sensors) are the leading flexible, reliable, high-performance, low-profile piezoelectric devices. Invented in 1999 by NASA, the MFC piezoelectric device was commercialized in 2002 by Smart Material [67]. MFC piezoelectric patches are composed of rectangular piezo ceramic rods, arranged between layers of adhesive, electrodes, and polyimide film as seen in Figure 2.4. The electrodes are interdigitated and arranged to facilitate in-plane poling, actuation, and sensing. Further discussion of interdigitated electrodes and their piezoelectric effect is seen in section 2.1.2.1. The MFC piezoelectric material can be embedded within a composite structure or applied as a thin, flexible sheet onto various structures. If voltage is applied, it acts as an actuator, suppressing or generating vibrations by bending or distorting materials. In the absence of voltage, it functions as a sensor by detecting deformations and vibrations. Additionally, the MFC is ideal for harvesting energy from vibrations. Some advantages of MFC piezoelectric actuators include their flexibility, durability, and reliability. They have increased efficiency in strain actuation, offer directional actuation and sensing, and are damage-tolerant. The MFC is available in elongator (d33 mode) and contractor (d31 mode) variations, conforms to surfaces, can be easily embedded, and comes in an environmentally sealed package. It has demonstrated high performance and is available in different piezo ceramic materials.

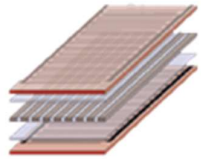


Figure 2.4 MFC actuator configuration [67]

2.1.2.1 Piezoelectric Actuator Model

In this work, two MFC P1-type piezoelectric patches are attached to the top of the beam structure assumably with some bonding agent, like epoxy glue. The MFC patches have an actuating capability, which is governed by the piezoelectric constant d_{33}. The d_{33} effect, also known as the longitudinal piezoelectric effect, is present in actuators with interdigitated electrodes, where the electric field is aligned in the plane of the structures, as opposed to the d_{31} effect where the electric field is aligned in the thickness direction [68]. Note that for actuators with either effect, the deformation occurs in the plane direction as illustrated in Figure 2.6. Therefore, applying a voltage in the same direction as the polarization of the MFC P1 piezoelectric actuators results in elongation along the x-axis, considering the coordinate axis systems shown in Figure 2.2 and Figure 2.6. If the voltage is applied in the opposite direction then the actuator contracts along the x direction. The voltage applied generates a moment that is applied to the structure. Considering a voltage V_a applied to one of the actuators, the moment generated (M_a) can be calculated as [11]:

$$M_a(t) = C_a V_a(t) \tag{2.10}$$

Where the constant C_a is calculated as [11]:

$$C_a = \frac{1}{2} E_a d_{33} b_a (t_b + t_a) \tag{2.11}$$

Where, E_a is the Young's modulus of the piezoceramic actuator, d_{33} is the electric charge constant, b_a is the width of the actuator, t_b is the thickness of the beam, and t_a is the thickness of the actuator.

Note that the interdigitated illustrations of the piezoelectric actuators shown in Figure 2.5(b) and Figure 2.6 represent the actuator chosen for this research work as they illustrate the d_{33} effect.

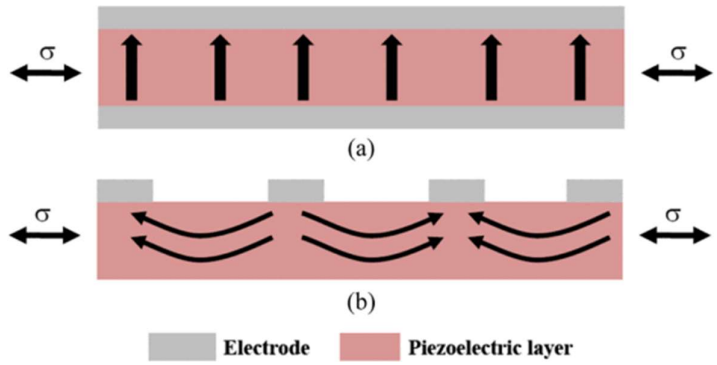


Figure 2.5: The poling direction observed in conventional (a) and interdigitated (b) electrode configuration in piezoelectric actuators [69]

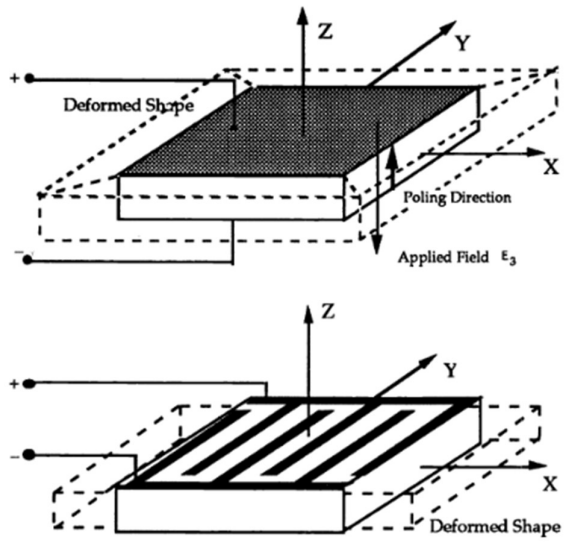


Figure 2.6: Conventional (top) and interdigitated (bottom) electrode configuration in actuators [68].

This section aims to present the derivation of the beam and piezoelectric actuator model. In order to generate the model, the transfer function of this system is derived. In section 2.1.1, it is established that the transverse beam displacement can be represented as [62]:

$$w(x,t) = \sum_{i=1}^{\infty} W_i(x)\eta_i(t) \tag{2.12}$$

Plugging this into the equation of motion;

$$\sum_{i=1}^{\infty} [\rho A W_i(x)\ddot{\eta}_i(t) + EIW_i''''(x)\eta_i(t)] = M_a(t)\frac{\partial^2 R(x)}{\partial x^2} \tag{2.13}$$

Where; $M_a(t)\frac{\partial^2 R(x)}{\partial x^2}$ represents the force applied to the system by an actuator and the expression $R(x)$ can be written as [11]:

$$R(x) = H(x - x_{a1}) - H(x - x_{a2}) \tag{2.14}$$

Where H is the Heaviside function and x_{a1}, x_{a2} represent the distance of the edges of a piezoelectric patch from the fixed coordinate system onto the body of a structure [70]. Note that the mode shape functions have the orthogonality property [62] and therefore:

$$\int_0^1 W_i^2(x)dx = 1 \tag{2.15}$$

While

$$\int_0^1 W_i(x)W_j(x)dx = 0, i \neq j \tag{2.16}$$

Also note that,

$$W_i''''(x) = \beta_i^4 W_i(x) \tag{2.17}$$

Substituting this expression into the equation of motion and multiplying through by $\int_0^1 W_i(x)dx$, the following equation is obtained:

$$\sum_{i=1}^{\infty}[\rho A\ddot{\eta}_i(t) + EI\beta_i^4\eta_i(t)] = M_a(t)\int_0^1 \frac{\partial^2 R(x)}{\partial x^2}W_i(x)dx \tag{2.18}$$

Evaluating the right hand side of the equation:

$$M_a(t)\int_0^1 \frac{\partial^2 R(x)}{\partial x^2}W_i(x)dx = C_a[W_i'(x_{a2}) - W_i'(x_{a1})]V_a \tag{2.19}$$

Substituting the expression into the equation:

$$\sum_{i=1}^{\infty}[\rho A\ddot{\eta}_i(t) + EI\beta_i^4\eta_i(t)] = C_a[W_i'(x_{a2}) - W_i'(x_{a1})]V_a \tag{2.20}$$

Dividing through by ρA and including damping into the system, the final form of the governing equation becomes:

$$\sum_{i=1}^{\infty}\left[\ddot{\eta}_i(t) + 2\zeta\omega_{n,i}\dot{\eta}_i(t) + \omega_{n,i}^2\eta_i(t)\right] = k_a[W_i'(x_{a2}) - W_i'(x_{a1})]V_a \tag{2.21}$$

Where $\omega_{n,i}^2$ and k_a are calculated as:

$$\omega_{n,i}^2 = \frac{EI}{\rho A}\beta_i^4 \tag{2.22}$$

$$k_a = \frac{C_a}{\rho A} \tag{2.23}$$

From the final form of the equation of motion, the transfer function of the beam-actuator system is obtained as:

$$G(s) = \sum_{i=1}^{\infty} \frac{k_a[W_i'(x_{a2}) - W_i'(x_{a1})]}{s^2 + 2\zeta\omega_{n,i}s + \omega_{n,i}^2} \tag{2.24}$$

Note that the numerator is a constant value that relies on the properties and placement of the actuator(s).

2.2 Finite Element Analysis

In this subsection, a brief explanation of the finite element method is provided. Then a detailed description of the finite element model of the beam created is given. The beam and actuator properties are summarized and presented. Finally, some preliminary results of numerical modal analysis are illustrated in graphs.

2.2.1 Finite Element Model

Finite element method analysis of the beam is carried out in ANSYS. The geometry of the beam, with and without actuators, is constructed in ANSYS Mechanical and thereafter, modal analysis is carried out. The material of the beam is assigned as an aluminium alloy. The length, width, and thickness of the beam are taken as 280mm, 20mm, and 1mm respectively, to satisfy the thin beam requirements. As discussed previously an MFC piezoelectric actuator is chosen to exert force over the structure. M2814-P1 is selected from the Smart Material array of actuators. With an overall length of 38mm, a width of 20mm, and a thickness of 0.5mm, two actuators are modeled on top of the beam, in the thickness direction. Figure 2.7 shows a schematic of the MFC P1 type, that operates with the d_{33} effect and is commonly used as a powerful actuator and sensitive sensor. The material properties of the aluminium alloy and the piezoelectric actuator used, like Young's Modulus, Poisson's ratio, density, and coupling coefficients, are summarized in Table 2.1.

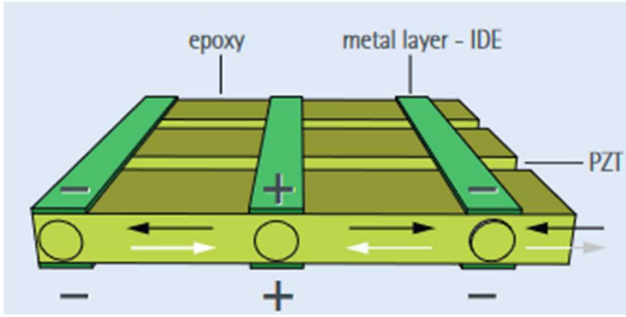


Figure 2.7: MFC P1 type schematic

Table 2.1: Geometrical measurements and material properties of the system

		Beam	*Actuator*
Material type		Aluminium alloy	MFC P1
Young's Modulus	$E\ (Pa)$	71e9	33.6e9
Poisson's ratio	υ	0.3	0.31
Material density	$\rho\ (kg/m^3)$	2770	5440
Length	$L\ (mm)$	280	38
Width $w\ (mm)$		20	20
Thickness	$t\ (mm)$	1	0.5
Coupling coefficient $d_{31}(pmV^{-1})$		-	-210
Coupling coefficient $d_{33}(pmV^{-1})$		-	460

The first three natural frequencies and modes of the pinned-pined beam are presented in this section. The modes were computed in MATLAB and ANSYS and the results were summarized in Table 2.2. The beam's natural frequencies were calculated first, without any actuators. The error difference between the analytically and numerically obtained natural frequencies is calculated and found as 1.18%. Because the error is

small, the modal analysis results of the beam with actuators can be assumed accurate enough for our purpose.

Table 2.2: Natural frequency of the pinned-pinned beam calculated on MATLAB and ANSYS

Natural Frequency	*MATLAB(Beam)*	*ANSYS(Beam)*	*ANSYS (Beam with actuator)*
$\omega_{n,1}\ (Hz)$	29.282	29.288	49.491
$\omega_{n,2}(Hz)$	117.13	117.22	155.15
$\omega_{n,3}(Hz)$	263.54	264.02	314.58

The model of the undeformed beam-actuator system and the mode shapes corresponding to the first three natural frequencies are shown in Figure 2.8 to Figure 2.11 as follows:

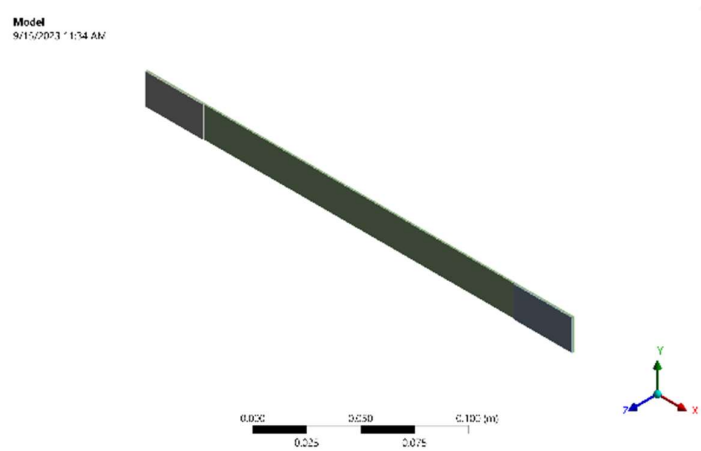


Figure 2.8: Undeformed model the beam-actuator system

Note that the natural frequencies calculated for the beam-actuator system on ANSYS are used in the transfer function to account for the effect of adding the piezoelectric actuators to the thin beam.

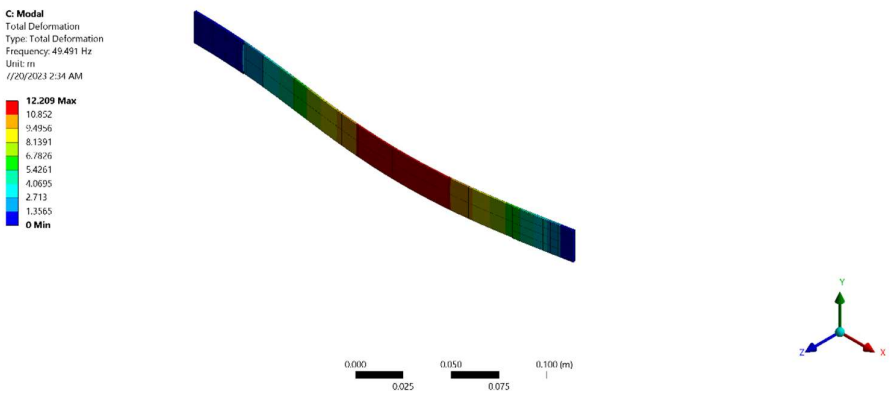


Figure 2.9: First mode shape of the beam-actuator system

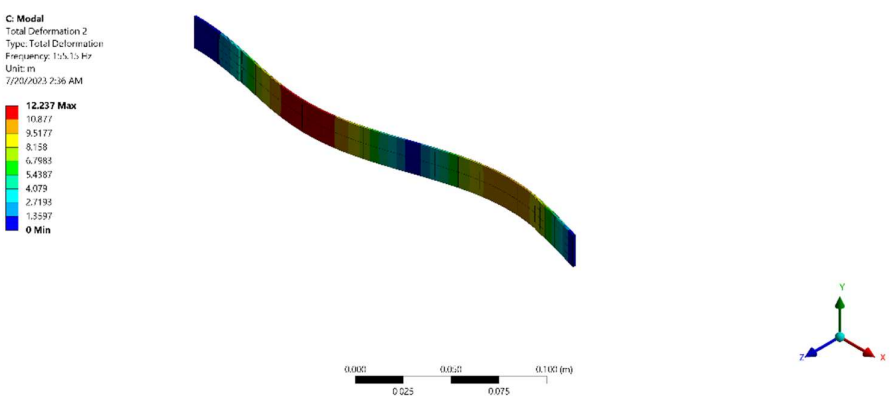


Figure 2.10: Second mode shape of the beam-actuator system

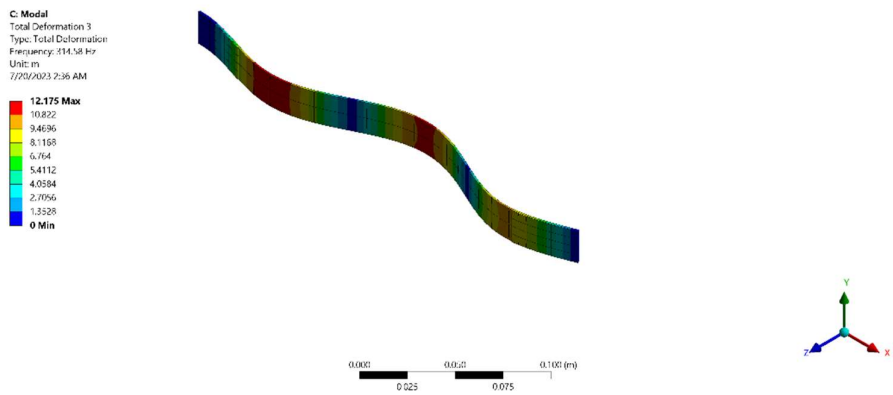


Figure 2.11: Third mode shape of the beam-actuator system

CHAPTER 3

SYSTEM ANALYSIS AND MODEL UNCERTAINTY

In this chapter, a system analysis is carried out after the system state space matrices are obtained to determine the observability and controllability of the system in question. The observability and controllability Gramian matrices are calculated and the rank is inspected to determine whether the system is fully observable and controllable. To fulfill the aim of this research, system uncertainty is introduced, discussed, and modeled in this section. The importance of considering uncertainty in our mathematical models and system parameters is discussed. As a final step, the parameter uncertain model for this work is defined.

3.1 State Space Model

For this section and the subsequent ones, a continuous-time state space system representation is considered in the form [71]:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{19}$$

$$y(t) = Cx(t) + Du(t) \tag{3.20}$$

where, $x(t)$ is the state vector, $\dot{x}$ is the first derivative of the state vector, $u(t)$ is the control signal, y is the output vector, A is the state matrix, B is the control matrix, C is the output matrix, and D is the feedthrough matrix.

The state space system that represents the beam-actuator dynamics is derived directly from the transfer function presented in 2.1.2.1. For the purpose of this work, only the first natural frequency is considered as it has the biggest effect on the vibration amplitude of the system. Based on these parameters, $A, B, C,$ and D matrices of the state space systems are obtained defined as: space systems are defined as:

$$A = \begin{bmatrix} -1.4719 & -264.45 \\ 128 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.6104e-4 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0.5844e-4 \end{bmatrix}, \qquad D = 0$$

3.2 Observability

In this subsection, the observability of the system is inspected using the Gramian matrix. If every change of the state affects every element of the output vector, the system is considered observable. This means that the system is considered as completely observable if every state $x(t_0)$ can be determined from the observation of $y(t)$ over a finite time interval, $t_0 < t < t_1$. The system's observability is inspected by computing the observability matrix and checking its rank. The observability matrix is defined as [71]:

$$W_O = \begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ \cdot \\ CA^{n-1} \end{bmatrix} \tag{3.3}$$

The observability matrix is calculated and obtained as:

$$\begin{bmatrix} 0 & 0.0001 \\ 0.0075 & 0 \end{bmatrix}$$

Its rank is evaluated. Because the dynamics of only the first mode are considered, the system obtained is of order n, where n is equal to 2, and the rank of the observability matrix is 2; therefore, the system is observable.

3.3 Controllability

The controllability of the system is also inspected using the Gramian matrix. If it is feasible to generate an unrestricted control signal, the system is considered to be controllable at time $t = t_0$. This means that it is possible to transition the initial state

to any desired final state within a finite time interval. When every state of the system is controllable in this manner, it is referred to as completely state controllable. The controllability matrix is calculated as [71]:

$$W_C = [B \quad AB \quad . \quad . \quad . \quad A^{n-1}B] \tag{3.4}$$

The controllability matrix is computed as:

$$\begin{bmatrix} 0.0001 & -0.001 \\ 0 & 0.0078 \end{bmatrix}$$

The rank is evaluated and a full rank of 2 is obtained and proves that the system is fully controllable.

3.4 System Uncertainty

Understanding and managing uncertainty is a crucial task for control engineers. A successfully designed controller should be robust in order to maintain performance and stability, even with some uncertainties. System uncertainty could be a result of disturbance signals e.g. measurement noise, external disturbances, and/or be caused by dynamic perturbations like changes or errors in plant parameters, error during modeling, neglected dynamics or nonlinearities, and a reality gap between the mathematical model and the real system. Because of this, and the uncertain dynamics and conditions of the space environment, it is paramount that the controllers designed for space vehicles and their subsystems are robust and adaptive.

At the core of robust control is the concept of uncertain Linear Time-Invariant (LTI) systems. Model uncertainty arises when the system gains or other parameters are not precisely known or can vary within a given range. Uncertain pole and zero locations and uncertain gains are some examples of real parameter uncertainties.

One of the main aims of this research work is to illustrate the benefits of using RL for vibration control instead of more conventional approaches. Because of its robust and adaptive nature, RL control can handle model or parameter uncertainty, and in

much simpler ways than other robust control approaches. This is vital for the thin structures that operate in space for two main reasons. Consider a case where the space vehicle adjusts its attitude or deploys structures, there will be low-frequency loads applied to the structure that are unknown and will likely trigger the system and induce vibration. In this case, the external disturbance to the system becomes an uncertainty and the RL controller response can still be tailored to suppressing vibration even without prior training for that specific force input. For the second case, structural changes can occur due to thermal loads on space structures setting off a change in length [72] and consequently width or thickness of the structure, because of the tendency to maintain volume. In this case, because of the change in the dimensions of the space structure the parameters dependent on them, like moment of inertia, also shift and change accordingly. Because of these sources of uncertainty in the systems, it is important to factor them in during the modeling process and develop a controller that is robust and has good performance.

3.4.1 Parameter Uncertainty

Considering the case described in section 3.4, where the variables representing some of the system properties, e.g., moment of inertia, can change, parameter uncertainty is modeled into the system to account for instances like these. Parameter uncertainty modeling within a system can be categorized into one of three groups:

- Additive Deviation: In this case, the parameter exhibits an additive deviation from its nominal value, i.e., $n = n_{nominal} \pm n_{deviation}$.
- Range: Where the parameter displays variability within a specified range around its nominal value, i.e., $n_1 \leq n_{nominal} \leq n_2$
- Percentage Deviation: Here, the parameter demonstrates a deviation from its nominal value expressed as a percentage.

In order to model the parameter uncertainty, the natural frequency which is directly affected by the moment of inertia, is modeled as a constant within a certain range.

The nominal value of the system's natural frequency is used initially to obtain the state space matrices given in section 3.1. The uncertainty is represented using a range deviation from the nominal natural frequency values. Assuming that the beam maintains its elastic properties is one of the assumptions of the work thus far, therefore, parameter ranges and even external disturbances that result in inelastic behavior of the system are not considered. In order to model parameter uncertainty, the system's natural frequencies are modeled as a range of values within a 5% range of the original values, i.e.:

$$0.95\omega_n \leq \omega_{n,uncertain} \leq 1.05\omega_n \tag{3.5}$$

By changing the value of natural frequency, the transfer function and consequently the state space model representing the system are affected. Note that in this work, only the parameter uncertainty is focused on. The other forms of uncertainty have yet to be justified for the scope of this work and were therefore left out.

CHAPTER 4

REINFORCEMENT LEARNING METHODOLOGY

Reinforcement learning control has been applied to various problems in aerospace engineering as highlighted in Section 1.1.2.3. However, limited research exists on the implementation of reinforcement learning for vibration control. For this work, a comparative study showing the effect of unknown system parameters on the effectiveness of a reinforcement learning vibration controller is studied. In this chapter, the reason reinforcement learning is selected as the vibration control approach is discussed. The reinforcement learning framework is described and the algorithms used for control are discussed. Furthermore, the use of reinforcement learning as the control method is discussed and justified.

4.1 Reinforcement Learning: a branch of Machine Learning

In machine learning, three major learning strategies exist; supervised, unsupervised, and reinforcement learning [73]. Supervised learning refers to the case where a model is trained given a set of inputs and expected responses then evaluated by only giving the model a set of inputs and comparing the response from the model with the true response [32]. Unsupervised learning describes a learning strategy where a data set for a model is given where the response is not known and the model then uses certain techniques to find associations between the features. Both supervised and unsupervised learning algorithms are trained on large sets of data [32].

RL employs a trial-and-error search learning strategy where an agent is trained by taking actions in an environment. The goal of the agent is to maximize a (usually delayed) numerical reward as it interacts with an uncertain environment. All reinforcement learning agents have explicit goals, can receive information from the environments, and can take actions that influence their environments [32]. Delayed

reward and the trial-and-error search are the major distinguishing features of reinforcement learning.

4.1.1 Elements of Reinforcement Learning

Briefly mentioned in the previous section, agents, rewards and environments are some of the key elements of reinforcement learning. Other elements; policy, value function, and model, and those previously mentioned, are described in this section.

An environment [74] is a representation of the problem or task, which an agent interacts with to learn how to behave, with an incentive to maximize the cumulative reward, consequently, solving the problem at hand. An agent [74] is a reinforcement learning algorithm that learns which actions to take within the environment in order to maximize the reward. A policy [75] describes how an agent behaves. It maps the states (from environment observations) to actions to be taken when in those states. A policy may be stochastic or deterministic. A value function [76] specifies the value of a state. It is the total amount of expected reward an agent can accumulate over the future, starting from a particular state. A reward signal [77] indicates which actions are valuable when in a given state. One of the core concepts behind reinforcement learning is that an agent's sole objective is to maximize the total reward it receives in the long run. A model [75] refers to something that mimics the environment, given a state and action, it tries to predict the next state and action and reward. Knowing the model of the environment is not always necessary, a lot of RL algorithms are model-free and learn agent policies accurately.

4.1.1.1 Markov Decision Process

The Markov Decision Process (MDP) [78] is a framework used to formally define reinforcement learning problems. As stated previously, in RL, an agent is placed inside an environment with the goal of maximizing a reward value. The agent interacts with the environment by performing an action determined by a policy. A

reward is then given by the environment to the agent when a particular state is attained.

The state describes the current situation that an agent is in at a particular point in time [32]. For this project, the state is defined as the structure's current vibration amplitude. The action is the possible operation that an agent can perform at a particular point in time to transition from one state to another [32]. The action that can be taken is the displacement applied to the structure by an actuator after a certain voltage is sent to the piezoelectric actuator. To determine the effect of an action at a particular state, a value function is used to estimate the reward of a state-action pair. The value is defined as the expected return in a state (s) when following a policy (π). The value function is derived from the Bellman equation, which is taken from dynamic programming expressing the total reward given by taking an action at a particular state. In terms of a MDP, the value function can be defined as [76]:

$$V^{\pi}(s) = E_{\pi}[R_t|s_t = s] = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\right\} \tag{4.1}$$

where the policy π, defines which action the agent will take at a given state, s_t. E_{π} is the expected value of a random variable when an agent follows a particular policy π, R is the reward value at a state at time t, and γ is a discount factor whose value is between zero and 1. The discount factor is an indication of how much influence future rewards have on the value function. For the thin structure, the value function is a function of the difference between the current and target structure vibration amplitude. Value functions can be represented by a linear or non-linear function. When an agent reaches a particular termination state, the current episode terminates. The agent replays multiple episodes until a reward value, maximum number of episodes, or a particular final state is obtained. With RL, the agent does not have to be aware of the dynamics of the environment to reach its goal, making RL algorithms capable of solving problems too complex to model analytically [32]. An illustration of an MDP is shown below.

4.1.1.2 Value-Based and Policy-Based Reinforcement Learning

Value-based RL methods are a category of reinforcement learning algorithms that have an implicitly defined policy in the form of a learned value function, which defines states and their expected long-term rewards. Therefore, in a given state, the agent chooses the action based on the value function to maximize the expected reward. The state- and action-value methods can both be used to calculate the value function. Some value-based methods are Q-learning, SARSA (state-action-reward-state-action), and value iteration [79]. The advantages of value-based methods are their ease of implementation, adaptability to the type of reinforcement learning problem, stability during training, and their efficiency in function approximation. However, they are very sensitive to environmental noise, converge slowly for larger state spaces, and are difficult to implement for continuous state space problems.

Policy-based RL describes the set of RL methods that have an explicitly defined policy that maps states to probability distributions over actions and saves this policy in their memories. This set of methods is effective in high-dimensional, continuous, stochastic action spaces, and learns stochastic policies. Policy gradient and actor-critic methods are both considered policy-based RL methods. For policy gradient methods, the agent's policy is updated directly while for actor-critic methods, a value function evaluates the actions chosen by a policy, and they both learn simultaneously. Examples of policy-based methods are REINFORCE, proximal policy optimization (PPO), and twin-delayed deep deterministic policy gradient (TD3) [80, 81]. Policy-based methods are sample efficient, can handle large continuous state spaces, and are more robust to noise. However, they are typically harder to understand, more sensitive to hyperparameters, and less stable while training.

4.1.1.3 Function Approximation in Reinforcement Learning

Function approximation is a method that is used to estimate a "true function" that governs the relationship between the inputs and outputs in a system. In RL, the agent chooses an action to perform that affects the environment. It then receives information through the observation (updated state) vector and reward signal, which it uses to pick its next action. The agent picks an action governed by an explicit or implicit policy. This policy maps a state to an action. In some cases, the policy is represented by a table with all possible state-value pairs that indicate which action has the potential largest value for the current state. However, for some reinforcement learning problems, e.g., control problems, the state space is too large to store such a table. For cases like these, function approximation is applied. Many different types of function approximators can be used in reinforcement learning. Some common examples include:

- Linear function approximators: simple functions that can be represented as a linear combination of features. A special important case in gradient descent function approximation [82]
- Neural networks: more complex function approximators that can learn non-linear relationships between states and rewards [83]
- Gaussian processes: a type of probabilistic function approximator that can be used to represent uncertainty in the value function [82]

Function approximation offers increased scalability by allowing a larger state space problem to be solved. Also, using function approximators can enable the reinforcement learning agent to generalize to new states that have not been seen before. Furthermore, function approximators can learn from smaller sets of data than what is required for table-based methods.

Function approximation is a powerful technique that can be used to improve the performance of reinforcement learning algorithms by allowing for large state space problem solutions and better generalization with less complex samples.

4.1.1.4 Neural Networks as Function Approximators

As mentioned in section 4.1.1.3, neural networks can be used as function approximators, to approximate the value function, in reinforcement learning [84]. In reinforcement learning, the value function is a function that maps states to their expected long-term rewards. A neural network can be used to approximate the value function by learning a set of weights that map states to their corresponding expected rewards.

Assuming that the neural network is parametrized by the weights $\bar{W}$, it can calculate the function $F(\bar{X}_t, \bar{W}, a)$, and the learned estimate of the action value function $\hat{Q}(s_t, a)$, for every action a [83].

$$F(\bar{X}_t, \bar{W}, a) = \hat{Q}(s_t, a) \tag{4.2}$$

Neural networks learn by being trained on a dataset of state-action pairs and rewards. The training dataset is used to calculate the error between the neural network's predictions and the actual rewards. The neural network's weights are then updated to minimize this error. There are several advantages to using neural networks as function approximators in reinforcement learning. First, neural networks can learn non-linear relationships between states and rewards. This is important because many reinforcement learning problems involve non-linear relationships. Second, neural networks can generalize well to new states that have not been seen before. This is because neural networks learn to represent the underlying structure of the state space, rather than just memorizing the training data. Third, neural networks can be trained on a relatively small dataset of state-action pairs. This is because neural networks can learn from the relationships between the states and rewards, rather than just the individual state-action pairs.

However, there are some drawbacks to using neural networks as function approximators. Neural networks can be computationally expensive to train. This is because neural networks require a lot of data to learn from and they can be difficult to interpret. This is because neural networks learn a complex set of weights that map states to their corresponding expected rewards. It can be difficult to understand how these weights affect the neural network's predictions, and therefore difficult to alter.

4.2 Actor-Critic Methods

Temporal Difference (TD) learning is one of the central ideas in reinforcement learning. TD methods combine aspects of dynamic programming and Monte Carlo methods, therefore, they bootstrap (i.e. they update estimates before the eventual outcome, based on other learned estimates) and even without having a model of the environment's dynamics, they can learn directly from raw experience [79]. As explained in section 4.1.1.2, policy-based RL computes an optimal policy by directly manipulating and improving the existing policy while value-based RL computes the optimal value function to implicitly find the optimal policy. Actor-critic methods are TD methods that combine the concepts of policy- and value-based RL. They have separate actor and critic memory structures. The actor represents the policy structure that is used to select actions while the critic evaluates the actions taken by the actor by calculating the estimated value function. Using the critic's estimated value function, the agent learns and gradually improves its policy [85]. This means that learning is always on-policy i.e. the critic must learn about and critique whatever policy is currently being followed by the actor. The critique takes the form of a TD error [85]:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{4.3}$$

where δ_t is the TD error, r_{t+1} is the reward at time $t+1$, γ is the discount factor,V is the current value function implemented by the critic, and s_t and s_{t+1} represent the state at time t and $t+1$, respectively. This TD error can be used to evaluate the

action taken in state (s). If the TD error is positive, it suggests that the tendency to select should be strengthened for the future, whereas if the TD error is negative it suggests the tendency should be weakened.

The main advantages of actor-critic methods are that minimal computation is required in order to select actions and they can learn the optimal probabilities for choosing multiple different actions.

The actor-critic algorithms that will be implemented in this project are described in the following sections. Note that one of the aims of this work is to present a comparative study of soft actor-critic (SAC), an off-policy algorithm, and PPO algorithm, an on-policy algorithm, on the solution of the vibration problem of thin structures. The comparative study is carried out to determine the difference in performance of the off- and on-policy algorithms.

4.2.1 Soft Actor-Critic

The soft actor-critic (SAC) algorithm is an off-policy actor-critic algorithm based on the maximum entropy RL framework. Entropy here, can be defined as how unpredictable the agent's actions are. In the maximum entropy framework, the algorithm has a goal to not only maximize rewards but also maximize entropy, i.e. to successfully perform a task while choosing actions randomly. Therefore, SAC presents a modified objective function for optimization. By combining off-policy updates with a stable stochastic actor-critic formulation, this approach proves to be sample efficient and has a more stable convergence [86].

Most off-policy RL algorithms display brittle convergence as they are sensitive to hyperparameters and require a lot of tuning. However, as mentioned above, SAC demonstrates a more stable convergence, all while being sample-efficient. With the increased entropy in our policy, the agent is encouraged to explore, which can advance learning later on. It can also prevent the policy from getting stuck taking a particular action and converging to a bad local optimum [81, 87].

For large continuous domains, like those of control problems, it is necessary for a practical approximation to soft policy iteration to be derived. SAC makes use of this policy iteration where the algorithm alternates between policy evaluation and policy improvement, with a goal to maximize entropy. The objective function that represents this concept can be expressed as [87]:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \alpha H\left(\pi(\cdot\,|s_t)\right)\right] \tag{4.4}$$

where $J(\pi)$ is the objective function based on the policy π, $r(s_t, a_t)$ represents the expected reward at from state s_t while taking an action a_t, H represents the entropy of the policy π, weighted by α. This objective function represents what the algorithm aims to maximize. For this optimization, SAC makes use of three separate networks. A soft Q function, Q parametrized by θ, a state value function V parametrized by ψ, and a policy function π parametrized by ϕ. Each of the networks is trained by minimizing the error in different objective functions.

The state value function is trained by minimizing the objective function shown in [88]:

$$J_V(\psi) = E_{s_t\sim\mathcal{D}}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}_{(s_t,a_t)\sim\pi_\phi}\left[Q_\theta(s_t,a_t) - log\pi_\phi(a_{t+1}|s_{t+1})\right]\right)^2\right] \tag{4.5}$$

where V_ψ is the parametrized state value function, Q_θ is the parametrized state-action value function, $\mathcal{D}$ is the distribution of previously sampled states and actions, and $\mathbb{E}_{(s_t,a_t)\sim\pi_\phi}$ is the expected return from the state s_t while taking the action a_t and following the policy π_ϕ. Note that the state-value function, considering soft policy iteration is given by the soft bellman iteration [88]:

$$Q(s_t,a_t) = r(s_t,a_t) + \gamma\mathbb{E}_{(s_t,a_t)\sim\pi_\phi}\left[Q_\theta(s_{t+1},a_{t+1}) - \alpha\log\left(\pi_\phi(a_{t+1}|s_{t+1})\right)\right] \tag{4.6}$$

where α is the adjusting factor to be tuned, $Q(s_t, a_t)$ is the state-value function, π_ϕ is the adopted policy with the distribution ϕ. The equation for the objective function

illustrates that the goal is to decrease the error between the value prediction and the Q function prediction and entropy, for all the sampled states.

The soft Q function is trained by minimizing the error calculated by taking the difference between the Q function prediction and the sum of the immediate reward from the current state and the expected target value of the successive state. This is expressed by the objective function [86]:

$$J_Q(\theta) = E_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta(s_t,a_t) - \left(r(s_t,a_t) + \gamma\mathbb{E}_{(s_t,a_t)\sim p}\left[V_{\bar{\psi}}(s_{t+1})\right]\right)\right)^2\right] \quad (4.7)$$

where p is the state transition probabiliy. While the policy function is trained by minimizing the error computed by taking the difference between the policy and the exponentiated Q_θ function normalized by another function [86]:

$$J_\pi(\phi) = E_{s_t\sim\mathcal{D}}\left[D_{KL}\left(\pi_\phi(\cdot\,|s_t)\middle|\frac{exp\left(Q_\theta(s_t,\cdot)\right)}{Z_\theta(s_t)}\right)\right] \quad (4.8)$$

The equation introduces the Kullaback-Lieber divergence function, D_{KL}, that determines how different the policy distribution is from the normalized Q_θ function ratio. After some mathematical manipulation, the objective function can be expressed in a more compact form [86]:

$$J_\pi(\phi) = E_{s_t\sim\mathcal{D},\epsilon_t\sim\mathcal{N}}\left[log\pi_\phi\left(f_\phi(\epsilon_t;s_t)|s_t\right) - Q_\theta\left(s_t, f_\phi(\epsilon_t;s_t)\right)\right] \quad (4.9)$$

where f_ϕ is the implicit definition of π_ϕ and ϵ_t here represents an input noise vector sampled from a fixed distribution.

4.2.2 Proximal Policy Optimization

The proximal policy optimization (PPO) is an on-policy actor-critic algorithm that is more sample efficient and simpler to tune than its counterpart policy gradient methods. The PPO agent samples data by interacting with the environment and optimizing a special kind of objective function using stochastic gradient ascent [89].

Policy gradient methods are popular options for continuous control problems but are sensitive to step sizes and have poor sample efficiency. With a step size that is too small, they are very slow, and with a step size too large there is too much noise for the signal and performance severely drops. This leads to high instability while training on-policy algorithms. PPO, however, offers a balance in easy implementation, tuning, and sample efficiency. It tries to limit the change in the agent's behavior to prevent a collapse in performance while trying to make the biggest possible improvement step on a policy using the data that is currently available. [80].

For this project, PPO is proposed as one of the algorithms for the vibration control system. As mentioned previously, PPO ensures that the change from the old policy to the updated policy is not too large. To do this, the PPO algorithm employs an adaptive Kullaback-Lieber divergence (KL) penalty or a special objective function, the clipped surrogate objective [89], in place of the objective function used in other on-policy algorithms. The respective objective functions for KL penalty and clipped surrogate are shown in the equations below:

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t[\min\,(r_t(\theta)\hat{A}_t - \beta KL[\pi_{\theta old}(\cdot\,|s_t), \pi_\theta(\cdot\,|s_t)] \tag{4.10}$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min\left(r_t(\theta)\hat{A}_t, clip\,(r_t(\theta), 1\,-\,\epsilon, 1\,+\,\epsilon)\hat{A}_t\right] \tag{4.11}$$

where L^{KLPEN} is the objective function using the KL penalty, L^{CLIP} is the objective function using the clip factor and:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)} \tag{4.12}$$

is the probability ratio of the updated and old policy outputs, $\hat{\mathbb{E}}_t$ is the mathematical expression for the expectation, r_t is the reward at time t, $\pi_{\theta old}$ represents the old policy, s_t is the state at time t, ϵ (epsilon) is a hyperparameter, and $\hat{A}_t$ is the estimator of the advantage function.

For this work, the clipped surrogate objective is implemented, therefore the rest of this subsection will solely focus on that form of the cost function. Note that the objective function expression denotes that the minimum between the clipped and unclipped objective is selected. Therefore, with this scheme, the change in probability ratio is ignored when it would make the objective improved, and is included when it makes the objective worse [89]. Also of importance is to note that where the probability ratio is clipped depends on whether the advantage is positive or negative, i.e. if the advantage is positive the ratio is clipped at $1 + \epsilon$ and if the advantage is negative then the ratio is clipped at $1 - \epsilon$. This way there is a ceiling that is placed to ensure that the new policy is not too different from the old policy.

PPO trains by sampling actions based on its current policy and relies on randomness generated by initial conditions and training procedures. Therefore, as training continues, PPO agents explore less and may get stuck in local optimum solutions to the problem. Fortunately, this problem can be solved by adding an entropy bonus to guarantee enough exploration [89].

4.3 Reinforcement Learning as a Vibration Control Approach

As presented in the preceding subsections, reinforcement learning is a branch of machine learning where an agent learns through direct interaction with an environment that represents the problem dynamics. It is important to establish and/or restate what the RL elements represent in the vibration control problem presented in this work and further explain why reinforcement learning was chosen as the control method. The environment in RL is the equivalent of the plant in the closed loop controller, where the dynamics equations summarize the physical problem that is being modeled for control. In this case, the plant is the beam-actuator system and the vibration dynamic equations that represent its motion. The RL agent is the controller that takes in the system outputs and produces an output to encourage the desired response from the system. The output produced by the agent is the equivalent of a control input into the beam-actuator system. The observation is a vector that is sent

to the agent as information about what is going on in the environment. As opposed to being the direct state vector/state output from the environment, the observation can include additional information like an error from a desired value and its integrated value at a particular timestep. This explains the distinction between the two. Note that Table 4.1 summarizes this information and presents it in a compact format for ease of reference for the reader. Also, the observation and state have two separate descriptions because of the change in the controller/reward function description as detailed and described in Section 5.1.1 and Section 5.3.

Table 4.1: The Reinforcement Learning Elements and what they represent in a conventional closed loop controller scheme

Reinforcement Learning Element	*Represents*
Environment	Beam-actuator vibration dynamics
Agent	Controller
Action	Control input
Observation	Integrated error, error, displacement
	Displacement and velocity
State	Displacement only
	Displacement and velocity
Reward function	Cost function

In this work, RL is selected to implement vibration control for the pinned-pinned beam. The author presents the case where uncertainties can affect the performance of the controller in question, as described in Section 3.4, and suggests the need to develop a robust controller that can handle such cases. In recent literature, many researchers have explored the possibility of using adaptive conventional classical or optimal controllers. Sutton et al. [90] state that in a case where the model parameters are unknown or have uncertainty, then an adaptive controller can be implemented. Even though RL is more computationally expensive [91] and less sample-efficient

than the conventional control methods, it provides a more robust controller [90] to systems with unmodeled dynamics and changes in the environment, unlike the conventional controllers that would not perform well with any significant changes in the modeled system and place stricter restrictions on the uncertainties. Furthermore, there are multiple examples in literature that illustrate the use of RL as a tool to convert the conventional controller to adaptive ones [91, 92, 93, 94, 95, 96]. Because one of the main aims is to illustrate whether the controller can suppress vibration with uncertain parameters in the model and unknown external inputs, and with the aim to provide novelty in existing literature then the RL controller is chosen as the control method.

CHAPTER 5

CONTROLLER DEVELOPMENT AND DESIGN

The main focus of this chapter is to detail the fundamental aspects of the controller development and design for the vibration problem of the beam-actuator system. The creation of the reinforcement learning environment is explained, and the reward function, stopping criteria, observation, and action space definitions are provided and their selection is justified. The training details of the RL agent and the architecture of the neural network structures are also given and explained. Finally, a detailed, logical explanation is provided of the reward-shaping process for the given system.

5.1 Reinforcement Learning Environment

The RL environment is a representation of the problem's dynamics. The vibrating beam-actuator system is taken as the core of the environment as it summarizes the environment dynamics. Here, the environment is created in MATLAB-Simulink, with the main architecture designed on Simulink as block diagrams and the required system variables and matrices imported from MATLAB. In order to create the RL environment, multiple subsystem blocks are created on Simulink, each with a specific purpose it fulfills. This is described in detail in the following sections.

5.1.1 Reinforcement Learning Environment Architecture

The main architecture of the general block diagram is shown in Figure 5.1. The block diagram represents the system environment comprising the beam-actuator plant system and some blocks providing information to the RL agent. The beam-actuator subsystem block consists of a state space model plant that represents the system dynamics. The state space matrices are derived from the transfer function that

represents the beam-actuator dynamics, which was presented in section 2.1.2.1 with Equation (2.24). The system representation is transformed into state space form and the resulting matrices are imported directly from the workspace to the Simulink environment. The RL controller block represents the RL agent that chooses and sends an action i.e., a control input, into the beam-actuator system. The RL controller obtains information about the effect of the control input on the beam-actuator environment through the reward signal, state observation vector, and stop signal. Each of these three sets of information is defined in a separate block. To indicate whether the episode training or simulation is done, the *"stop simulation"* subsystem block evaluates whether the criteria set to end training have been met. This means that if a given control input generates a system response from the beam-actuator environment that has been set as any of the stop criteria, the subsystem block will send a signal that will indicate the end of that simulation run. The simulation then starts afresh, i.e., another episode begins. For the stopping criteria given in this section, shown in Appendix Figure 4, if the beam-midpoint displacement exceeds the value 0.1, the simulation will be stopped.

The subsystems that generate observations comprise equations calculating the relevant environment observations. These are chosen by the author and can only be calculated from the available state variables, i.e., the plant output. In this case, the observation is defined as a three-state vector comprising the error computed by finding the difference between the beam-midpoint displacement and the desired reference value 0, the integral of the error, and the displacement value itself. These three are sent to the agent (controller) after each time step, once the action is evaluated and a new state is obtained. This is summarized in Appendix Figure 2.

For the agent to learn, it requires a reward signal from the environment, to indicate how good the actions it takes are for the successful completion of the task at hand. This process is akin to the tuning of conventional controllers. The RL controller adjusts its neural network weights and biases according to the reward signal obtained. For positive rewards, the controller reinforces the actions taken in that step,

while for negative rewards, the agent discourages taking the same action in the next simulation while in that state.

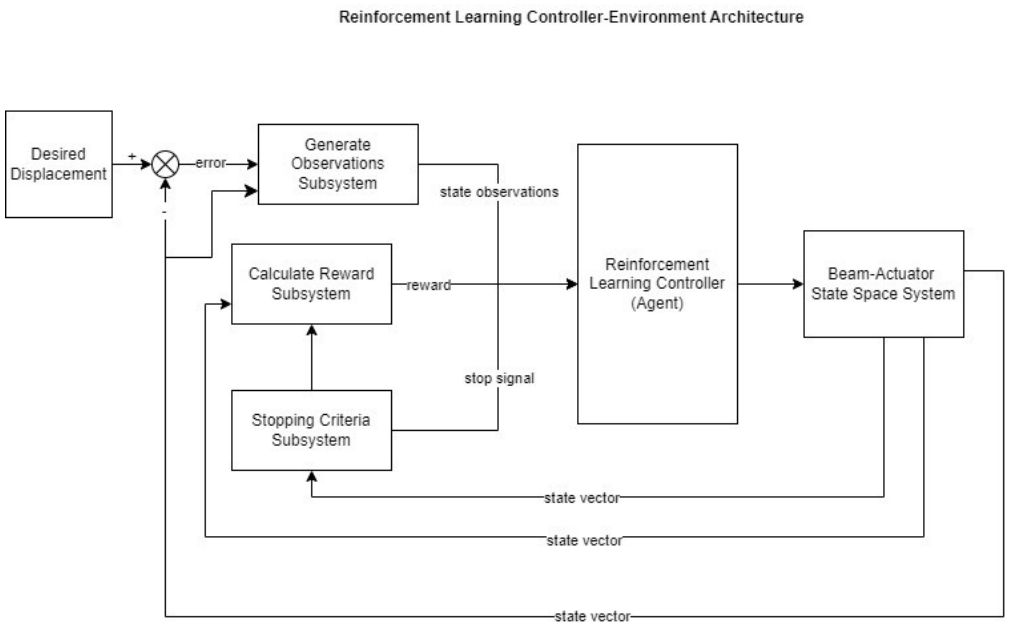


Figure 5.1: Block Diagram of RL Controller and Environment Architecture.

The *"calculate reward"* subsystem, shown in Appendix Figure 3, is set up to evaluate the reward signal according to the function specifically crafted for this problem. The reward function chosen to train the agents, after trial and error runs, is represented by the equations:

$$if\ |u| \leq 1 \times 10^{-9}, r = 10 \tag{5.1}$$

otherwise;

$$r = 10^3\left(1 - e^{|u|}\right) \tag{5.2}$$

where r represents the reward and u is the displacement of the beam-actuator system midpoint. This simple reward function indicates that the environment will send a reward of 10 if the displacement value is less than 1 nanometer. If that is not the case

then the reward is calculated by the equation shown. This encourages the agent to get the reward closer and closer to zero which is the desired state value. The reward function is manually designed through reward-shaping. This will be further discussed in Section 5.3.

However, the reward function provided in this section could prevent the agent/RL controller from learning an optimal policy. Since the desired value for displacement is zero, it proves counterintuitive that the reward function includes a reward for a displacement value less than a nanometer. In RL, the agent learns directly from the reward function, therefore, having obtained a displacement of less than a nanometer and obtaining a reward of 10, the agent will reinforce any action that attains positive reward even though the behavior to attain the reward is undesired.

5.1.2 Reinforcement Learning Agent

The neural network structure and the properties of the reinforcement learning SAC and PPO agents are briefly presented and discussed in this section. The deep learning neural networks used for training were specific to each case. For the SAC agent, as established in section 4.2.1, two different critic networks are used to estimate the value function and the soft Q function as well. The PPO agent is comprised of only a critic- and actor-network. The PPO critic network is defined differently from the SAC critic network, with the PPO critic having just input, hidden, and output layers and the SAC critic network having an observation path, a common path, and an action path. Note that the number of nodes in the input and output layers are determined by the dimensions of the observation and state spaces, and are therefore 3 and 1, respectively. Because of the nature of the problem, a small neural network is used for the agent, with hidden layers of only 12 nodes for the SAC agent and 15 nodes for the PPO agent. This choice was made considering similar cases in the literature [46]. Both the PPO and SAC actor networks are comprised of a common path, a standard deviation path, and a mean path. The common path has an input layer of size 1, again as determined by the action space, and is constructed as a fully

connected layer with 100 nodes and a rectified linear unit (ReLU) layer. Figure 5.2 illustrates the neural network structure for the PPO and SAC actors.

The SAC and PPO agent options are summarized in

Table 5.1 and Table 5.2, respectively. The discount factor was set as 0.99 for the SAC agent and 0.997 for the PPO agent, prioritizing long-term reward over immediate reward. For both agents, the sample time is chosen as 0.1. Sample time indicates how often the actions are chosen, in this case, how often the control input is sent into the beam-actuator system.

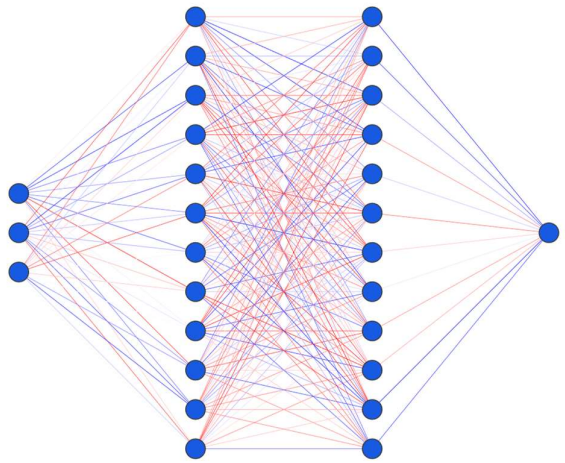


Figure 5.2: The actor's neural network structure for both PPO and SAC agent

For the SAC agent, the experience buffer length, where samples of the experience are stored for the off-policy update is chosen as 10^6, while the mini batches that are used to store groups of samples throughout training to update the agent's policy are of size 32.

Table 5.1: Soft Actor Critic Agent Options

SAC Agent Options	*Value*
Sample Time (s)	0.1
Discount Factor (γ)	0.99
Target Smooth Factor	1e-3
Experience Buffer Length	1e6
Mini Batch Size	32

As discussed in section 4.2.2, the PPO agent utilizes a clip factor in its surrogate objective. For this case, this factor is chosen as 0.2 while the entropy loss weight is selected as 0.01 to ensure the agent sets out to maximize entropy within the policy. The experience horizon value indicates the number of steps an agent interacts with the environment before learning from its experience. This is one of the parameters to consider for on-policy agents like PPO. For this case, this value is chosen as 600. The SAC and PPO optimizer agent options are presented in Table 5.3 and Table 5.4, respectively. The learning rate given in these tables is for both the actor and critic networks. The tables contain information on the type of optimizers used in the simulation. SAC uses Adam which is a method that only needs first-order gradients for efficient stochastic optimization [97].

Table 5.2: Proximal Policy Optimization Agent Options

PPO Agent Options	*Value*
Experience Horizon	600
Discount Factor (γ)	0.997
Clip Factor	0.2
Entropy Loss Weight	0.01
Sample Time	0.1

Table 5.3: Soft Actor Critic Agent Optimizer Options

SAC Agent Optimizer Options	*Value*
Optimizer	Adam
Learning Rate	1e-3
Gradient Threshold	1
L2 Regularization Factor	2e-4

For the PPO algorithm, the generalized advantage estimator is used to efficiently calculate the advantage while giving the option to control the bias-variance tradeoff [98]. It has a smoothing factor of 0.95 assigned for this training case. An epoch refers to a cycle through a single data point [99], therefore the number of epochs indicate the number of times the actor and critic will learn through cycling the current experience state. The results obtained from these training settings are illustrated in Chapter 6.

Table 5.4: Proximal Policy Optimization Agent Optimizer Options

PPO Agent Optimizer Options	*M*
Advantage Optimizer	Generalized Advantage Estimate
GAE factor	0.95
Learning Rate	1e-4
Number of epochs	3

5.2 Reinforcement Learning Controller Training Specifications

For each of the cases, the agents were trained for 1000 episodes each, with a maximum of 100 timesteps in each episode.

Table 5.5: Soft actor critic and proximal policy optimization training options

Training Options	*SAC*	*PPO*
Maximum Episodes	1000	1000
Maximum Steps per Episode	100	100
Stopping Criteria	Average Reward	Average Reward
Stopping Value	8000	400

The criterion for training termination was set as an average reward of 8000 for the SAC agent and 400 for the PPO agent. Note that these values do not hold any major significance except to ensure that the agent trains for all episodes. For agent exploration, the initial ϵ was set as 1, with a decay factor of 0.005 and a minimum ϵ of 0.001.

5.3 Reinforcement Learning Controller Reward-Shaping

With the reward function described in Section 5.1.1, the results obtained and illustrated in Section 6.3 satisfy the criteria intended, which is to ensure the displacement is of order $\leq 10^{-7}m$. However, the results also indicate a residual vibration that remains indefinitely. Hence a new goal is defined, to eliminate the residual oscillation of the beam-actuator midpoint and drive it to zero in time. For this goal to be attained, a reward-shaping procedure is undertaken.

Reward-shaping is the process of defining a reward function that is specific to the problem and guides a policy to a desired behavior by providing more frequent updates on the performance of the agent. In this case, the controller input effect on the beam-actuator displacement is evaluated at each time step and the consequent reward is sent back to the agent with the same frequency. This way, the RL controller will be able to adjust the actor and critic neural networks without waiting for a whole episode to end, leading to faster learning for the agent, i.e. faster tuning for the RL

controller. Note that the reward function presented in Section 5.1.1 is also a result of reward-shaping. The reward function defined in this section offers better performance based on the additional criteria introduced in this section. The block diagram representing the reward function is shown in Figure 5.3.

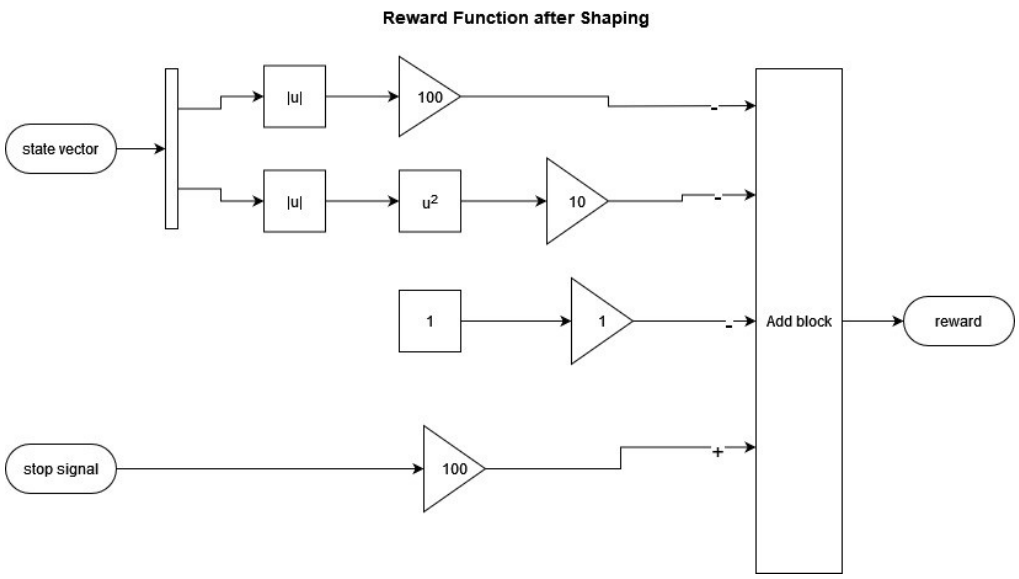


Figure 5.3: The revised reward function, to eliminate the residual vibration of the beam-actuator system.

The shaped reward function, in this section, contains a terminal reward, a punishment for the agent for the longer it takes to get to the terminal state, and small negative rewards on the way to the terminal state. To encourage the controller to reduce or eliminate the residual oscillation, the stop criterion is changed. The new stop signal is sent once a displacement of value 0 is achieved, which is the desired terminal state. This is well illustrated in Figure 5.4. Once the RL controller gives an input to the beam-actuator system that achieves the terminal state, the agent will receive a reward of 100, signifying the importance of reinforcing that action or sequence of actions when facing similar states in another simulation/episode. Note also that there is a constant penalty of -1 after each step that indicates to the controller that it should aim

to reach the terminal state as fast as possible. Otherwise, the penalty compounds the longer the simulation runs.

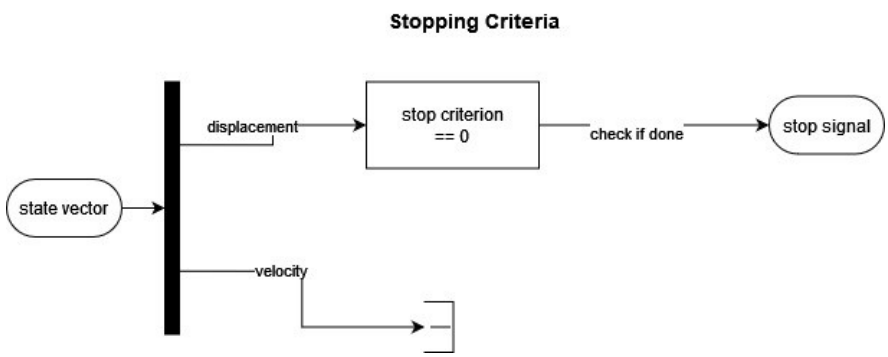


Figure 5.4: The stop criterion that contributes to the reward function

Furthermore, the shaped reward function also contains smaller negative rewards for non-terminal states. The further away from the terminal state, the greater the penalty as indicated in Figure 5.3. In this case, the desired states are displacement and velocity values are zero. Considering the order of the initial displacement and the uncontrolled response of the beam-actuator system to it, a trial-and-error search for suitable penalty and reward coefficients is carried out. The choice of the reward function greatly affects how well the agent can learn. This means that the tuning of the RL controller is directly influenced by the reward function. In reinforcement learning, the reward function provides the basis of learning. Therefore, reward-shaping limits how well the RL controller performs by limiting how well the RL agent learns.

The equations representing the improved reward function are given as:

$$if\ stop\ signal == yes, r = 100 \tag{5.3}$$

else:

$$-100|x| - 10|v|^2 - 1 = r \tag{5.4}$$

where $|x|$ and $|v|$ represents the absolute value of the displacement and velocity of the beam-actuators midpoint respectively.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter includes the results obtained from training and simulating the reinforcement learning controller on several environments that represent the beam-actuator system dynamics. In section 6.1 the results obtained from simulating a PID controller for vibration suppression of the system are presented alongside the uncontrolled and the RL-controlled response. In section 6.2 the training results for the RL agent are depicted and discussed. Section 6.3 contains the results obtained from simulating the trained agents on various selected beam-actuator environments.

6.1 Conventional Controller Results

This section contains the results obtained from applying a PID controller to suppress the vibration within the beam-actuator system. These results are provided as a reference, to see how the RL controller compares, in terms of performance, to a conventional controller. For better reference, the uncontrolled and RL-controlled responses to the same input are provided alongside the PID result. Note, however, that it is not the goal of this study to focus on this comparison, rather, this section aims to just give the reader a frame of reference.

The PID controller gains are chosen by using the pidTuner in MATLAB as a starting point, to avoid the otherwise tedious trial-and-error process. The transient performance criteria and steady-state error are selected and input into the app to generate the response that is desired. For this case, the performance criterion is chosen as a settling time of 20 seconds. The final selection of the proportional, integral, and derivative gains is made as 13.64, 10000, and 7.98, respectively. For these performance criteria and respective gains, the uncontrolled and PID-controlled response to an 8 mm beam displacement, at the midpoint, is obtained and shown in

Figure 6.1. The plot represents the beam-actuator system midpoint displacement over time. Since the initial excitation is given as 8 mm, the beam settles in residual vibration by 40s and maintains a steady state error that is determined by the derivative gain that was provided for the PID controller. Note that with the gains chosen, the PID controller overlaps the uncontrolled response. By using the tuner, the suggested integral gain is of order 10^7 and therefore this prompts such a controlled response from the gains chosen by the author. This can be explained by the order of the system output. Since the beam dimensions and the displacement are in the order of mm, the controller gain must compensate for the integral error that compounds the longer the simulation is run, making the PID controller ineffective in this case.

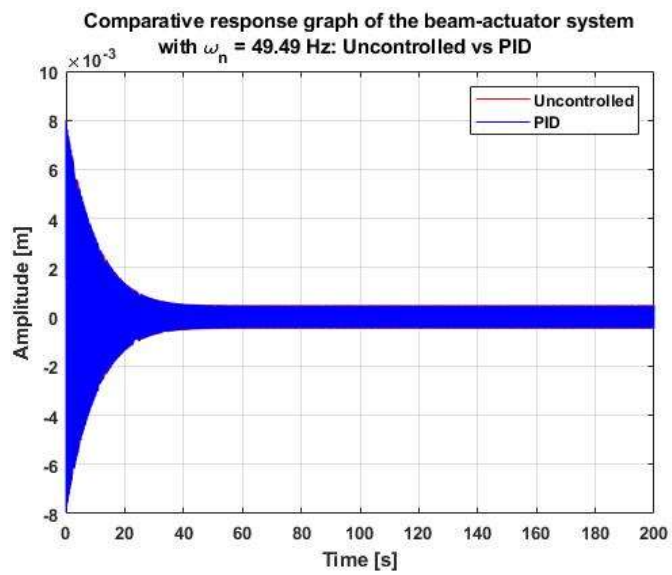


Figure 6.1: Beam-actuator uncontrolled response and the PID controlled response to an initial displacement of 8mm, for beam-actuator system with $\omega_n = 49.49Hz$

The uncontrolled response of the beam-actuator system is also compared to the reinforcement learning controlled response for the PPO and SAC algorithms, respectively. The resulting graphs are shown in Figure 6.3 and Figure 6.4. It is

observed that both RL controllers suppress the system vibration and eliminate the residual oscillation of the beam-actuator midpoint that is still present for the uncontrolled system even at the 200-second mark. Note that these RL-controlled results are obtained from the agents trained using the improved reward function.

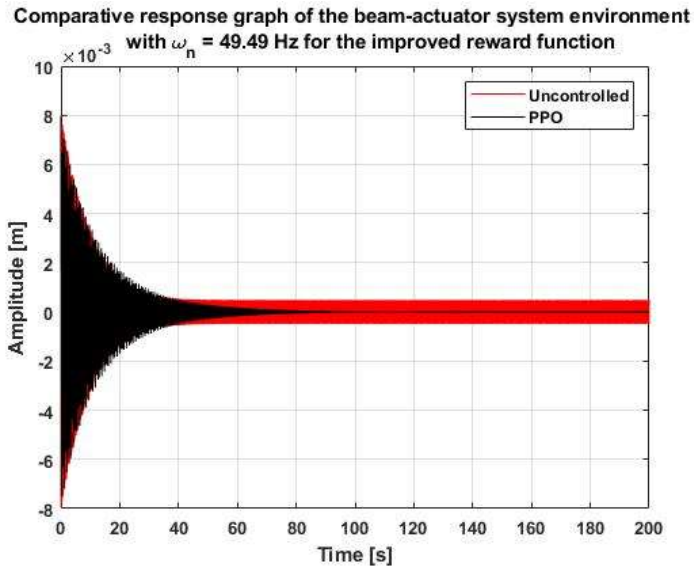


Figure 6.2: The uncontrolled response and the PPO controlled response to an initial displacement of 8mm, for the beam-actuator system with $\omega_n = 49.49Hz$

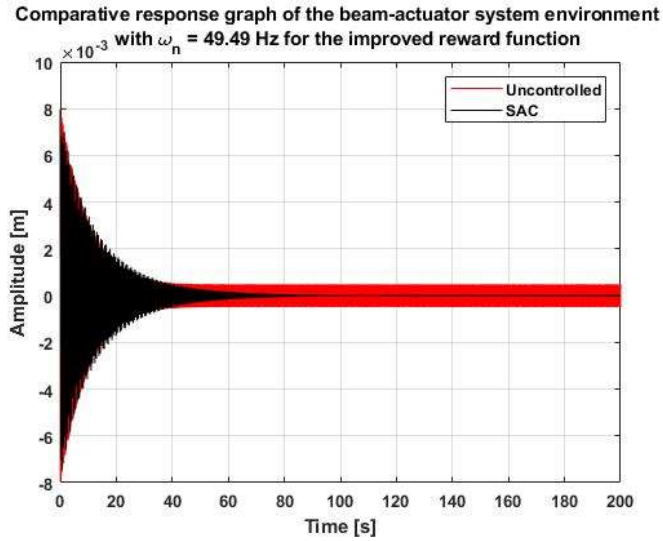


Figure 6.3: The uncontrolled response and the SAC controlled response to an initial displacement of 8mm, for the beam-actuator system with $\omega_n = 49.49Hz$

6.2 Reinforcement Learning Controller Training Results

The training results of the reinforcement learning control agent on the beam-actuator environments are given in this section. Since two algorithms were selected as the RL agents for this research, the results in the subsequent sections will present results for each separately. A conclusive discussion comparing the performance of the two will be provided in section 7.1. Note that the graphs illustrating the training results have the same plot settings. Therefore for all the training plots, the blue line represents the episodic reward, the black line represents the average episode reward while the red line represents the episode Q0. Given the initial observation of the environment, Q0 is defined as the estimated discounted long-term reward at the start of each episode. Q0 is calculated for all agents that have a critic and indicates a well designed critic, by converging to the estimated long-term reward, as training progresses. From the reward function defined in Equation (5.1) and (5.2), the maximum reward that the agent can get from an episode of 100 steps is 1000, considering the case that the

displacement is always within the desirable limit. Note however, that for the improved reward function, summarized in Equation (5.3) and (5.4), the maximum reward an agent can get within an episode is 100, which is the terminal reward of that function. The episode reward, Q0 value and average reward, are all plotted against the number of episodes the agent has completed. These three values are recorded for each episode and illustrated in Figure 6.4 to Figure 6.18.

6.2.1 Soft Actor-Critic Agent Training Results

This section contains the training results of the soft actor critic agent, for the original and improved reward functions. Note that the training results for the original reward function, presented in Section 5.1.1, are detailed first. Figure 6.4 shows the results of the SAC agent training in the beam-actuator environment with varying initial displacement for 1000 steps, with the original reward function. The initial displacement of the beam-actuator midpoint, x_0, is randomized at the beginning of each episode in the reset function as;

$$x_0 = 0.01 - 0.005 * rand\ [m] \tag{5.4}$$

Note that the agent is first trained on the beam-actuator environment that is derived from the first natural frequency of the system, where $\omega_n = 49.49\ Hz$, as established in Section 2.2.1.

As observed in Figure 6.4, the episodic reward is bound between [-140, -60] for the whole training simulation and exhibits very noisy behavior. Considering the form of the reward function presented in Equation (5.1) and (5.2), the average reward indicates that the controller has not learnt the optimal behavior policy. The episodic reward also indicates a cap on the ability of our agent to learn given the original reward function. Note that the Q0 estimate approaches the average reward as training goes on, indicating that the critic is being well designed as the agent trains. High variance is observed in the episodic rewards.

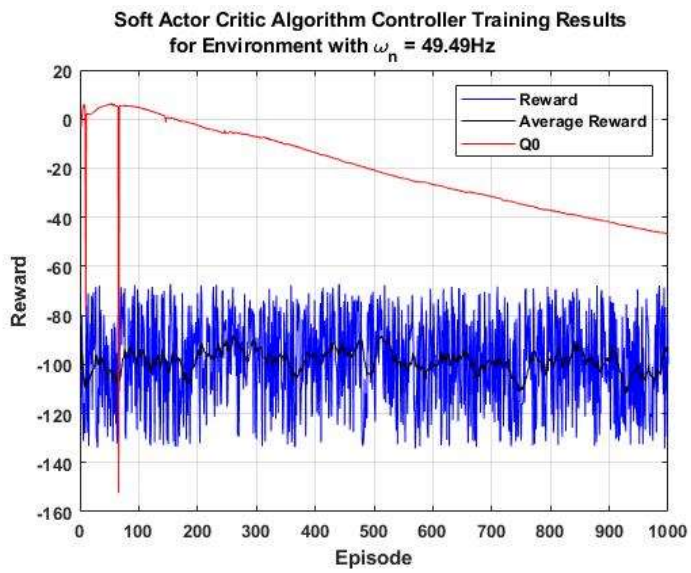


Figure 6.4: Training results for SAC agent on beam-actuator environment with $\omega_n = 49.49Hz$ for 1000 episodes.

The agent trained on the beam-actuator environment with $\omega_n = 49.49Hz$ is also trained other environments derived from different natural frequencies, to introduce the agent to uncertain dynamics. The agent is trained on 5 different beam-actuator environments for 200 episodes each, with the system matrices changing after every training session.

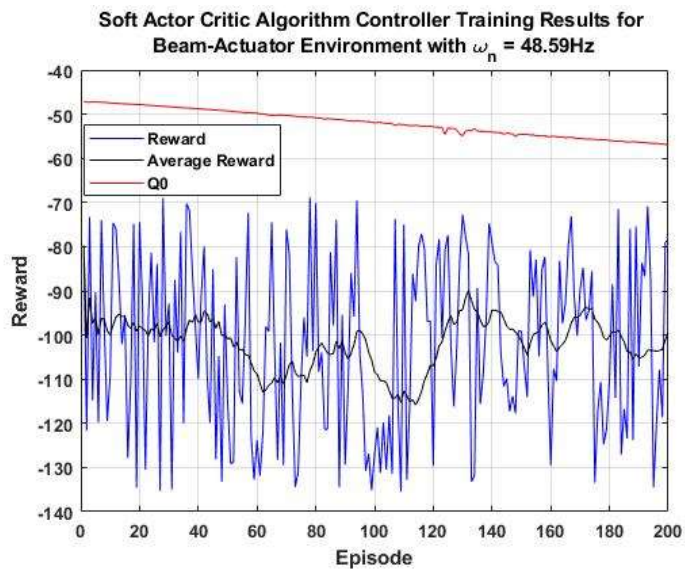


Figure 6.5: Training session results for SAC agent on the beam-actuator environment with $\omega_n = 48.59Hz$, for varying initial displacement

For the consecutive training sessions, the pre-trained agent from the previous session is loaded as the SAC agent and trained on an environment with different dynamics. For all the training sessions, the initial excitations to the system are randomized during training, as it was in the preceding case, to increase the robustness of the controller developed. The results obtained for the all five training sessions of the agent on the different beam-actuator environments, are illustrated through Figure 6.5 to Figure 6.9. It is observed that the agent's reward values obtained are bounded within the same limits as the training on the modeled beam-actuator environment with $\omega_n = 49.49Hz$.

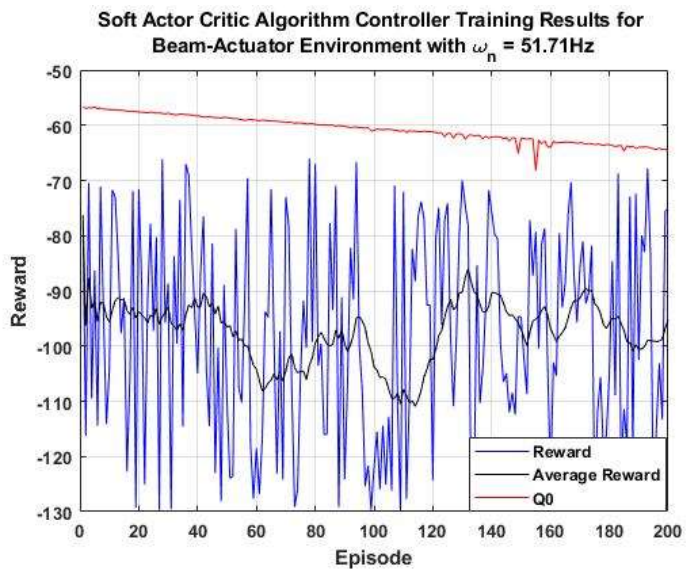


Figure 6.6: Training session results for SAC agent on the beam-actuator environment with $\omega_n = 51.71Hz$, for varying initial displacement

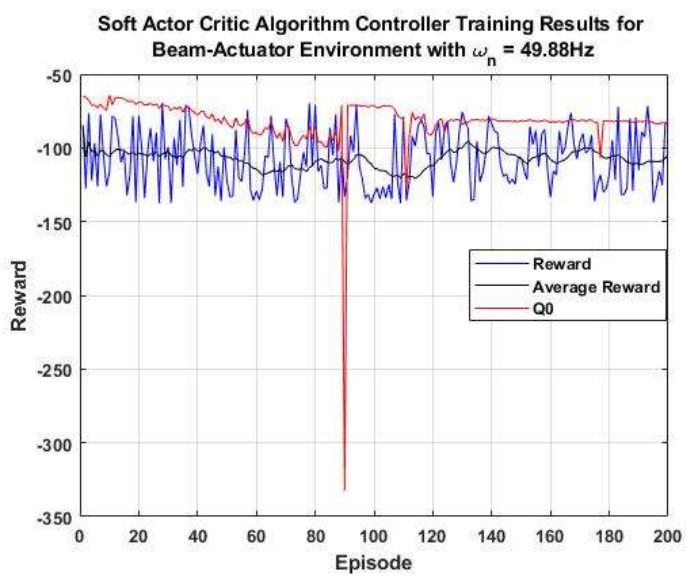


Figure 6.7: Training session results for SAC agent on the beam-actuator environment with $\omega_n = 49.88Hz$, for varying initial displacement

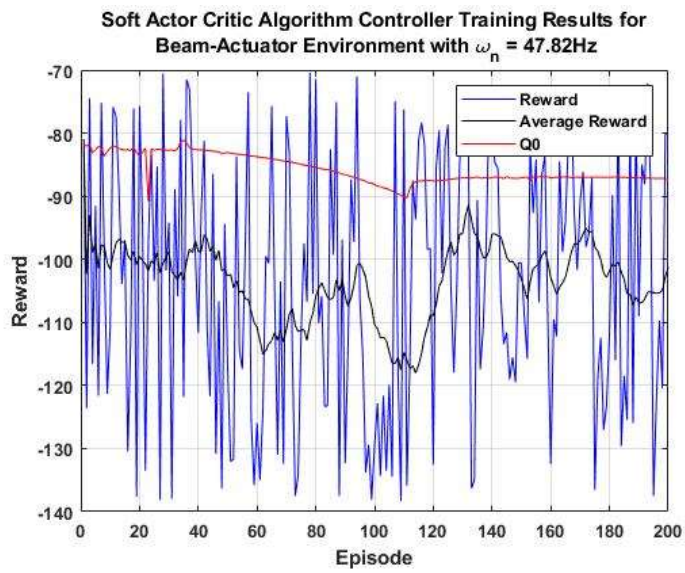


Figure 6.8: Training session results for SAC agent on the beam-actuator environment with $\omega_n = 47.82Hz$, for varying initial displacement

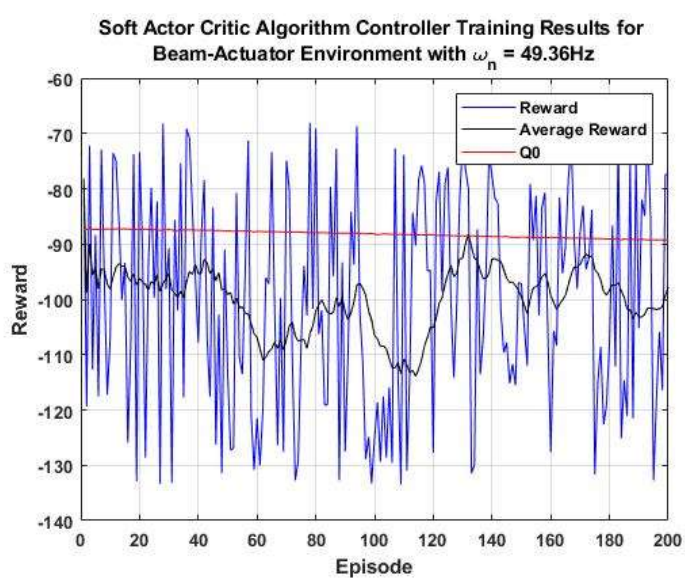


Figure 6.9: Training session results for SAC agent on the beam-actuator environment with $\omega_n = 49.36Hz$, for varying initial displacement

From Figure 6.5 to Figure 6.9, as the agent trains for a longer period, the Q0 value converges to the average reward. This is an indication that the critic network weights are changing as the agent trains, so that they can make better value estimates and actions to maximize reward. Despite the change in the beam-actuator environment dynamics, the Q0 value gradually converges to the average reward. This indicates that the Q0 approximation gets more accurate with the consecutive training sessions

The average reward resulting from training the SAC agent over time is summarized in Table 6.1. Even with the change in environment dynamics, the value of the average reward remains bounded between [-97.34, -107.23].

Table 6.1: SAC training session episodes and average reward for the changing environments

SAC Training Session (ω_n)	*Average Reward*
48.59 Hz	-101.79
51.71 Hz	-97.343
49.88 Hz	-107.23
47.28 Hz	-103.47
49.36 Hz	-99.89

The training results for the improved reward function are illustrated in Figure 6.10 and Figure 6.11. The figures indicate an improvement in the training of the agent because of the higher average reward noted. The training results also illustrate that the RL agent is stuck in a local optimum as they are unable to obtain the maximum reward defined, which is 100. The improved performance shows that the changes made to the reward function improved the learning ability of the RL controller and consequently the behavior policy. The reward function still has more room for improvement, but satisfies the requirements in this case and is therefore suitable for the given problem. For this case, instead of plotting separate graphs for the training of the agent for various environments, the results are summarized in Figure 6.11.

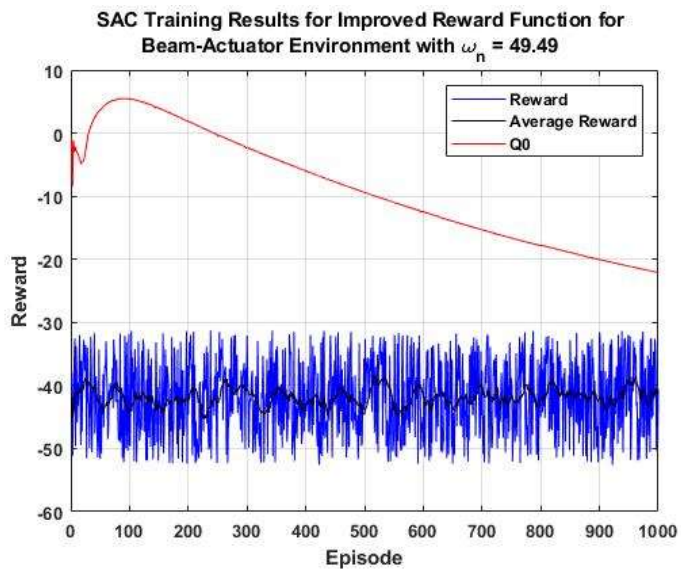


Figure 6.10: Training results for SAC agent on beam-actuator environment with $\omega_n = 49.49Hz$ for 1000 episodes, for improved reward

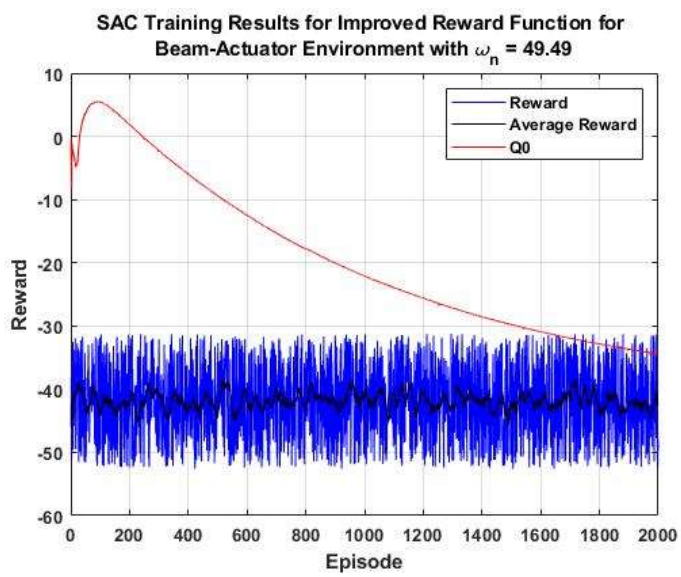


Figure 6.11: Training results for SAC agent on beam-actuator environment with $\omega_n = 49.49Hz$ for 2000 episodes, for improved reward

6.2.2 Proximal Policy Optimization Agent Training Results

This section contains the results from training the proximal policy optimization agent for the original and improved reward functions. Note that the training results for the original reward function, presented in Section 5.1.1, are detailed first. The PPO agent records a similar average reward bound as the SAC agent as can be seen in Figure 6.12 and Figure 6.13. For these training results, the initial displacements are randomized and the base model, i.e. the beam-actuator environment derived from $\omega_n = 49.49Hz$ is used for training. The average reward and the bounded episodic reward close to the SAC results shown in Figure 6.4 is unexpected as the PPO agent learns online while the SAC agent employs offline learning by sampling from its experiences and is therefore more likely to illustrate better performance for shorter training lengths. With this result, it is safe to presume that there exists the possibility that the agent is not learning well because of the designed reward function.

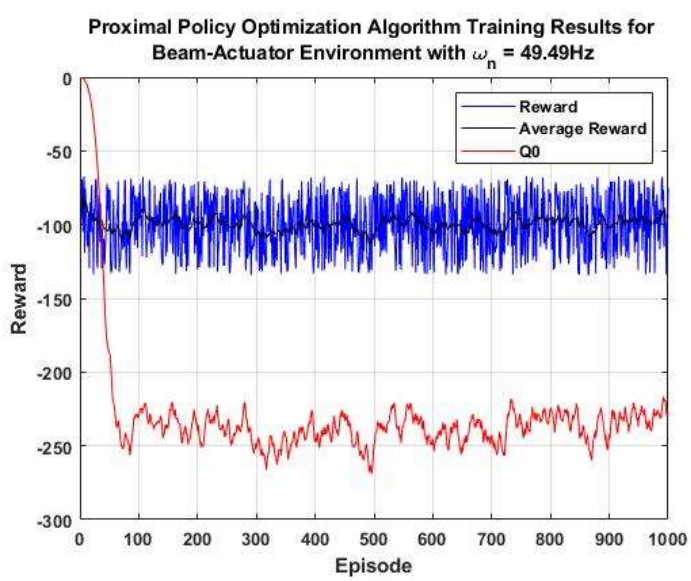


Figure 6.12: Training results for Proximal Policy Optimization Controller (agent) on environment with $\omega_n = 49.49\ Hz$ for 1000 episodes

It is also expected, as discussed in Section 4.2.2, that the PPO agent limits the change in the policy for more stable training and therefore, a tightly bounded average reward follows this principle. However, this is not observed for the PPO agent training, rather a similar bound as that seen for the SAC agent training with the original reward function given in Equation (5.1) and (5.2). High variance is observed in the individual episode rewards while the average reward remains bounded between [-150,-50]. The Q0 estimate does not move towards the average reward at the end of the 1000 or 5000 episodes, which indicates poor design of the critic network or the reward function. This exhibits the need for longer training sessions for on-policy algorithms that do not learn from sampled experiences and a change in either the reward function or the critic network architecture. As opposed to the SAC algorithm training in Figure 6.4 where the Q0 estimate starts slowly converging towards the episode reward after the first 100 episodes, the PPO agent training in Figure 6.13 maintains a value of -225 for Q0 for most of the training session.

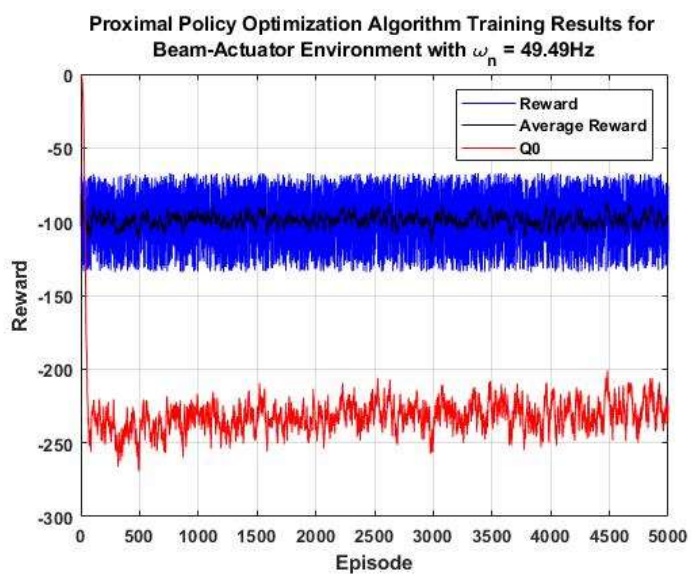


Figure 6.13: Training results for Proximal Policy Optimization Controller (agent) on environment with $\omega_n = 49.49\ Hz$ for 5000 episodes

The same agent, trained on the beam-actuator environment with $\omega_n = 49.49Hz$, is then trained on the other environments derived from different natural frequencies, to introduce the agent to uncertain dynamics. The agent is trained on 5 different beam-actuator environments for 200 episodes each, with the system matrices changing after every training session. The training results are illustrated in Figure 6.14 to Figure 6.18. Note that the average reward is maintained within a tight range. This is expected because the PPO agent limits the policy update. Therefore, even if pre-trained agents are loaded and trained for different dynamics, because the dynamics themselves are slightly altered, the policy will only have small updates for the PPO agent. Note that also here, the initial excitations to the system are randomized during training, as it was in the preceding cases, to increase the robustness of the controller developed. As observed in Figure 6.14 through to Figure 6.18, the Q0 value still does not converge to the average reward, unlike what is observed for the SAC agent in Section 6.2.1.

The average reward resulting from training the PPO agent over time is summarized in Table 6.2. Even with the change in environment dynamics, the value of the average reward remains bounded between [-95.51, -105.41].

Table 6.2: PPO training session episodes and average reward for the uncertain environment

PPO Training Session	*Average Reward*
48.59 Hz	-99.271
51.71 Hz	-95.513
49.88 Hz	-105.41
47.28 Hz	-101.53
49.36 Hz	-98.00

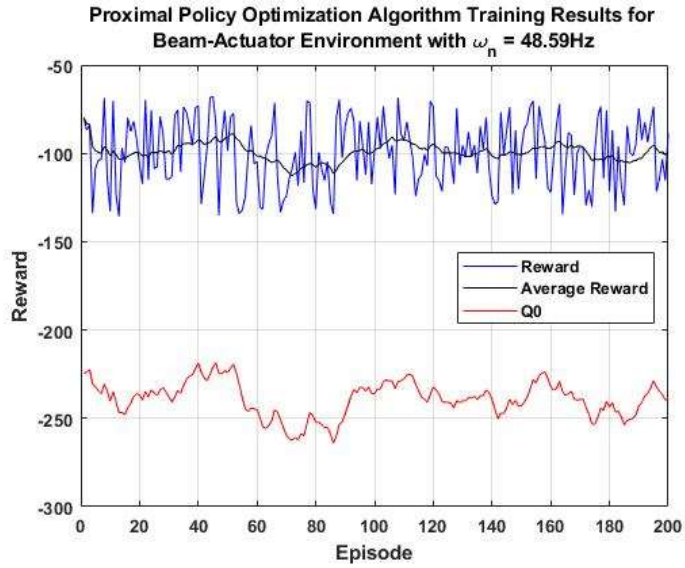


Figure 6.14: Training session results for PPO agent on the beam-actuator environment with $\omega_n = 48.59Hz$, for varying initial displacement

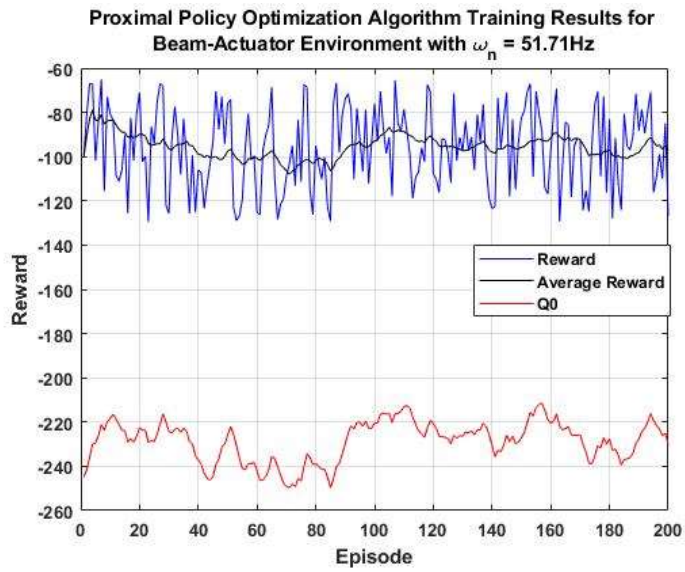


Figure 6.15: Training session results for PPO agent on the beam-actuator environment with $\omega_n = 51.71Hz$, for varying initial displacement

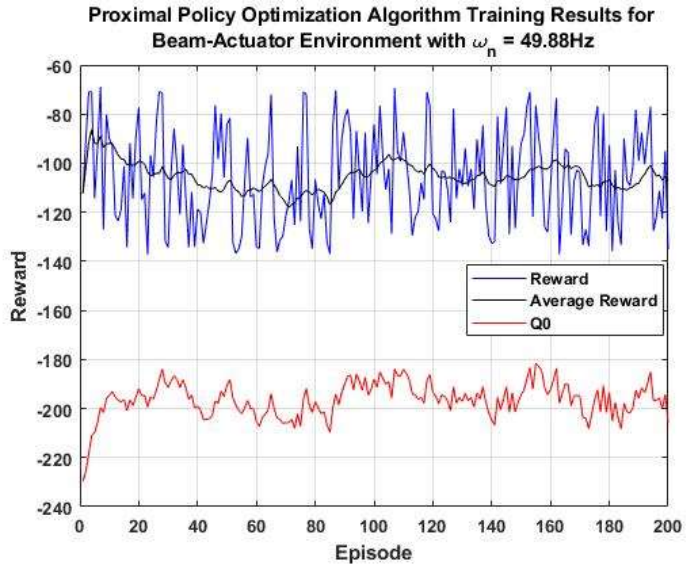


Figure 6.16: Training session results for PPO agent on the beam-actuator environment with $\omega_n = 49.88Hz$, for varying initial displacement

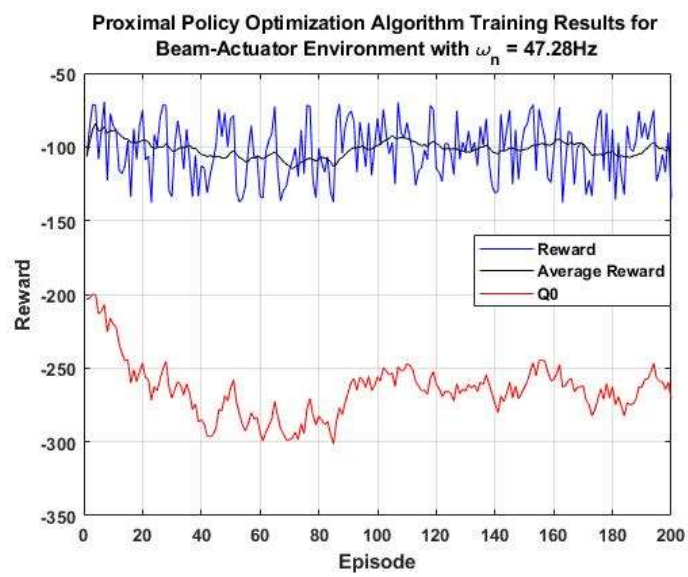


Figure 6.17: Training session results for PPO agent on the beam-actuator environment with $\omega_n = 47.28Hz$, for varying initial displacement

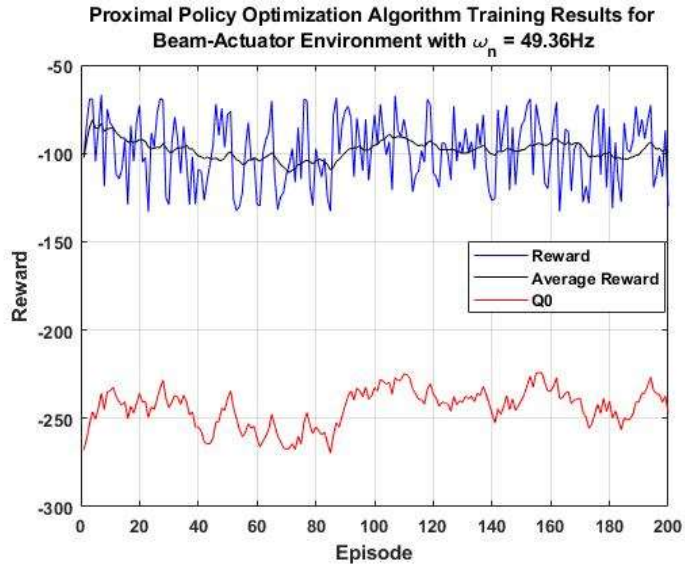


Figure 6.18: Training session results for PPO agent on the beam-actuator environment with $\omega_n = 49.36Hz$, for varying initial displacement

The training results for the PPO agents with improved reward function are given in Figure 6.19 and Figure 6.20. The figures indicate an improvement in the training of the agent because of the higher average reward noted. A higher average than that of the SAC training is observed. However, the training results still illustrate that the RL agents is stuck in a local optimum as they are unable to obtain the maximum reward defined, which is 100. Note that for the improved training the results are compiled into the Figure 6.20 instead of plotting 5 additional graphs with 200 episodes of training. With the improved reward function, the Q0 estimate tracks the episode reward and this indicates that the critic network is well designed for the given reward function.

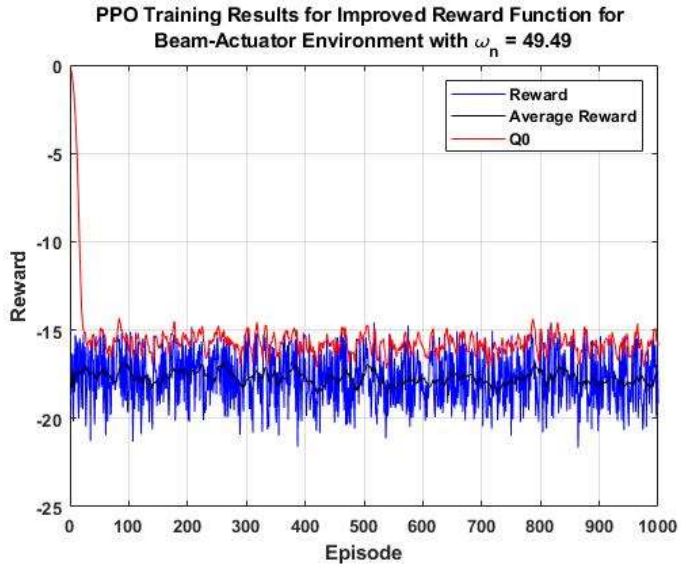


Figure 6.19: Training results for PPO agent on beam-actuator environment with $\omega_n = 49.49Hz$ for 1000 episodes, for improved reward

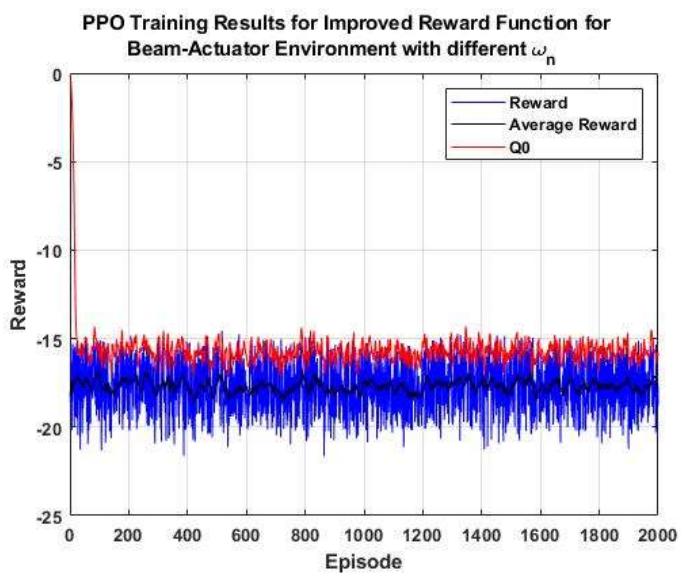


Figure 6.20: Training results for PPO agent on beam-actuator environment for different ω_n, for 2000 episodes, for improved reward

6.3 Reinforcement Learning Controller Simulation Results

In Section 6.3.1 and Section 6.3.2, the simulation results of the trained reinforcement learning agents are shown and discussed, for both reward functions. The displacement vs. time graphs are presented for the controlled beam-actuator system derived from $\omega_n = 49.49Hz$ and thereafter the simulation results for multiple environment dynamics cases are also provided. Note that the results, for the original reward function, are presented first for both sections. Thereafter, the simulation results of the SAC and PPO agents trained with the improved reward function are illustrated. Note that the simulation results are obtained from three different types of agents; one simulated on the environment it is trained on, one simulated on a different environment from what it is trained on, and finally, one simulated on a random environment after being trained on multiple environments. The discussion of the results obtained will clarify what agent was used, for each case. For clarity, the environments are the state space systems that represent the beam-actuator dynamics, derived from a specified value of natural frequency and the agent refers to the reinforcement learning controller obtained after training.

6.3.1 Soft Actor-Critic Agent Simulation Results

This section contains the simulation results for the trained soft actor-critic vibration controllers, for the response to the initial displacement of a beam-actuator system midpoint. Figure 6.21 to Figure 6.24 display the performance of the SAC reinforcement learning controller on the vibrating beam system with $\omega_n = 49.49Hz$. It can be seen that the vibration settles within steady state error in about 10 seconds. An initial displacement of 8 mm is quickly suppressed and the beam retains some minimal oscillations indefinitely. The amplitude of the residual oscillations is one order less than the initial displacement given to the beam. Considering the duration the RL agent was trained for and the simple form of the original reward function [Equation (5.1) and Equation (5.2)], the agent exhibits suboptimal performance and

is unable to achieve the positive reward criterion of achieving a displacement absolute value of less than or equal to 10^{-9}. Tuning the reward function more and training the agent on the environment for longer could result in the vibration suppression within acceptable bounds or complete elimination. However, as discussed in Section 5.3, the agent, i.e. the reinforcement controller, forms its policy centered around the reward function, therefore, the policy generated could be suboptimal after even longer training sessions. To prove if this was the case, the agent was trained for 5000 episodes and simulated for the same case and the results shown in Figure 6.23. From the graph, the same behavior as that in Figure 6.21 is observed. Therefore, it is established that the original reward function can only produce a suboptimal control scheme that results in a residual vibration of order 10^{-7} after 200 seconds.

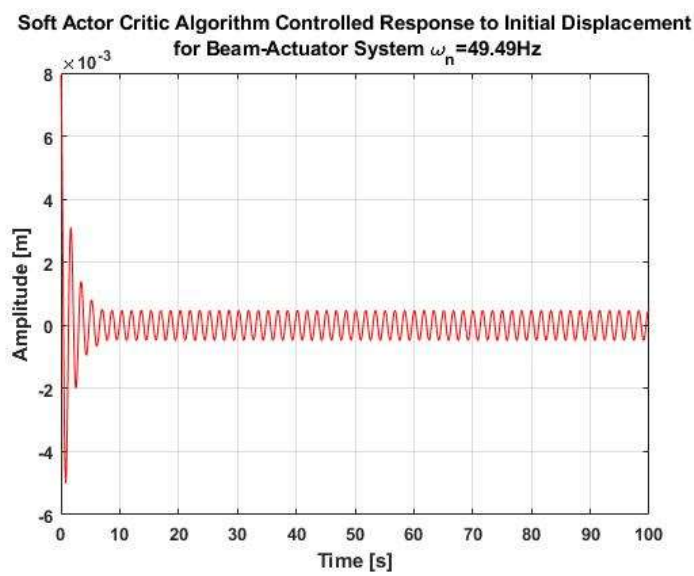


Figure 6.21: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

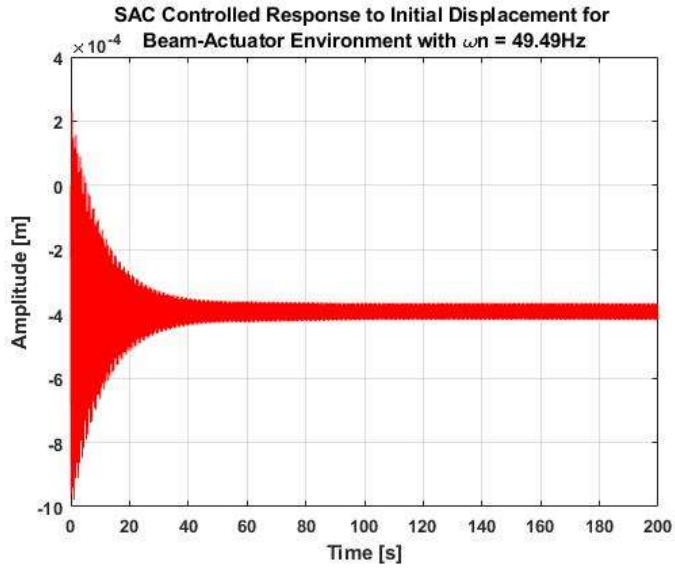


Figure 6.22: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$, after longer training

The controlled response of the agent trained on multiple environments (SAC-MET) is presented in Figure 6.24. The graph indicates no change in the order of displacement after 100 seconds, even with training on multiple environments. This supports the results obtained in Figure 6.23 as well and indicates that the original reward function limits the performance of the RL controller as well.

To further investigate the performance of the RL controller, it is trained afresh, with the original reward function, in new environments with different dynamics.

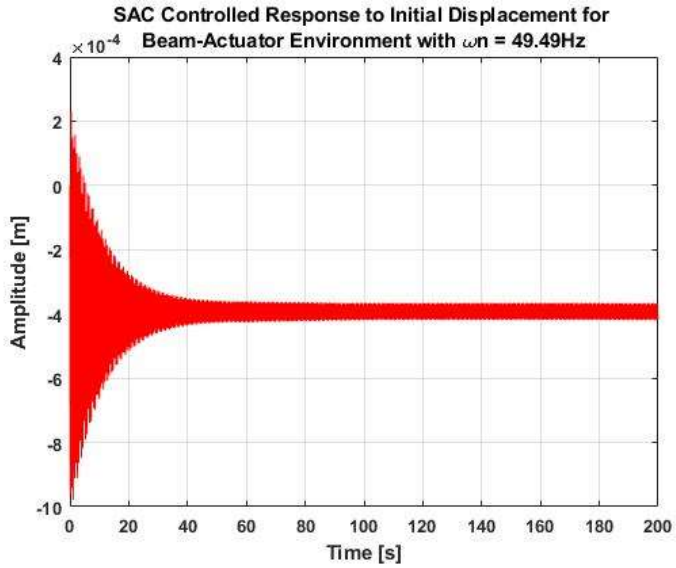


Figure 6.23: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$, after longer training

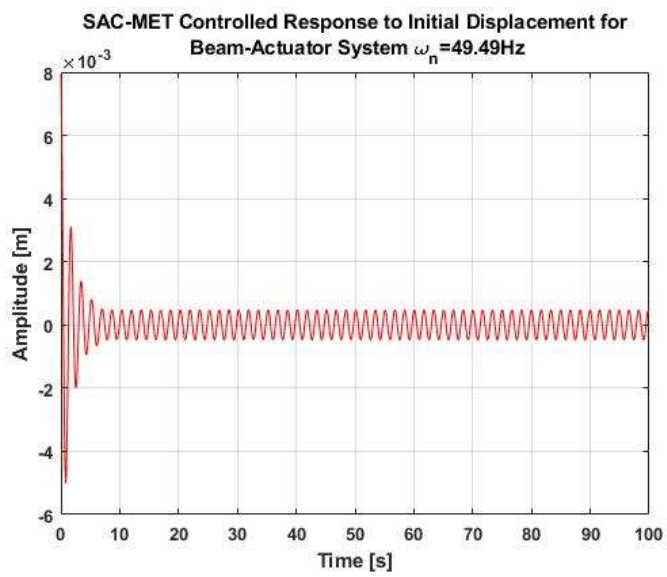


Figure 6.24: SAC-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

The results obtained from simulating the RL controllers for these different cases are illustrated in Figure 6.25 to Figure 6.28. Figure 6.25 shows the RL controlled response of the agent trained on the environment dynamics with $\omega_n = 48.59Hz$ while Figure 6.26 shows the RL controlled response simulated by the agent/controller trained on multiple environments. A similar pattern to the preceding results is observed. The controller scheme is suboptimal and the controlled amplitude retains a residual absolute value of order 10^{-7}. The responses shown exhibit the potential of the reinforcement controllers to dampen the vibration, even with a simple reward function. However, the performance is suboptimal and in need of improvement. This presents one of the challenges in tuning an RL controller.

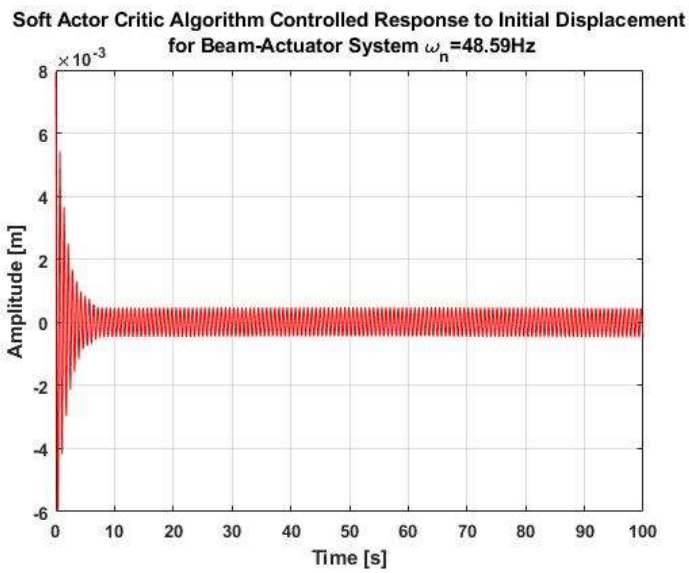


Figure 6.25: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 48.59Hz$

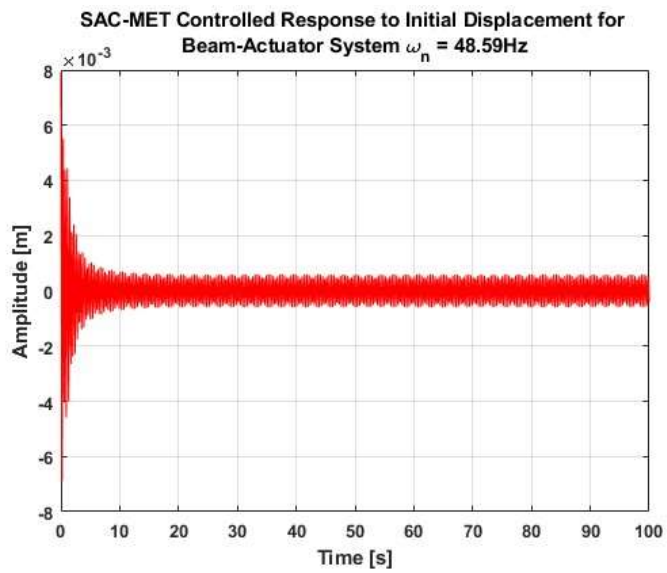


Figure 6.26: SAC-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 48.59Hz$

The results illustrated in Figure 6.27 and Figure 6.28 also indicate a similar pattern to the controlled responses seen in Figure 6.21 through to Figure 6.26. Note also that Figure 6.27 indicates the RL controlled response of the agent trained on the environment dynamics with $\omega_n = 51.71Hz$ while Figure 6.28 shows the RL controlled response simulated by the agent/controller trained on multiple environments. The robust nature of the RL controller is not demonstrated through these results. The RL controller is unable to illustrate a difference even after training for longer durations and on multiple different beam-actuator environments, as highlighted in 6.2.1. This can be attributed to the limitations set on the RL controller by the original reward function. To further illustrate this, the simulation results for the RL controllers developed using the improved reward function are also provided. As explained in Section 5.3, the goal of the improved reward function, described in Equation (5.3) and (5.4), is to eliminate the residual oscillation of the beam-actuator system midpoint about zero and attain the desired displacement value, 0.

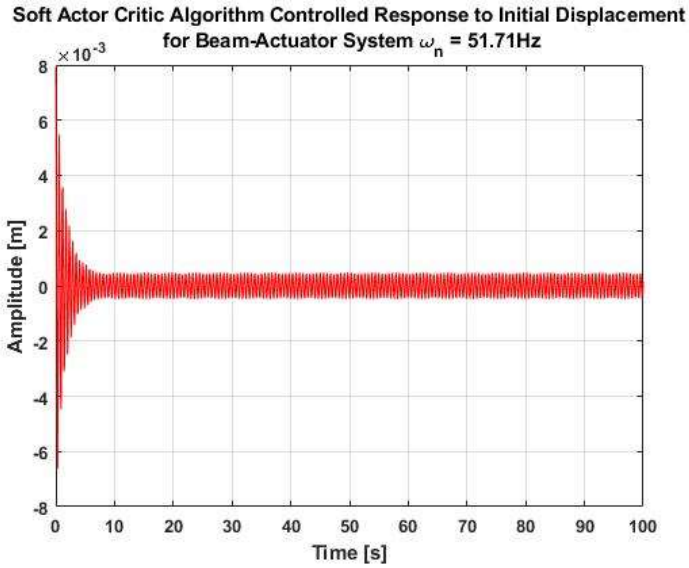


Figure 6.27: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 51.71Hz$

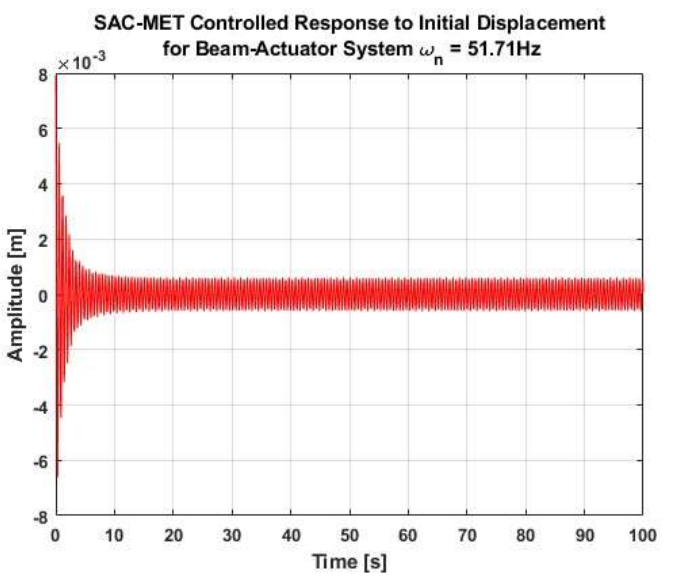


Figure 6.28: SAC-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 51.71Hz$

The RL agents trained using the improved reward function are simulated in various environments and the results obtained are shown in Figure 6.29 through to Figure 6.33. Figure 6.29, Figure 6.30, and Figure 6.31 show the simulated results of the RL controllers obtained after training on the respective environments with the beam-actuator dynamics derived from $\omega_n = 49.49Hz, \omega_n = 47.01Hz,$ and $\omega_n = 48.66Hz$, in that order. The order of amplitude displacement of the beam-actuator system midpoint after 200 seconds is summarized in Table 6.3. The values presented in Table 6.3 indicate that the controller performance is much better for the improved reward function compared to the original reward function. The controlled response tends to zero as time goes to infinity. The improvement of the reward function allows for a better control scheme to be learnt and implemented on the system.

Table 6.3: The order of displacement of the beam-actuator controlled response at 100 and 200 seconds

Dynamics SAC controller trained on	*Order of displacement at 100s [m]*	*Order of displacement at 200s [m]*
49.49 Hz	10^{-8}	10^{-11}
48.66 Hz	10^{-8}	10^{-10}
47.01 Hz	10^{-8}	10^{-10}

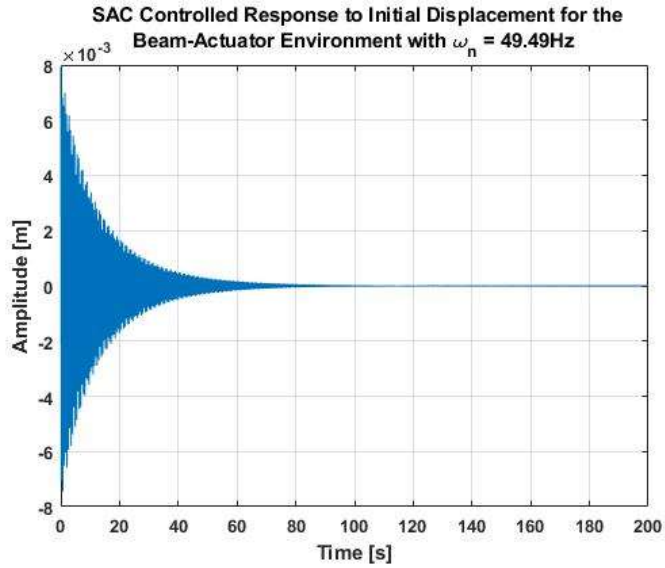


Figure 6.29: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

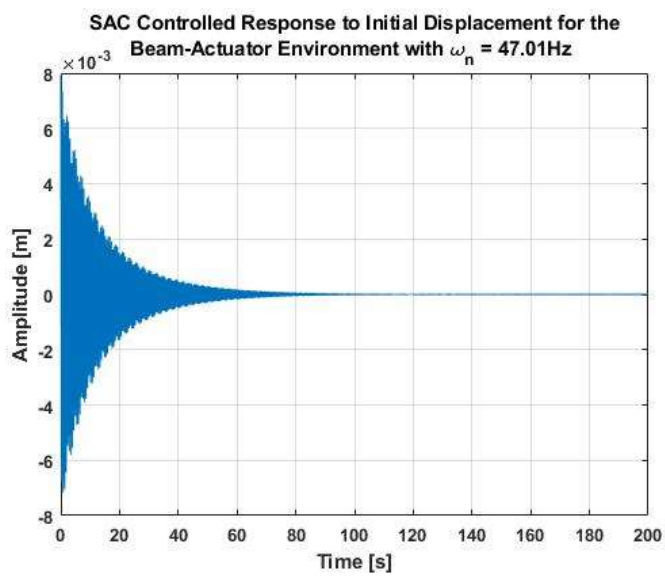


Figure 6.30: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 47.01H$

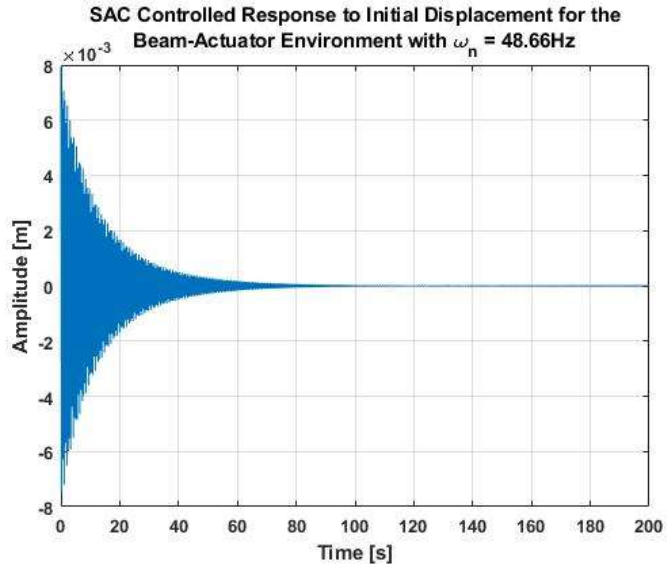


Figure 6.31: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 48.66Hz$

The agent/controller trained on the beam-actuator environment with $\omega_n = 49.49Hz$ is then simulated on the environment derived from $\omega_n = 47.01Hz$. The displacement results obtained are then shown in Figure 6.32. The results indicate that the controller is still able to suppress vibration but only up to an order of 10^{-7}, even after 200 seconds of simulation. An agent trained on multiple environments, SAC-MET, is also simulated on the same environment with $\omega_n = 47.01Hz$ and the results obtained are illustrated in Figure 6.33. The results demonstrate that with the improved reward function, the controller learns a policy that enables it to generalize a control scheme after training on different dynamics. The robust nature of the RL controller trained on multiple environments is also

illustrated in the results, as the SAC-MET, Figure 6.33, demonstrates performance comparative to the SAC controller trained on the environment itself, Figure 6.30.

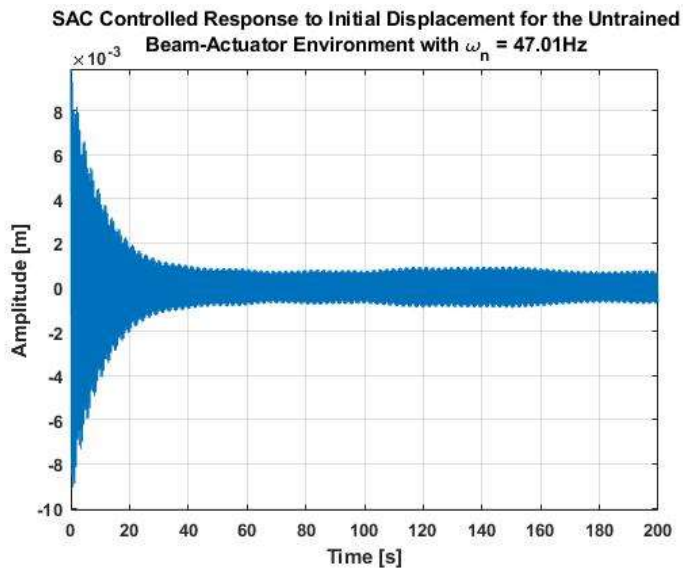


Figure 6.32: SAC controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 47.01Hz$

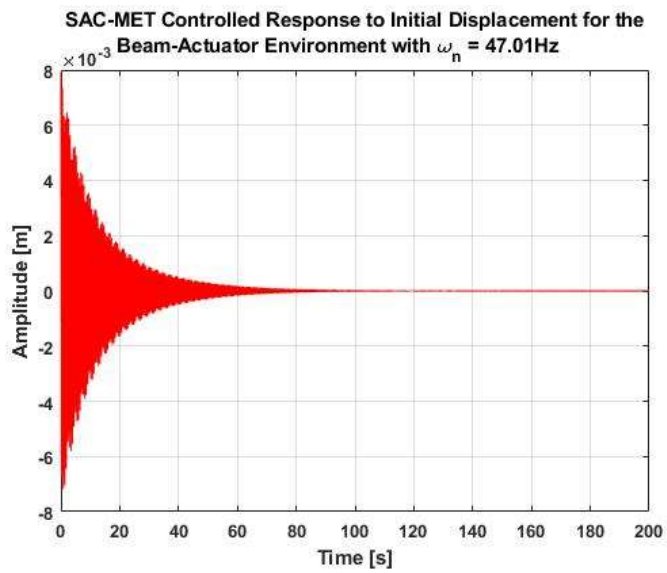


Figure 6.33: SAC-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 47.01Hz$

6.3.2 Proximal Policy Optimization Agent Simulation Results

This section contains the simulation results from the trained proximal policy optimization vibration controller, for the response to the initial displacement of a beam-actuator system midpoint. The performance of the PPO agent, trained with the original reward function, is similar to the performance of the SAC agent. Figure 6.34 and Figure 6.35 display the performance of the PPO reinforcement learning controller on the vibrating beam system with $\omega_n = 49.49Hz$. Here as well, as can be seen in Figure 6.34, the vibration of the beam, to an 8 mm initial displacement, is suppressed by the PPO controller and the steady state value is attained within the first 10 seconds.

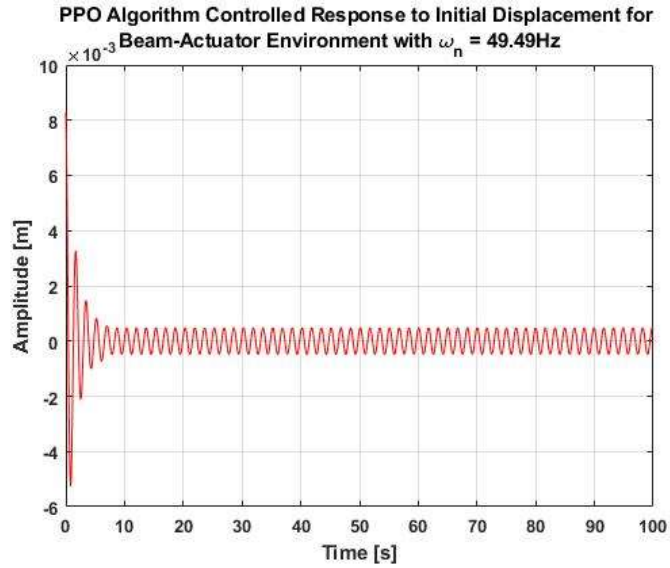


Figure 6.34: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

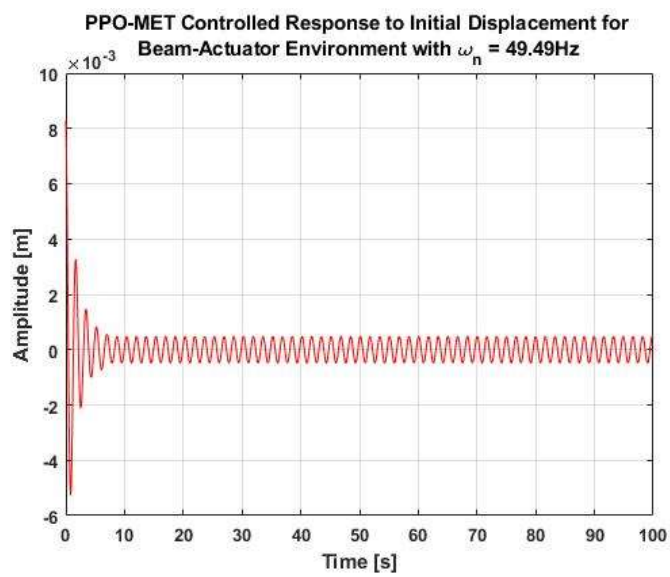


Figure 6.35: PPO-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

The controller, however, is unable to eliminate the residual oscillations of the beam-actuator midpoint about its equilibrium point. This again, is explained by the limitations the reward function places on the agent's ability to learn. After the agent is trained on multiple beam-actuator environments, the controller is still unable to generate an optimal scheme, as reflected in the results shown in Figure 6.35.

The performance of the RL PPO controllers is inspected once more by training each of them for different sets of matrices that represent different system dynamics. The results are plotted in Figure 6.36 to Figure 6.39. Figure 6.36 shows the RL controlled response of the agent trained on the environment dynamics with $\omega_n = 48.59Hz$ while Figure 6.37 shows the RL controlled response simulated by the agent/controller trained on multiple environments. A similar pattern to the preceding results is observed. The controller scheme is suboptimal and the controlled amplitude retains a residual absolute value of order 10^{-7}.

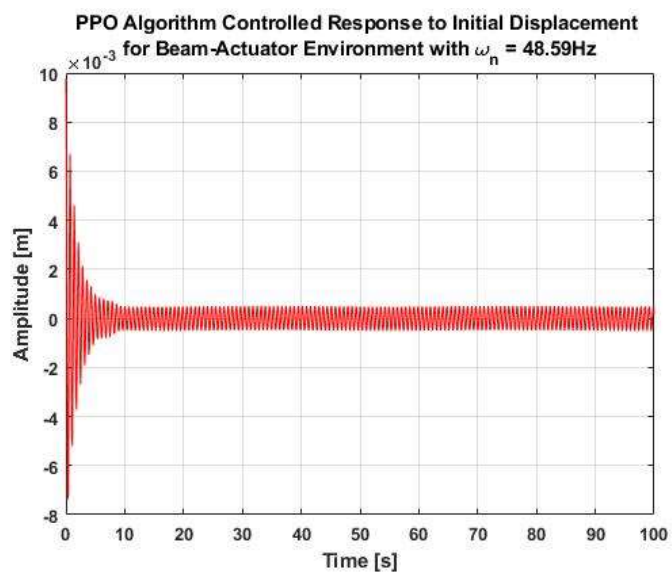


Figure 6.36: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 48.59Hz$

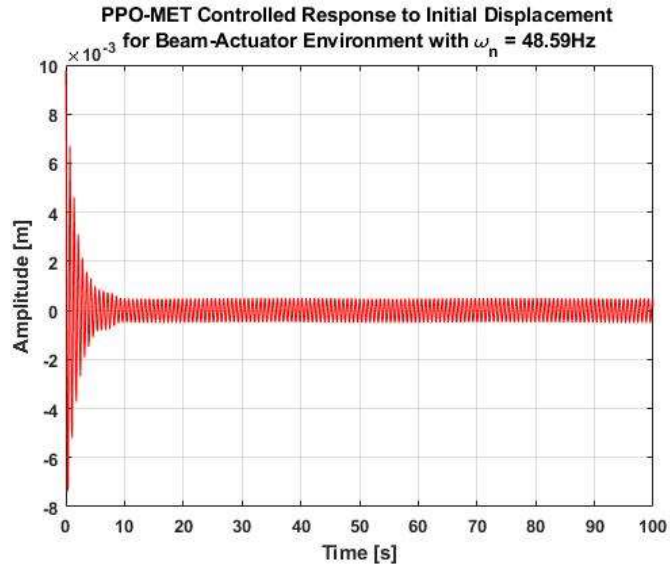


Figure 6.37: PPO-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 48.59Hz$

The results illustrated in Figure 6.38 and Figure 6.39 also indicate a similar pattern to the controlled responses seen in the simulation results for the other PPO controllers for the preceding cases. Note also that Figure 6.38 indicates the RL controlled response of the agent trained on the environment dynamics with $\omega_n = 49.88Hz$ while Figure 6.39 shows the RL controlled response simulated by the agent/controller trained on multiple environments. The robust nature of the RL controller is not demonstrated through these results, confirming that the original reward function was poorly shaped for this task.

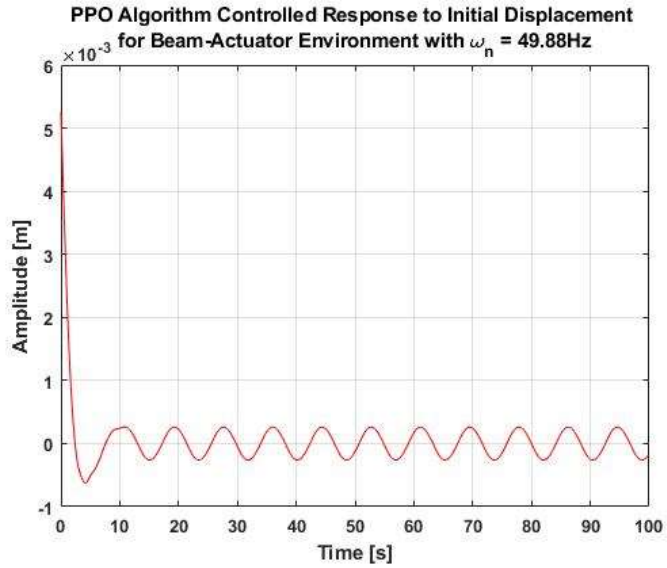


Figure 6.38: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.88Hz$

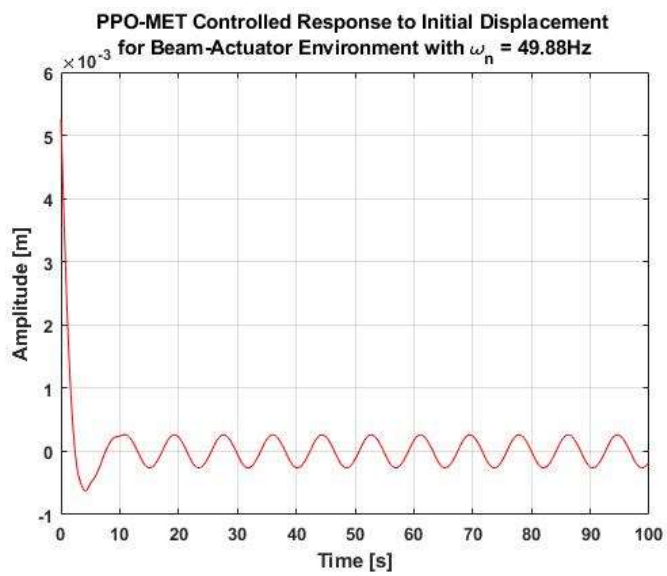


Figure 6.39: PPO-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.88Hz$

For the case of the improved reward function, represented by Equation (5.3) and Equation (5.4), the robust nature of the RL controller is demonstrated through the results obtained. The RL controller is able to handle both external disturbances and model uncertainty, within the range defined. Figure 6.40, Figure 6.41, and Figure 6.42 show the simulated results of the PPO RL controllers obtained after training on the respective environments with the beam-actuator dynamics derived from $\omega_n = 49.49Hz, \omega_n = 49.21Hz,$ and $\omega_n = 50.31Hz$, in that order. The order of amplitude displacement of the beam-actuator system midpoint after 200 seconds is summarized in Table 6.4. Again, the values presented in Table 6.4 indicate that the controller performance is much better for the improved reward function compared to the original reward function. The improvement of the reward function allows for a more optimal control scheme to be learnt and implemented on the system.

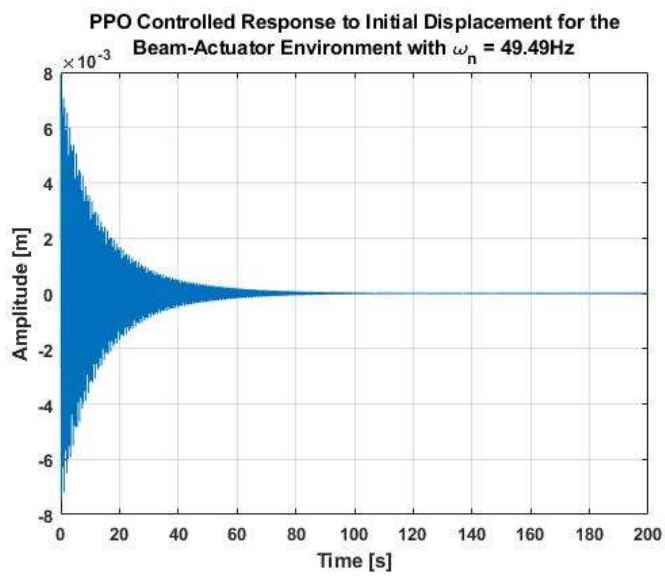


Figure 6.40: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

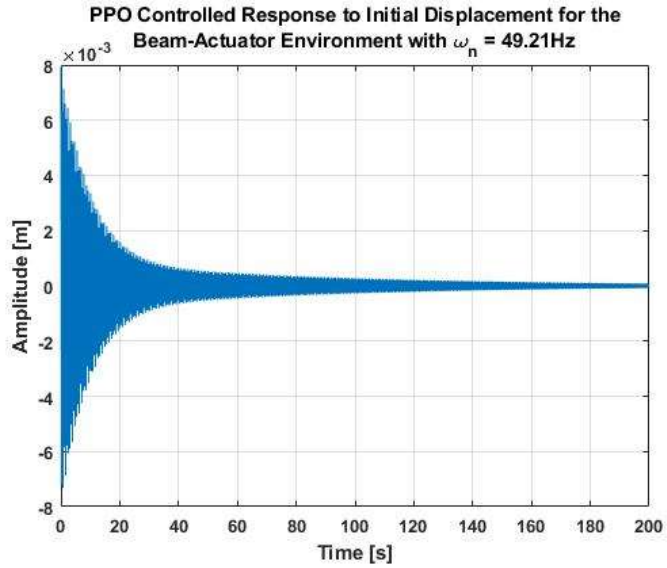


Figure 6.41: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.21Hz$

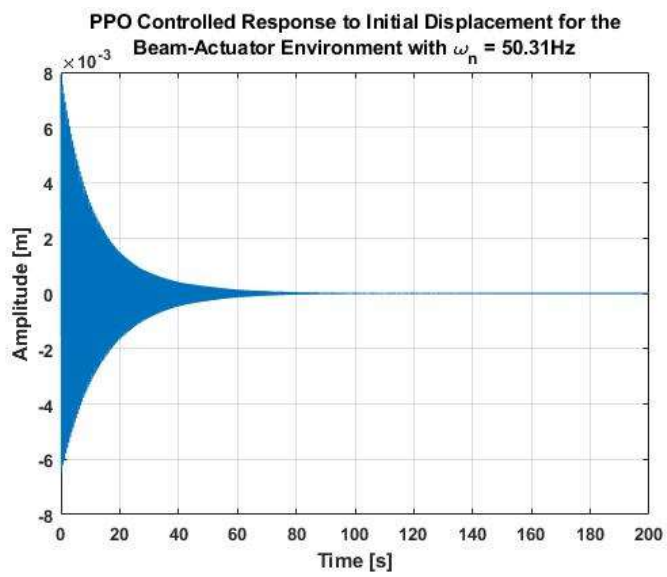


Figure 6.42: PPO controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.21Hz$

Table 6.4: The order of displacement of the beam-actuator controlled response at 100 and 200 seconds

Dynamics SAC controller trained on	*Order of displacement at 100s [m]*	*Order of displacement at 200s [m]*
49.49 Hz	10^{-8}	10^{-10}
49.21 Hz	10^{-7}	10^{-10}
50.31 Hz	10^{-8}	10^{-12}

Note that after training the PPO controllers on multiple different beam-actuator environments, the PPO-MET is simulated on a randomly chosen environment. To compare the effect of training on multiple environments, the same environment that is randomly selected for the PPO-MET is simulated for using the agent trained only on the first, unaltered beam-actuator environment with a natural frequency, $\omega_n = 49.49Hz$. The randomly chosen beam-actuator environment, in this case, is derived from $\omega_n = 50.31Hz$. The controlled response of the agent trained on a different environment is shown in Figure 6.43. The results indicate that the controller is still able to suppress the vibration but only up to an order of 10^{-7}, even after 200 seconds of simulation. The PPO-MET controller is also simulated on the same environment with $\omega_n = 47.01Hz$ and the results obtained are illustrated in Figure 6.44. The results demonstrate that with the improved reward function, the controller learns a policy that enables it to develop a general control scheme after training on different dynamics. The robust nature of the RL controller trained on multiple environments is also illustrated in the results, as the PPO-MET, Figure 6.44, demonstrates better performance compared to the PPO controller trained on the environment itself, Figure 6.42. Even if untrained for these randomly selected environments, the performance of the RL controller is maintained with the agent being able to suppress the vibration for those environments. Within a reasonable range, the controller can be applied with confidence, to take care of the uncertainties in real life where the variables may not be known or accurate, for a thin beam with similar properties.

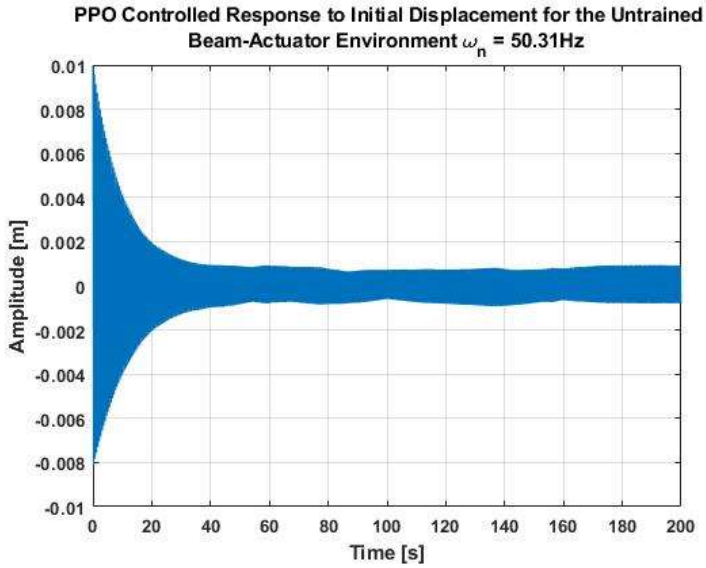


Figure 6.43: PPO controlled response to an initial displacement of 8mm for the untrained beam-actuator system with $\omega_n = 49.49Hz$

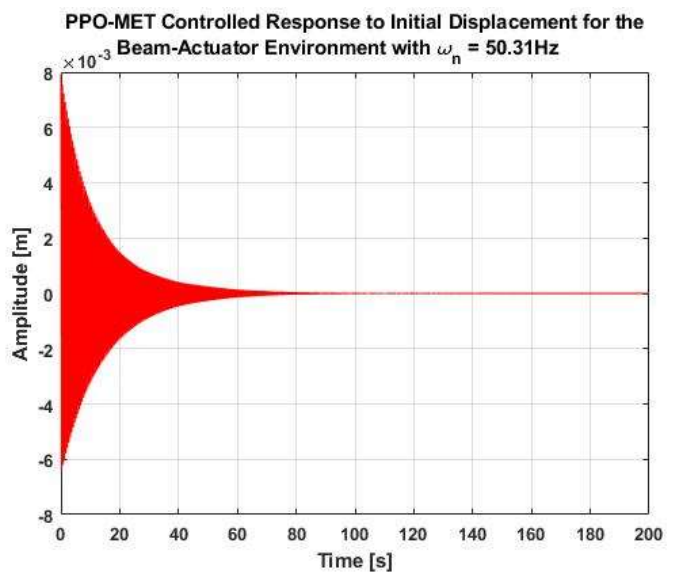


Figure 6.44: PPO-MET controlled response to an initial displacement of 8mm for the beam-actuator system with $\omega_n = 49.49Hz$

For the improved reward function, it is observed that the PPO and SAC agents are well suited for the vibration control of the uncertain parameter systems that they were trained on and even those that they are interacting with anew. This is possible because the agent interacts with the environment and acquires information to update the policy that represents the governing function. The observations that the environment sends to the agent along with the reward signal influence the actor and critic weights. As the agent learns and interacts with an environment, it can adjust the function that governs the control input into the system. This means that even with a change in the environment dynamics the agent is still able to learn as it simulates and interacts with the environment. The agent chooses actions based on the reward incentive and therefore is reliant on the reward shape to attain desired behaviour when the initial conditions or environment dynamics change.

CHAPTER 7

CONCLUSION

7.1 General Research Conclusion

This thesis presents the vibration control of thin structures using a reinforcement learning approach. With an aim to develop a robust controller to suppress the vibration of a pinned-pinned beam system, a reinforcement learning controller is developed. Two algorithms are selected as the reinforcement controllers, Soft Actor-Critic (SAC) and Proximal Policy Optimization (PPO). In order to generate a robust and adaptive controller, the initial displacements to the system midpoint, which represent the external disturbances, are randomized during agent training. It is well established that the reinforcement agents represent the controllers and the environments represent the beam-actuator systems. Considering the changes that occur once a space structure encounters thermal loading, in this case, a change in length and cross-sectional area, the natural frequency of the beam is subject to change. For this reason, the need to account for even more system uncertainty arises. In this thesis, parameter uncertainty is modeled into the system through the beam-actuator transfer function by modeling the natural frequency of the system with 5% uncertainty. In the scheme presented, the natural frequency is assumed to vary from its first value by ±5%. The RL agents first trained on the beam-actuator system derived from the first natural frequency of the chosen beam dimensions. Thereafter, they are trained for the different environment dynamics to investigate the performance of the controller considering the designed reward function. The same agent is then trained on multiple different environments by maintaining the beam-actuator system dynamics for every 200 episodes then and loading the pre-trained agent for a different set of system matrices and repeating the process. The training results obtained for the originally crafted reward function indicate that the controller

develops a suboptimal scheme for the vibration problem. They show a similar bounded average reward for the SAC and PPO cases, and the training averages provided for the different beam-actuator systems are close in magnitude for both cases as well. The simulation results are obtained from three different types of agents; one simulated on the environment it is trained on, one simulated on a different environment from what it is trained on, and finally, one simulated on a random environment after being trained on multiple environments. For the agents trained using the original reward function, all three cases indicate responses that exhibit the potential of the reinforcement controllers to dampen the vibration, even with a simple reward function. However, the performance is suboptimal and in need of improvement. This presents one of the challenges in tuning an RL controller.

A detailed account of the reward-shaping process to improve the performance of the controller is provided. Here, emphasis is placed on eliminating the residual vibration that is observed with the simulated results from the controllers trained with the original reward function. With this, an improved reward function is developed, the agents are re-trained, and the simulation results for the new scheme are obtained. Despite illustrating bounded episode rewards that indicate that the agent is stuck in a local optimum, the training results indicate better training with higher average rewards. The re-trained agents are simulated on various environments as well and with the same three groups presented in the previous paragraph. The displacement vs. time results obtained indicate much better performance, with the amplitude tending to 0 the longer the simulation runs. The results illustrate that the agents trained on those environments and the agents trained on multiple environments perform comparatively. This exhibits the robustness of the RL controllers developed from the improved reward function.

Through all the results, it is observed that the PPO and SAC agents are well suited for the vibration control of the original system as well as the uncertain parameter systems, even those that they were not trained on. This is possible because the agent interacts with the environment and updates the policy with information about the observations and reward signals. Slowly, the agent tries to learn a function that

governs the dynamics of the system by interacting with the environment. Even with a change in the environment dynamics, the agent is still able to learn as it simulates and interacts with the environment. The agent chooses actions based on the reward incentive and therefore is reliant on the reward shape to attain desired behaviour when the initial conditions or environment dynamics change. Both SAC and PPO agents exhibit comparable results for the vibration suppression of the pinned-pinned beam and can mitigate the model uncertainties and the random initial excitation.

Finally, this work presents the vibration control of a thin beam using a reinforcement control approach. It is to be noted that this work is motivated by research on the suppression of thin space structures. However, because of the choice of beam dimensions, the work presented here is not reflective of the vibration control of real-life space structures. This work aims to provide a reference for works considering structures of that size and acts as a starting point to just illustrate if the reinforcement learning controllers developed can handle system uncertainty. It also aims to provide a logical explanation for reward shaping that can be used in cases with similar problems. With this in mind, the field of reinforcement learning vibration control has only slightly been explored.

7.2 Future Work

The author suggests the following items for future work related to this thesis topic:

- Reinforcement learning control can be applied for the vibration control of a system that represents more vibration modes. In this work, only the first mode is considered, which captures a considerable amount of the vibration dynamics of the system. However, expanding the system to include a few more vibration modes will make the controller more suited for real-life applications.
- A deeper comparative study of the reward function shaping, with an aim to attain optimal behavior, can be done. Reward-shaping is one of the most

crucial parts of developing a reinforcement learning agent. How well the agent learns relies on how well the reward function is shaped to facilitate learning. A study solely focused on the development of a good reward function is vital for the continued success of research related to reinforcement learning for vibration control.

- Model uncertainty can be expanded to include more than just parameter uncertainty and external disturbances. If the inclusion of more uncertainty within the vibrating system can be justified, then modeling the additional uncertainty will result in increased robustness of the developed controller.
- A different model for the system can be considered. In this thesis, the state space system matrices obtained to represent the system dynamics were directly transformed from the beam-actuator system's transfer function. Alternatively, the state system matrices can be obtained from model order reduction of the mass, damping, and stiffness matrices obtained using finite element modeling.
- Experimental validation of the use of reinforcement learning for vibration control of a pinned-pinned beam with changing initial excitations or some structural changes can be carried out.

REFERENCES

[1] T. S. Lee and E. A. Alandoli, "A critical review of modelling methods for flexible and rigid link manipulators," *Journal of Brazilian Society of Mechanical Sciences and Engineering,* vol. 42, no. 508, 2020.

[2] X. Liu, G. Cai, F. Peng and Z. F, "Piezoelectric Actuator Placement Optimization and Active Vibration Control of a Membrane Structure," *Acta Mech. Solida Sin.,* vol. 31, p. 66–79, 2018.

[3] I. Ferhat and C. Sultan, "System Analysis and Control Design for a Membrane with Bimorph Actuators," *AIAA Journal,* vol. 53, no. 8, 2015.

[4] I. Ferhat and C. Sultan, "LQR Using Second Order Vector Form for a Membrane with Bimorph Actuators," in *23rd AIAA/AHS Adpative Structures Conference*, 2015.

[5] I. Ferhat and C. Sultan, "System Analysis and Control Design for a Membrane with Bimorph Actuator," *AIAA,* vol. 53, no. 8, pp. 2110-2120, 2015.

[6] V. Gupta, M. Sharma and N. Thakur, "Active structural vibration control: Robust to temperature variations," *Mechanical Systems and Signal Processing,* vol. 33, pp. 167-180, 2012.

[7] Y. Liu, Y. Fu, W. He and Q. Hui, "Modeling and Observer-Based Vibration Control of a Flexible Spacecraft With External Disturbances," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 11, pp. 8648-8658, 2019.

[8] A. S. Deshpande and S. B. V., "Passive Damping of Large Space Structures," *AIAA,* vol. 31, no. 8, pp. 1511-1516, 1993.

[9] D. Williams, H. H. Khodaparast, S. Jiffri and C. Yang, "Active vibration control using piezoelectric actuators employing practical components," *Journal of Vibration and Control,* vol. 25, no. 21-22, pp. 2784-2798, 2019.

[10] S. Zhang, S. Rüdiger and X. Qui, "Active vibration control of piezoelectric bonded smart structures using PID algorithm," *Chinese Journal of Aeronautics,* vol. 28, no. 1, pp. 305-313, 2015.

[11] S. Le, "Active vibration control of a flexible beam," San Jose State University, UMI Dissertation Publishing, San Jose, 2009.

[12] G. C. Goodwin, S. F. Graebe and M. E. Salgado, "Design via Optimal Control Techniques," in *Control Systems Design*, Valparaiso, 2000, pp. 708-710.

[13] T. Y. Aksoy, "Active Vibration Control of a Smart Sandwich Plate via Piezoelectric Sensors and Actuators," Middle East Technical University Master's E-theses, Aerospace Engineering, Ankara, 2015.

[14] D. J. Mead, "The Approach to Controlling Vibration," in *Passive Vibration Control*, Sussex, John Wiley & Sons, 2000, pp. 4-8.

[15] E. M. Kerwin Jr., "Damping of Flexural Waves by a Constrained Viscoelastic Layer," *The Journal of the Acoustical Society of America,* vol. 31, no. 952, 2005.

[16] T. Pranoto, K. Nagaya and H. Atsushi, "Vibration suppression of plate using linear MR fluid passive damper," *Journal of Sound and Vibration,* vol. 276, no. 3-5, pp. 919-932, 2004.

[17] G. Takacs, G. Batista, M. Gulan and B. Rohal-Ilkiv, "Embedded explicit model predictive vibration control," *Mechatronics,* vol. 36, pp. 54-62, 2016.

[18] N. W. Hagood and E. F. Crawley, "Experimental Investigation of Passive Enhancement of Damping for Space Structures," *AIAA,* vol. 14, pp. 1100-1109, 1991.

[19] P. J. Lynch and S. S. Banda, "Active Control for Vibration Damping," in *Large Space Structures: Dynamics and Control*, 1988, pp. 239-261.

[20] I. Ferhat and C. Sultan, "LQG Control and Robustness Study for a Prestressed Membrane with Bimorph Actuators," in *ASME International Engineering Technical Conferences* , 2014.

[21] E. R. Ruggiero and D. J. Inman, "Modeling and Control of a 1-D Membrane Strip with an Integrated PZT Bimorph," in *International Mechanical Engineering Congress and Exposition*, Orlando, 2005.

[22] J. D. Inman and E. J. Ruggiero, "Modeling and vibration control of an active membrane mirror," *Smart Materials and Structures,* vol. 18, 2019.

[23] C. Guo, D. Lu, M. Zhang, Q. Zhang and C. Chen, "Active control technology for flexible solar array disturbance suppression," *Aerospace Science and Technology,* vol. 106, 2020.

[24] F. M. Giovanni and M. Amabili, "Active vibration control of a sandwich plate by non-collocated positive position feedback," *Journal of Sound and Vibration,* vol. 342, no. 44-56, 2015.

[25] I. Ferhat, "Development and Application of Modern Optimal Controllers for a Membrane Structure Using Vector Second Order Form," ETDs: Virginia Tech Electronic Theses and Dissertations , Virginia, 2015.

[26] F. An, W.-d. Chen and M.-q. Shao, "Dynamic behavior of time-delayed acceleration feedback controller for active vibration control of flexible structures," *Journal of Sound and Vibration,* vol. 333, pp. 4789-4809, 2014.

[27] W. He, T. Wang, X. He, L.-J. Yang and O. Kaynak, "Dynamical Modeling and Boundary Vibration Control of a Rigid-Flexible Wing System," *IEEE/ASME Transactions on Mechatronics,* vol. 25, no. 6, pp. 2711-2721, 2020.

[28] J. A. Gustafson and M. P. S, "Flexible Spacestructure Control Via Moving-Bank Multiple Model Algorithms," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 30, no. 3, pp. 750-757, 1994.

[29] R. Xu, D. Li and J. Jiang, "Online learning fuzzy vibration control of smart truss structure," *Journal of Aerospace Engineering,* vol. 231, no. 3, pp. 548-557, 2016.

[30] M. S. Yang and S. G. Lee, "Vibration Control of Smart Structures by Using Neural Networks," *Journal of Dynamic Systems, Measurement, and Control,* vol. 119, pp. 34-39, 1997.

[31] A. Homaifar, Y. Shen and B. V. Stack, "Vibration Control of Plate Structures Using PZT Actuators and Type II fuzzy logic," in *American Control Conference*, Arlington, 2001.

[32] R. S. Sutton and A. G. Barto, "Introduction: Reinforcement Learning," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[33] E. Bøhn, E. M. Coates, S. Moe and T. A. Johansen, "Deep Reinforcement Learning Attitude Control of Fixed-Wing UAVs Using Proximal Policy Optimization," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019.

[34] G. M. Barros and E. L. Colombini, "Using Soft Actor-Critic for Low-Level UAV Control.," arXiv, São Paulo, 2020.

[35] K. Hovell and S. Ulrich, "Deep Reinforcement Learning for Spacecraft Proximity Operations Guidance," *Journal of Spacecraft and Rockets,* vol. 58, no. 2, pp. 254 - 264, 2021.

[36] L. Federici, B. Benedikter and A. Zavoli, "Deep Learning Techniques for Autonomous Spacecraft Guidance During Proximity Operations," *Journal of Spacecraft and Rockets,* vol. 58, no. 6, 2021.

[37] W. Koch, R. Mancuso, R. West and A. Bestavros, "Reinforcement Learning for UAV Attitude Control," arXiv, 2018.

[38] S. Li, T. Lu, C. Zhang, D. T. Yeung and S. Shen, "Learning Unmanned Aerial Vehicle Control for Autonomous Target Following," arXiv, 2017.

[39] C. Wilson and A. Riccardi, "Improving the efficiency of reinforcement learning for a spacecraft powered descent with Q-learning," *Optimization and Engineering,* vol. 24, pp. 223-255, 2021.

[40] G. C. Lopes, M. Ferreira, A. Silva Simoes and E. L. Colombini, "Intelligent Control of a Quadrotor with Proximal Policy Optimization Reinforcement Learning," in *IEEE Latin American Robotics Symposium, LARS*, 2018.

[41] F. R. Gaudet, "Deep reinforcement learning for six degree-of-freedom planetary landing," *Advances in Space Research,* vol. 65, no. 7, pp. 1723-1741, 2020.

[42] S. Willis, D. Izzo and D. Hennes, "Reinforcement Learning for Spacecraft Maneuvering Near Small Bodies," in *AAS/AIAA Space Flight Mechanics Meeting*, Napa, 2016.

[43] Vedant, J. T. Allison, M. West and A. Ghosh, "Reinforcement Learning for Spacecraft Attitude Control," in *International Astronautical Congress*, Washington D.C., 2019.

[44] T. Long, E. Li, Y. Hu, L. Yang, F. Junfeng, Z. Liang and R. Guo, "A Vibration Control Method for Hybrid-Structured Flexible Manipulator Based on Sliding Mode Control and Reinforcement Learning," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 32, no. 2, pp. 841-852, 2021.

[45] Z. C. Qiu, J. H. Du and X. M. Zhang, "Vibration control of three coupled flexible beams using reinforcement learning algorithm based on proximal policy optimization," *Journal of Intelligent Material Systems and Structures,* pp. 1-26, 2022.

[46] Z. C. Qiu, J. H. Du and X. M. Zhang, "Reinforcement learning vibration control for a flexible hinged plate," *Aerospace Science and Technology,* vol. 118, 2021.

[47] S. N. Wanyonyi, I. Ferhat and D. F. Kurtuluş, "Comparative Study on Vibration Control Using Reinforcement Learning," in *Recent Advances in Air and Space Technologies*, Istanbul, 2023.

[48] Y. F. Comez, K. D. F and K. B. Arikan, "Unsteady Aerodynamic Analysis of a Flapping Wing Actuated with PZT Material," in *Workshop on Non-Intrusive Measurements of Unsteady Flows and Aerodynamics*, Poitiers, 2015.

[49] Y. F. Comez and D. F. Kurtulus, "Effect of Forward Velocity on Rectangular Wings Flapping with Piezoelectric Actuators," in *International Symposium on Sustainable Aviation*, Rome, 2018.

[50] O. Harputlu and D. F. Kurtulus, "Development and Analysis of flat plate in flapping motion using piezoelectric actuators," in *7th Ankara International Aerospace Conference*, Ankara, 2013.

[51] Y. Ekici, "Nonlinear vibration analysis of L-shaped beams and their use in vibration reduction," Middle East Technical University Master's E-Theses, Mechanical Engineering, Ankara, 2022.

[52] K. Kopşa, "Reinforcement Learning Control for Autorotation of a Simple Point-Mass Helicopter Model," Middle East Technical University Master's E-Theses, Aerospace Engineering, Ankara, 2018.

[53] İ. H. Uğurlu, "A Comparative Study of Learning Based Control Policies and Conventional Controllers on 2D Bi-Rotor Platform with Tail Assistance," Middle East Technical University Master's E-Theses, Electrical and Electronic Engineering, Ankara, 2019.

[54] M. S. Camcı, "Structural modifications in optomechanical systems for vibration reduction by using sequential model," Middle East Technical University Master's E-Theses, Mechanical Engineering, Ankara, 2020.

[55] Ü. Özalp, "Bipedal robot walking by reinforcement learning in partially observed environment," Middle East Technical University Master's E-Theses, Scientific Computing, Ankara, 2021.

[56] K. B. Ünal, "A comparative study of deep reinforcement learning methods and conventional controllers for aerial manipulation," Middle East Technical University Master's E-Theses, Computer Engineering, Ankara, 2021.

[57] R. E. Şenöz, "Evaluation of the Robustness Performance of a Fuzzy Logic Controller for Active Vibration Control of a Piezo-Beam via Tip Mass Location Variation," Middle East Technical University Master's E-Theses, Aersopace Engineering Department, Ankara, 2019.

[58] A. B. Chmielewski and C. H. Jenkins, "Gossamer Spacecraft," in *Compliant Structures in Nature and Engineering*, WIT Press, 2005, pp. 203-243.

[59] E. Vitug, "Deployable Composite Booms," NASA, 24 May 2022. [Online]. Available: https://www.nasa.gov/directorates/spacetech/game_changing_development/projects/dcb. [Accessed 28 June 2022].

[60] F. Angeletti, P. Gasbarri and M. Sabattini, "Optimal design and robust analysis of a net of active devices for microvibration control of an on-orbit large space antenna," *Acta Astronica,* vol. 164, pp. 241-253, 2019.

[61] S. S. Rao, "Continuous Systems: Lateral vibration of beams," in *Mechanical Vibrations*, Saddle River, Pearson Education Inc., 2011, pp. 721-739.

[62] S. S. Rao, "Transverse Vibration of Beams," in *Vibration of Continuous Systems*, New Jersey, John Wiley & Sons, 2007, pp. 317-390.

[63] A. Erturk and D. J. Inman, Piezoelectric Energy Harvesting, West Sussex: John Wiley & Sons, 2011.

[64] E. Balmes and A. Deraemaeker, "Basics of piezoelectricity," in *Modeling structures with piezoelectric materials. Theory and SDT Tutorial*, Paris, SDTools, 2001, pp. 4-34.

[65] S. M. F. Moghaddam and H. Ahmadi, "Active vibration control of truncated conical shell under harmonic excitation using piezoelectric actuator," *Thin-Walled Structures,* vol. 151, 2020.

[66] W. Voigt, Lehrbuch der Kristallphysik (mit Ausschluss der Kristalloptik), Wiesbaden: Springer Fachmeiden, 1966.

[67] Smart Material, "Smart Material - Home of the MFC," Smart Material, [Online]. Available: https://www.smart-material.com/MFC-product-mainV2.html. [Accessed 4 June 2023].

[68] N. W. Hagood, R. Kindel, K. Ghandi and P. Gaudenzi, "Improving transverse actuation of piezoceramics using interdigitated surface electrodes," in *Smart Structures and Materials 1993: Smart Structures and Intelligent Systems*, Albuquerque, NM, 1993.

[69] S.-B. Kim, H. Park, S.-H. Kim, C. H. Wikle, J.-H. Park and D.-J. Kim, "Comparison of MEMS PZT Cantilevers Based on d31 and d33 Modes for Vibration Energy Harvesting," *Journal of Microelectromechanical Systems,* vol. 22, no. 1, pp. 26-33, 2013.

[70] H. R. Pota, R. S. Moheimani and M. Smith, "Resonant controllers for smart structures," *Smart Materials and Structures,* vol. 11, no. 1, pp. 1-8, 2002.

[71] K. Ogata, "Control Systems Analysis in State Space," in *Modern Control Engineering*, Saddle River, Pearson Education Inc., 2012, pp. 648-721.

[72] University of Hawaii, "A gude to cubesat mission and design: 4.6 Structural Analysis," Pressbooks.

[73] D. S. Kolosa, "A Reinforcement Learning Approach to Spacecraft Trajectory," Western Michigan University, Kalamazoo, 2019.

[74] R. S. Sutton and A. G. Barto, "The Reinforcement Learning Problem: The Agent-Environment Interface," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[75] R. S. Sutton and A. G. Barto, "Introduction: Elements of Reinforcement Learning," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[76] R. S. Sutton and A. G. Barto, "The Reinforcement Learning Problem: Value Functions," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[77] R. S. Sutton and A. G. Barto, "The Reinforcement Learning Problem: Goals and Rewards," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[78] R. S. Sutton and A. G. Barto, "The Reinforcement Learning Problem: Markov Decision Processes," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[79] R. S. Sutton and A. G. Barto, "Temporal Difference Learning: Introduction," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[80] OpenAI, "Proximal Policy Optimization," OpenAI Spinning Up, 2018. [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/ppo.html. [Accessed 4 September 2022].

[81] OpenAI Spinning Up, "Soft Actor-Critic," 2018. [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/sac.html. [Accessed 4 September 2022].

[82] R. S. Sutton and A. G. Barto, "Generalization and Function Approximation," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[83] C. C. Aggarwal, "Deep Reinforcement Learning: Deep Learning Models as Function Approximators," in *Neural Networks and Deep Learning*, Cham, Springer International Publishing, 2018, pp. 384-386.

[84] R. S. Sutton and A. G. Barto, "Value Prediction with Function Approximation," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[85] R. S. Sutton and A. G. Barto, "Temporal Difference Learning," in *Reinforcement Learning: An Introduction*, Cambridge, MIT Press, 1998.

[86] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," arXiv, 2018.

[87] V. V. Kumar, "Soft Actor-Critic Demystified: An intuitive explanation of the theory and a PyTorch implementation guide," 9 January 2019. [Online]. Available: https://towardsdatascience.com/soft-actor-critic-demystified-b8427df61665. [Accessed 21 May 2022].

[88] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel and S. Levine, "Soft Actor-Critic Algorithms and Applications," arXiv, 2018.

[89] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal Policy Optimization Algorithms," OpenAI, 2017.

[90] R. S. Sutton, A. G. Barto and R. J. Williams, "Reinforcement Learning is Direct Adaptive Optimal Control," in *American Control Conference*, Boston, 1991.

[91] B. R. Beck, J. Tipper and S. Su, "Comparison of Constant PID Controller and Adaptive PID Controller via Reinforcement Learning for a Rehabilitation Robot," in *Control Conference, Australia and New Zealand*, Gold Coast, 2022.

[92] W. Caarls, "Deep Reinforcement Learning with embedded LQR Controllers," *IFAC-PapersOnLine,* vol. 53, no. 2, pp. 8063-8069, 2020.

[93] I. Carlucho, M. De Paula and G. G. Acosta, "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," *ISA Transactions,* vol. 102, no. July, pp. 280-294, 2020.

[94] J. Shuprajhaa, K. S. Sujit and K. Srinivasan, "Reinforcement learning based adaptive PID controller design for control of linear/nonlinear unstable processes," *Applied Soft Computing,* vol. 128, no. 109450, 2022.

[95] Q. Sun, C. Du, Y. Duan, H. Ren and H. Li, "Design and application of adaptive PID controller based on asynchronous advantage actor–critic learning method," *Wireless Networks,* vol. 27, pp. 3537-3547, 2021.

[96] C. W. Wong and J. H. Lee, "A Reinforcement Learning-Based Scheme for Adaptive Optimal Control of Linear Stochastic Systems," in *American Control Conference*, Seattle, 2008.

[97] D. P. Kingma and L. J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, San Diego, 2015.

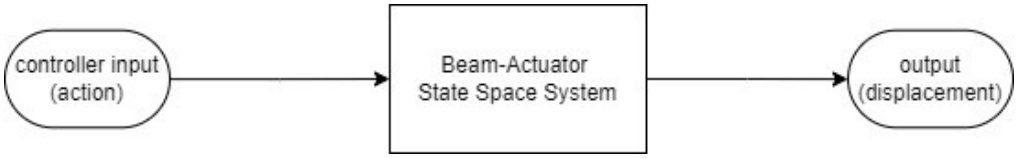
[98] J. Schulman, P. Moritz, S. Levine, M. Jordan and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *International Conference on Learning Representations*, San Juan, 2016.

[99] C. C. Aggarwal, "An Introduction to Neural Networks: The Basic Architecture of Neural Networks," in *Neural Networks and Deep Learning*, Cham, Springer International Publishing, 2018, pp. 4-21.

APPENDICES

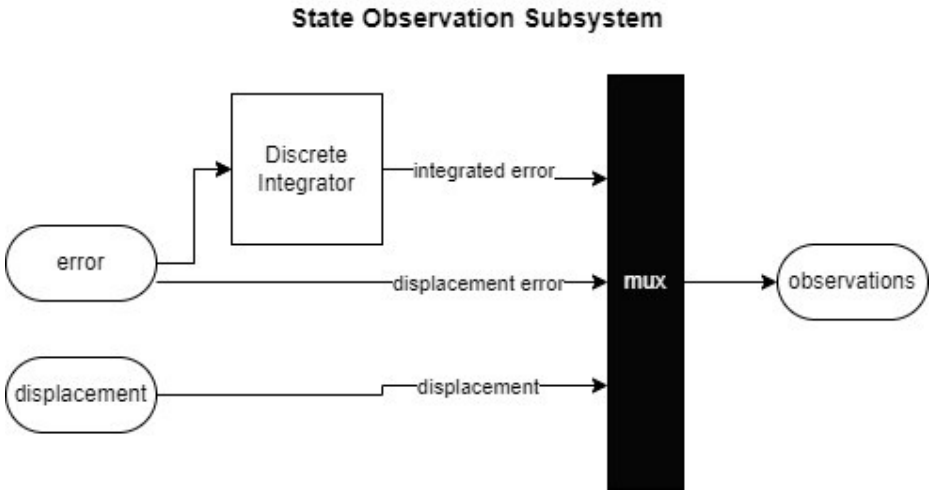
A. Reinforcement Learning Beam-Actuator and Controller Subsystems

The subsystems in the beam-actuator and reinforcement learning controller are presented here. The beam-actuator state space system within the "Beam-Actuator" subsystem block is displayed below:



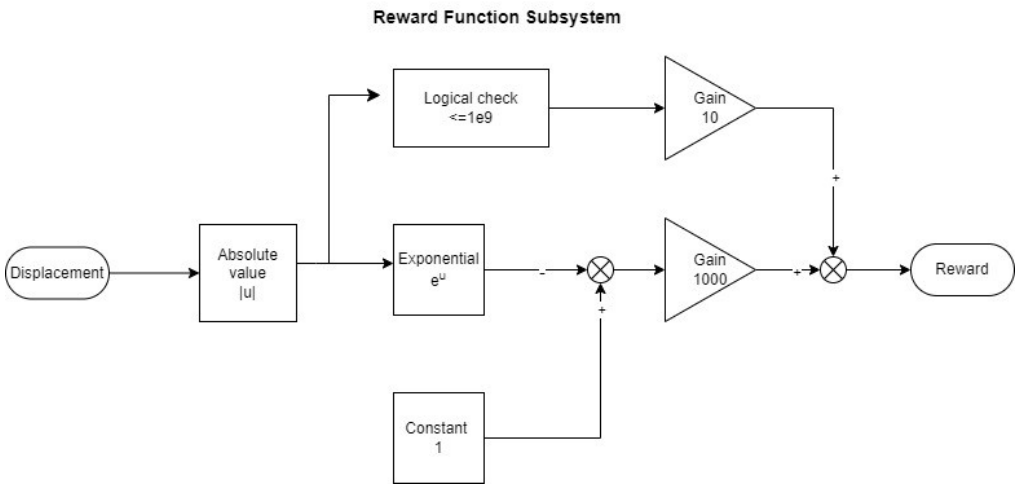
Appendix Figure 1: Beam-Actuator System

The observation calculation blocks within the "get observations" subsystem block is shown below:



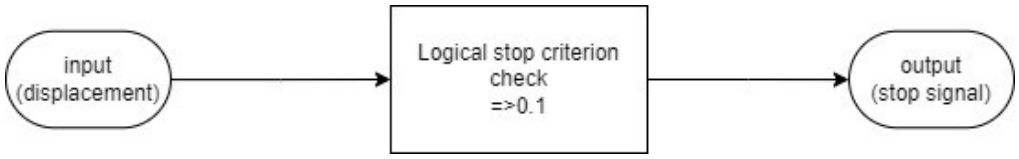
Appendix Figure 2: Observation generation subsystem. Integrated displacement error, displacement error and displacement.

The reward signal computation blocks within the “calculate reward” subsystem block is shown below:



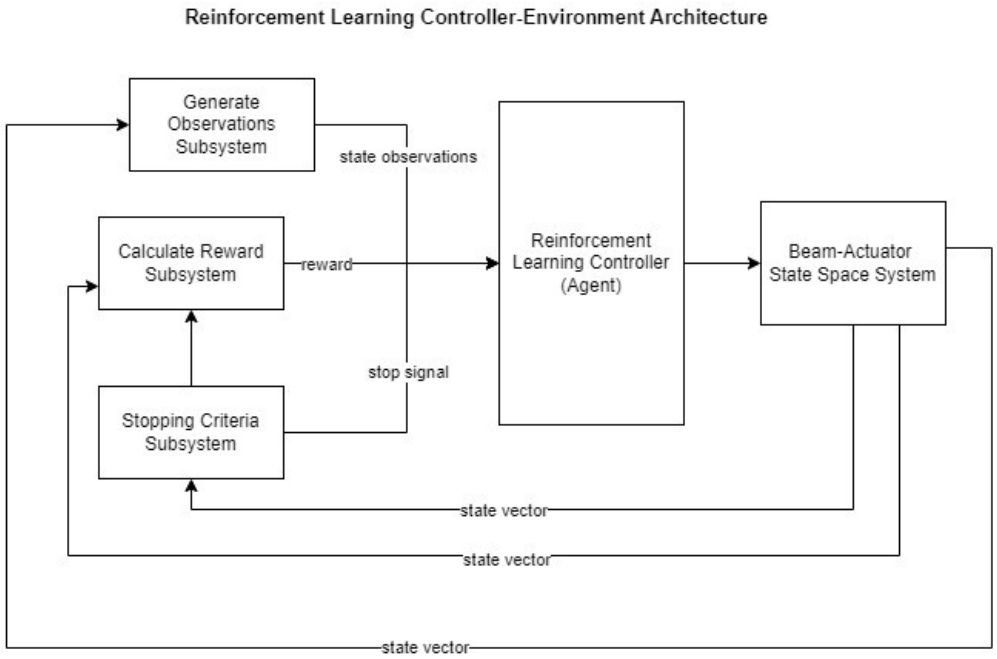
Appendix Figure 3: Reward function subsystem that determines how well the reinforcement learning controller (agent) learns

The stopping criteria logical blocks within the “stop simulation” subsystem block is shown below:



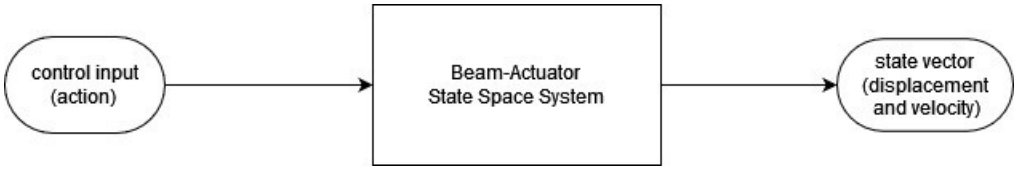
Appendix Figure 4: Stopping criteria subsystem. Sets the conditions for the episode (simulation run) to end

B. Reinforcement Learning Beam-Actuator and Controller Subsystems – Improved



Appendix Figure 5: Reinforcement Learning Controller-Environment Architecture. Shows the controller and environment subsystems as defined in Simulink.

The beam state space subsystem is represented as:



Appendix Figure 6: Beam-Actuator System. Inside the beam-actuator subsystem in Simulink.

C. Reinforcement Learning Controller Code

The randomized initial excitations to the system are coded in the reset function in MATLAB as:

```
function in = localResetFcn(in)
% Set the reference signal
blk = sprintf('rlpinbeam3/Desired \nDisplacement');
x = 0;
in = setBlockParameter(in,blk,'Value',num2str(x));

% Randomize initial displacement
x = 0.01-0.005*randn;
blk = 'rlpinbeam3/Beam-Actuator System/State-Space System';
in = setBlockParameter(in,blk,'InitialCondition',num2str(x));
end
```

The pseudocode of the soft actor critic algorithm is given as:

Soft Actor Critic Algorithm	
Input: $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{\phi}$	Initial parameters
$\boldsymbol{\theta_1} \leftarrow \boldsymbol{\theta_1}, \boldsymbol{\theta_2} \leftarrow \boldsymbol{\theta_2}$	Initialize network target weights
$\boldsymbol{D} \rightarrow \emptyset$	Initialize an empty replay pool
For each iteration do	
For each environment step do	
$\boldsymbol{a_t} \sim \boldsymbol{\pi_\phi}(\boldsymbol{a_t}\|\boldsymbol{s_t})$	Sample action from the policy
$\boldsymbol{s_{t+1}} \sim \boldsymbol{p}(\boldsymbol{s_{t+1}}\|\boldsymbol{s_t}, \boldsymbol{a_t})$	Sample transition from the environment
$\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{D}} \cup \{(\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{r}(\boldsymbol{s_t}, \boldsymbol{a_t}), \boldsymbol{s_{t+1}})\}$	Store the transition in the replay pool
End for	
For each gradient step do	
$\boldsymbol{\theta_i} \leftarrow \boldsymbol{\theta_i} - \boldsymbol{\lambda_Q} \hat{\boldsymbol{\nabla}}_{\theta_i} \boldsymbol{J_Q}(\boldsymbol{\theta_i}), \boldsymbol{i} \in \{\mathbf{1}, \mathbf{2}\}$	Update Q-function parameters
$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \boldsymbol{\lambda_\pi} \hat{\boldsymbol{\nabla}}_{\phi} \boldsymbol{J_\pi}(\boldsymbol{\phi})$	Update policy weights
$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \boldsymbol{\lambda} \hat{\boldsymbol{\nabla}}_{\alpha} \boldsymbol{J}(\boldsymbol{\alpha})$	Adjust temperature
$\overline{\boldsymbol{\theta}}_{\boldsymbol{i}} \leftarrow \boldsymbol{\tau}\boldsymbol{\theta_i} + (\mathbf{1} - \boldsymbol{\tau})\overline{\boldsymbol{\theta}}_{\boldsymbol{i}}, \boldsymbol{i} \in \{\mathbf{1}, \mathbf{2}\}$	Update target network weights
End for	
End for	
Output: $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{\phi}$	Optimized parameters

The pseudocode of the proximal policy optimization algorithm is given as:

Proximal Policy Optimization Algorithm

For iteration 1, 2, ... do
  For iteration 1, 2, ..., N, do
    Run policy $\pi_{\theta old}$ in environment for T timesteps
    Compute advantage estimates $\widehat{A}_1, \dots, \widehat{A}_T$
  End for
  Optimize for surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{old} \leftarrow \theta$
End for