



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**VIBRATION REDUCTION ALGORITHMS BASED
ON MODEL PREDICTIVE CONTROL FOR LARGE
FLEXIBLE SPACE STRUCTURES**

by

Patrick S. Clouse

December 2023

Thesis Advisor:

Co-Advisor:

Second Reader:

Jennifer Hudson

Carson R. Vogt

Marcello Romano

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 2023	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE VIBRATION REDUCTION ALGORITHMS BASED ON MODEL PREDICTIVE CONTROL FOR LARGE FLEXIBLE SPACE STRUCTURES			5. FUNDING NUMBERS	
6. AUTHOR(S) Patrick S. Clouse				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) This thesis investigates the efficacy of Model Predictive Control (MPC) algorithms in reducing vibrations in large flexible space structures, focusing on the Space Solar Power Incremental Demonstrations and Research (SSPIDR) project. It hypothesizes that MPC algorithms can enhance stability and performance in space environments, particularly in managing spacecraft dynamics and vibrations. The study uses a simulated spacecraft model connected to a flexible solar structure through a revolute joint to test the MPC algorithms. These algorithms demonstrated improved vibration control, confirming the hypothesis by adapting to the dynamic complexities of the spacecraft-beam system and enhancing stability. This research highlights the potential of MPC algorithms in aerospace engineering, especially for space-based solar power systems, and suggests further development and application in more complex scenarios, along with future research in algorithm refinement and real-world testing.				
14. SUBJECT TERMS Model Predictive Control, MPC, Space Solar Power Incremental Demonstrations and Research, SSPIDR, spacecraft dynamics, vibration control, spacecraft-beam system, aerospace engineering, space-based solar power systems			15. NUMBER OF PAGES 103	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**VIBRATION REDUCTION ALGORITHMS BASED ON MODEL PREDICTIVE
CONTROL FOR LARGE FLEXIBLE SPACE STRUCTURES**

Patrick S. Clouse
Lieutenant, United States Navy
BS, Brescia College, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
December 2023**

Approved by: Jennifer Hudson
Advisor

Carson R. Vogt
Co-Advisor

Marcello Romano
Second Reader

Brian S. Bingham
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis investigates the efficacy of Model Predictive Control (MPC) algorithms in reducing vibrations in large flexible space structures, focusing on the Space Solar Power Incremental Demonstrations and Research (SSPIDR) project. It hypothesizes that MPC algorithms can enhance stability and performance in space environments, particularly in managing spacecraft dynamics and vibrations. The study uses a simulated spacecraft model connected to a flexible solar structure through a revolute joint to test the MPC algorithms. These algorithms demonstrated improved vibration control, confirming the hypothesis by adapting to the dynamic complexities of the spacecraft-beam system and enhancing stability. This research highlights the potential of MPC algorithms in aerospace engineering, especially for space-based solar power systems, and suggests further development and application in more complex scenarios, along with future research in algorithm refinement and real-world testing.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	BACKGROUND	5
A.	APPROACH.....	6
B.	FACILITIES	7
C.	EXPERIMENTAL SETUP	8
D.	ERA OVERVIEW	9
E.	MPC OVERVIEW.....	15
F.	ERA/MPC UTILIZATION PROBLEM.....	18
III.	METHODS	27
A.	PROBLEM STATEMENT	27
B.	LINEAR CONTROL USING MPC	30
C.	NONLINEAR CONTROL USING MPC	35
D.	OPTIMAL VIBRATION REDUCTION.....	41
IV.	ANALYSIS	45
A.	ROS GAZEBO RESULTS.....	45
B.	IMPLEMENTATION OF MPC IN FSS	50
1.	Experimental set-up.....	54
C.	FSS RESULTS	56
1.	MPC Test 1	56
2.	MPC Test 2	58
V.	CONCLUSION	63
A.	SUMMARY	63
B.	FUTURE WORK.....	64
APPENDIX A. ERA STATE SPACE WITH MPC CONTROLLER MATLAB		
	CODE	67
APPENDIX B. LINEAR MPC CONTROLLER MATLAB CODE		71
APPENDIX C. NONLINEAR MPC CONTROLLER MATLAB CODE.....		75

APPENDIX D. NONLINEAR DYNAMICS OF THE SYSTEM MATLAB CODE	81
LIST OF REFERENCES	83
INITIAL DISTRIBUTION LIST	85

LIST OF FIGURES

Figure 1.	Artist Rendition of SSPIDR. Source: [3].	3
Figure 2.	Schematic of the FSS setup.	8
Figure 3.	Bridge sensor impulse response data. Source: [2].	11
Figure 4.	Visual representation of truncating values during SVD. Source: [2].	14
Figure 5.	Feedback Control Diagram of MPC. Source: [11].	17
Figure 6.	Spring-Mass-Damper problem model. Source: [14].	18
Figure 7.	The vibrations of the two masses in the system based of the dynamical equations.	20
Figure 8.	Estimated system overlayed with the true system	21
Figure 9.	The state trajectories using MPC.	24
Figure 10.	The step response of the individual states of the system	25
Figure 11.	Schematic illustration of the spacecraft and beam illustrated by Dr. Romano	28
Figure 12.	Feedback loop of the simulated system	30
Figure 13.	Model predictive controller (MPC) of the linearized dynamic equations	33
Figure 14.	Linear Quadratic Controller (LQR) of the linearized dynamic equations	34
Figure 15.	Representation the state trajectories over time	40
Figure 16.	Torque profile	41
Figure 17.	Vibrations of the spacecraft due to its dynamics during a maneuver	42
Figure 18.	Nonlinear MPC controlling the vibrations of the spacecraft and beam system	43
Figure 19.	The FSS and Rigid beam in the ROS simulation (side view)	46
Figure 20.	The FSS and Rigid beam in the ROS simulation (top view)	47

Figure 21.	Top view of the FSS utilizing the MPC torque profile over 20 seconds.....	48
Figure 22.	Graph of the vibration of the beam at the start of the simulation up to 10 seconds.....	49
Figure 23.	Aerial view of the granite table of the POSEIDYN system.....	52
Figure 24.	FSS Beam assembly with markers.....	53
Figure 25.	Spring hinge mechanism.....	55
Figure 26.	Relative beam vibration for test 1	57
Figure 27.	Path of the FSS for test 1	58
Figure 28.	Relative beam vibration for test 2.....	59
Figure 29.	Path of the FSS for test 2	60

LIST OF TABLES

Table 1.	Initial values k and u	12
Table 2.	Parameter descriptions and values used for the simulation. Source [14].	19
Table 3.	Values of the established parameters	31
Table 4.	The FSS and beam system parameters	38
Table 5.	Initial conditions of the state values	38

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AFRL	Air Force Research Laboratory
CGT	Controlled Gas Thruster
COM	Center of Mass
DOF	Degrees of Freedom
ECR	Extended Control Horizon Rate
EOM	Equations of Motion
ERA	Eigensystem Realization Algorithm
FSS	Floating Spacecraft Simulator
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
MISO	Multi Input Multi Output
MOI	Moment of Inertia
MPC	Model Predictive Control
MV	Manipulated Variables
NLMPC	Nonlinear Model Predictive Control
ROS	Robot Operating System
SISO	Single Input Single Output
SRL	Spacecraft Robotics Laboratory
SSPIDR	Space Solar Power Incremental Demonstrations and Research
SVD	Singular Value Decomposition

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I extend my deepest gratitude to Dr. Jennifer Hudson, whose invaluable assistance was crucial in helping me comprehend the complexities of my research problem and deepen my understanding of the project. I am also immensely thankful to Dr. Shambo Bhattacharjee for his guidance and expertise in navigating through the intricacies of my code and for his indispensable support during my experiments. My sincere appreciation goes to Dr. Carson Vogt for his vital contributions to the development and refinement of my simulations.

A special note of thanks to my wife, Sarah Clouse, and my family, whose unwavering support and encouragement have been my pillars of strength throughout this journey. Their patience and understanding have been a constant source of motivation and have played a significant role in the completion of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Spacecraft operations present many complex challenges, ranging from the spacecraft dynamics to the requirements of mission objectives. The ongoing advancements in technology have allowed the use of powerful tools to tackle these challenges, paving the way for innovative solutions that redefine the approach to space exploration. This research project is the culmination of meticulous investigation and innovation, aiming to address one of the most fundamental concerns in spacecraft operations.

The central premise of this research builds upon the potential of model predictive control (MPC) to minimize vibrations during attitude-changing maneuvers of spacecraft. Simultaneously, the study seeks to leverage the capabilities of the Floating Spacecraft Simulator (FSS) and advanced motion capture systems to empirically validate these predictive control strategies.

Space-based solar power (SBSP) and its application through solar power satellites (SPS) offer a revolutionary approach to renewable energy generation, as detailed in the paper “A Contemporary Analysis of the O’Neill – Glaser Model for Space-based Solar Power and Habitat Construction” by Peter A. Curreri and Michael K. Detweiler [1]. This concept involves placing large solar collecting arrays in space to harness solar energy, which is then beamed down to Earth via microwaves, providing a consistent and pollution-free power source. The challenges in implementing SBSP are multifaceted. One of the primary hurdles is the high cost and logistical complexity of launching and assembling large-scale solar arrays in space. Curreri and Detweiler discuss the economic considerations of using materials launched from Earth versus those derived from space, such as lunar or asteroid materials. Additionally, the technological demands for efficient energy transmission from space to Earth, and the construction and maintenance of these large structures in space, present significant engineering challenges.

Despite these challenges, SBSP via SPS offers substantial benefits. SPS can supply continuous solar power, unaffected by Earth’s atmospheric or weather conditions, thus offering a more reliable and consistent energy source compared to terrestrial solar power.

Furthermore, SBSP has the potential to play a crucial role in mitigating global climate change by providing a substantial amount of carbon-free energy. The scalability of SBSP could meet the growing global energy demands while reducing dependence on fossil fuels, making it a pivotal technology for sustainable energy in the future.

SSPIDR, an acronym for Space Solar Power Incremental Demonstrations and Research, is an innovative project spearheaded by the Air Force Research Laboratory's (AFRL) Space Vehicles Directorate. It is a multi-phase, integrated research initiative designed to explore, develop, and ultimately mature the necessary technology for space-based solar power collection and transmission.

The objective of this project is to develop and demonstrate the key steps for SPS attitude control: identification of system parameters using the eigensystem realization algorithm (ERA) [2], control of spacecraft attitude using MPC, and simulation of attitude control of flexible space structures using physics-based modeling software and hardware-in-the-loop experiments. While the direct applications of this work are aligned with the goals of the SSPIDR Program, the implications extend far beyond [3]. By enhancing the attitude change capabilities of large spacecraft through advanced control algorithms, this research can propel future space missions to new frontiers of performance and capability. Furthermore, the rigorous approach to modeling spacecraft dynamics, featuring a dual-reference frame system and parameterization, lays a solid theoretical foundation for this study. The careful blend of theoretical modeling, algorithmic development, and experimental validation ensures that the research outcomes can contribute meaningfully to the broader academic discourse and practical applications in spacecraft operations. Figure 1 shows an artist rendition of SSPIDR satellite.

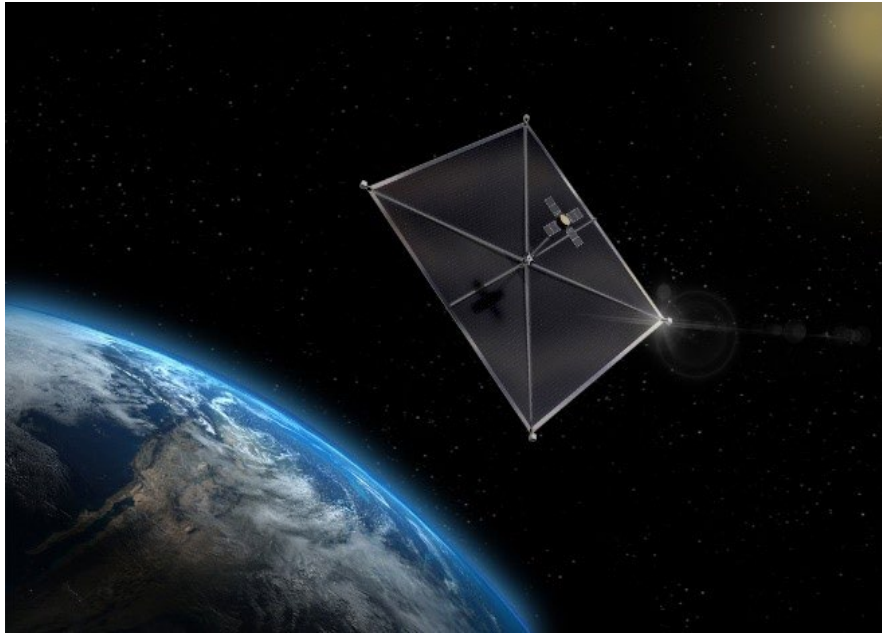


Figure 1. Artist Rendition of SSPIDR. Source: [3].

In essence, this research represents the concerted effort to harness the power of advanced control strategies for space-based applications. It highlights the potential of integrating cutting-edge technology and innovative thinking to address complex challenges in space exploration. The insights collected from this endeavor promise to contribute to the ongoing evolution of spacecraft technology, helping to chart a course towards a future of more stable, efficient, and reliable space missions.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

Solar Power Satellites (SPS), a concept supporting space-based power generation, has remained part of scientific and technological debate for several decades. Despite the apparent longevity of this idea, only in recent years has the requisite technology evolved sufficiently to support such an ambitious undertaking [3]. This advancement not only represents a significant stride in harnessing extraterrestrial resources, but also presents a potential solution to the pressing issues of geographical constraints, resource allocation, and environmental sustainability. These areas of impact converge upon one critical global challenge: our ever-growing energy needs.

The targeted beneficiaries of the SSPIDR technology are predominantly stationed at forward operating bases. Here, the necessity for a stable, reliable power supply often clashes with logistical realities, such as resource transportation and the time-consuming setup of power grid infrastructures. By utilizing space-based solar power, SSPIDR aims to significantly reduce the logistical burden and expedite the operational readiness of these power grids.

In addition to providing a reliable power source, SSPIDR also holds significant implications for the environment. By harnessing solar power from space, it offers a clean, renewable alternative to conventional energy sources, which are known contributors to greenhouse gas emissions. Hence, it could play a transformative role in not only revolutionizing the energy infrastructure at forward operating bases but also mitigating the impacts of global climate change.

Overall, SSPIDR represents a pivotal initiative within the broader sphere of space exploration and technology. Its success could be a game-changer, opening up new horizons in our understanding and utilization of space-based resources while also addressing some of the most pressing challenges of our time. The primary attitude control goal of the SSPIDR mission is to ensure the stability of its payload during orbital cycles and slewing maneuvers. This stability is critical due to the precision required for the mission's tasks and the need to keep error margins minimal for peak performance. However, specific

challenges arise in maintaining this stability. Vibrations, induced by torques applied to the spacecraft's structure for attitude control, are a significant concern. These vibrations are particularly problematic because of the spacecraft's large, thin solar panels, which are prone to experiencing substantial vibrations. Moreover, these panels are at risk of encountering resonances, which could significantly interfere with the accuracy of power-beaming operations. Addressing these vibration issues is therefore essential for the success of the SSPIDR mission.

This study aims to delve into the realm of reducing vibrations, with particular emphasis on attitude-changing maneuvers of the spacecraft. The primary instrument of investigation is the utilization of model predictive control (MPC), a modern control strategy based on the iterative optimization of a plant model over a finite horizon.

A. APPROACH

The primary function of MPC in spacecraft design involves forecasting the spacecraft's motion trajectory and the anticipated torques. These forecasts enable the controller to predict and mitigate potential sources of vibration and instability. Integrating MPC significantly improves the precision of the spacecraft's navigation and control systems, thereby enhancing the mission's robustness.

In this framework, ERA is also incorporated. ERA plays a pivotal role in refining the understanding and analysis of the spacecraft's vibrational dynamics. This enhanced modeling capability allows for more effective vibration mitigation strategies, further minimizing the impact of these vibrations.

This thesis develops a method in which ERA can be used to find the equations of motion for a space structure, and then MPC can be used to control the orientation of that structure, aiming to reduce these vibrations to levels at or below pre-established constraint values. This ensures that the vibrational effects remain within acceptable limits, safeguarding the spacecraft's structural integrity and optimizing operational efficiency.

The proposed study, incorporating both MPC and ERA, extends beyond theoretical exploration. It is poised to make a substantial contribution to the understanding of

spacecraft vibration dynamics. This approach not only promises to elevate the precision and reliability of future space missions but also provides critical insights into the effective implementation of advanced predictive control strategies. In a broader sense, this study is a vital step in harnessing the full potential of space-based solar power and its myriad applications.

B. FACILITIES

To demonstrate attitude control of a flexible space structure, this study will leverage the robust infrastructure available within the Spacecraft Robotics Lab. Specifically, employing Floating Spacecraft Simulators (FSS).

The FSS, designed to recreate the dynamics of spacecraft in a controlled environment, will be outfitted with a cantilever beam. This beam, by virtue of its mechanical properties, will serve as a flexible element of the spacecraft's structure. The simulation will replicate the vibrational response of the actual spacecraft to various maneuvers, providing an analogue for the real-world conditions encountered in space. It is important to consider that the spacecraft's solar panels are complex 3D structures, and their vibrations resemble those of a planar membrane within a frame. This characterization is crucial for accurately simulating and understanding the vibrational dynamics. However, the initial testing will focus on vibrations about one axis, as this is what the current capabilities of the Spacecraft Robotics Laboratory's (SRL) granite table allow. This approach provides a starting point for analyzing and mitigating vibrational effects, with the intention of gradually expanding the scope to encompass more complex vibrational behaviors as the research progresses.

To capture and quantify the vibrations that ensue from simulated maneuvers, we will employ a Vicon motion capture system. Known for its precision and accuracy, the Vicon system will allow for the detailed, high-resolution tracking of the cantilever beam's movements. This, in turn, will yield empirical data concerning the vibrational characteristics of the spacecraft.

Furthermore, it should be emphasized that these testing procedures not only have immediate implications for the SSPIDR mission, but they also pave the way for future

advancements in spacecraft technology. This endeavor will contribute to a body of knowledge that will underpin the design and operation of more stable, reliable, and efficient spacecraft in the years to come. This research effort represents a synergistic continuation of the ongoing academic pursuit spearheaded by esteemed professors at the Naval Postgraduate School (NPS). It is principally geared towards the application and enhancement of the SSPIDR. The broader scope of this program extends to fortifying the power supply for forward-deployed forces, a goal which holds substantial strategic importance.

C. EXPERIMENTAL SETUP

The ensuing testing phase in the Spacecraft Robotics Lab will revolve around a carefully constructed apparatus. It will include a 1 Degree of Freedom (DOF) cantilever arm that will play a crucial role in quantifying vibrations during attitude change maneuvers, thus providing tangible data for this research. The spring is a torsional spring with axis normal to the granite surface. Figure 2 shows a schematic representation of the FSS experimental setup.

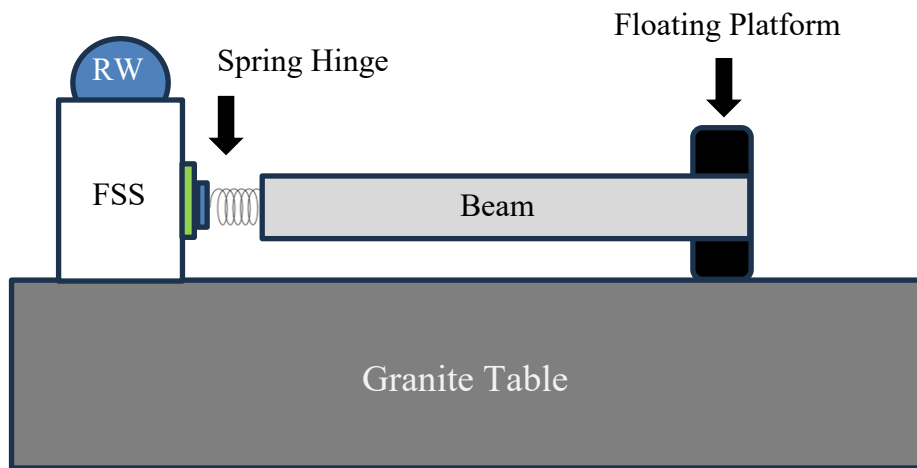


Figure 2. Schematic of the FSS setup

A reaction wheel will be utilized to apply a specific amount of torque on the Floating Spacecraft Simulator (FSS). Reaction wheels are devices used commonly in

spacecraft to control orientation without the use of thrusters, as they operate on the principle of conservation of angular momentum. This method enables the accurate simulation and control of forces exerted on a spacecraft during real-world maneuvers.

The task of measuring the vibrations of the cantilever arm during the simulated spacecraft maneuvers will be undertaken by the Vicon motion capture system. The Vicon system is known for its high precision tracking capabilities, making it an ideal tool for capturing the intricate details of the cantilever arm's vibrational response. This data will be invaluable in assessing the performance of our vibration reduction strategy.

The core of the analysis will be aimed at the first resonance frequency, which is the highest frequency that a 1 DOF system such as this will encounter [5]. The first resonance frequency often serves as a critical point of analysis, given its propensity to trigger the most substantial vibrational response in the system. By focusing attention on this frequency, effective assessment and refinement of strategies for vibration reduction can be achieved.

In essence, the experimental setup and testing strategy embody the commitment to comprehensive, real-world applicable research. The data derived from these tests will feed directly into efforts to optimize the MPC algorithms, ultimately supporting the broader goal of the SSPIDR Program. Moreover, the results obtained from this study will significantly contribute to the body of knowledge in spacecraft dynamics, establishing a foundation for future advancements in the field.

D. ERA OVERVIEW

ERA stands as a pivotal method in system identification, specializing in the analysis of dynamic systems. It examines the motion of an object, transforming it into a Linear Time Invariant (LTI) state space using both input and output data [2]. The algorithm utilizes a Hankel matrix to accurately represent a linear system, employing data that is free from external 'noise' or interference to ensure precision and reliability. This approach facilitates the development of mathematical models that translate the intricate dynamics of physical systems into coherent input, output, and state variables represented by first-order differential equations. The inception of this state-space realization technique is attributed to the groundbreaking work of Ho and Kalman, whose contributions were instrumental in

advancing the theory of minimum time realization. Their foundational framework has undergone extensive refinements and enhancements by Juang and Pappa [6], evolving ERA into a robust and reliable system that is integral to numerous contemporary applications. The evolved ERA method is now a cornerstone in various fields, enabling practitioners to delve deeper into the understanding and modeling of complex systems with unparalleled accuracy and insight.

The Ho-Kalman system identification method is employed to estimate the realization by utilizing the Markov parameters, denoted as G where the current step depends on the previous step [7]. This method ensures the graceful degradation of the output as a function of $\|G - \hat{G}\|$. In this process, it takes a noisy estimate $\hat{G}$ of G and aims to derive optimal state-space form system matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ from $\hat{G}$. The algorithm involves the construction of the Hankel matrix $\hat{H}$ (defined when each row is shifted one time step down from the previous row) from $\hat{G}$ and employs Singular Value Decomposition (SVD) (defined when the factorization of one matrix into three matrices $A = U\Sigma V^T$) on a rank- n approximation of a submatrix of $\hat{H}$ to obtain the matrices $\hat{O}$ and $\hat{Q}$. Utilizing these matrices, the state space realization matrices $\hat{A}, \hat{B}$ and $\hat{C}$ are subsequently derived.

Juang and Pappa [6] approached the problem similarly, developing the ERA. In this system the idea is to construct a state space model utilizing the structure of the Hankel matrix which is formed from the impulse response data. They then used the time domain of the data to construct the Hankel matrices. With this new Hankel matrix, a singular value decomposition is performed to extract the system matrices $\hat{A}, \hat{B}, \hat{C}, \hat{D}$. This application is used when there is more accurate data that can be procured from the system for a more accurate estimation. While this system is very similar to the Ho-Kalman method, the difference relies on how the 2 construct their systems. The Ho-Kalman method wants to achieve a more balance realized system through the use of Markov parameters where the ERA leans heavily on constructing state space models utilizing impulse response data.

ERA will be used to find equations of motion for the FSS-spring-beam system, but to introduce the method, first an example will be presented from Systems Identification

Technologies [2], which describes how ERA is used to find the equations of motion of motion of a bridge structure with distributed sensors.

The assumption is that the dynamic system can be described by the motion in discrete time

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$

$$y(k) = Cx(k) + Du(k), \quad (2)$$

where $x(k)$ is the system state, $u(k)$ is the input, and $y(k)$ is the output at time step k . To describe the system dynamics, the objective is to estimate the values of the matrices A , B , C , D .

Step 1. Impulse response data must be collected from the system, both input and output and measured to the specifics of the system. Figure 3 shows sensors along a bridge and the response data that is collected.

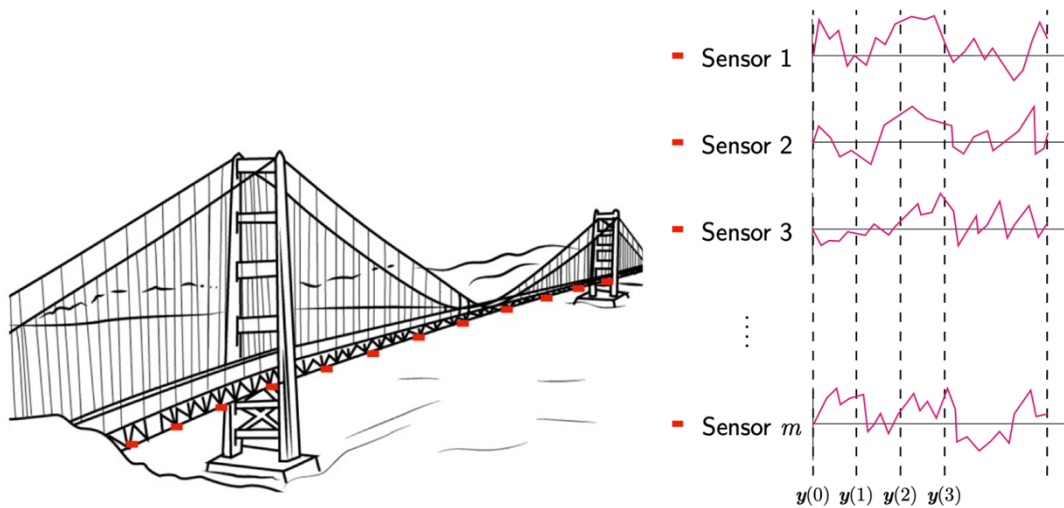


Figure 3. Bridge sensor impulse response data. Source: [2].

Step 2. With this collected data, a Hankel matrix and a shifted Hankel matrix must be formulated. Equation 3 is the shifted Hankel matrix ($k = 0$) is constructed from the sensor data collected:

$$H(0) = \begin{bmatrix} y(1) & y(2) & \cdots & y(q) \\ y(2) & y(3) & \cdots & y(q+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(p) & y(p+1) & \cdots & y(p+q-1) \end{bmatrix}. \quad (3)$$

When a pulse input is applied to the system, the values of $u(k)$ at time steps $k=1$ through $k=p$ are shown in Table 1.

Table 1. Initial values k and u .

k	u	y
0	1	D
1	0	CB
2	0	CAB
$\vdots$	$\vdots$	$\vdots$
p	0	$CA^{p-1}B$

Utilizing the discrete state space equations, and assuming $x(0) = 0$, the state $x(k)$ can be calculated and subsequently applied in the shifted Hankel matrix to determine the system's output. This process subsequently results in

$$\begin{bmatrix} x(0) = 0 \\ x(1) = Ax(0) + Bu(0) = B \\ x(2) = Ax(1) + Bu(1) = AB \\ x(3) = Ax(2) + Bu(2) = A^2B \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} y(0) = D \\ y(1) = Cx(0) + Du(0) = CB \\ y(2) = Cx(1) + Du(1) = CAB \\ y(3) = Cx(2) + Du(2) = CA^2B \end{bmatrix}, \quad (5)$$

creating the matrix

$$H(0) = \begin{bmatrix} CB & CAB & \dots & CA^{q-1}B \\ CAB & CA^2B & \dots & CA^qB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^pB & \dots & CA^{p+q-2}B \end{bmatrix}. \quad (6)$$

Step 3. Obtain the controllability (O_p) and observability (R_q) vectors by decomposing $H(0)$ as seen in Equation 7.

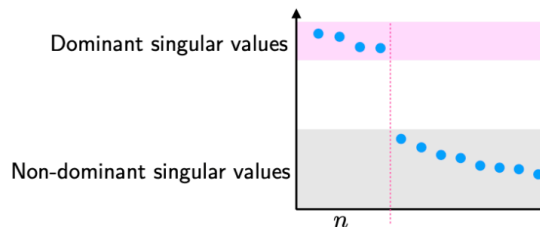
$$H(0) = \begin{bmatrix} C \\ CA \\ \vdots \\ \underbrace{CA^{p-1}}_{O_p} \end{bmatrix} \underbrace{\begin{bmatrix} B & AB & \dots & A^{q-1}B \end{bmatrix}}_{R_q} \quad (7)$$

Step 4. Next perform a singular value decomposition of the Hankel matrix, denoted as H , in accordance with equation 8. This decomposition takes $H = U\Sigma V^T$, where both U and V are orthonormal matrices, and Σ is a diagonal matrix that encapsulates the singular values. More explicitly, this can be represented as

$$H_0^{(p,q)} = U\Sigma V^T = \begin{bmatrix} U^{(n)} & U^{(0)} \end{bmatrix} \begin{bmatrix} \Sigma^{(n)} & 0 \\ 0 & \Sigma^{(0)} \end{bmatrix} \begin{bmatrix} V^{(n)T} \\ V^{(0)T} \end{bmatrix} \simeq U^{(n)}\Sigma^{(n)}V^{(n)T}. \quad (8)$$

In this formulation, the Hankel matrix $H_0^{(p,q)}$ is approximated by the product of the truncated matrices $U^{(n)}$, $\Sigma^{(n)}$, and $V^{(n)T}$, where the superscript (n) indicates the truncated version of each matrix.

Step 5. Truncate the matrices. This is to take the significant singular values (typically based on the number of states in a system in this case (n)) and truncate them to the n -order. This can be seen visually in Figure 4.



$$\hat{D} = h_0 \quad (14)$$

Using $k = 1$ to form $H(1)$ (shifting Equation 3 by one) allows the initial systems dynamics to accurately estimate the state matrix $\hat{A}$ by capturing the early stage responses for defining the system's evolution over time.

The ERA is a versatile and widely adopted method, finding applications across various industries, including chemical and aerospace engineering. Its prominence in diverse fields is due to its ability to directly incorporate system responses, ensuring high accuracy and reliability in modeling dynamic behaviors. ERA's focus on time-domain data offers unparalleled adaptability, facilitating its application to a wide range of systems with different complexities and characteristics. This adaptability is essential for meeting the unique challenges and requirements of various engineering disciplines.

In the context of this thesis on vibration reduction algorithms for large flexible space structures, ERA's role becomes even more significant. By using ERA as a system identifier, it can effectively transform complex vibrational data from these structures into a coherent state-space model. This model will be used for the application of MPC, which is designed to manage and reduce vibrations in these large, flexible structures. The precise state-space representation provided by ERA enables the MPC to accurately predict and counteract unwanted vibrations, optimizing the structure's performance and stability in space. ERA allows for a deeper understanding of the vibrational dynamics of large space structures, leading to more effective control strategies. The integration of ERA with MPC in this thesis exemplifies the algorithm's utility in developing robust and efficient solutions, optimizing system performance, and mitigating risks in sophisticated engineering applications.

E. MPC OVERVIEW

At the heart of this thesis lies the development and application of robust model predictive control (MPC) algorithms. It is incumbent upon this to establish these algorithms effectively to facilitate meaningful evaluation and testing.

MPC has established itself as a leader in the field due to its exceptional proficiency in managing constraints and executing multivariable controls with high precision. This capability has led to its widespread adoption, greatly benefiting from the significant strides made in computational power. As processors have become more powerful, MPC has found new applications in various sectors, notably in aerospace and astronautical engineering, where it addresses intricate control challenges [8].

Originally developed for the petrochemical industry, where it managed processes spanning several minutes [9], MPC's effectiveness in handling multi-variable systems and integrating constraints was evident. Today, technological advancements have significantly boosted MPC's processing speeds, enabling its application in more dynamic and challenging environments. A prime example is its use in reducing spacecraft vibrations, the central aspect of this research. This evolution highlights MPC's growing versatility and its potential in predictive control, especially in fields like space exploration.

MPC, also known as receding horizon control, stands out for its real-time optimization capabilities within the intervals of time sampling [10]. This sophisticated approach to control involves establishing an optimal control problem at every time interval, resolving it over a set finite horizon, and then applying just the initial part of the optimal control sequence derived. With the acquisition of new data, the horizon is adjusted forward—this advancing process is why it's termed 'receding horizon' control.

In executing this controller, the MPC perpetually resolves an optimization problem at predefined time intervals, determining the optimal control or trajectory for the system to follow. This continual refinement and adjustment process allows for precise and responsive control actions, adapting to the evolving states and conditions of the system. Moreover, MPC is adept at managing the intricate interactions between the inputs and outputs of a system. This proficiency in coordinating multiple variables concurrently makes it an exemplary candidate for Multi Input Multi Output (MIMO) systems, where the complexity of interactions requires advanced control strategies for effective management and operation. The versatility and precision of MPC in handling diverse and complex systems underscore its significance in advancing control engineering solutions [11]. References

[12] and [13] demonstrated that MPC can be used to control spacecraft relative motion and rendezvous and docking maneuvers.

Figure 5 shows an example of a simple MPC controller that references the dynamical model and receives feedback from the plant outputs to be circulated back into the controller, allowing MPC to optimize control input. This sequence computes control actions to minimize costs over the horizon by solving a constrained optimization problem based on an internal plant model and current system state. The controller applies only the first control action to the plant, ignoring subsequent ones, and repeats the process in the next time step [11]. By minimizing the cost function of the system, MPC creates an output which is closer to optimal.

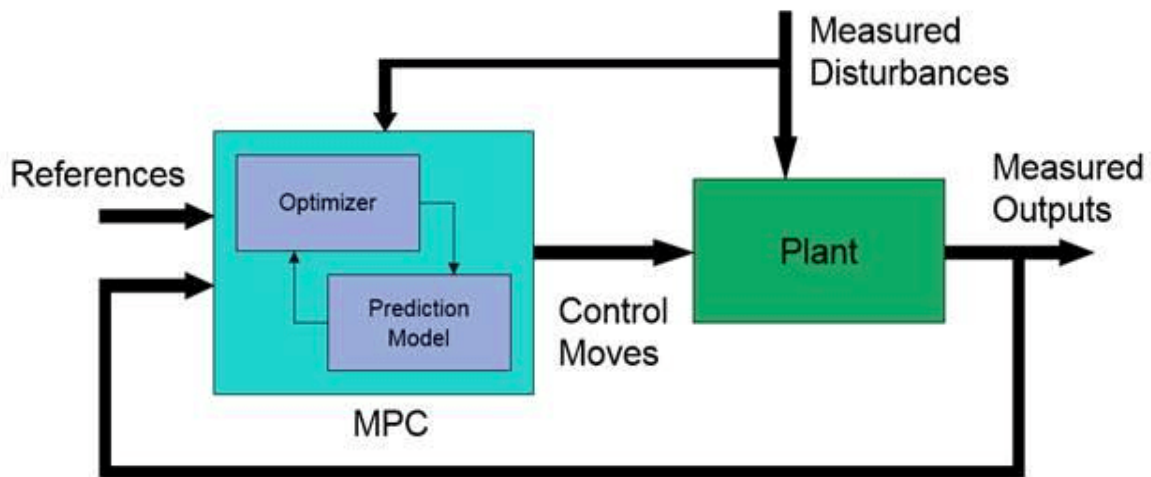


Figure 5. Feedback Control Diagram of MPC. Source: [11].

In this depiction, a plant model represents the system and produces an output, denoted as y . The measured output y is subsequently fed back into the MPC controller. This feedback mechanism enables the controller to recurrently solve the optimization problem at predetermined time intervals. Within each of these intervals, the controller selects the optimal control action, u , based on the defined constraints and the current state of the system. MPC is solving the optimization problem from the current time step to the end of the time horizon, but it only applies the control over the next time step, and then repeats

the process allowing the MPC to dynamically adjust to variations in the system and external disturbances, thereby maintaining optimal control over the system.

F. ERA/MPC UTILIZATION PROBLEM

As a simple model to introduce the combined applications of ERA and MPC, consider the Spring-Mass-Damper problem. This combination is essential for gaining insights into how these two algorithms can collaboratively function in a given problem scenario. A simplistic Spring-Mass-Damper model was formulated, drawing examples from the works of Chen and Negrut [14] from the University of Wisconsin-Madison. This model serves as a foundational reference to explore the synergistic operation of ERA and MPC in addressing the complexities associated with flexible structures like beams in the SSPIDR project.

Figure 6 illustrates the model employed to design a code capable of solving the spring-mass-damper problem using both ERA and MPC.

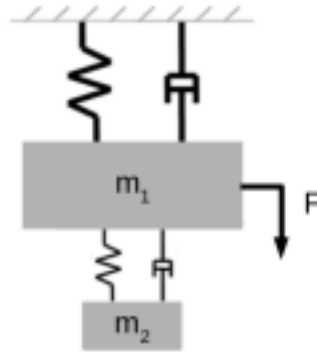


Figure 6. Spring-Mass-Damper problem model. Source: [14].

This model represents a 4 Degrees of Freedom (DOF) system, where force F is applied in a single direction and the system is subject to gravity. It diverges from the model presented in the referenced paper, as it omits the pendulum system included in the original, focusing solely on providing a clear understanding of ERA and MPC. The dynamic equations governing this system are

$$m_1\ddot{x}_1 = F(t) + m_1g - K_1x_1 - C_1\dot{x}_1 + K_2(x_2 - x_1) + C_2(\dot{x}_2 - \dot{x}_1) \quad (15)$$

$$m_2\ddot{x}_2 = m_2g - K_2(x_2 - x_1) - C_2(\dot{x}_2 - \dot{x}_1). \quad (16)$$

These are derived directly from the paper to ensure accurate simulation and analysis.

The variables included in the equation are K_1 and K_2 , representing the stiffness constants; C_1 and C_2 , which are the damping coefficients for m_1 and m_2 respectively; and F , denoting the total external force applied to the system. Table 2 presents the parameters used for the simulation.

Table 2. Parameter descriptions and values used for the simulation.
Source [14].

Symbol	Parameter Description	Value
m_1	Mass of m_1	1 (kg)
m_2	Mass of m_2	0.01 (kg)
K_1	Spring stiffness for m_1	20 (N/m)
K_2	Spring stiffness for m_2	1 (N/m)
C_1	Damping for m_1	0.02 (Ns/m)
C_2	Damping for m_2	0.02 (Ns/m)
g	Gravity Acceleration	9.8 (m/s ²)

Appendix A displays the problem executed in MATLAB, beginning with the creation of a function utilizing the dynamical equations. Within this function, the variables, input force, and dynamical equations are formulated into ordinary differential equations (lines 90–115). Subsequently, the initial conditions are applied as $x_0 = [1 \ 1 \ 0 \ 0]$.

The simulation, set with a 0.1s time step over 10 s, displays the object’s trajectory without a controller, representing the “truth” system or the actual physical system. This baseline is established by solving ordinary differential equations with MATLAB’s ode45 solver, and the results are shown in Figure 7. In the ERA process, this “truth” system data

is used for deriving equations of motion using only the measured output, this ensures the accuracy of the system's dynamic modeling.

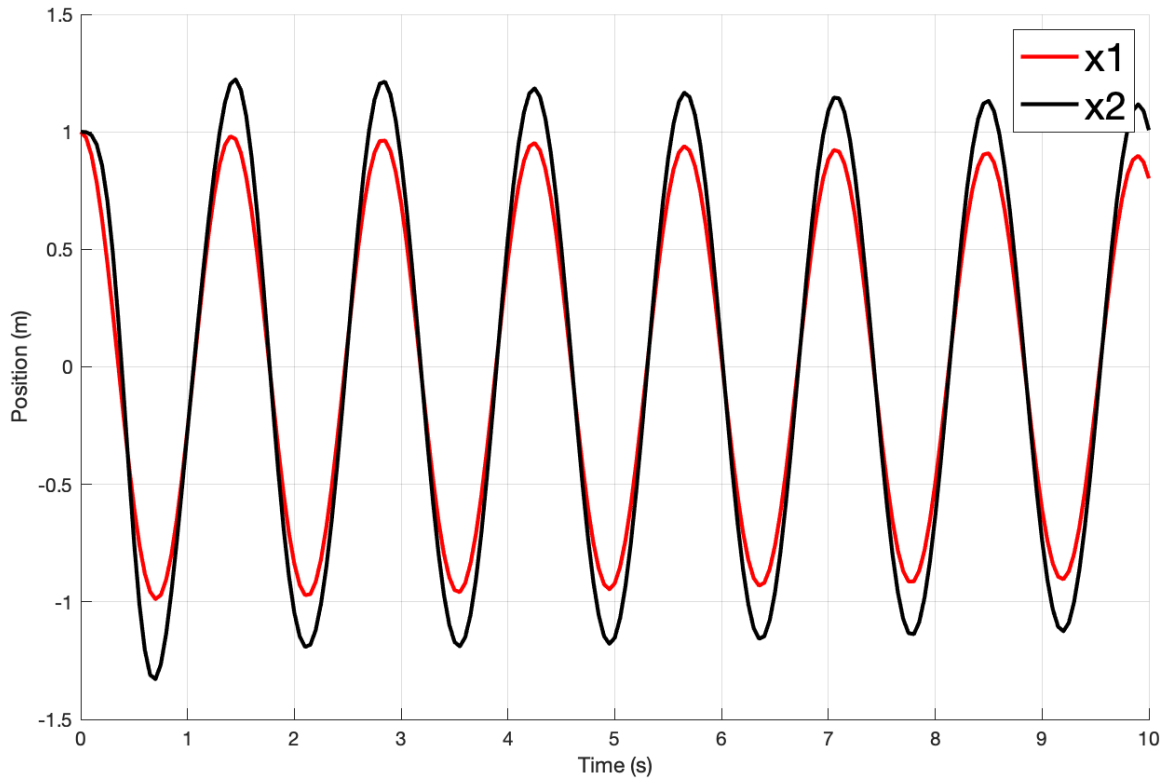


Figure 7. The vibrations of the two masses in the system based of the dynamical equations

After executing the system, vibrations were observed, affecting the two masses, and causing variations in the positions of x_1 and x_2 . Beginning at line 35, the system underwent computation through the (ERA). Given the presence of four states within the system, four singular values were selected. This choice is pivotal as it enables the system to filter out the inherent noise, ensuring the accuracy and reliability of the results.

Subsequently, a Hankel matrix was constructed in line 38 of the code, followed by the formation of a shifted Hankel matrix in line 39. These matrices are crucial for analyzing the system dynamics and extracting essential information about its behavior. The subsequent step involved performing the Singular Value Decomposition (SVD) of the Hankel matrix, as depicted in line 40. This decomposition is integral to the ERA process,

allowing for the extraction of system modes and the realization of the state-space representation. This approach, involving the creation and manipulation of matrices and the careful selection of singular values, is instrumental in understanding the dynamics of the system and in accurately capturing the vibrations and variations in the positions of the masses. Figure 8 shows the system using the new estimated state space formed using ERA overlayed with the true model from Figure 7.

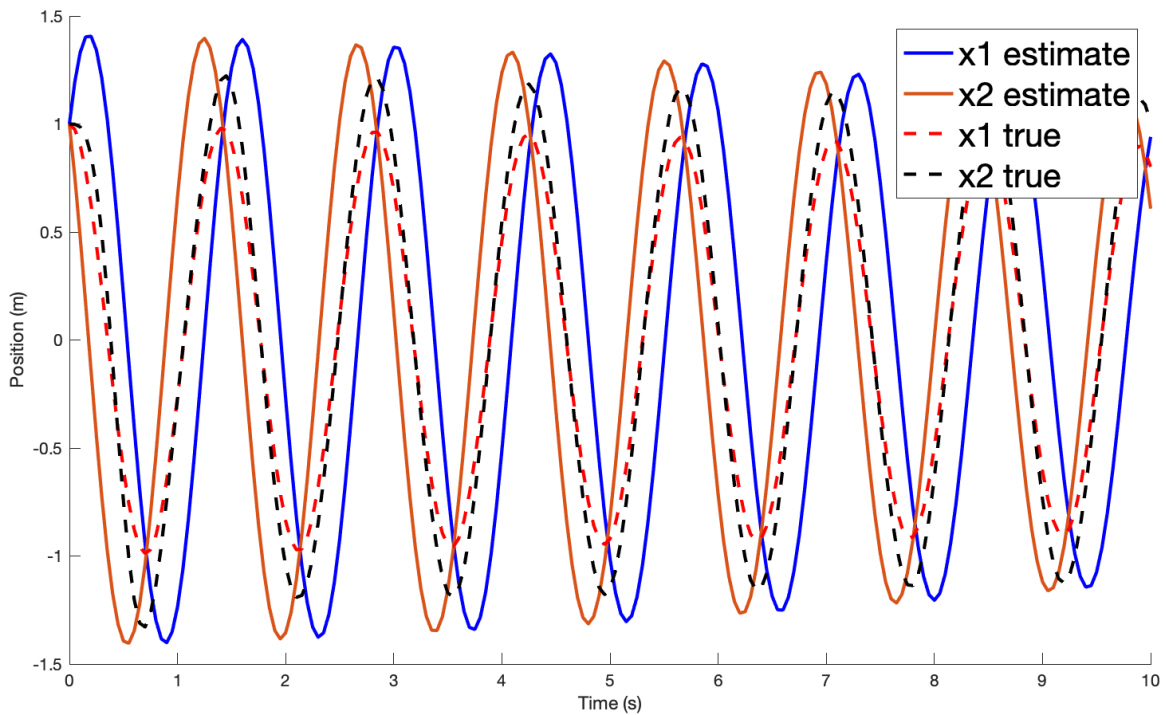


Figure 8. Estimated system overlayed with the true system

The subsequent step involved formulating the estimated state spaces of the equations. Lines 44–47 of the code delineate the computations for the estimated state spaces, with the Equations 11–14.

It is noteworthy that in line 47, $\hat{D}$ is populated with all zeros, indicative of the absence of direct feedback between the input and the output of this system. The result was a continuous state space estimation of the systems dynamics where each state space representation was a 4x4 matrix.

To implement the MPC, it is essential to discretize the system's state-space representation and convert it into a form that is compatible with the controller. This process is crucial for ensuring that the controller can effectively manage the system's dynamics and optimize its performance. In lines 51–52 of the MATLAB code, the system is formulated into its state-space representation using the `'ss'` function in MATLAB and is assigned the name `'sys'`. Initially, this representation is in continuous time. To adapt it for the controller, it is discretized to discrete time using the `'c2d'` MATLAB function. This discretization facilitates the application of digital control strategies, which are more adaptable to real-world scenarios.

Once the system is properly set up, it is necessary to input the specific MPC signals. These signals are fundamental as they dictate how the MPC will interact with the system to compute and apply control actions accurately. The signals for the MPC are categorized as follows:

- MV (Manipulated Variables)

These are the variables that the controller can adjust to drive the system's behavior.

- MD (Measured Disturbances)

These are the external disturbances that are measured and compensated for by the controller.

- UD (Unmeasured Disturbances)

These are the disturbances that are not measured but can affect the system's output.

- MO (Measured Outputs)

These are the outputs of the system that are measured and used by the controller to adjust the manipulated variables.

- UO (Unmeasured Outputs)

These are the outputs of the system that are not measured or used by the controller.

In the simulation, the MPC signals are configured with four Manipulated Variables (MV) and four Measured Outputs, corresponding to the four states within the system. Line

71 of Appendix B instantiates the MPC object, incorporating a timestep of 0.5 seconds. The following provides a visual representation of the created MPC object through a screenshot.

MPC object (created on 27-Oct-2023 13:21:44):

```
-----
Sampling time:      0.5 (seconds)
Prediction Horizon: 10
Control Horizon:    2
```

Plant Model:

```
-----
1 manipulated variable(s) -->| 4 states |
                                |         | --> 4 measured output(s)
0 measured disturbance(s) -->| 1 inputs  |
                                |         | --> 0 unmeasured output(s)
0 unmeasured disturbance(s) -->| 4 outputs |
-----
```

Weights are applied to various elements within the control framework,

```
Weights:
    ManipulatedVariables: 100
    ManipulatedVariablesRate: 1
    OutputVariables: [10 10 10 10]
    ECR: 10000000
```

namely the Manipulated Variables (MV), the rates of change of MVs (MV rates), and the Output Variables. Additionally, the Extended Control Horizon Rate Penalty (ECR) is also subjected to a weighting scheme. While these weights can be selectively assigned based on specific control objectives or system requirements, in the observed instance, they were not explicitly chosen but were instead adopted as the system's inherent weight penalties.

After establishing the MPC object and applying the relevant weights, the subsequent phase involves simulating the system. In this instance, the number of control intervals is designated as 20, and a reference trajectory of $[1 \ 1 \ 0 \ 0]$ is specified corresponding with the position of mass 1, position of mass 2, velocity of mass 1, and velocity of mass 2 respectively. This trajectory indicates the target state to which the controller aims to steer the

system. Specifically, the controller is tasked with guiding the x_1 and x_2 positions to a value of 1, while ensuring that the velocities $\dot{x}_1$ and $\dot{x}_2$ converge to 0.

Figure 9 simulates the controlled trajectory of the ERA state space system utilizing the MPC. This provides a comprehensive visual insight into its dynamic evolution over time. When the model predictive controller is introduced into the system, it guides the states towards the predefined reference trajectory, ensuring a precise and controlled transition through the state-space. It is visible that the states trajectories stabilize in various locations, however noting the scale of the graph these are acceptable levels of 10^{-3} .

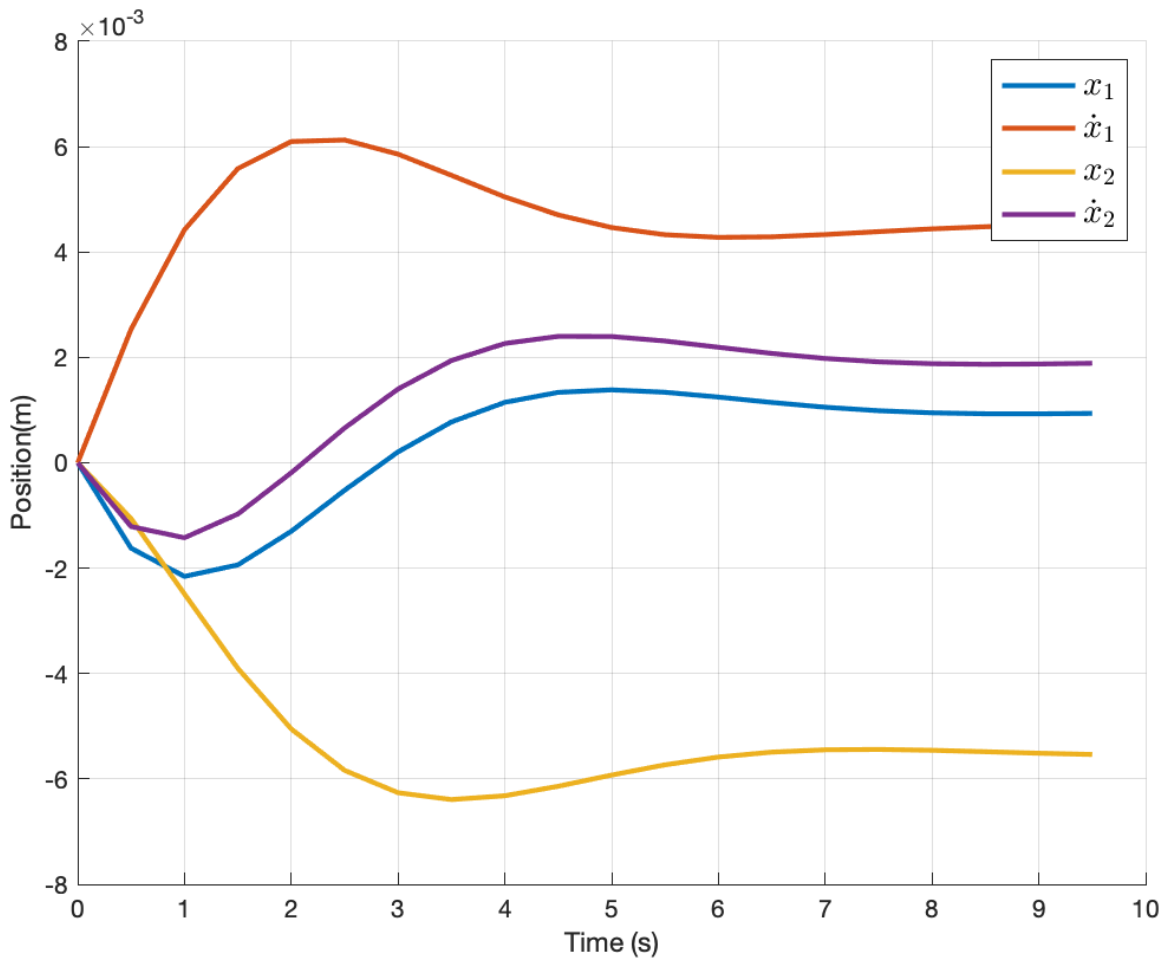


Figure 9. The state trajectories using MPC.

In contrast, Figure 10 provides a depiction of the step response of the states, showing a perspective of the system's response to sudden changes in input. The ability of the controller is displayed here, as it successfully navigates the system through the state-space with proficiency. Furthermore, the system exhibits minimal percentage overshoot, highlighting the controller's ability to mitigate excessive system deviations beyond the desired setpoints. Ultimately, the states level at the predetermined points, highlighting the controller's adept capability in ensuring system stability and adhere to the defined reference trajectory.

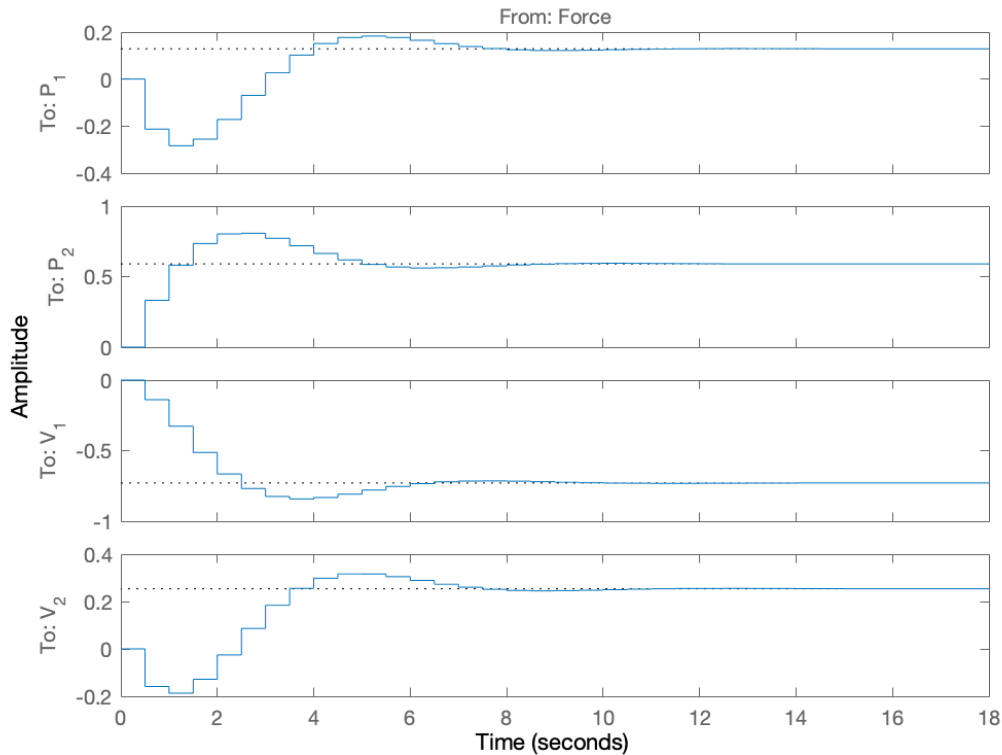


Figure 10. The step response of the individual states of the system

This contrast of the system trajectory and step response not only validates the effectiveness of the MPC controller but also underscores its pivotal role in maintaining system stability and tracking accuracy amidst dynamic changes and potential disturbances.

The spring-mass-damper example serves as a robust model, revealing the synergistic application of the ERA and MPC when used in tandem. Moreover, it demonstrates the capabilities of the MPC controller, particularly within a vibrating mechanical system. This exploration facilitated a deeper comprehension of both ERA and MPC by employing a straightforward system as a precursor to tackling more intricate challenges. The spring-mass-damper, with its inherent simplicity, amplifies the understanding of pivotal coding and lays a foundational understanding of the mathematical complexities embedded within the ERA-MPC system.

III. METHODS

The solar power satellite (SPS) satellite system is designed to optimize solar energy capture. The expansive nature of the solar arrays introduces susceptibility to mechanical vibrations, particularly when torque is applied for spacecraft maneuvers. These vibrations, if not adequately controlled, can lead to cumulative structural degradation. To address this, the application of MPC is proposed to dampen the vibrational response. By leveraging MPC's predictive capabilities and real-time optimization, the control strategy aims to minimize the vibrational impact of operational torques, thereby enhancing the structural integrity and power beaming capability of the satellite system. The challenge is to tailor the MPC algorithm to accommodate the nonlinear dynamic responses of the spacecraft-beam-array assembly, ensuring precise attitude control and system stability throughout its mission life cycle.

A. PROBLEM STATEMENT

The SPS satellite encompasses a spacecraft with a lengthy solar panel array of photovoltaic cells, strategically engineered to harness solar energy. Given the significant size requirements imposed on the solar cells, the system is predisposed to experiencing vibrations when torque is applied to maneuver the spacecraft for various operational necessities. This applied torque, over an extended period, harbors the potential to degrade the system and, as such, necessitates damping. The implementation of a controller, such as MPC, is anticipated to mitigate the system's vibrations effectively. This, in turn, could potentially enhance the spacecraft's longevity, as well as its ability to reliably point the power beam to the ground station. The dynamics of the system discussed are from notes by Dr. Marcello Romano's observation of the system.

The complex multi-body system of a SBSP spacecraft is simplified, as a first-order model to a system composed of two rigid bodies, interconnected through a specialized revolute joint. This system, embodying a spacecraft and a flexible solar structure (referred to as the beam), operates under a specific assumption termed as "free floating." Under this

assumption, it is envisaged that the system will be influenced only by its internal torque and a torsional spring force at the revolute joint that connects the two bodies.

The spacecraft and the beam are restricted to planar motion with the connecting links between the bodies maintaining a perpendicular orientation within the plane of motion.

The system's movement is not merely linear, but is characterized by a series of complex mechanical interactions between the interconnected bodies. The control torque is applied to the system with a reaction wheel on the spacecraft body. Figure 11 shows a Schematic representation of a planar, free-floating system comprising two rigid bodies interconnected by a single revolute joint, situated in a flat space. The designated plane of motion is congruent with the plane of the schematic illustration.

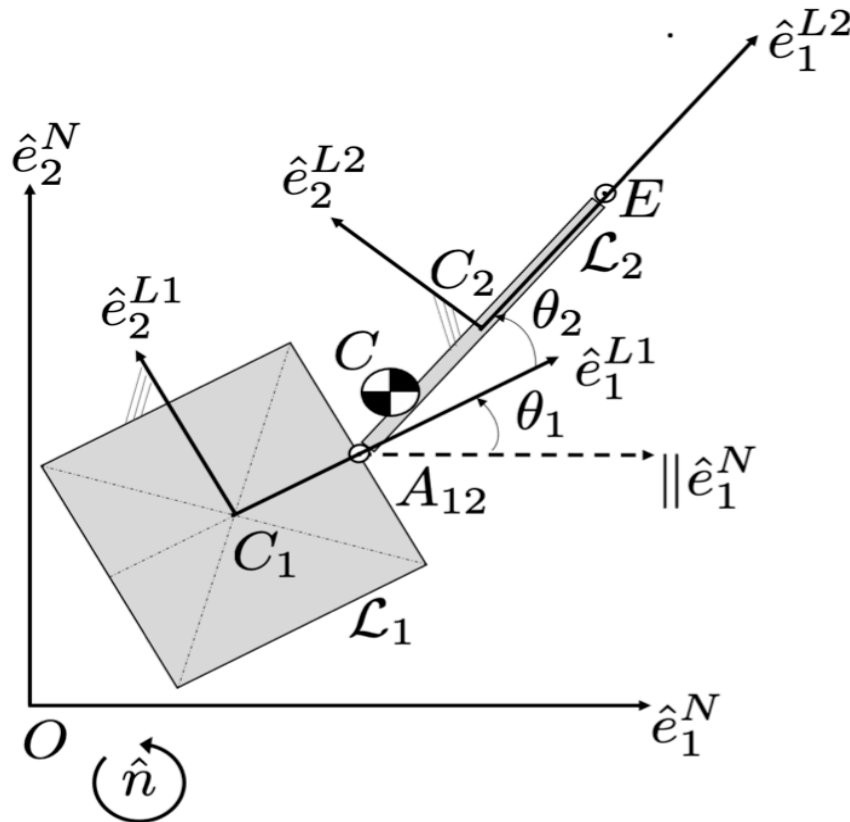


Figure 11. Schematic illustration of the spacecraft and beam illustrated by Dr. Romano

The schematic provides a detailed representation of a rigid body system, wherein the spacecraft and the beam are denoted by $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. In this illustrative example, both bodies are homogeneous, implying that their Centers of Mass (COM) are strategically positioned at the geometric centers of their respective masses, labeled as C_1 for the spacecraft and C_2 for the beam.

By analyzing the COM of both the beam and the spacecraft, the overall system's COM can be discerned and is symbolically represented by C . The unit vector normal to the plane of motion, denoted as $\hat{n}$, is parallel to the revolute joint. The point A_{12} signifies the joint intersection between the two bodies and is commonly referred to as the 'articulation point'.

Moreover, point E is designated as the 'end effector' of the manipulator, situated at the extremity of $\mathcal{L}_2$. The inertial frame N is fixed and serves as a foundational reference, from which a Cartesian coordinate system is affixed, with $O \in N$ and a vector base $\hat{e}^N \in N$ facilitating precise navigation and control of the system within its operational environment.

Explicit expression of the equations of motion for the above system can be expressed by

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} = G. \quad (17)$$

In Equation 14, the symbol (q) represents the state vectors, with $\dot{q}$ and $\ddot{q}$ denoting the first and second-order derivatives, respectively. Here, C is synonymous with the COM, H corresponds to the mass matrix, and G is the conservative force matrix, each playing a pivotal role in the mathematical representation and physical interpretation of the system dynamics. Figure 12 represents the flow of the system's feedback controller. Here the plant is the FSS and the beam, which are represented by the nonlinear equations of motion, the control is the reaction wheel torque, and the prediction model is a linearization of the equations of motion.

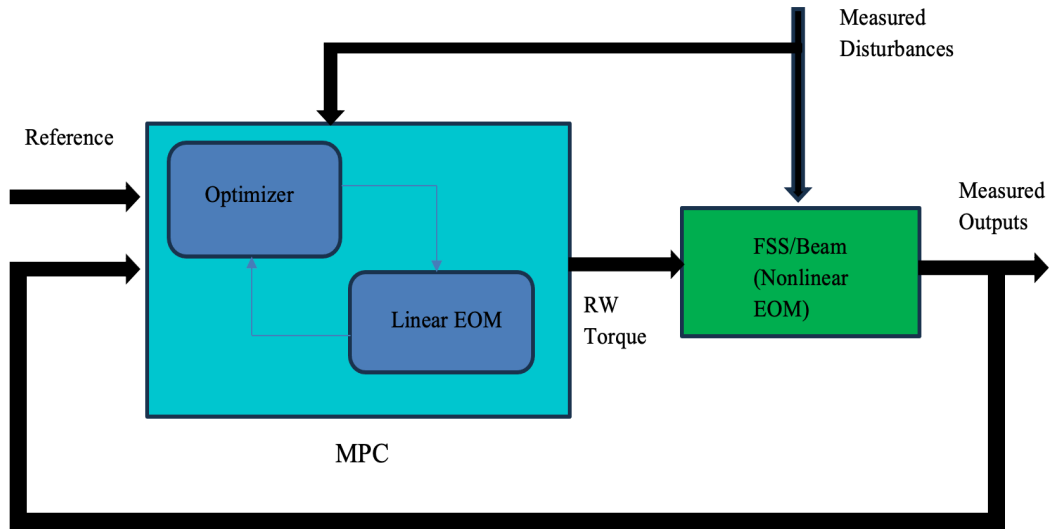


Figure 12. Feedback loop of the simulated system

The problem statement for the SPS satellite system centers on the development of an advanced control mechanism capable of mitigating the inherent vibrational dynamics induced by operational torques in a free-floating, rigid multi-body configuration. The system, characterized by its interconnected spacecraft and flexible beam assembly, must perform attitude change maneuvers while maintaining vibrations within operating constraints. The challenge is to devise an MPC strategy that can accurately model the complex interactions encapsulated by Equation 17, accounting for the nonlinearities introduced by the revolute joint and the system's response to control inputs. The MPC must effectively manage the system's dynamics, ensuring precise manipulation of the spacecraft and beam. The ultimate goal is to enhance the satellite's functional stability, enabling efficient solar energy harnessing while navigating the exigencies of space operations.

B. LINEAR CONTROL USING MPC

The system underwent its initial testing phase employing linearized equations, serving as a simplified representation of the system's dynamics. Linearized models, while providing a foundational understanding, often necessitate a comparative analysis with their non-linear counterparts to ensure comprehensive insight into system behavior. T

The linear equations are articulated in the state-space form

$$\ddot{\theta} = A(\theta) + B(u) \quad (18)$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & C_1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -C_3^2 & 0 \end{bmatrix} \quad (19)$$

$$B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ b_4 \end{bmatrix}. \quad (20)$$

The state vector is

$$\theta(t) = \begin{bmatrix} \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{bmatrix}. \quad (21)$$

θ_1 and $\dot{\theta}_1$ represent the spacecraft position and rotational velocity, and θ_2 and $\dot{\theta}_2$ represent the beam's position and rotational velocity. For the small-scale experimental on the POSEIDYN granite table, the given constants are provided in Table 3.

Table 3. Values of the established parameters

Established Parameters	Values
C_1	0.822
C_3^2	1.66
b_2	6.24
b_4	-7.48

From the above information, the simulation ran in MATLAB (code in Appendix C) the initial and target conditions are

$$\theta_0(t) = [\pi \quad 0 \quad 0 \quad 0] \quad (21)$$

$$\theta_f(t) = [0 \quad 0 \quad 0 \quad 0]. \quad (22)$$

Utilizing MATLAB, the linearized equations were input as delineated above. The full state output is elucidated in lines 24 and 25 of Appendix B. The matrix C is selected as an identity matrix, signifying that all the states are directly measurable and thereby represented as outputs. Concurrently, matrix D is constituted as a vector of zeros, underscoring the absence of a direct pathway from the input to the output within this particular system.

Upon establishing the state-space representation for the system, a pivotal step involved discretizing the continuous-time model, which was accomplished using MATLAB's `'c2d'` function. The `'setmpcsignals'` function was employed, configured with one Manipulated Variable (MV) and two Measured Outputs (MO). The simulation is executed for a total of 100 control intervals, with each step representing a time increment of 0.5 seconds. The simulation spans a duration of 50 seconds, providing a timeframe to observe and analyze the system's dynamics as seen in Figure 13.

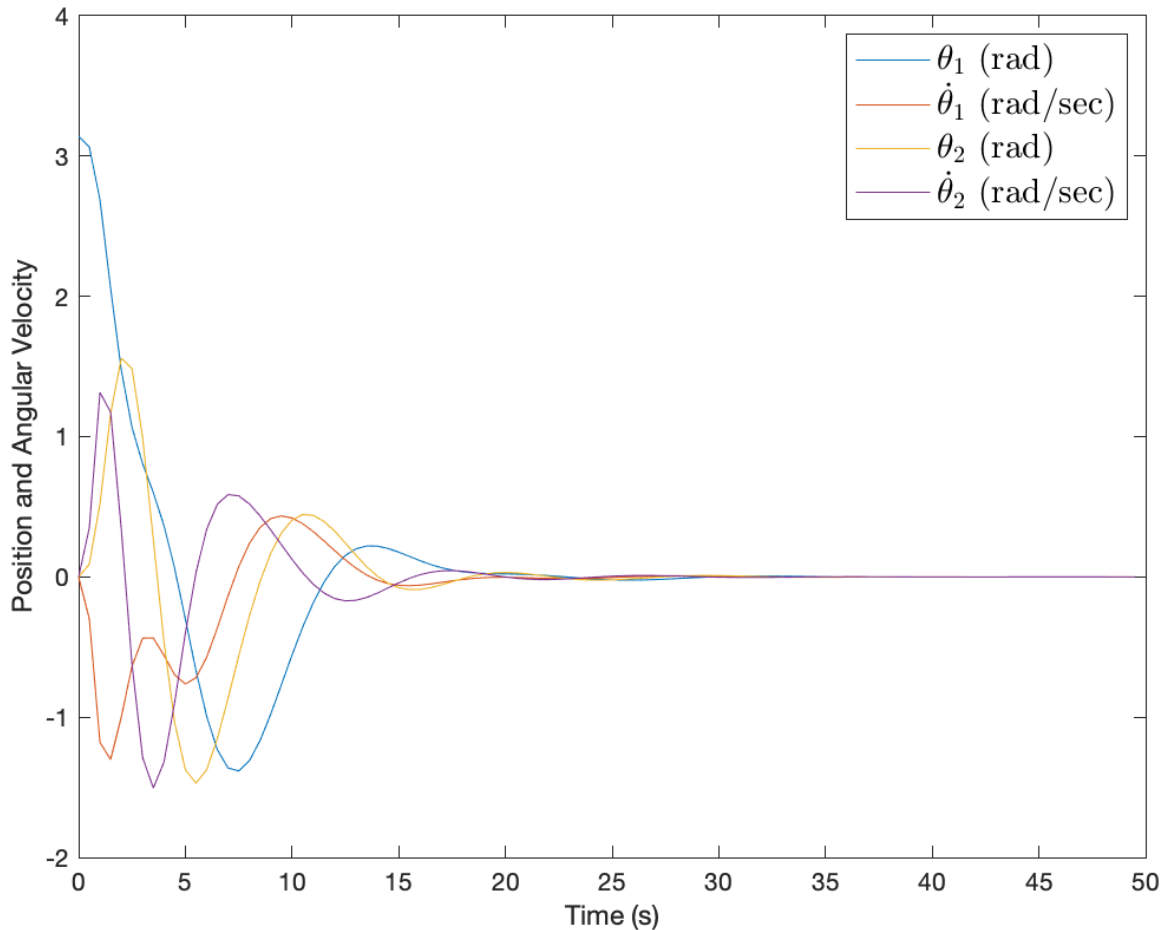


Figure 13. Model predictive controller (MPC) of the linearized dynamic equations

Linear quadratic regulator (LQR) was also used to compare the results. LQR is a control strategy that minimizes a quadratic cost function that penalizes the states and control efforts. LQR differs from MPC in that MPC allows for a more flexible definition of the cost function and can handle constraints, while LQR traditionally has a quadratic cost function without explicit constraint handling.

Leveraging the identical linear equations utilized in the MPC and maintaining a consistent time step of 0.5 seconds across 100 control intervals, the Linear Quadratic Regulator (LQR) demonstrated notably promising results. The LQR managed to guide the system to its final position with a swifter convergence compared to its MPC counterpart when applied to the linearized equations. This expedited attainment of the final position by

the LQR could potentially be attributed to the penalties encapsulated within the Q and R matrices, which represent the state and control penalties, respectively. The Q matrix, which penalizes deviations in the state variables, and the R matrix, which penalizes the use of control inputs, were tuned to balance system performance and control effort. The LQR (Figure 14) with its inherent ability to minimize the quadratic cost function, navigates the state trajectory toward the desired setpoint, potentially offering a more aggressive control action in contrast to the MPC which is seen in Figure 13.

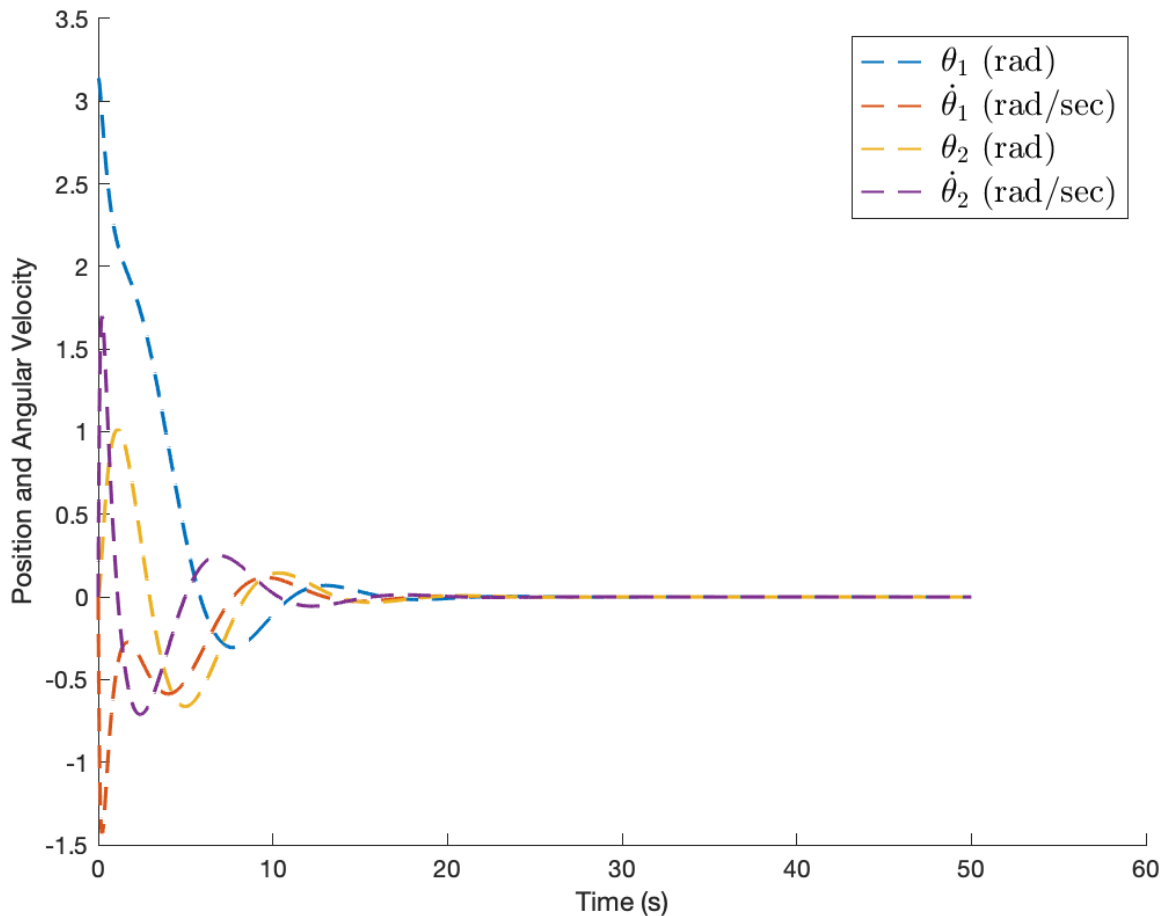


Figure 14. Linear Quadratic Controller (LQR) of the linearized dynamic equations

While LQR showed faster convergence in the linearized system comparison, it is important to consider factors like robustness, constraint handling, and real-world

applicability for an evaluation of controllers, especially given that the actual system dynamics are nonlinear.

C. NONLINEAR CONTROL USING MPC

Nonlinear control methodology has been employed as an alternative approach to derive torque profiles for subsequent testing purposes. This method accounts for the nonlinear relationships between input and output variables, allowing for a more comprehensive capture of inter-variable interactions. Such a nuanced relationship is particularly advantageous in the context of our system, given the inherent complexity involved in the design of both the spacecraft and the associated beam structure.

For the system explained in the problem statement, nonlinear equations applied to that system are as follows:

The state vector $q(t)$ is

$$q(t) = \begin{bmatrix} r \\ s \\ \theta_1 \\ \theta_2 \end{bmatrix}. \quad (23)$$

Here the r and s are the Cartesian coordinates of the systems COM, and θ_1 and θ_2 are the angular positions of the hub and beam.

The mass matrix H is Equation 24:

$$H = \begin{bmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & H_{33} & H_{34} \\ 0 & 0 & H_{34} & H_{44} \end{bmatrix}. \quad (24)$$

In this expression are

$$H_{33} = \frac{1}{4M} [m_1 m_2 (l_1^2 + l_2^2 + 2l_1 l_2 \cos \theta_2)] + (i_1 + i_2) \quad (25)$$

$$H_{34} = \frac{1}{4M} [m_1 m_2 (l_2^2 + l_1 l_2 \cos \theta_2)] + i_2 \quad (26)$$

$$H_{44} = \frac{1}{4M} [m_1 m_2 l_2^2] + i_2. \quad (27)$$

The centrifugal forces matrix is denoted by

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C_{33} & -(C_{33} + C_{43}) \\ 0 & 0 & C_{43} & 0 \end{bmatrix}. \quad (28)$$

In this expression are

$$C_{33} = -\frac{1}{4M} (m_1 m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1) \quad (29)$$

$$C_{43} = -\frac{1}{4M} (m_1 m_2 l_1 l_2 \sin \theta_2 \dot{\theta}_1). \quad (30)$$

In the case of this problem the explicit format to solve for $\ddot{q}$ is

$$\ddot{q} = H^{-1} (G + F - C\dot{q}). \quad (31)$$

G represents the torque from the torsional spring and the F is the applied torque:

$$G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -k\theta_2 \end{bmatrix} \quad (32)$$

$$F = \begin{bmatrix} 0 \\ 0 \\ T \\ 0 \end{bmatrix}. \quad (33)$$

Given that both H and C are functions that utilize $q(t)$ and $\dot{q}(t)$, it is possible to determine $\ddot{q}(t)$ rewritten as

$$\ddot{q} = \begin{bmatrix} \ddot{r} \\ \ddot{s} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = H^{-1} \left(\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -k \end{bmatrix}}_D \underbrace{\begin{bmatrix} r \\ s \\ \theta_1 \\ \theta_2 \end{bmatrix}}_G + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_F u - C \begin{bmatrix} \dot{r} \\ \dot{s} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right). \quad (34)$$

Let

$$y = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}. \quad (35)$$

Integrating all these variables results in

$$\dot{y} = \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{s} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{r} \\ \ddot{s} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0_{4 \times 4} & I_{4 \times 4} \\ H^{-1}D & -H^{-1}C \end{bmatrix} \begin{bmatrix} r \\ s \\ \theta_1 \\ \theta_2 \\ \dot{r} \\ \dot{s} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + H^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u. \quad (37)$$

Equations 23–37 provide a detailed mathematical representation of these nonlinear relationships inherent to the model. Table 4 shows the values for the FSS and beam system parameters for the experimental setup in the POSEIDYN facility.

Table 4. The FSS and beam system parameters

Symbol	Parameter Description	Value
m_1	Mass of m_1	10 (kg)
m_2	Mass of m_2	2 (kg)
l_1	Spacecraft length (FSS)	0.27 (m)
l_2	Length of beam	0.4 (m)
l_3	Beam thickness	0.005 (m)
k	Damping for m_2	0.02 (Ns/m)
M	Total mass	12 (kg)
i_1	MOI for the FSS	$\frac{1}{6}m_1l_1^2$
i_2	MOI for the beam	$\frac{1}{12}m_2(l_2^2 + l_3^2)$

The nonlinear equations, were numerically integrated in MATLAB to simulate the system dynamics. This simulation, as detailed in Appendix C, serves as the foundation for implementing the MPC for nonlinear control.

Next is the initial conditions for the state variables are in Table 5.

Table 5. Initial conditions of the state values

Established Parameters	Initial Conditions
r_0	0 (m)
s_0	0 (m)
θ_{1_0}	3.14 (rad)
θ_{2_0}	0.1 (rad)
$\dot{r}_0$	0.1 (m/s)
$\dot{s}_0$	0 (m/s)
$\dot{\theta}_{1_0}$	0 (rad/s)
$\dot{\theta}_{2_0}$	0 (rad/s)

Lines 44–46 show the configuration of the NLMPC using MATLAB’s dedicated function. This NLMPC controller is designated as ‘*nlobj*’. The state and output functions are defined by ‘*MyStateFcn*’ and ‘*MyOutputFcn*’, respectively. The prediction horizon and control horizon are 50 and 2, respectively. To fine-tune the controller’s performance, weights are assigned to various components: the controls, the outputs, and the rate of change of the controls, which are termed as manipulated variable rates. After a process of trial and error, the weights are identified in lines 59–61. The code also sets constraints on the manipulated variable, the torque, limiting its values between -0.05 and 0.05. The following steps align with the code in Appendix C, lines 82–120.

1. Matrices ‘*H*’ and ‘*C*’ are calculated based on the current state. These matrices represent system dynamics related to inertia and Coriolis/centrifugal forces.
2. A nonlinear model predictive control (NLMPC) strategy is employed to compute the control input, or torque, applied to the system. This torque is determined based on the current measurements and the previous control input.
3. The system’s behavior, given the computed torque, is then simulated over a short time span using the ‘*ode45*’ function.
4. Results from the simulation, including angles, angular velocities, and time, are stored.
5. The loop updates the initial conditions for the next iteration based on the end state from the current simulation.
6. The loop continues until the simulation period is covered.

After the loop completes, the code compiles the stored time and torque data into a table for easy reference and analysis.

The outcomes of the simulation are illustrated in Figures 15 and 16. Figure 15 presents the state trajectories for $\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2$. These trajectories are depicted in degrees or degrees per second. The simulation commences with initial conditions set at [3.14 rad, 0

rad/s, 0.1 rad, 0 rad/s] and progressively transitions to the final state of [0 rad, 0 rad/s, 0 rad, 0 rad/s]. This transition is indicative of the system's objective perform a rotation while mitigating the vibrations in the beam (represented by θ_2) over the course of time.

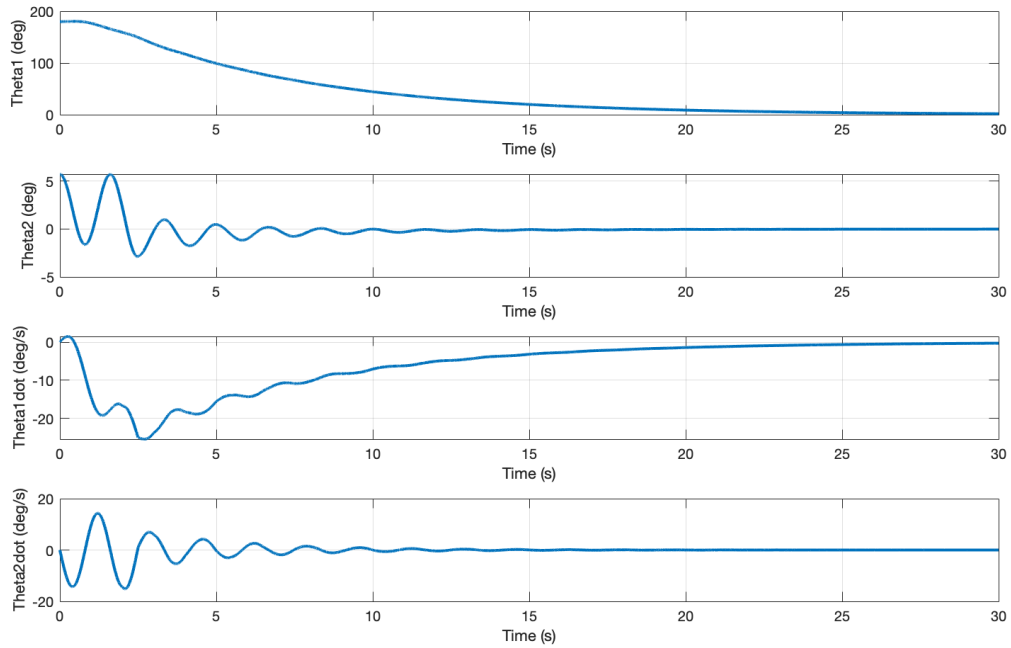


Figure 15. Representation the state trajectories over time

Figure 16 showcases the optimal torque profile applied to the spacecraft for this specific scenario. This profile, delineated in distinct time steps, offers a visual representation of the MPC in action. At each time step, the MPC updates the torque inputs, continually seeking the most efficient value. The ultimate goal of this dynamic adjustment is to maintain the beam, represented by θ_2 , in a state of minimal vibration from the initial state to the final state.

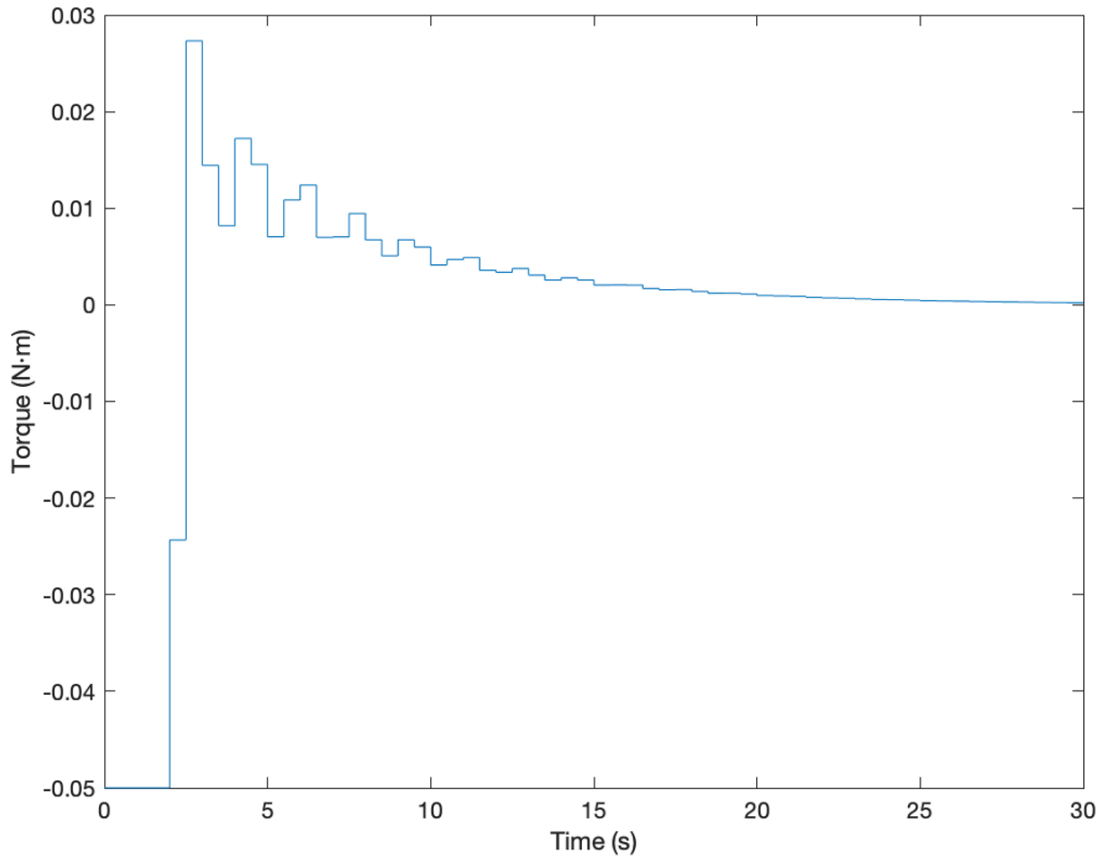


Figure 16. Torque profile

D. OPTIMAL VIBRATION REDUCTION

The primary objective of this thesis is to employ an MPC controller to minimize vibrations during slewing maneuvers. As demonstrated in the nonlinear MPC section, the controller effectively achieves this goal. However, to validate the effectiveness of the MPC controller, it's essential to compare its performance against a system without a controller. The simulation code for the uncontrolled system can be found in Appendix D. Without the controller, as anticipated, the system exhibits continuous vibrations, characteristic of such systems. The system dynamics were simulated using the 'ode45' solver in MATLAB using the same dynamics that were used in the nonlinear MPC in Chapter 3 part C.

The ODE was run with the same initial conditions for 30 seconds and the positions θ_1 and θ_2 were plotted against time as seen in Figure 17.

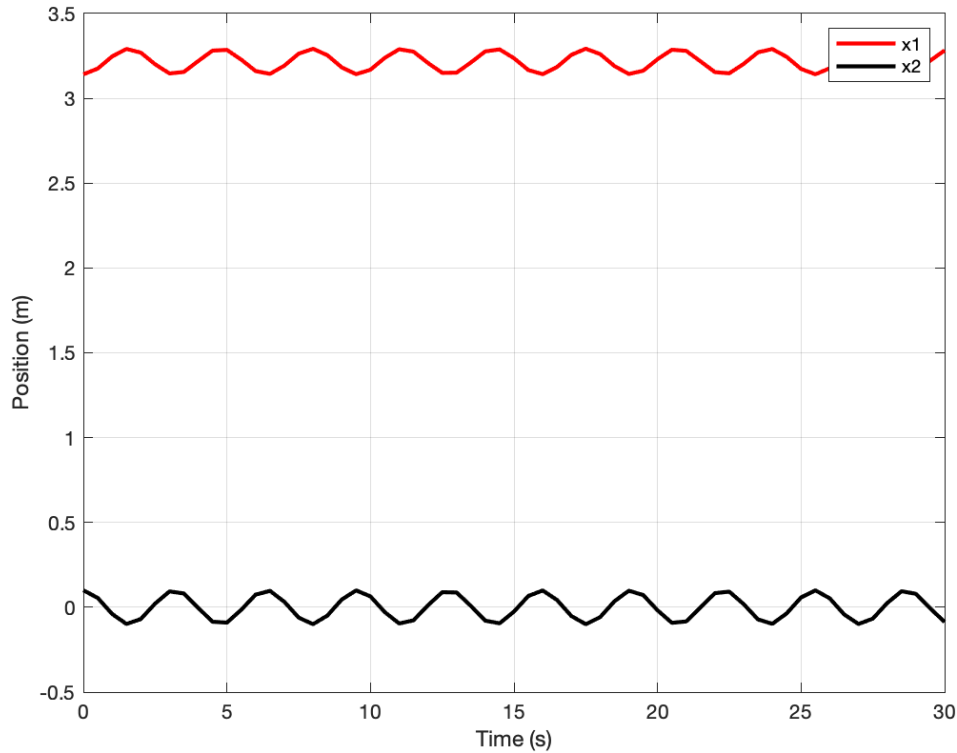


Figure 17. Vibrations of the spacecraft due to its dynamics during a maneuver

The vibrations of the system are evident from the Figure 16 which shows the requirements that are necessary to use a controller to stabilize this system. Employing the nonlinear controller outlined in part C, it has been tailored to match the system's parameters, specifically targeting the system's vibrations. By analyzing this controller, one can discern its functionality and ascertain that MPC is an effective design for reducing vibrations in this system.

The weights of this simulation played a major role and can be seen in Appendix D lines 61–63. Through trial and error, the output variables $[r, s, \theta_1, \theta_2, \dot{r}, \dot{s}, \dot{\theta}_1, \dot{\theta}_2]$ vary. ' r ' and ' s ' have weights of 100, ' θ_1 ' and ' θ_2 ' have weights of 1000, while ' $\dot{r}$ ' and ' $\dot{\theta}_1$ ' are also weighted at 1000. ' $\dot{s}$ ' has the highest weight of 10,000. ' $\dot{\theta}_2$ ' is weighted similarly to ' θ_1 ' and ' θ_2 ' at 1000.

In Figure 18, the impact of the MPC controller on the system's dynamics is illustrated. The graph captures the damping effect on the system's vibrations. Initially, the spacecraft's position was set at 3.14. However, as the graph clearly depicts, there's a

progressive decline in this value. Over time, and under the influence of the MPC controller, the position steadily approaches zero, which is its final resting position. This demonstrates the efficacy of the MPC controller in stabilizing the system and guiding the spacecraft to its desired state.

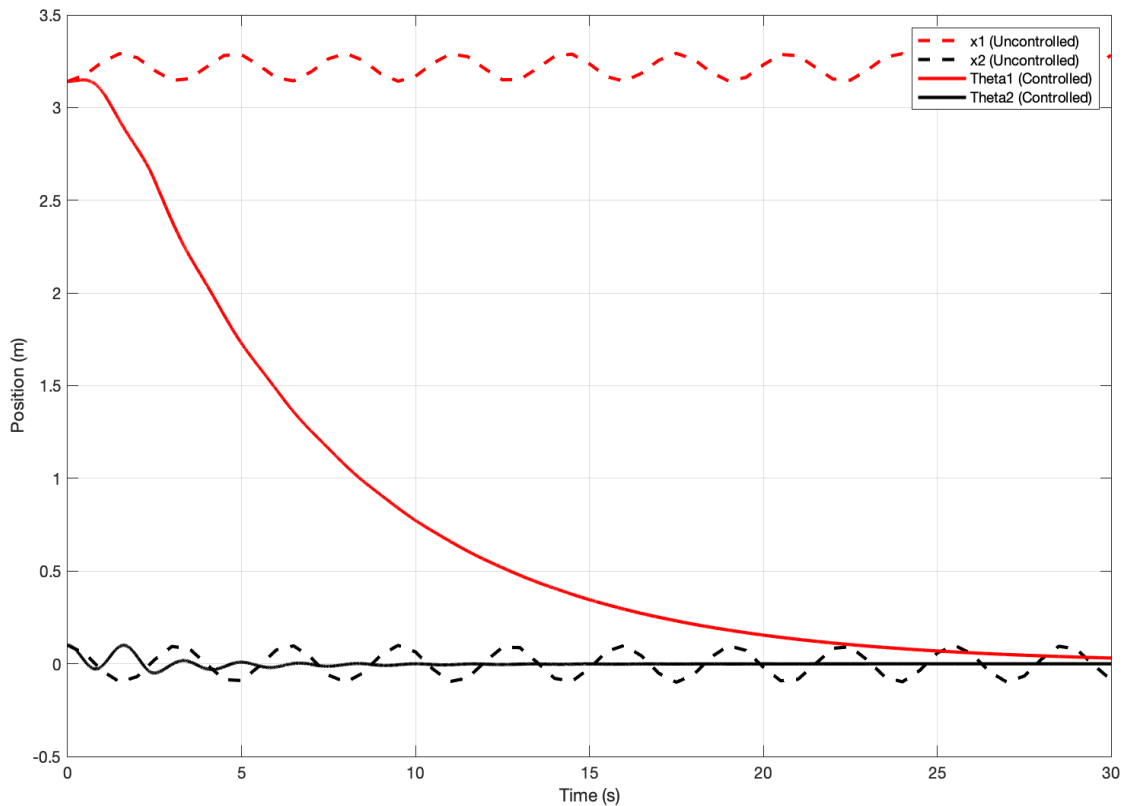


Figure 18. Nonlinear MPC controlling the vibrations of the spacecraft and beam system

The torque profiles derived from MATLAB for the MPC are set to be tested within the ROS Gazebo and the FSS floating space simulator environments. The application of a nonlinear MPC torque profile is expected to provide a nuanced understanding of the system's behavior under various operational conditions. The next section of this paper will transition into the testing phase, where these MATLAB-generated torque profiles will undergo simulation. This phase is crucial for verifying the nonlinear MPC's performance, bridging the gap between theoretical predictions and their practical implications. The

insights gleaned from these simulations will not only validate the control strategies for the SSPIDR Program but will also lay the groundwork for subsequent empirical studies, thereby enriching the domain of spacecraft dynamic control.

IV. ANALYSIS

In the following section, the analysis focuses on the ROS Gazebo simulation of the Floating Spacecraft Simulator (FSS), applying specific torque profiles in an open-loop implementation. This method is utilized to accurately simulate the system's behavior and dynamics, rather than running model predictive control (MPC) with feedback in the Gazebo simulations or hardware experiments.

A. ROS GAZEBO RESULTS

The field of robotics has witnessed significant advancements over the past few decades, with the development of sophisticated software tools and platforms that facilitate the design, testing, and deployment of robotic systems. Among these tools, the Robot Operating System (ROS) stands out as a “flexible framework for writing robot software. It provides libraries and tools to help software developers create robot applications from simple hobbyist projects to complex industrial robots” [15].

Gazebo, integrated with ROS, serves as a powerful simulation environment that allows for the testing of robots in intricate scenarios without the need for physical prototypes. Gazebo offers realistic rendering of environments, complete with physics-based interactions, sensor feedback, and dynamic changes [15]. This realism is crucial, especially when simulating scenarios that are challenging or impossible to recreate in the real world.

For spacecraft simulations, the utility of ROS Gazebo becomes even more pronounced. Spacecraft operate in environments that are inherently difficult to replicate on Earth, such as microgravity, extreme temperatures, and the vacuum of space. Using Gazebo, researchers and engineers can model these conditions and test how spacecraft systems respond, all without leaving the confines of a laboratory. This not only reduces the risks associated with space missions but also leads to cost savings, as potential issues can be identified and addressed in the simulation phase rather than post-launch.

Furthermore, the modular nature of ROS allows for easy integration of various sensors and actuators commonly found in spacecraft. This means that spacecraft

simulations in Gazebo can be incredibly detailed, encompassing everything from thruster outputs to solar panel inputs. The combination of ROS and Gazebo offers a robust platform for the simulation of mobile robots, including spacecraft.

Figures 19 and 20 show screenshots of the simulated FSS and rigid beam in a ROS/Gazebo simulation. This model used in Gazebo is based on the small-scale hub and beam system that matches closely with the experimental setup on the POSEIDYN table.

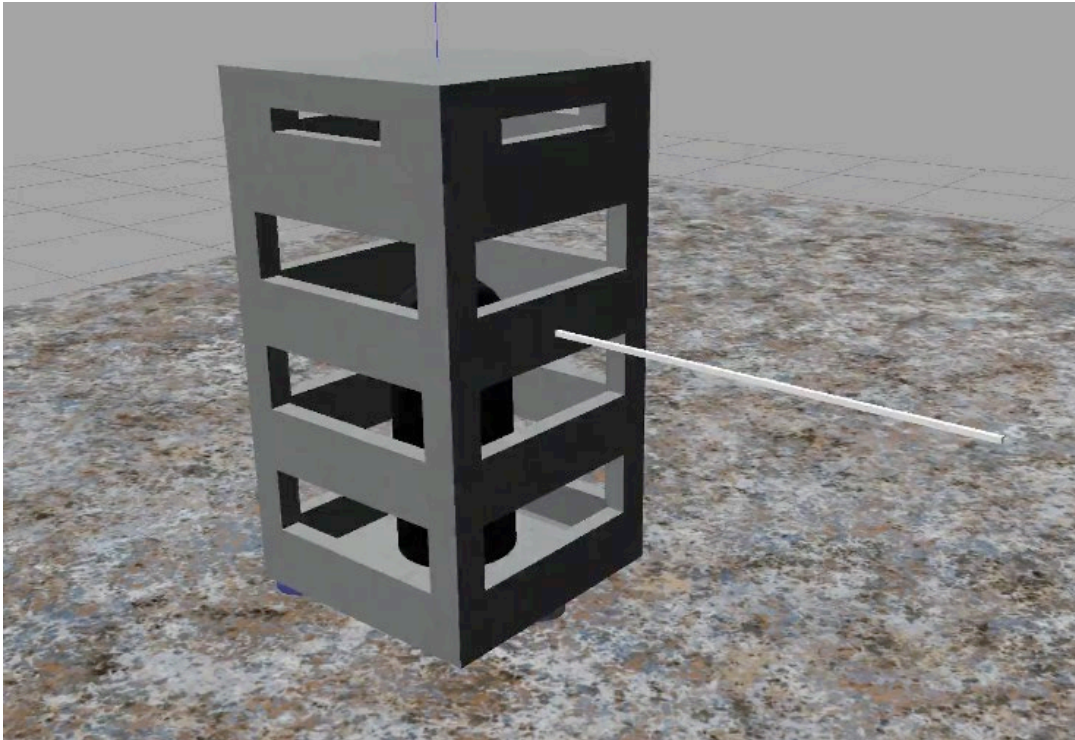


Figure 19. The FSS and Rigid beam in the ROS simulation (side view)

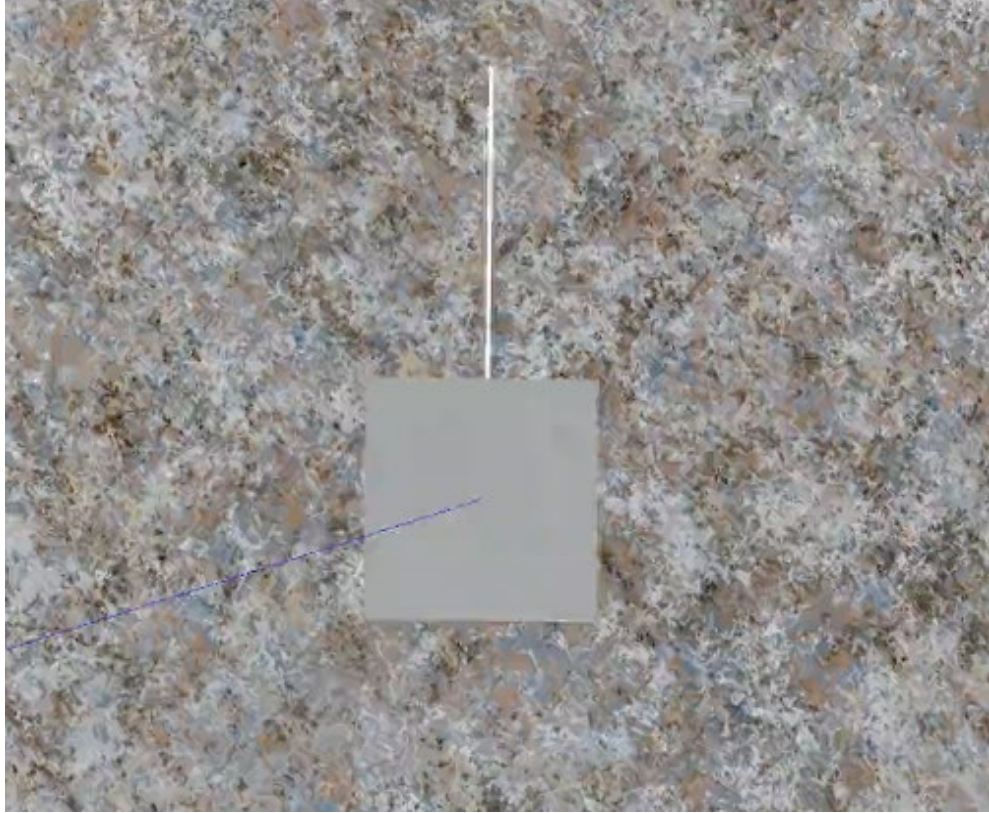


Figure 20. The FSS and Rigid beam in the ROS simulation (top view)

The analysis presented in Chapter III, Part C, derived a robust torque profile using advanced nonlinear MPC techniques. This specific torque profile was not merely a theoretical construct; it was subsequently employed as the primary input for our Floating Spacecraft Simulator (FSS) within the ROS Gazebo simulation environment. The FSS, designed to emulate real-world spacecraft dynamics in a virtual setting, applied the torque profile by channeling it through a simulated reaction wheel. This resulted in the spacecraft undergoing rotations based on the inputs from the nonlinear MPC. The entire simulation sequence was completed in a span of approximately 20 seconds. For a more visual and intuitive understanding of this dynamic process, Figure 21 shows a time-lapse representation captures the spacecraft's motion in response to the applied torque.

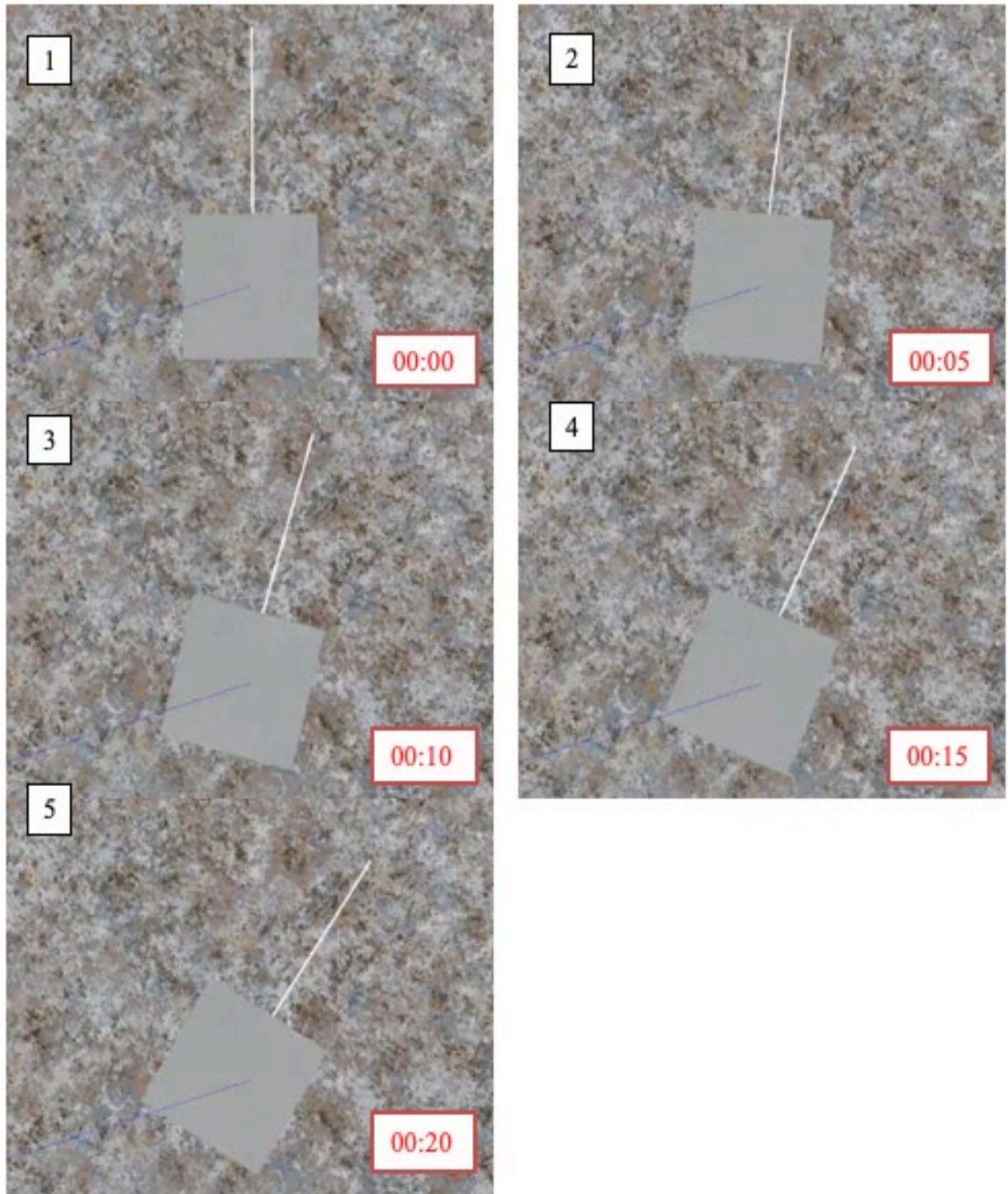


Figure 21. Top view of the FSS utilizing the MPC torque profile over 20 seconds

Upon close examination of Figure 21, one can discern the motion dynamics of the FSS. The visual representation clearly illustrates that the FSS undergoes a rotational movement pattern, which aligns with the anticipated behavior derived from the output torque of the nonlinear MPC controller.

Figure 22 captures the beam's response to the initiation of rotational movement by the FSS. The analysis revealed an immediate deflection of the beam following the application of nonlinear torque to the FSS, marking the initial excitation response. This phase is crucial in vibrational analysis as it sets the stage for the system's subsequent vibrational behavior. The simulation indicated that while the torque did cause the beam to deflect, the oscillations that followed remained consistent, without significant damping the system swiftly reaches a steady-state amplitude. This behavior is indicative of the system's capacity to stabilize into a consistent oscillation pattern following the initial excitation. Throughout the test, the amplitude of these oscillations were maintained below 6 degrees, a limitation arising from the specific parameters and constraints of the test design. This consistent amplitude and the lack of significant damping highlight the focus of the test on understanding the physical response of the system, rather than evaluating damping mechanisms.

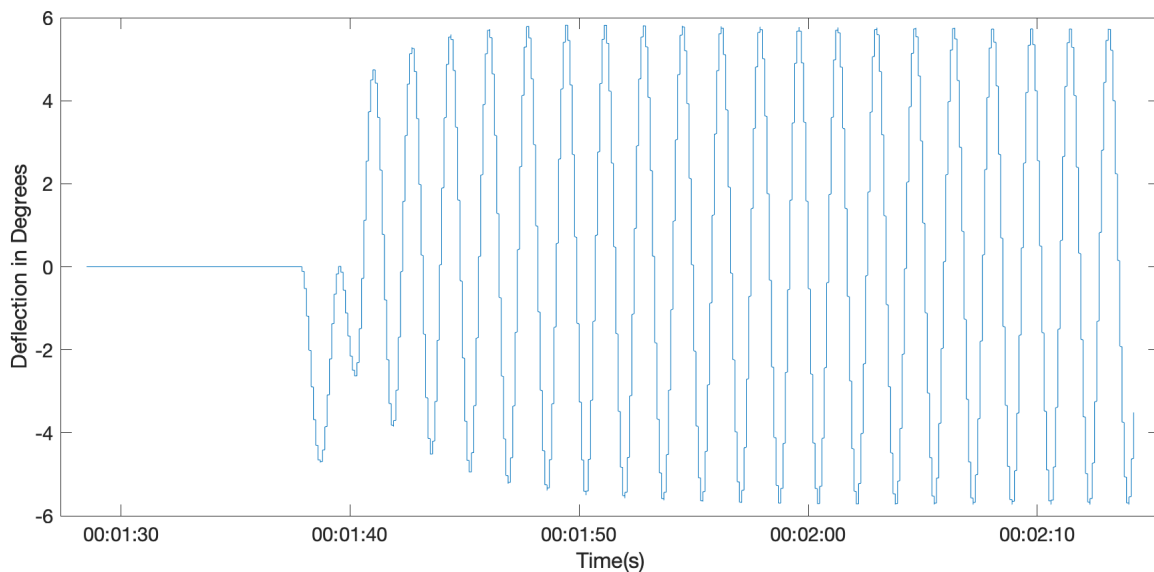


Figure 22. Graph of the vibration of the beam at the start of the simulation up to 10 seconds

The real SPS system's stability traits will differ from those observed in this simple two-body model. The attainment of a steady-state conditions without external damping forces highlights the system's stability and the effect of Nonlinear MPC in controlling its response to applied torques.

It is crucial to acknowledge that the results obtained from this simulation may be imperfect, primarily due to the limitations of the simulation's inertia model. Specifically, the onboard sensors, such as the Inertial Measurement Unit (IMU) and the Reaction Wheel, possess their own inertia, which can introduce discrepancies in the simulation outcomes. These discrepancies are particularly evident in the context of the system's response to applied torques and its vibrational characteristics. The inertia of these components can subtly alter the system's dynamics, leading to variations in the observed deflection and oscillation patterns when compared to a real-world scenario.

The ROS simulation offers a perspective on the system's physical dynamics. The initial excitation response, followed by sustained oscillations with a confined amplitude and the absence of noticeable damping, shows a steady-state amplitude. This scenario, however, must have an understanding of the potential imperfections arising from the simulation's inertia model. The use of ROS Gazebo for this experiment has been an inexpensive but effective means to begin to understand the physics of the system. It has served as a proof of concept, bridging the gap between theoretical predictions and experimental observations. Thus, ROS Gazebo highlights key areas for further research, especially in reconciling simulation outcomes with real-world physical behaviors.

B. IMPLEMENTATION OF MPC IN FSS

This chapter focuses on the POSEIDYN Floating Satellite Simulator Testbed at the Naval Postgraduate School, a cutting-edge facility for spacecraft simulation and dynamics analysis. The POSEIDYN system represents a significant advancement in the field, offering a unique platform for conducting experiments that closely mimic the conditions of space [16].

Central to the POSEIDYN system is a granite monolith table, measuring 4 meters by 4 meters with a thickness of 0.3 meters and a mass of 15,200 kilograms. This table

provides a stable and precise base for the simulation of microgravity conditions, essential for accurate and realistic space environment replication.

The system's tracking capabilities are facilitated by a ten-camera Vicon system, utilizing infrared markers. This setup ensures high-precision tracking of the Floating Spacecraft Simulators (FSS) on the table. Each FSS, with dimensions of 0.2 meters by 0.2 meters and a height of 0.8 meters, weighing 10 kilograms, is propelled by eight Controlled Gas Thrusters (CGTs), demonstrating advanced maneuverability and control.

Experiments conducted using the POSEIDYN system have primarily focused on interactions between chaser and target FSS units. The specific experiment detailed in this chapter involves an FSS equipped with a rigid beam, simulating a spacecraft structure. The aim is to investigate the dynamics of this spacecraft-beam system and evaluate the effectiveness of control strategies for vibration mitigation.

This section explores the application of MPC in controlling the Floating Spacecraft Simulator (FSS) with an attached rigid beam. It is important to clarify that the focus here is on testing an open-loop torque profile. This investigation aims to demonstrate the use of the POSEIDYN system in advancing research in spacecraft dynamics and control, using an open-loop approach to simulate control scenarios. Figure 23 provides a picture showing the granite table used by the FSS for planar motion in the lab setting.

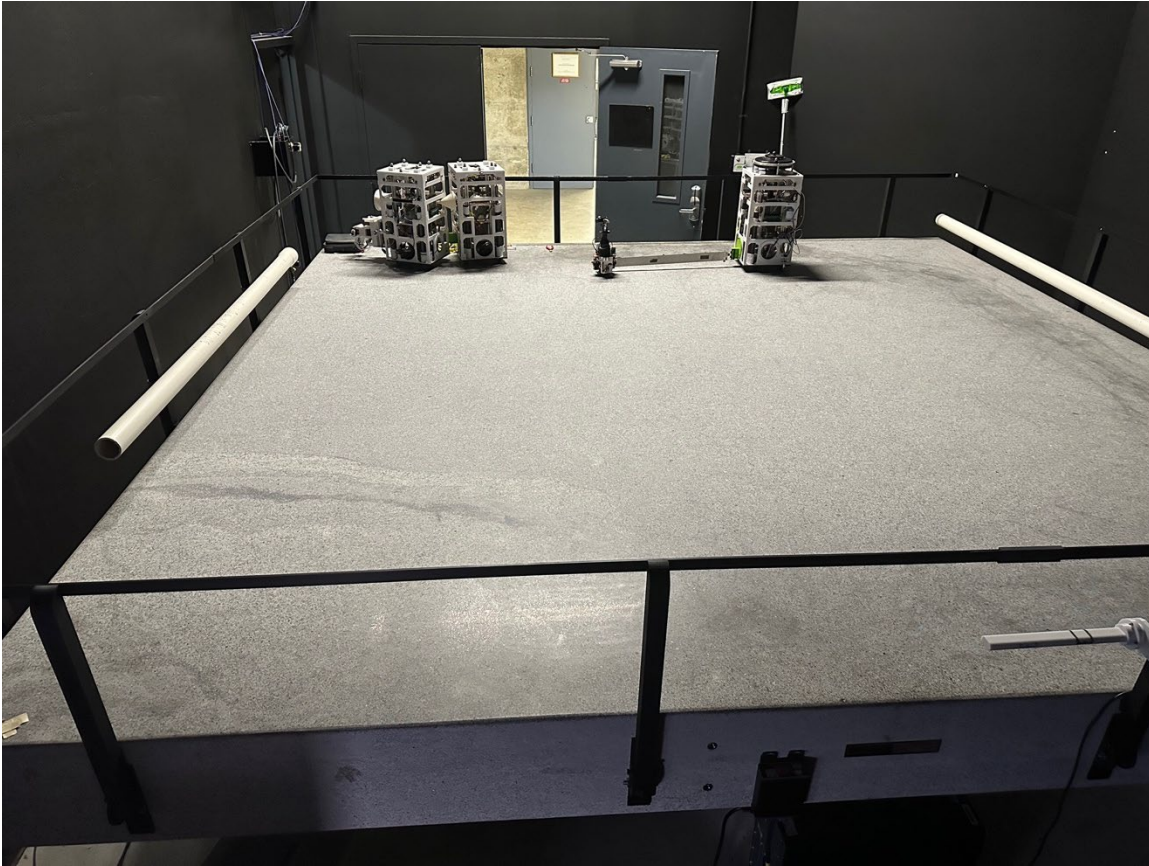


Figure 23. Aerial view of the granite table of the POSEIDYN system

The POSEIDYN Floating Satellite Simulator Testbed incorporates a sophisticated control and data acquisition system, central to which is a Simulink file designed for processing torque profiles. This system enables the conversion of uploaded torque profiles into corresponding voltage outputs. These voltage signals are then transmitted to the FSS model, where they are input into the reaction wheel. The reaction wheel, in response, generates a precise torque, facilitating controlled movement of the system.

A key feature of the POSEIDYN system is its comprehensive sensor array, which plays a crucial role in motion detection and analysis. Vicon system markers are placed along the beam and on the FSS itself. These markers are integral to capturing motion data, which is subsequently tracked by the Vicon camera system.

The motion data captured by the Vicon system is relayed to a MATLAB file, where it undergoes further processing. This step involves the generation of a visual representation

of the vibrations occurring within the system. By analyzing these visualizations, researchers can assess the dynamics of the system under various conditions and control inputs. Figure 24 shows the FSS and the Beam. Notice the markers on both. With these sensors the Vicon cameras can capture the movement of the FSS and beam for further analysis.

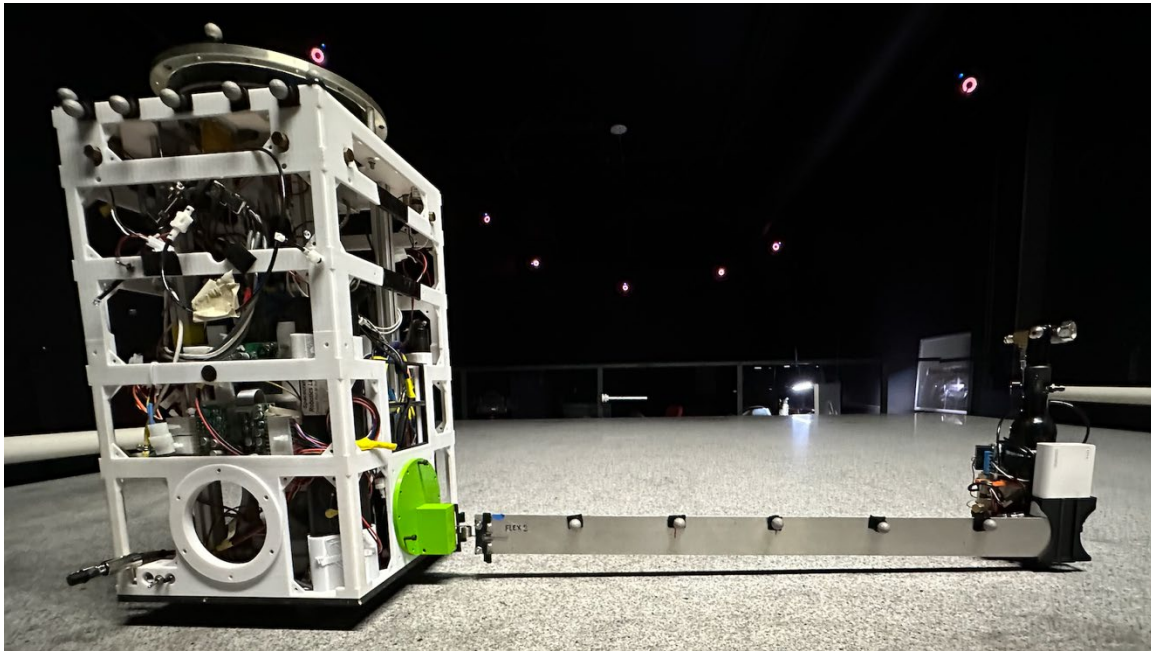


Figure 24. FSS Beam assembly with markers

This setup allows for an in-depth evaluation of the controller's performance, particularly in terms of its vibration reduction capabilities. By analyzing the response of the system to different torque inputs and observing the resultant vibration patterns, the efficiency and effectiveness of the control strategy, specifically the implementation of MPC, can be quantitatively and qualitatively assessed. This assessment is crucial for understanding the dynamics of spacecraft structures and for refining control algorithms to enhance stability and performance in real-world space missions.

1. Experimental set-up

To replicate a large beam structure, a slender aluminum piece was cut, measuring 30” in length, 0.05” in thickness, and 2” in width, with a weight of 183g. A spring hinge mechanism was designed at the revolute joint between the FSS and the beam. To accurately simulate the dynamics of a large beam structure, a meticulous conversion of the beam’s physical properties into relevant mechanical forces was undertaken.

Following a comprehensive evaluation of various torsional springs, a spring was identified that closely aligns with the experimental requirements. In an adaptation of a door hinge mechanism, the pin was removed, resulting in two distinct plates. These plates were then utilized as anchor points for the spring mechanism: one plate was securely fastened to the beam, while the other was attached to the FSS adapter. Figure 25 provides a visual representation of the assembly process, showing the integration of the hinge mechanism within the experimental setup. This figure serves to show the mechanical configuration and the approach employed in affixing the spring mechanism to the beam, which is supported by a smaller floating platform to maintain the beams weight.

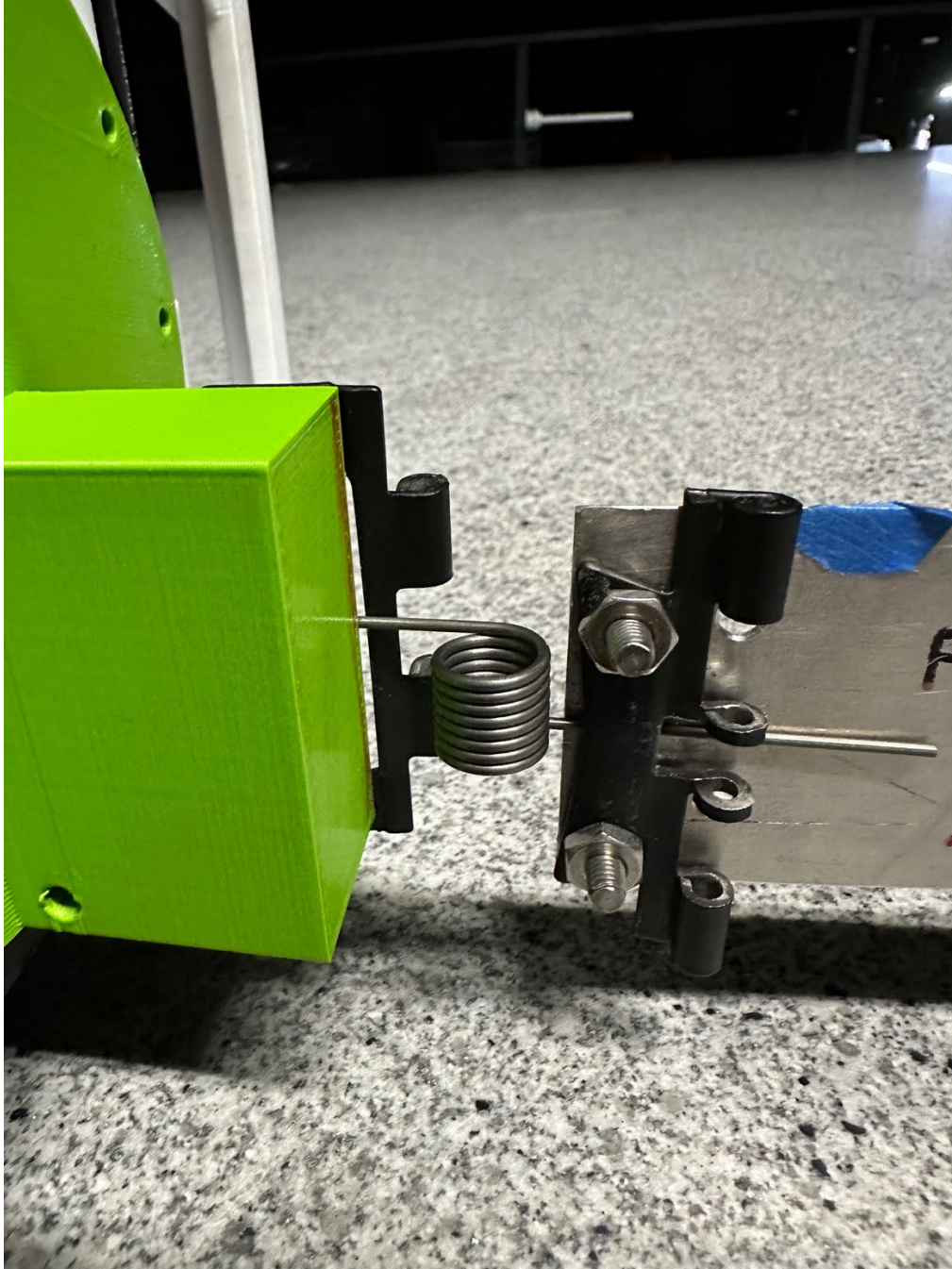


Figure 25. Spring hinge mechanism

Due to the absence of feedback regarding the position of the beam and FSS flowing through the Vicor system, this investigation constitutes an open-loop test. This approach is pivotal as a proof of concept and in comprehending the performance characteristics of the FSS and beam, laying the groundwork for future research endeavors. The open-loop

nature of the test is instrumental in elucidating the vibrational behaviors inherent in the beam and the test setup. Such an understanding is crucial for the development of a more sophisticated closed-loop system in subsequent studies. This methodology not only facilitates a deeper insight into the current system dynamics but also establishes a solid foundation for future advancements in control system integration and performance optimization.

C. FSS RESULTS

The conducted experiments involved applying the torque profile generated by the Nonlinear MPC directly to the FSS. These tests were repeated to ensure a comparison of results, particularly focusing on the vibrational responses elicited by the system.

1. MPC Test 1

Figure 26 presents a graphical representation of the beam vibrations observed during the simulation. The graph reveals an initial offset between the beam and the FSS, due to the position of the markers on the two structures. Notably, significant deflections are observed at two distinct intervals: approximately at the 60-second mark and just prior to the 80-second threshold. These deflections are consistent with previous test results conducted on this system and are hypothesized to be attributable to systemic noise, potentially emanating from the Vicon tracking system, at these specific junctures. This could also be caused by the orientation of the markers on the FSS and beam that may not be visible enough to the Vicon cameras.

As the simulation advances, the beam demonstrates a pronounced vibration pattern emerging after the 100-second mark, aligning with the test's culmination, which entailed a full 360-degree rotation of the FSS. This oscillatory behavior is primarily ascribed to the dynamic response of the spring hinge mechanism. Notably, this mechanism initiates a back-and-forth oscillation following the cessation of its rotational movement. This observed phenomenon highlights the intricate mechanical interactions among the structural components of the system, especially in reaction to the halting of rotational motion. Such observations are crucial in shedding light on the vibrational behavior of the beam, particularly within the operational context of the system's parameters.

Moreover, it is discernible from the data that minimal deflections occur outside of the identified noise intervals, up until the point where the torque applied to the FSS is deactivated. This observation suggests a direct correlation between the application of torque and the vibrational response of the system, indicating that the system maintains a relatively stable state in the absence of external torque influences, except for the noted disturbances potentially attributable to noise factors.

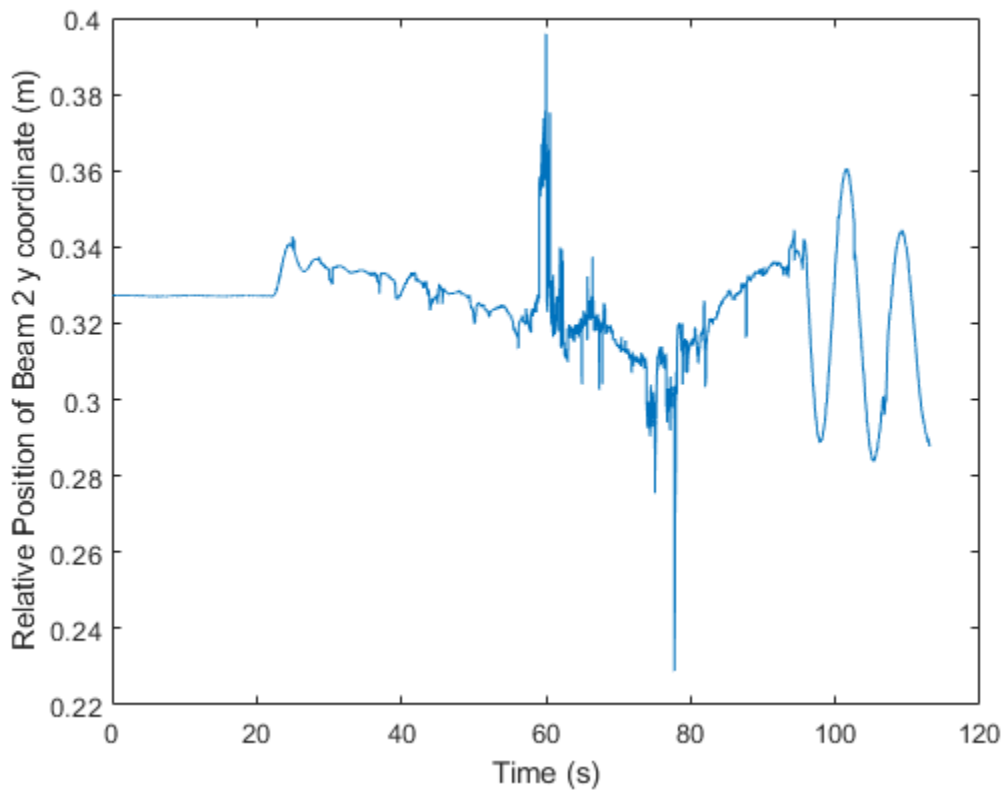


Figure 26. Relative beam vibration for test 1

Figure 27 provides a comprehensive visualization of the complete 360-degree trajectory undertaken by the FSS during the course of this test. This depiction offers a holistic view of the FSS's path, encapsulating the entire range of motion executed in the experimental setup.

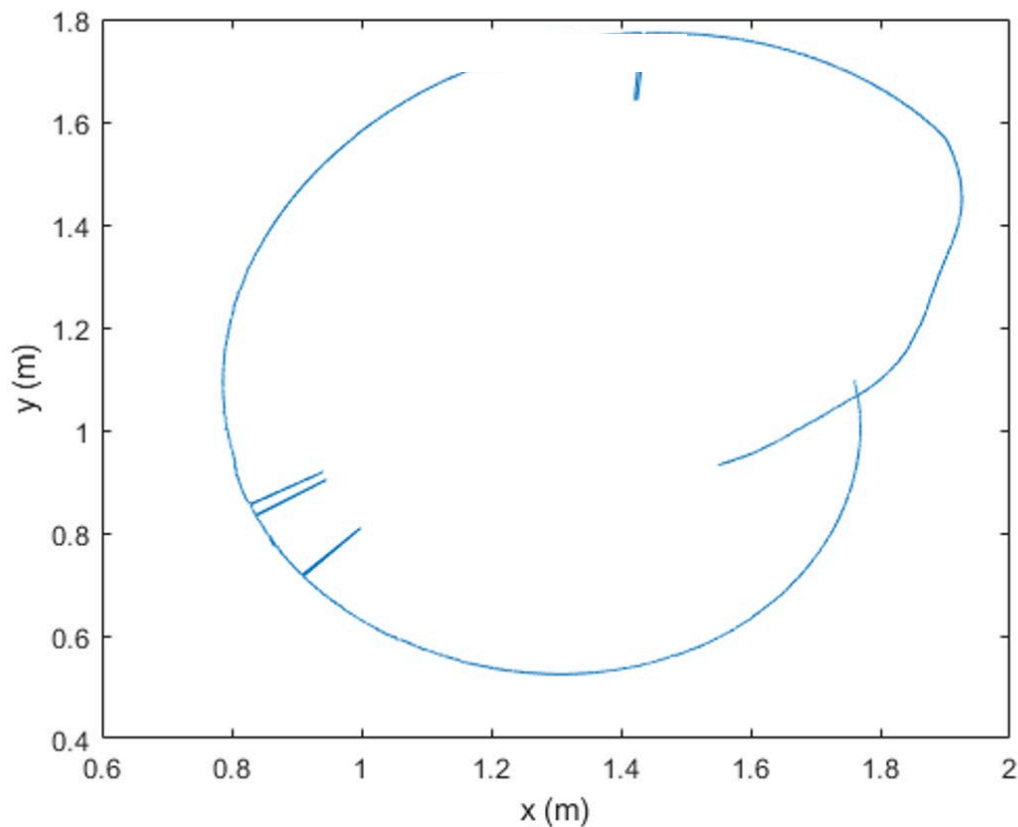


Figure 27. Path of the FSS for test 1

2. MPC Test 2

Test 2 was conducted as a means of corroborating the observations made during the initial test. The results of this subsequent test, as depicted in Figure 28, affirm the consistency of the findings. Similar to the first test, the onset of torque application is immediately followed by the deflection of the oscillating beam. This test also replicates the noise-related anomalies observed in the first test, occurring at approximately the same positions within the system's operation. The test concludes with the oscillations of the FSS as the system halts upon completing a 360-degree rotation.

A notable aspect of this test is the quantification of the beam's deflection. Excluding the periods affected by system noise, the largest deflection recorded for the beam is approximately 37 cm. However, considering an initial offset of 33 cm, the net deflection attributable to the post-torque oscillations is reduced to a mere 4 cm. The initial

offset of ~ 33 cm is due to the way that markers are positioned on the FSS and the beam. This observation is significant as it highlights the beam's behavior in response to the cessation of torque application, providing a clearer understanding of the system's dynamic response and the mechanical interactions at play once the applied torque is discontinued.

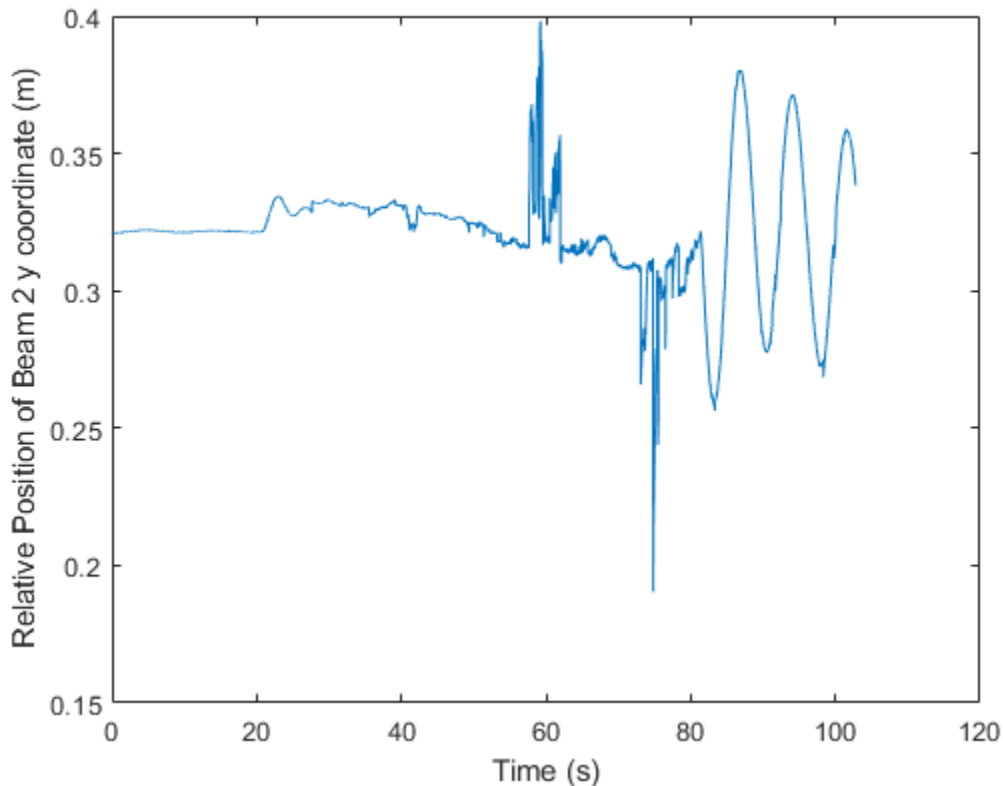


Figure 28. Relative beam vibration for test 2

Figure 29 illustrates the trajectory followed by the FSS during the second test. This test involved executing a 360-degree maneuver, after which the torque was deactivated at the 360-degree mark, facilitating the observation of beam deflection. It is important to clarify that the cessation of the beam's motion was not a direct result of turning off the torque. Instead, the absence of torque allowed the system's inherent momentum and the beam's dynamics to gradually decelerate and eventually bring the beam to a halt, during which it exhibited oscillatory behavior. This phase of the experiment is critical in

understanding the natural damping characteristics of the system and the beam's response to the removal of external driving forces. The path analysis provided by Figure 29 offers insights into the mechanical behavior of the system in a torque-free state, highlighting the inertia, structural dynamics, and the resultant vibrational patterns post-torque application.

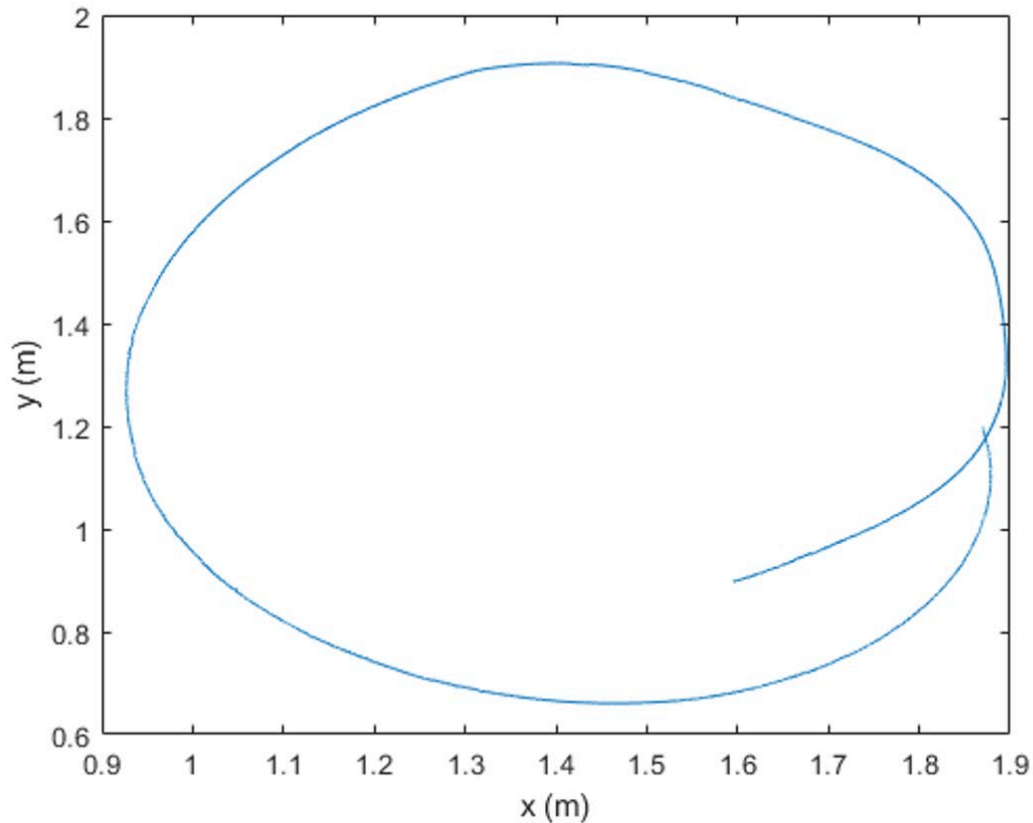


Figure 29. Path of the FSS for test 2

Despite the absence of feedback mechanisms in the testing of the MPC with the FSS, the experiments conducted have yielded significant and insightful data. These tests have been instrumental in identifying key constants and parameters within the system that require optimization. More importantly, they have provided an overview of the system's behavior, offering crucial insights into the underlying problem being addressed. While not all planned tests were completed, the ones that were carried out have been far from futile. They have laid a solid foundation for understanding the dynamics at play and have

generated meaningful data that is essential for future developments. This endeavor has not only validated the effectiveness of the MPC in the FSS but has also highlighted areas for improvement and further investigation. The knowledge gained from these tests is invaluable, forming a cornerstone for the ongoing refinement of the system and contributing significantly to the broader field of spacecraft simulation and control.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

A. SUMMARY

This study embarked on an exploration of the use of MPC and ERA for attitude control of flexible space structures, alongside practical testing using ROS/Gazebo and the Floating Spacecraft Simulator (FSS). The project delved into the foundational principles and intricate understanding of the system, while also examining the complexities inherent in advanced algorithmic approaches.

Despite the project not reaching its full completion, the research conducted has been illuminating, uncovering significant insights into the challenges and limitations of the current methodologies. The findings and the areas where the project fell short will be analyzed and elaborated upon in the section dedicated to future work. A key revelation from this study is the potential synergy between MPC and ERA, particularly when applied in a closed-loop system. Such an integration is pivotal for achieving noticeable reductions in system vibrations.

This work has laid down essential building blocks, paving the way for further advancements in this field. The integration of ERA with MPC has not only demonstrated the necessary steps and processes for this control method but has also proven to be invaluable in several key aspects of system development. One of the critical areas of contribution is in the system identification of states, which is fundamental for understanding and controlling the system's behavior.

Moreover, the creation of the beam hinge mechanism emerged as a significant aspect of this research, involving extensive trial and error to refine and perfect the design. This process was crucial in understanding the mechanical and dynamic interactions within the system.

In summary, this research has not only enhanced the understanding of MPC and ERA in the context of spacecraft simulation but has also provided invaluable insights into system design and control. The knowledge and experience gained through this project are instrumental for future explorations and innovations in this rapidly evolving field, setting

a strong foundation for the next steps in spacecraft system development and control optimization.

B. FUTURE WORK

The ongoing project with the (FSS) system has identified several key areas for future development and enhancement. While the FSS is operational, it is an older system that would benefit significantly from modernization and technological upgrades. Despite its current functionality, the system requires further refinement and additional functionality testing, tasks that were beyond the scope of this thesis due to time constraints.

A critical aspect yet to be explored in this thesis is the application of ERA to the FSS system. While the theoretical foundation and practical relevance of ERA are recognized, a comprehensive testing and analysis of these methods were not feasible within the available timeframe, primarily due to the extensive data collection and coding needed for constructing an example for analysis.

Moreover, the field of space-based solar power and vibration reduction in large structures is continuously evolving and presents significant challenges. The complexity and scale of these structures demand persistent research and development efforts. Future work in this area is crucial for advancing our understanding and capabilities in managing and mitigating vibrations, which are vital for the longevity and efficiency of space-based systems.

Incorporating the need for a feedback controller, particularly using the Vicon camera system, is an essential aspect of this future work. The Vicon system, with its advanced capabilities, can provide real-time data, enabling the development of a more effective feedback controller. This integration would not only update the FSS system with cutting-edge technology but also enhance the precision and effectiveness of vibration control. Implementing a feedback controller using the Vicon system could lead to significant improvements in the system's dynamics and overall performance.

Therefore, subsequent research should focus on updating the FSS-beam control algorithm use of the Vicon camera system for feedback control. Comprehensive

functionality tests and exploration of ERA's potential in this context are also recommended. These efforts will contribute substantially to the broader field of space technology and structural vibration control, enhancing our understanding and management of these complex systems.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. ERA STATE SPACE WITH MPC CONTROLLER

MATLAB CODE

```
clear all
close all
clc

% Initial conditions
x0 = [1; 1; 0; 0];

% Simulation time
dt = 0.5; % Simulation time step [s]
tf = 10; % Max simulation time [s]
tspan = 0:dt:tf; % ode45() time vector [s]

% Solve the ODEs using ode45
[t,x] = ode45(@myODE, tspan, x0);

% Initialize input as a row vector of zeros
u = zeros(1, length(tspan));

% Simulate input
for i = 2:length(tspan)
    u(i-1) = 0.5*sin(2*pi*tspan(i)); % Assign the input force value at each time step
end

% Plot the results
figure;
plot(t, x(:,1),'r',t, x(:,2),'k','Linewidth',2);
grid on
xlabel('Time (s)');
ylabel('Position (m)');
legend('x1', 'x2','FontSize',20);
title('4 DOF Spring Mass Damper');

% Compute Singular Value Decomposition (SVD) and estimate state-space matrices

k = 4; % The number of singular values and vectors to compute
hx = hankel(x); % Create Hankel matrix from the position data
hx1 = [hx(2:end,:); zeros(1,size(hx,2))]; % Shift the Hankel matrix down by one row
[U,S,V] = svds(hx,k); % Compute the singular value decomposition of the Hankel matrix

j = k;
r = 1;
```

```

Er = [eye(r); zeros(length(V)-r,r)];
En = [eye(j); zeros(length(V)-j,j)]; % needed to be adjusted due to the size of the matrix for input
and output
A_hat = S^(-0.5)*U'*hx1*V*S^(-0.5); % Estimate the state-space matrix A
B_hat = S^(0.5)*V*Er; % Estimate the state-space matrix B
C_hat = En'*U*S^(0.5); % Estimate the state-space matrix C
D_hat = [0]; % No direct feed through from input to output in this system

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Check ERA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
zest = zeros(length(tspan),k);
zest(1,:) = x0';
for i = 2:length(tspan)
    zest(i,:) = (A_hat*zest(i-1,:)' + B_hat*u(tspan(i)))';
end
% Plot the results
figure(2);
plot(t, zest(:,1),'b',t, zest(:,2),'g','Linewidth',2);
grid on
xlabel('Time (s)');
ylabel('Position (m)');
legend('x1 true', 'x2 true', 'x1 estimate', 'x2 estimate','FontSize',20);
title('4 DOF Spring Mass Damper');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MPC Controller %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sys = ss(A_hat,B_hat,C_hat,D_hat);
CSTR = c2d(sys,0.5);

% CSTR = ss(A_hat,B_hat,C_hat,D_hat); % Create state-space model

% Set names, units, and signals for MPC
CSTR.InputName = {'Force'};
CSTR.OutputName = {'P_1', 'P_2', 'V_1', 'V_2'};
CSTR.StateName = {'P_1', 'P_2', 'V_1', 'V_2'};
CSTR.InputUnit = {'N'}; % Ensure this has one entry per input
CSTR.OutputUnit = {'m', 'm', 'm/s', 'm/s'};
CSTR.StateUnit = {'m', 'm', 'm/s', 'm/s'};
CSTR = setmpcsignals(CSTR, 'MV', 1, 'MO', [1 2 3 4]);

damp(CSTR) % Display damping of system
step(CSTR) % Display step response of system

mpcobj = mpc(CSTR,0.5) % Create MPC object

```

```

Q = diag([100,100,100,100]);    %4x4 tuning

mpcobj.Weights.OV = {Q};

review(mpcobj) % Review MPC object
SoDC = cloffset(mpcobj) % Compute steady-state offset


T = 20; % Number of control intervals
r = [1 1 0 0];
[y,t,u] = sim(mpcobj,T,r); % Simulate MPC and get outputs, time, and input


% Plotting the results after simulation
figure; % Create a new figure
plot(t, y(:,1), 'LineWidth', 2); % Plot the first state variable against time
xlabel('Time (s)'); % Label x-axis
ylabel('Meter'); % Label y-axis
title('MPC of Spring Mass Damper System'); % Add title
grid on; % Add grid


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function dxdt = myODE(t,x)
% Parameters
m1 = 1; % Mass 1
m2 = 0.01; % Mass 2
g = 9.81; % Gravitational constant
K1 = 20; % Spring constant 1
K2 = 1; % Spring constant 2
C1 = 0.02; % Damping constant 1
C2 = 0.02; % Damping constant 2


% Input force
F = 0.5*sin(2*pi*t);


% State variables
x1 = x(1);
x2 = x(2);
x1_dot = x(3);
x2_dot = x(4);


% ODE equations
dxdt = zeros(4,1);
dxdt(1) = x1_dot;

```

```
dxdt(2) = x2_dot;  
dxdt(3) = (F + m1*g - K1*x1 - C1*x1_dot + K2*(x2 - x1) + C2*(x2_dot - x1_dot)) / m1;  
dxdt(4) = (m2*g - K2*(x2 - x1) - C2*(x2_dot - x1_dot)) / m2;  
end
```

APPENDIX B. LINEAR MPC CONTROLLER MATLAB CODE

```
clear all
close all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% System Parameters %%%%%%%%%
c1 = 0.822;
c3_2 = 1.66;
b2 = 6.24;
b4 = -7.48;

% Initial conditions
x0 = [pi 0 0 0];
xf = [0 0 0 0];

A = [0 1 0 0;
     0 0 c1 0;
     0 0 0 1;
     0 0 -c3_2 0];

B = [0; b2; 0; b4];

% Full state output
C = eye(4);
D = [0; 0; 0; 0];

sys = ss(A,B,C,D);
CSTR = c2d(sys,0.5);

%% MPC 1 %%

% Set names
CSTR.InputName = {'Force'}; % Set names of input variables
CSTR.OutputName = {'P_1', 'P_2', 'V_1', 'V_2'}; % Set names of output variables
CSTR.StateName = {'P_1', 'P_2', 'V_1', 'V_2'}; % Set names of state variables

% Set units
CSTR.InputUnit = {'N'}; % Set units of input variables
CSTR.OutputUnit = {'m', 'm', 'm/s', 'm/s'}; % Set units of output variables
CSTR.StateUnit = {'m', 'm', 'm/s', 'm/s'}; % Set units of state variables

CSTR=setmpcsignals(CSTR,'MV',1,'MO',2); % Set signals for MPC
```

```

damp(CSTR) % Display damping of system
step(CSTR) % Display step response of system

Weights.ManipulatedVariables = 1000; % Scalar if single input
Weights.ManipulatedVariablesRate = 1;
Weights.OutputVariables = [100, 100, 100, 100]; % 4x4 matrix if two outputs

mpcobj = mpc(CSTR,0.5,10,4,Weights); % Create MPC object

% Apply constraints
% mpcobj.MV(1).Min = -1;
% mpcobj.MV(1).Max = 1;

% review(mpcobj) % Review MPC object
% SoDC = clffset(mpcobj) % Compute steady-state offset

% Initialize simOptions
simOptions = mpcsimopt;
simOptions.PlantInitialState = x0;

% Simulate MPC controller
% [y, t] = sim(mpcobj, 50, x0, [], simOptions);

T = 100; % Number of control intervals
r = [pi 0 0 0; 0 0 0 0]; % 1 2 0 0; 1 2 0 0;

[y, t] = sim(mpcobj,T,r,[],simOptions); % Simulate MPC

figure; % Create a new figure
plot(t, y); % Plot the simulation results
title('Linear MPC');
%ylim([-1 1]); % Set the y-axis limits
legend('MPC FSS_1', 'MPC FSS_2', 'MPC B_1', 'MPC B_2')

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LQR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% LQR Controller Design
Q_LQR = diag([100, 100, 100, 100]); % State penalty matrix
R_LQR = 100; % Control penalty matrix

% Use the MATLAB lqr function to compute the LQR gain matrix
K = lqr(A, B, Q_LQR, R_LQR);

```

```

T_ = 50;

% Define the LQR controller
A_ = [(A-B*K)]; % u = -K*x
B_ = [B];
C_ = [C];
D_ = [D];

% Form the closed-loop system
sys_lqr = ss(A_, B_, C_, D_);

% Simulate LQR
[t_lqr, y_lqr, x_lqr] = initial(sys_lqr, x0, T_);

figure;
hold on;
%plot(t, outputs, 'LineWidth', 1);
title('Linear LQR');
plot(y_lqr, t_lqr, '--', 'LineWidth', 1); % Plot the LQR simulation results
%ylim([-5 5]);
legend('LQR FSS_1', 'LQR FSS_2', 'LQR B_1', 'LQR B_2'); % Add legend to the figure
hold off;

% Print LQR
disp('LQR Initial state:');
disp(x_lqr(1, :));

disp('LQR Final state:');
disp(x_lqr(end, :));

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. NONLINEAR MPC CONTROLLER MATLAB CODE

```
clear all
close all
clc

% FSS + beam system parameters
m1 = 10; % kg
m2 = 2; % kg
l1 = 0.27; % m FSS side
l2 = 0.4; % m beam length
l3 = 0.005; % m beam thickness
k = 0.5; % N/rad spring constant

M = m1+m2;
i1 = 1/6*m1*l1^2;
i2 = 1/12*m2*(l2^2+l3^2);

p = deg2rad(45);

% Initial conditions
r0 = 0; % m
s0 = 0; % m
theta10 = pi; % rad
theta20 = 0.1; % rad
rdot0 = 0.1; % m/s
sdot0 = 0; % m/s
theta1dot0 = 0; % rad/s
theta2dot0 = 0; % rad/sec

% Problem parameters
t0 = 0; % start time
tf = 30; % final time
dt = 0.5; % time increment between control updates

% Start MPC loop
% Initialize loop parameters
tk = t0;
theta1_true = [];
theta2_true = [];
theta1dot_true = [];
theta2dot_true = [];
T_true = [];
Torque_history = [];
D = diag([0 0 0 -k]);
yk = [r0 s0 theta10 theta20 rdot0 sdot0 theta1dot0 theta2dot0];
```

```

yksmall = [theta10 theta20 theta1dot0 theta2dot0];
options = odeset('reltol',1e-12,'abstol',1e-12);

% Define the NLMPC object
nx = 8; % Number of States:
nu = 4; % Number of Inputs
ny = 8; % Number of Outputs

% NLMPC controller
nlobj = nlmpc(nx,ny,nu);

% Define Sample Time
nlobj.Ts = dt;

% Define Prediction and Control Horizons
nlobj.PredictionHorizon = 50;
nlobj.ControlHorizon = 2;

% Adjusting weights
nlobj.Weights.ManipulatedVariables = [100 100 100 100]; % Weight on control inputs
nlobj.Weights.OutputVariables = [100 100 1000 100 1000 10000 1000 1000]; % Weight on
outputs
nlobj.Weights.ManipulatedVariablesRate = [0.5 0.5 0.5 0.5]; % Weight on rate of change of
control inputs

% Define state and output functions
nlobj.Model.StateFcn = @(x, u) MyStateFcn(x, u, m1, m2, l1, l2, i1, i2, k,M);
nlobj.Model.OutputFcn = @(x, u) MyOutputFcn(x, u);

% Initialize Controller States
x0 = [r0; s0; theta10; theta20; rdot0; sdot0; theta1dot0; theta2dot0]; % Initialize all states
u0 = [0; 0; 0; 0];
validateFcns(nlobj, x0, u0);

% Set constraints on the manipulated variable (torque)
nlobj.MV = struct('Min',{-0.05, -0.05, -0.05, -0.05}, 'Max',{0.05, 0.05, 0.05, 0.05});

% Loop
while tk<tf
    % Calculate H and C at tk
    theta2k = yk(4);
    theta1dotk = yk(7);
    theta2dotk = yk(8);
    H = findH(m1,m2,l1,l2,i1,i2,theta2k);
    C = findC(m1,m2,l1,l2,theta2k,theta1dotk,theta2dotk);

```

```

% compute control input using NLMPC
measurements = yksmall;
lastMV = [0;0;0;0]; % initialize lastMV if it is not initialized
[mv, info] = nlmpcmove(nlobj, x0, measurements);
Torque = mv; % The computed control input

% Limiting the torque to the desired range
%Torque = max(-0.005, min(0.005, mv)); % The computed control input

% Simulate the system
tspan = [tk tk+dt]; % s
[T, Y] = ode45(@(t,y) MyStateFcn(y, Torque, m1, m2, l1, l2, i1, i2, k, M), tspan, yk, options);

% Store the results
theta1_true = [theta1_true; Y(:,3)];

theta2_true = [theta2_true; Y(:,4)];
theta1dot_true = [theta1dot_true; Y(:,7)]; % Store theta1dot
theta2dot_true = [theta2dot_true; Y(:,8)]; % Store theta2dot
T_true = [T_true; T];
Torque_history = [Torque_history; repmat(Torque', length(T), 1)];

% torque values between 0.005 and -0.005 Nm
% Update the initial state
yk = Y(end, :);
x0 = yk'; % Update x0 to the new state

tk = T(end);
end
% Create a table with time and torque data
T_true = T_true(:); % Ensure T_true is a column vector
Torque_history = Torque_history(:,1); % Assuming you want to save the first torque value, adjust accordingly
data_table = table(T_true, Torque_history, 'VariableNames', {'Time', 'Torque'});

% Write the table to a .csv file
writetable(data_table, 'torque_time_data.csv');

% Create a structure with the data for theta1 and theta1dot
save_data.theta1 = theta1_true;
save_data.theta1dot = theta1dot_true;
save_data.Torque = Torque_history;

```

```

% %%%%create .mat file%%%%%%%%
save('torque_theta1_theta1dot.mat', 'save_data');

% At the beginning of mpc_control.m
load('uncontrolled_data.mat');

% Plotting the uncontrolled system's behavior (loaded from dynamics.m)
figure;
plot(t, x(:,3), 'r--', t, x(:,4), 'k--', 'LineWidth', 2); % Using dashed lines for uncontrolled behavior
hold on;

% Plotting the controlled system's behavior (from the MPC code)
plot(T_true, theta1_true, 'r', 'LineWidth', 2); % Solid red line for controlled theta1
plot(T_true, theta2_true, 'k', 'LineWidth', 2); % Solid black line for controlled theta2

xlabel('Time (s)');
ylabel('Position (m)');
legend('x1 (Uncontrolled)', 'x2 (Uncontrolled)', 'Theta1 (Controlled)', 'Theta2 (Controlled)');
grid on;
hold off;
%%%%%%%%

figure()
plot(T_true, Torque_history);
title('Torque over Time');
xlabel('Time (s)');
ylabel('Torque (N·m)');

figure()
% Plotting theta1
subplot(4,1,1);
plot(T_true, rad2deg(theta1_true), 'LineWidth', 2); % Convert to degrees
title('Theta1 over Time');
xlabel('Time (s)');
ylabel('Theta1 (deg)'); % Update label to degrees
grid on;

% Plotting theta2
subplot(4,1,2);
plot(T_true, rad2deg(theta2_true), 'LineWidth', 2); % Convert to degrees
title('Theta2 over Time');
xlabel('Time (s)');
ylabel('Theta2 (deg)'); % Update label to degrees
grid on;

% Plotting theta1dot
subplot(4,1,3);
plot(T_true, rad2deg(theta1dot_true), 'LineWidth', 2); % Convert to degrees

```

```

title('Theta1dot over Time');
xlabel('Time (s)');
ylabel('Theta1dot (deg/s)'); % Update label to degrees per second
grid on;

% Plotting theta2dot
subplot(4,1,4);
plot(T_true, rad2deg(theta2dot_true), 'LineWidth', 2); % Convert to degrees
title('Theta2dot over Time');
xlabel('Time (s)');
ylabel('Theta2dot (deg/s)'); % Update label to degrees per second
grid on;

sgtitle('State Trajectories over Time');

function dxdt = MyStateFcn(x, u, m1, m2, l1, l2, i1, i2, k, M)

% Extracting states.
theta2 = x(4);
theta1dot = x(7);
theta2dot = x(8);
M = m1+m2;

% Finding C and H matrices.
H33 = (1/(4*M))*(m1+m2)*(l1^2+l2^2+2*l1*l2*cos(theta2))) + (i1+i2);
H34 = (1/(4*M))*(m1+m2)*(l2^2+l1*l2*cos(theta2))) + i2; % Corrected H34
H44 = (1/(4*M))*((m1+m2)*l2^2) + i2;
H = [M 0 0 0; 0 M 0 0; 0 0 H33 H34; 0 0 H34 H44];

C33 = -(1/(4*M))*(m1*m2*l1*l2*sin(theta2)*theta2dot);
C43 = (1/(4*M))*(m1*m2*l1*l2*sin(theta2)*theta1dot);
C = [0 0 0 0; 0 0 0 0; 0 0 C33 (C33-C43); 0 0 C43 0];

v = x(5:8, 1);
G = [0; 0; 0; -k * x(4)];
%F = [u(1); u(2); u(3); u(4)];
F = [0;0;u(1);0];

% Solving for q_ddot
q_ddot = H \ (G + F - C*v);

dxdt = [x(5:8); q_ddot];
end

function y = MyOutputFcn(x, u)
y = x;
end

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. NONLINEAR DYNAMICS OF THE SYSTEM MATLAB CODE

```
%% MPC_NONLINEAR

% clear all;
% close all;
% clc;

v = pi;

% Initial conditions
x0 = [0; 0; v; 0.1; 0.1; 0; 0; 0];
u0 = 0; % initial control input

% Simulation time
dt = 0.5; % Simulation time step [s]
tf = 30; % Max simulation time [s]
tspan = 0:dt:tf; % ode45() time vector [s]

% Solve the ODEs using ode45
[t,x] = ode45(@yODE, tspan, x0);

% Plot the results
figure;
plot(t, x(:,3), 'r', t, x(:,4), 'k', 'Linewidth', 2);
grid on
xlabel('Time (s)');
ylabel('Position (m)');
legend('x1', 'x2');

% At the end of dynamics.m
save('uncontrolled_data.mat', 't', 'x');

%% non linear MPC Controller

%%
function dxdt = yODE(t,x)

M = 3.0; %kg
m1 = 1.0; %kg
```

```

m2 = 2.0; %kg
l1 = 1.0; %m
l2 = 1.5; %m
i1 = (1/3)*m1*l1^2; % 0.3333; kg.m^2
i2 = (1/3) * m2 * l2^2; % 1.5; %kg.m^2
k = 0.5; %N/m
r = 1;
s = 1;
theta1 = x(3);
theta2 = x(4);

theta1_dot = x(7);
theta2_dot = x(8);

G = [0; 0; 0; -k*theta2];

H33 = (1/(4*M))*(m1+m2)*(l1^2+l2^2+2*l1*l2*cos(theta2))) + (i1+i2);
H34 = (1/(4*M))*(m1+m2)*(l1^2+l2^2+cos(theta2))) + i2;
H44 = (1/(4*M))*((m1+m2)*l2^2) + i2;

H = [M 0 0 0; 0 M 0 0; 0 0 H33 H34; 0 0 H34 H44];

C33 = -(1/(4*M))*(m1*m2*l1*l2*sin(theta2)*theta2_dot);
C43 = (1/(4*M))*(m1*m2*l1*l2*sin(theta2)*theta1_dot);
C = [0 0 0 0; 0 0 0 0; 0 0 C33 (C33-C43); 0 0 C43 0];

% ODE equations
dxdt = zeros(4,1);
dxdt(1) = x(5);
dxdt(2) = x(6);
dxdt(3) = x(7);
dxdt(4) = x(8);
dxdt(5:8) = inv(H) * (G - C * x(5:8));

end

```

LIST OF REFERENCES

- [1] P. A. Curreri and M. K. Detweiler, “A Contemporary Analysis of the O’Neill – Glaser Model for Space-based Solar Power and Habitat Construction,” [Online]. Available: <https://ntrs.nasa.gov/api/citations/20120001744/downloads/20120001744.pdf>.
- [2] D. Gueho, P. Singla, M. Majji, and J.-N. Juang, “Advances in System Identification: Theory and Applications,” in *2021 60th IEEE Conf. on Decision and Control (CDC)*, Austin, TX, USA, Dec. 2021, pp. 22–30, doi: 10.1109/CDC45484.2021.9683394.
- [3] J. A. Vedda and K. L. Jones, “Space-Based Solar Power: A Near-Term Investment Decision,” Center for Space Policy and Strategy, Space Agenda 2021, 2021.
- [4] R. Rodrigues, A. Murilo, R. V. Lopes, and L. C. G. D. Souza, “Hardware in the Loop Simulation for Model Predictive Control Applied to Satellite Attitude Control,” *IEEE Access*, vol. 7, pp. 157401–157416, 2019, doi: 10.1109/ACCESS.2019.2949731.
- [5] G. Takács, T. Polóni, and B. Rohal’-Ilkiv, “Adaptive Model Predictive Vibration Control of a Cantilever Beam with Real-Time Parameter Estimation,” *Shock and Vibration*, vol. 2014, pp. 1–15, 2014, doi: 10.1155/2014/741765
- [6] J.N. Juang and R. Pappa, “An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction,” *Journal of Guidance, Control, and Dynamics*, vol. 8, Nov. 1985, doi: 10.2514/3.20031.
- [7] S. Oymak and N. Ozay, “Revisiting Ho–Kalman-Based System Identification: Robustness and Finite-Sample Analysis,” *IEEE Trans. Automat. Contr.*, vol. 67, no. 4, pp. 1914–1928, Apr. 2022, DOI: 10.1109/TAC.2021.3083651.
- [8] U. Eren, A. Prach, B. B. Kocer, S. V. Rakovic, E. Kayacan, and B. Acikmese, “Model Predictive Control in Aerospace Systems: Current State and Opportunities,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017, doi: 10.2514/1.G002507.
- [9] G. Takács and B. Rohal’-Ilkiv, *Model Predictive Vibration Control*, Springer-Verlag London Limited, 2012, doi: 10.1007/978-1-4471-2333-0 [Online]. Available: <https://link.springer.com/content/pdf/10.1007/978-1-4471-2333-0.pdf>. [Accessed: Jan. 5, 2023].

- [10] Ø. Hegrenæs, J. T. Gravdahl, and P. Tøndel, “Spacecraft attitude control using explicit model predictive control,” *Automatica*, vol. 41, no. 12, pp. 2107–2114, Dec. 2005, doi: 10.1016/j.automatica.2005.06.015.
- [11] “*What is Model Predictive Control?*” MATLAB & Simulink, MathWorks, Accessed: Oct. 26, 2023 [Online]. Available: <https://www.mathworks.com/help/mpc/gs/what-is-mpc.html>
- [12] J. Virgili-Llop, C. Zagaris, H. Park, R. Zappulla II, and M. Romano, “Experimental Evaluation of Model Predictive Control and Inverse Dynamics Control for Spacecraft Proximity and Docking Maneuvers,” *CEAS Space Journal*, vol. 10, no. 1, pp. 37–49, Mar. 2018.
- [13] C. Zagaris, H. Park, J. Virgili-Llop, M. Romano, and I. Kolmanovsky, “Model Predictive Control of Spacecraft Relative Motion with Convexified Keep-Out-Zone Constraints,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 9, pp. 2054–2062, Sep. 2018.
- [14] S. Chen and D. Negrut, “*The Basics of Model Predictive Control in Multibody Dynamics*,” Simulation-Based Engineering Lab, University of Wisconsin-Madison, Tech. Rep. TR-2019-03-2, Oct. 2019 [Online]. Available: <https://myaidrive.com/iBe4oik6sudciD8h/TR-2019-03-2.pdf>. [Accessed: May 3, 2023].
- [15] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, “Simulation environment for mobile robots testing using ROS and Gazebo,” in *2016 20th Int. Conf. on System Theory, Control and Computing (ICSTCC)*, Oct. 2016, pp. 96–101, doi: 10.1109/ICSTCC.2016.7790647.
- [16] R. Zappulla II, J. Virgili-Llop, C. Zagaris, H. Park, and M. Romano, “Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers,” *Journal of Spacecraft and Rockets*, vol. 54, no. 4, pp. 825–839, Apr. 2017.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Fort Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California



DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE