ECE 445

SENIOR DESIGN LABORATORY

SENIOR DESIGN REPORT

MEMS-Based Feedback Controller

Team #33

PEIHENG YAO

(Peiheng5@illinois.edu)

JIANBO XU

(jianbox2@illinois.edu)

ZHIYUAN XIE

(zx30@illinois.edu)

TA: Xuyang Bai

May 18, 2023

Acknowledgement

We would like to express my sincere gratitude to the individuals and organizations who have contributed to the successful completion of our project. Their support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, we extend my deepest appreciation to our project sponsor Binbin Li and his PhD students for their unwavering support and invaluable guidance. We lack expertise in civil engineering. We are grateful to Professor Li and his students for their guidance and assistance in enhancing our professional knowledge. Their expertise and mentorship have been instrumental in shaping the direction of this project and pushing me to strive for excellence.

We would also like to acknowledge the generous support provided by our ECE445 indicator, Professor Mark Butala, who provided us invaluable help and guidance, and helpful advice from our TA Xuyang Bai. Professor Mark recommended Professor Cristoforo to help us. Thanks for Professor Cristoforo, with his new design ideas and solutions, we achieved our goals within the final two weeks.

We are grateful to my dedicated team members who have worked tirelessly alongside me. Their collaborative spirit, technical expertise, and unwavering commitment have played a vital role in the project's success. Their valuable insights and suggestions have significantly improved the quality of our work.

Lastly, we are very thankful to Beijing University of Science and Technology who provided us with technical support throughout this project. We are very grateful to the selfless assistance from there professor and students.

Abstract

This study aimed to investigate the use of a Micro-Electro-Mechanical Systems (MEMS) accelerometer for detecting and controlling the vibrations of a two-story building model. The initial experimental setup involved collecting acceleration data using the MEMS accelerometer, transmitting this data via an NI acquisition system to a computer, and processing the data using Simulink for active control algorithm implementation.

However, due to challenges with our actuator selection and algorithm complexity, we employed a simplified control system, controlling a small servomotor using an Arduino board interfaced with a Python-Tkinter GUI. We built a two-story building model using wooden blocks, bamboo sticks, and an L-shaped iron base, designed to vibrate primarily in one direction.

We collaborated with professors at the University of Science and Technology Beijing, who utilized a Quanser system for a similar experiment. This advanced setup involved a small car that acted as a vibration source on a single-story model, with control executed through Simulink.

Our findings indicate that using MEMS accelerometers for vibration detection in structures holds promise, although our setup had limitations due to hardware and algorithmic constraints. Comparison with the Quanser system's results provided valuable insights into the viability of our approach, contributing to our understanding of active vibration control in building structures.
Key words: Micro-Electro-Mechanical Systems (MEMS), Accelerometer, Vibration detection, Active control algorithm, Simulink, Arduino, Python-Tkinter GUI, Servomotor, Two-story building model, Quanser system, Vibration control in building structures, NI acquisition system, Control Systems.

Contents

1 Introduction

1.1 Problem

The problem we set out to address in this senior design project involves the creation and control of a two-story building model to simulate real-life scenarios of building movement caused by external stimuli such as earthquakes. Real buildings can suffer significant structural damage if subjected to intense oscillations, which can lead to catastrophic consequences. This is a common problem in regions with high seismic activity.

Micro-Electro-Mechanical Systems (MEMS) technology, especially MEMS accelerometers, have been identified as potential tools for detecting such vibrations. MEMS accelerometers are capable of detecting acceleration changes in all three dimensions (x, y, and z), making them ideal for comprehensive building movement analysis. However, a challenge arises in the practical application of MEMS accelerometers to control building vibrations.

The primary issue in this context is the complexity of control algorithms required to appropriately respond to the accelerometer's feedback and, in turn, actuate an effective countermeasure to balance the building's oscillations. Additionally, there is a technical challenge in selecting and deploying suitable actuator systems that can effectively respond to the control signals.

The final aspect of this problem lies in integrating the control algorithm, the MEMS accelerometer, and the actuator into a real-time vibration control system for a two-story building model. The system should be able to accurately detect vibrations, compute the necessary countermeasures, and implement them effectively to minimize the building's overall movement.

1.2 Solution

To address the problem defined, we have designed and developed a MEMS-based feedback control system for a two-story building model. The solution comprises several key elements.

First, a two-story building model was built from blocks of wood connected by bamboo sticks, allowing for a simulation of structural vibrations in one direction. An L-shaped metal plate was placed at the base to provide stability.

Second, a MEMS accelerometer was used to detect the acceleration in three dimensions (x, y, and z) of each part of the building. The acceleration data was acquired through a NI acquisition system and transferred to a computer via an amplifier.

Third, a control algorithm was developed in Simulink to process the acceleration data from the MEMS accelerometer and generate an appropriate response to counteract the detected vibrations.

However, due to the complexity of the control algorithm and the technical challenge in deploying a suitable actuator, a simpler control system was implemented for the initial testing phase. A small servo motor was used as an actuator, controlled by an Arduino to counteract the detected vibrations. A user-friendly interface was designed to allow easy control of the servo motor's speed.

Further, we collaborated with professors from Beijing University of Technology who have a fully equipped Quanser shake table. This shake table consists of a single-story building model with a cart on top. The vibration excitation from the shake table was transferred to the cart, causing the entire model to sway. The control of the cart was also managed by Simulink, aiming to reduce the overall sway of the model.

Like the block diagram we draw above, we have built two control systems. The first one uses

Wooden Bamboo Structure without feedback control, and the second one uses one floor flexible structure with feedback control.

In the first one, our shaker will perform a periodic vibration according to a sinusoidal signal which are provided by a wave generator, then go through an amplifier. And MEMS sensor will detect the acceleration of the ground, send signals to NI acquisition system which can be connected to an external display screen to show the changes in acceleration.

In the second one, we have implemented an observer-based state-feedback controller that dampens out the vibrations in the one floor flexible structure by driving the active mass (i.e., linear cart). Moreover, instead of using a shaker to generate vibration, we use linear cart itself to implement the structure self-excitation.

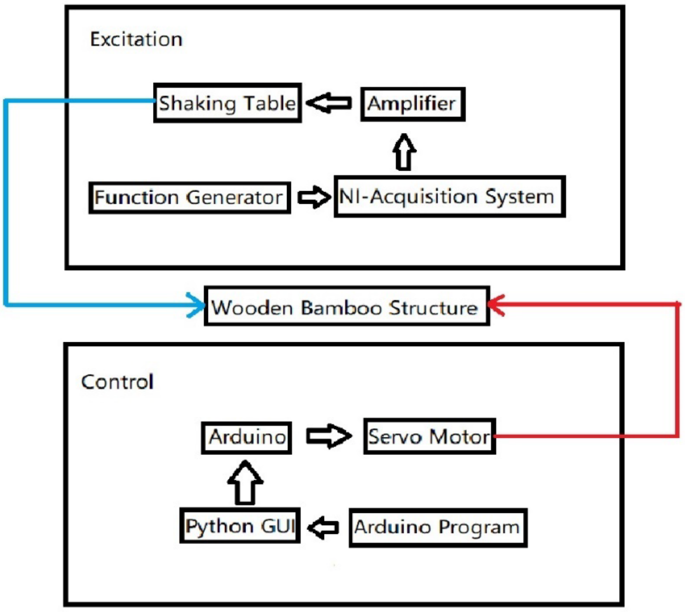


Figure 1: Block Diagram of Simplified System

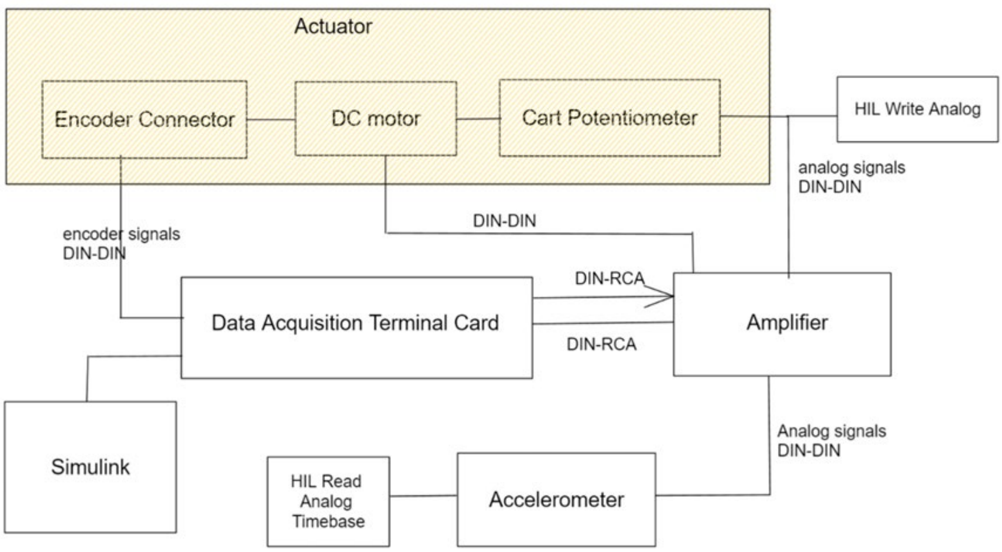


Figure 2: Block Diagram of Advanced System

1.3 High-level Requirements List

The high-level requirements for this project have been established to ensure the successful design and implementation of the MEMS-based feedback controller for the two-story building model. These requirements act as the guiding principles during the entire process.

MEMS Accelerometer: The MEMS accelerometer used in the project must be capable of accurately measuring the acceleration in three dimensions (x, y, and z) of each part of the building model.

Data Acquisition and Amplification: The NI acquisition system and amplifier must successfully collect and transmit the acceleration data to the computer system for further processing.

Control Algorithm: The control algorithm must be designed to effectively process the acceleration data and generate suitable control signals to counteract the vibrations.

Actuator System: The actuator system, specifically the servo motor, should respond accurately to the control signals generated by the control algorithm. It must successfully mitigate the detected vibrations.

Building Model: The two-story building model should be robustly constructed and capable of realistically simulating the vibrations typically experienced by real buildings due to external stimuli.

Real-time Control: The entire system, from data acquisition to vibration control, should operate in real-time to ensure timely detection and mitigation of any detected vibrations.

Collaborative Testing: Collaboration with Beijing University of Technology must result in successful testing and validation of the vibration control system on their Quanser shake table.

User Interface: The user interface designed for controlling the servo motor's speed should be intuitive and easy to use, allowing smooth control of the vibration control system.

The successful achievement of these high-level requirements will ensure the effective operation of the MEMS-based feedback controller for the two-story building model.

2 Design

The complete system comprises four distinct subsystems: MEMS Accelerometer, Data Acquisition and Amplification, Control Algorithm, and Actuator System. The Data Acquisition and Amplification serves as the central component, receiving, processing, and transmitting information to facilitate interaction among the other three subsystems.

Specifically, the MEMS Accelerometer subsystem is primarily responsible for accurately measuring the acceleration in three dimensions (x, y, and z) of each part of the building model and sending this data. The Control Algorithm subsystem is responsible for processing the acceleration data and generating suitable control signals to counteract the vibrations. The Actuator System subsystem is equipped to respond accurately to the control signals generated by the control algorithm, successfully mitigating the detected vibrations.

This system is further supplemented by the Building Model and the User Interface. The Building Model, representing the real-world application of the system, experiences vibrations that the system must control. The User Interface allows for easy control of the servo motor's speed and system parameters, providing an intuitive and user-friendly interaction with the system.

2.1. Initial Experimental Setup

2.1.1 MEMS Accelerometer

The Micro-Electro-Mechanical System (MEMS) accelerometer plays a crucial role in our vibration control system, functioning as the primary sensor for detecting accelerations. This device is essentially a small, low-cost sensor that measures acceleration based on changes in capacitance or piezo resistivity. It can accurately measure the accelerations in three dimensions (x, y, and z) experienced by different parts of the two-story building model.

The MEMS accelerometer has several characteristics that make it an excellent choice for this application:

Sensitivity: The device is very sensitive and can detect even minute changes in acceleration. This is vital for our project as it allows us to detect small vibrations in the building model that could potentially cause damage.

Size and Weight: MEMS accelerometers are tiny and lightweight. Their compact size means they can be placed in various parts of the building model without significantly affecting its dynamics.

Low Power Consumption: MEMS accelerometers require very little power to operate, which makes them ideal for applications where power resources might be limited.

Robustness: The devices are very robust and can withstand a range of operating conditions, making them well suited for a variety of applications.

Cost-effectiveness: MEMS accelerometers are relatively inexpensive, making them an affordable solution for our project.

In our project, MEMS accelerometer will be used to detect the ground acceleration (i.e., xag) or detect the building's vibration. In our initial design, giving the ground acceleration, we can output the control signal "u", using evaluation model to forecast the state of model. In this way, we can mitigate the impact of latency and indirectly control the vibration state of the model. It is a pity that without special hardware support, we cannot achieve the loop control in our wooden model.

By the way, we can still use this accelerometer to detect the effectiveness of our control system.

As shown in the image below, it provides a visual representation of the current acceleration through NI acquisition system, allowing us to easily interpret the data.

Figure 3 MEMS Accelerometer

2.1.2 NI Acquisition System

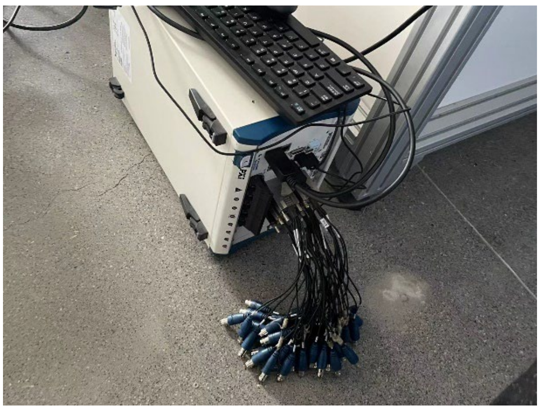
Data acquisition is a crucial part of our system as it allows us to retrieve the real-time data from the MEMS accelerometer. This process is facilitated by the NI acquisition system, which is a high-performance device capable of acquiring signals from multiple sensors simultaneously and at a high rate. This allows us to gather acceleration data in real-time from all the three directions (x, y, z) of each part of the two-story building model.

The NI acquisition system also allows for signal conditioning, filtering, and amplification, which helps enhance the signal's quality before it is processed. This is particularly important given that the signal from the MEMS accelerometer might be relatively weak and noisy.

After the signals are amplified, they are transmitted to the computer via an amplifier. The use of an amplifier is necessary to ensure that the signal is strong enough to be processed by the subsequent stages of the system. It boosts the signals from the MEMS accelerometer, thus ensuring that the control algorithm can accurately interpret the data.

Overall, the Data Acquisition and Amplification subsystem plays a crucial role in processing and transmitting the data from the MEMS accelerometer to the computer for further analysis and control. It serves as a bridge that links the physical world (the accelerometer and the building model) to the digital world (the control algorithm and the actuator system).

In the previous design, the shaker will simulate realistic seismic wave. Because the realistic seismic wave is changing in a rapid rate, it is too difficult for us to control the vibration of our model in such unpredictable wave. We choose to generate a sinusoidal periodical wave, and for the realistic seismic wave, we put it in the simulation part.



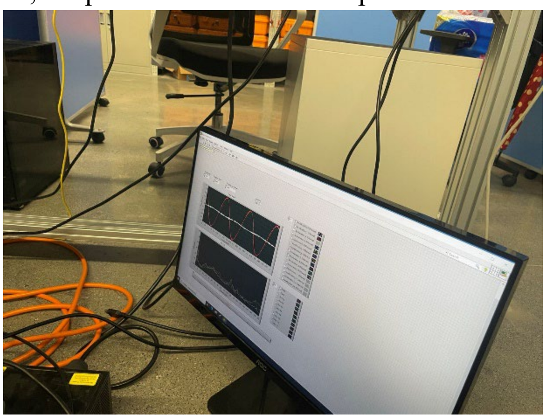


Figure 4&5 NI acquisition system and display screen

2.1.3 Simulink Processing and Control

1. Update for DiscreteTransferFcn Blocks:
    - The code updates the states of multiple DiscreteTransferFcn blocks, including '<S2>/Discrete Transfer Fcn', '<S3>/Discrete Transfer Fcn', '<S4>/Discrete Transfer Fcn', '<S5>/Discrete Transfer Fcn', and '<S6>/Discrete Transfer Fcn'.
    - For each block, the code calculates the intermediate variable 'denAccum' or 'denAccum_#' based on the current value of 'Sum[#]' and the previous states of the respective DiscreteTransferFcn block.
    - The updated states are then stored in 'Arduino1_DW.DiscreteTransferFcn_states[#]' for each block, where '[#]' represents the index.
2. Update for UnitDelay Block:
    - The code checks if the current task is the first task (TaskCounters.TID[1] == 0) and updates the UnitDelay state 'Arduino1_DW.UnitDelay_DSTATE' with the value of 'Arduino1_B.Quantizer1'.
3. Update for DiscreteStateSpace Block:
    - The code checks if the current task is the second task (TaskCounters.TID[2] == 0) and performs an update operation for the DiscreteStateSpace block.
    - The update operation involves calculating a new state vector 'xnew' based on the current input 'Arduino1_B.Quantizer' and the previous state vector 'Arduino1_DW.DiscreteController_DSTATE'.
    - The updated state vector 'xnew' is then copied back to 'Arduino1_DW.DiscreteController_DSTATE'.
4. Matfile Logging:
    - The code includes operations related to logging data for the Simulink model.
    - It updates the time and state variables to be logged using the 'rt_UpdateTXYLogVars' function.
    - It also checks if the simulation has finished and sets an error status if necessary.
5. Update of Absolute Time and Rate Scheduler:
    - The code updates the absolute time for the base rate and increments the clockTick counters.
    - It calculates the task time based on the clockTick values and the step size.
    - The rate_scheduler function is called to determine the scheduling of subsequent tasks.
6. Model Initialization and Termination:
    - The code includes initialization and termination functions for the model.
    - The initialization function sets up the model, initializes variables, and starts data logging.
    - The termination function handles any required cleanup.
    -

2.2 Simplified Control System

2.2.1 Servomotor and Arduino

To control a servo motor, you need to use a different approach. Servo motors have specific requirements for controlling their position and movement.

```
#include <Servo.h>
Servo myservo;    // Create a servo object
void setup() {
   myservo.attach(9);    // Attach the servo to pin 9
}
void loop() {
   myservo.write(90);    // Set the servo to the 90-degree position
   delay(1000);             // Wait for 1 second
   myservo.write(180); // Set the servo to the 180-degree position
   delay(1000);             // Wait for 1 second
}
```

In this code, We are using the Servo library to control a servo motor connected to pin 9 of Arduino board.

First, we include the Servo library, which provides functions to control servo motors. We create a Servo object named myservo to interface with the servo motor.

In the setup() function, we initialize the servo motor by calling the attach() function. The attach(9) statement attaches the myservo object to pin 9 of the Arduino board, indicating that the servo motor is connected to this pin.

In the loop() function, we perform a sequence of actions to control the servo motor. The write() function is used to set the position of the servo motor. In the first write(90) statement, we set the servo to the 90-degree position, which corresponds to a specific angle of rotation. The servo motor will move to this position.

After setting the servo to the 90-degree position, we introduce a delay of 1 second using the delay(1000) statement. This causes the program to pause execution for 1 second, allowing the servo motor to remain in that position for the specified duration.

Next, we use the write(180) statement to set the servo to the 180-degree position, which corresponds to a different angle of rotation. The servo motor will move to this new position.

Again, we introduce a delay of 1 second using delay(1000) to maintain the servo at the 180-degree position for the specified duration.

This sequence of setting the servo to different positions and introducing delays is repeated indefinitely in the loop() function.

2.2.2 Tkinter-based GUI for Arduino Control

1. Importing Libraries: The script imports necessary libraries such as **tkinter**, **ttk**, and **messagebox** to create the GUI and handle user interactions.
2. **update_and_upload_code()** Function: This function is responsible for updating the Arduino code and uploading it to the Arduino board. It performs the following steps:
   - Reads the Arduino code from a specified file.
   - Modifies the code by replacing a specific line that sets the duty cycle of an analog output pin with the value selected from a scale widget in the GUI.
   - Saves the modified code back to the file.
   - Uses the Arduino CLI (Command Line Interface) to compile and upload the code to the Arduino board connected to the specified port.

- Displays error messages if the compilation or upload process fails.

3. **on_scale_change()** Function: This function is called whenever the value of the scale widget changes. It updates a label in the GUI to reflect the selected value.
4. **on_window_close()** Function: This function is triggered when the user closes the GUI window. It sets the scale value to the maximum (255) and calls the **update_and_upload_code()** function to ensure that the latest code is uploaded before closing the application.
5. GUI Setup: The script creates a Tkinter root window and defines the mainframe and grid layout. It configures the scale widget, label, and button in the mainframe, specifying their positions and behavior.
6. Child Widget Configuration: A loop is used to configure padding for all the child widgets in the mainframe, ensuring consistent spacing.
7. **root.protocol()** Call: This line sets a protocol for the root window. It binds the **on_window_close()** function to the window's close button, ensuring that the specified behavior occurs when the user attempts to close the application.
8. **root.mainloop()** Call: This initiates the main event loop, allowing the GUI to be displayed and user interactions to be processed.

2.3 Simulation

For the simulation part, there will be a 28-state model with the following state variables:

x(i): displacement of i-th floor relative to the ground (cm)

xm: displacement of AMD relative to 3rd floor (cm)

xv(i): velocity of i-th floor relative to the ground (cm/s)

xvm: velocity of AMD relative to the ground (cm/s)

xa(i): acceleration of i-th floor relative to the ground (g)

xam: acceleration of AMD relative to the ground (g)

d(1) =x(1), d(2)=x(2)-x(1), d(3)=x(3)-x(2): inter-story drifts

according to the formula [1] below [2].

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{E}\ddot{x}_g \tag{1}$$

$$\mathbf{y} = \mathbf{C}_y\mathbf{x} + \mathbf{D}_y u + \mathbf{F}_y \ddot{x}_g + \mathbf{v} \tag{2}$$

$$\mathbf{z} = \mathbf{C}_z\mathbf{x} + \mathbf{D}_z u + \mathbf{F}_z \ddot{x}_g \tag{3}$$

The result is.

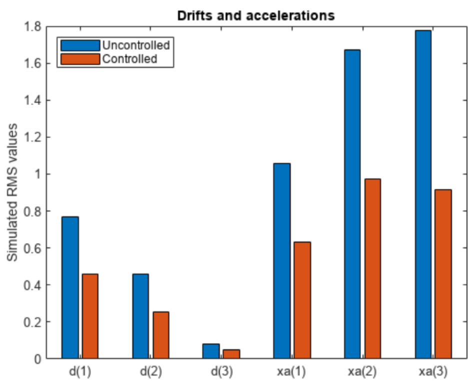


Figure 6 Plot from MATLAB

Another simulation [1] in Simulink:

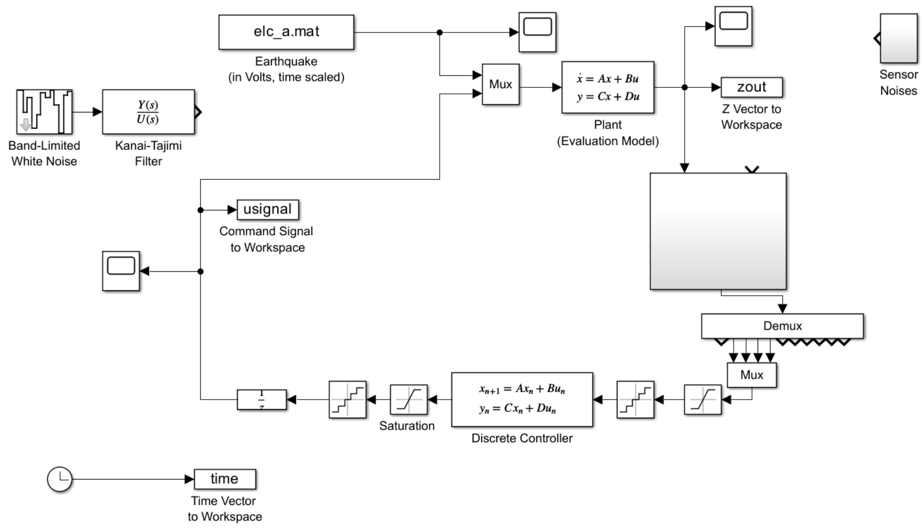


Figure 7 Simulation in Simulink

The seismic wave and floor's drift under AMD control:

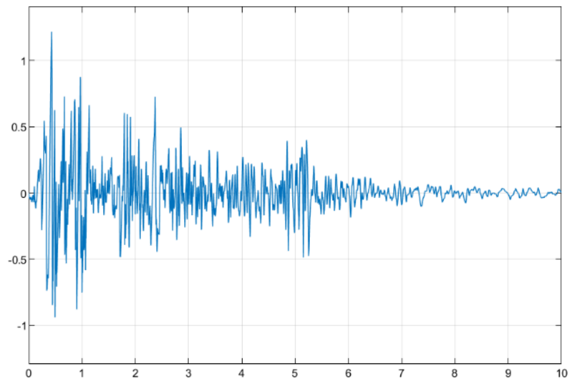


Figure 8 Floor's drift wave

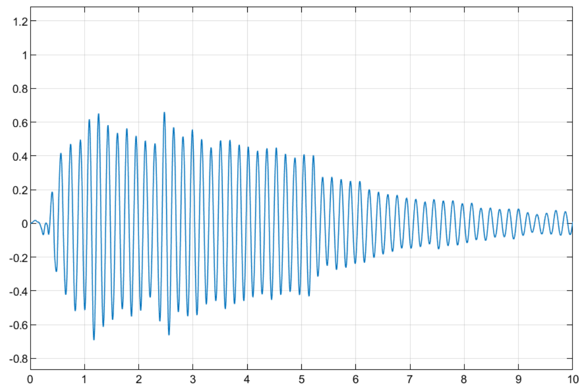


Figure 9 Seismic wave

2.4 Control Program

For the second one, We did the software part.

Figure 10 The second one control system

The AMD-1 plant is a bench-scale model to emulate a building controlled by an Active Mass Damper (AMD). The plant consists of a single-story building-like structure on top of which a linear cart (i.e., active mass) is driven by a rack and pinion mechanism, The top floor is instrumented with an accelerometer to measure the acceleration of the “roof” relative to ground. The structure frame is made of steel and is flexible along its façade.

According to the system’s state-space representation, we represent the state-space matrices A, B, C and D to be as follows:

```
A( 1, 1 ) = 0;
A( 1, 2 ) = 0;
A( 1, 3 ) = 1;
A( 1, 4 ) = 0;
A( 2, 1 ) = 0;
A( 2, 2 ) = 0;
A( 2, 3 ) = 0;
A( 2, 4 ) = 1;
A( 3, 1 ) = 0;
A( 3, 2 ) = Mc*r_mp^2*Kf/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
A( 3, 3 ) = -(Mc*Eff_g*Kg^2*Eff_m*Kt*Km+Mc*Beq*Rm*r_mp^2+Mf*Eff_g*Kg^2*Eff_m*Kt*Km+Mf*Beq*Rm*r_mp^2)/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
A( 3, 4 ) = 0;
A( 4, 1 ) = 0;
A( 4, 2 ) = -Kf*(Mc*r_mp^2+Jm*Kg^2)/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
A( 4, 3 ) = Mc*(Eff_g*Kg^2*Eff_m*Kt*Km+Beq*Rm*r_mp^2)/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
A( 4, 4 ) = 0;

B( 1, 1 ) = 0;
B( 2, 1 ) = 0;
B( 3, 1 ) = Eff_g*Kg*Eff_m*Kt*r_mp*(Mc+Mf)/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
B( 4, 1 ) = -Mc*Eff_g*Kg*Eff_m*Kt*r_mp/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);

C( 1, 1 ) = 1;
C( 1, 2 ) = 0;
C( 1, 3 ) = 0;
C( 1, 4 ) = 0;
C( 2, 1 ) = 0;
C( 2, 2 ) = -Kf*(Mc*r_mp^2+Jm*Kg^2)/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
C( 2, 3 ) = Mc*(Eff_g*Kg^2*Eff_m*Kt*Km+Beq*Rm*r_mp^2)/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
C( 2, 4 ) = 0;

D( 1, 1 ) = 0;
D( 2, 1 ) = -Mc*Eff_g*Kg*Eff_m*Kt*r_mp/Rm/(Mc*r_mp^2*Mf+Jm*Kg^2*Mc+Jm*Kg^2*Mf);
```

Figure 11 state-space matrices

There are some basic model parameters associate with the active mass damper (part of technical support).

```
% Floor 1 Height (m)
Hf = 21 * K_IN2M; % = 0.5334;
% Rack Mass (kg)
Mr = 0.7;
% Flexible Structure Total Mass (with accelerometer) (kg)
Ms = 1.60;
% Structure Top Floor Mass (kg)
Mtf = 0.68;
% Top Floor/Roof (i.e. floor 1 + rack) Total Mass (kg)
Mf = Mr + Mtf; % = 1.38
% AMD-1's Top Floor Natural Frequency (Hz)
fn = 2.5;
% AMD-1's Top Floor Natural Angular Velocity (rad/s)
wn = 2 * pi * fn;
% AMD-1's Top Floor Linear Stiffness Constant (N/m)
Kf = ( Mf + Mc ) * wn^2; % = 500.9
% Floor Accelerometer Sensitivity (m/s^2/V)
global K_ACC
% accelerometer calibration: 1.0 g/V
% positive floor acceleration is defined to the right when facing the AMD-1
K_ACC = 9.81;
% end of 'setup_amd1_parameters( )'
```

Figure 12 model parameters

And the controller file for AMD system in Simulink:

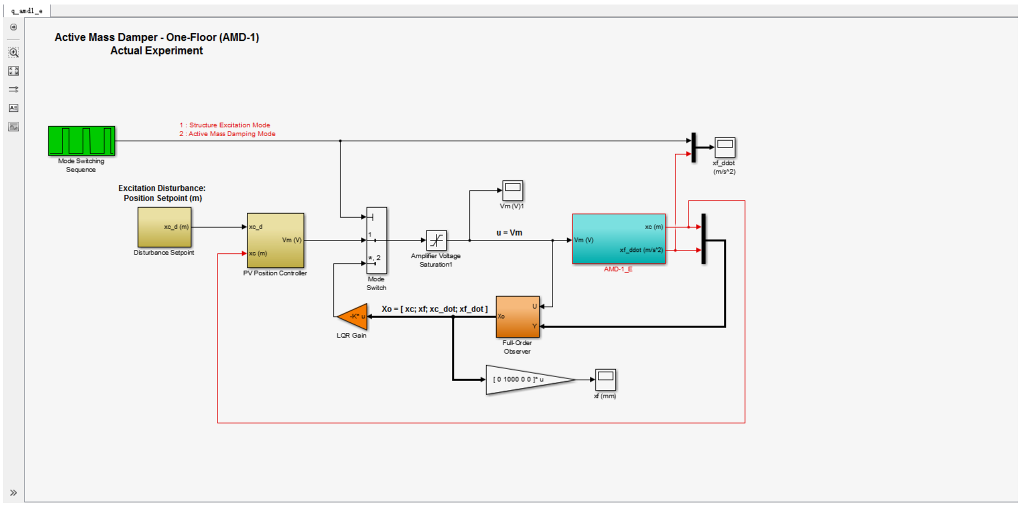


Figure 13 Simulink Controller

The experiment results show that with the AMD feedback control system, the vibration is significantly reduced as show below.

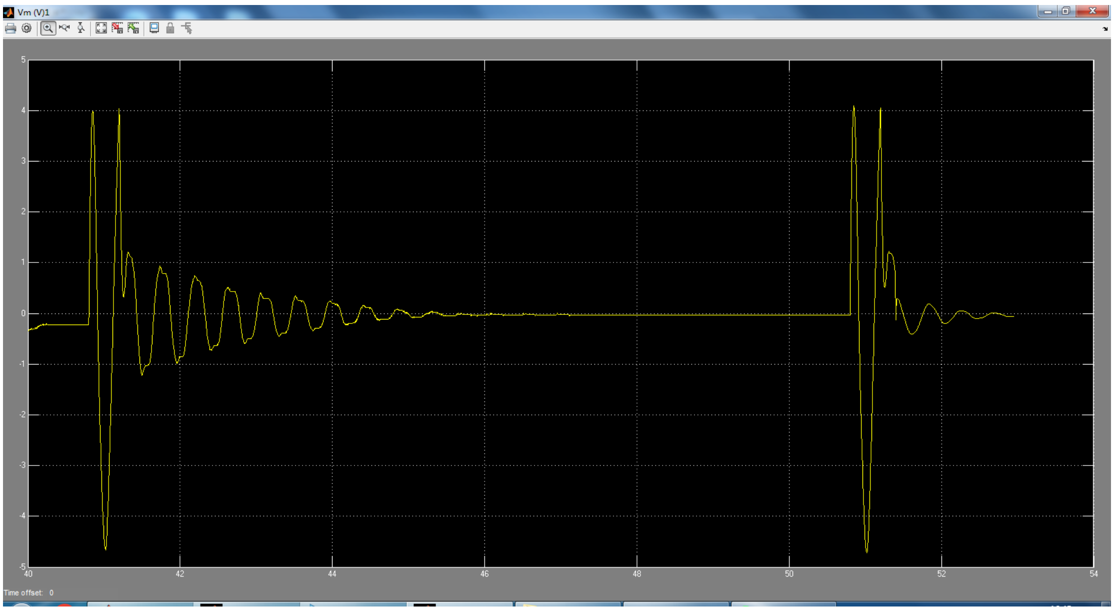


Figure 14 Comparison about floor's velocity

3 Verification and Experimentation

3.1 Two-Story Building Model

We collectively created a two-story model building. This structure was made entirely from wooden blocks, meticulously designed and constructed to meet our requirements. The wooden blocks were connected using bamboo strips, representing a creative use of readily available, eco-friendly materials.

To ensure the stability and rigidity of our building model, we employed self-tapping screws. These screws were specifically used to secure the bamboo strips to the wooden blocks. However, in an interesting design decision, we only fixed the bamboo strips to two sides of each wooden block. This was done with the intention of allowing the entire structure to vibrate in a specific direction, corresponding to the sides where the bamboo strips were affixed. This unique design approach brought about both structural and aesthetic considerations to the forefront of our project.

In the base of the model building, an L-shaped iron plate was utilized. This served as a solid and sturdy foundation for our two-story structure, ensuring it was adequately supported and could stand upright. This choice of material for the base brings a contrast to the primarily wooden design, highlighting our ability to incorporate various materials into our project.

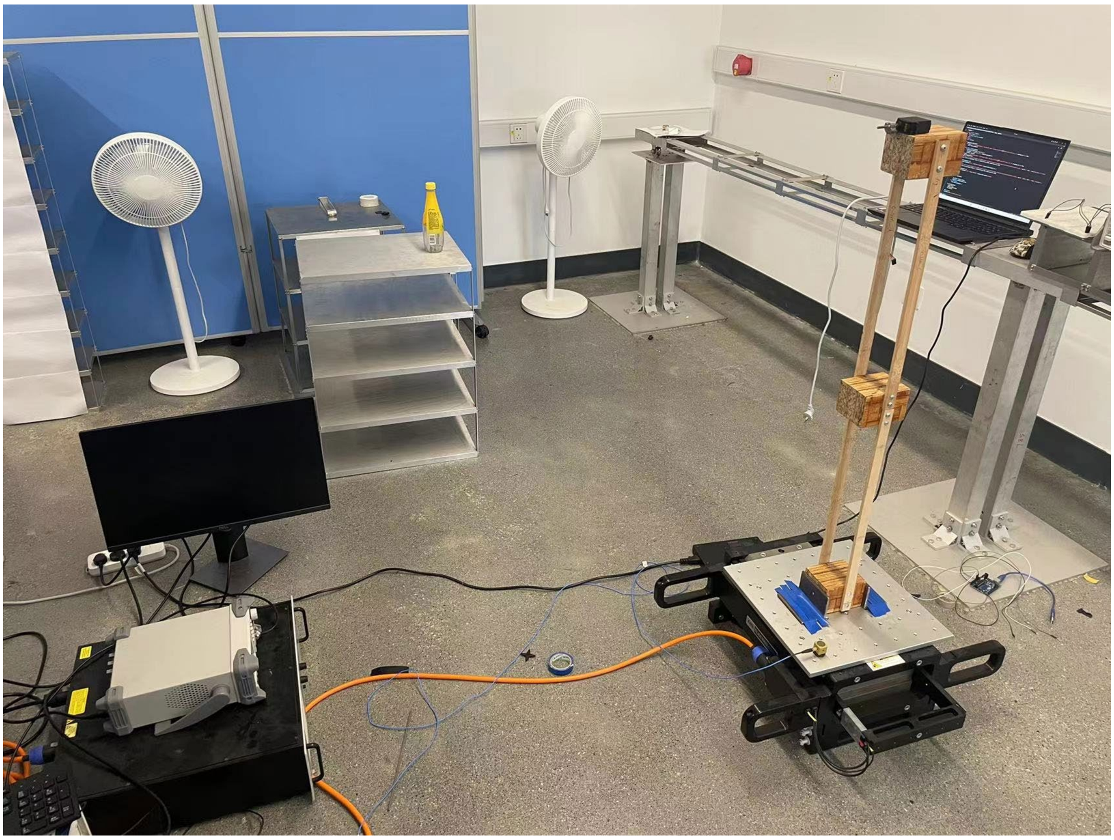


Figure 15: 2-story Building and the Verification Experiment

3.2 Collaborative Experiment with Quanser System

Due to certain limitations inherent in our experimental setup, we sought assistance from a professor at the University of Science and Technology Beijing. The institution possessed a complete Quanser experimental setup, which proved instrumental in the completion of our project. This setup consisted of a single-story building model with a small car mounted at the top. In this modified system, the shaking table's role was replaced by the small car. The car's vibration acted as a stimulus, causing the entire model to shake. The primary objective of the experiment was to control the motion of the car in order to minimize the overall shaking of the building model. Similar to our initial approach, the entire control mechanism was governed by Simulink. This collaborative experimentation allowed us to overcome our setup constraints and achieve more robust results.

4 Conclusion

4.1 Summary of Findings

Throughout the course of this project, a variety of significant findings were made. At the inception of the project, the design centered around the use of a MEMS accelerometer and an NI acquisition system to measure and process data relating to acceleration in three directions: X, Y, and Z. The data collected was then processed using Simulink, a flexible and versatile platform that facilitated effective and efficient analysis.

The implementation of an actuator on a two-story building model, designed to balance vibration through the control of a sliding rail, revealed insights about the dynamic interaction between active control algorithms, structural behavior, and actuation mechanisms. Despite the limitations of the actuator, a servo motor controlled via Arduino was successfully employed to induce vibration in the model.

Our simplified control system, while not as sophisticated as initially envisioned, provided essential insights into the operation and potential limitations of such systems. The system's responsiveness, speed, and precision are critical aspects that directly influence the effectiveness of vibration reduction strategies.

Furthermore, our collaborative experiment with Beijing University of Technology using the Quanser system added another layer of understanding. Their model, which incorporated a cart at the top of a single-story building model to stimulate vibration, further illustrated the intricacies of active vibration control.

However, we faced significant challenges concerning the control algorithm's complexity and the actuator's speed and precision. A higher speed actuator that met our requirements was too complex to be controlled using 220v voltage, presenting safety and operability issues.

Overall, while we encountered obstacles, the project offered valuable insights into active vibration control, control algorithm development, and the balance between actuator precision, speed, and controllability in real-world applications.

4.2 Limitations and Challenges

One of the main limitations of our project was the intricate nature of the control algorithm. Implementing a robust control mechanism required a keen understanding of various factors including system dynamics, noise levels, and sensitivity of the feedback loop. These factors had to be considered when programming the control commands, making it a demanding task given the resources and time constraints of an undergraduate project.

In addition, the requirements for the actuator presented significant challenges. The successful operation of our experiment relied heavily on the actuator's precision and speed. It was essential that the actuator had the capability to move the small car at the required speed to effectively counterbalance the shaking of the building model.

However, acquiring an actuator that satisfied these criteria proved to be a daunting task. The available options either lacked the necessary speed or, if they could reach the desired speed, presented a new set of complications. For instance, high-speed actuators typically required control at a voltage level of 220V. Such a high voltage control scheme can be quite complex and poses a safety concern, especially for an undergraduate-level project where safety is paramount. This meant that the use of such an actuator would have added an unwelcome layer of complexity to our project.

Therefore, despite our best efforts, these constraints hindered our ability to fully realize the control algorithm and effectively mitigate the vibrations in the building model. It was a valuable lesson in the practical difficulties that can be encountered in implementing theoretical concepts, reminding us that real-world engineering is often a balance between ideal outcomes and practical limitations.

4.3 Future Work

In light of our findings, our future work aims to integrate real seismic wave samples into the testing process to further evaluate the robustness of our control system. This advancement represents a significant step towards reproducing the complexity and unpredictability of real-world conditions. Utilizing actual seismic data will challenge our system in ways that simulated or manually induced vibrations cannot, offering a more comprehensive understanding of its performance and resilience. This would further highlight potential areas of improvement in the design, accuracy, and responsiveness of our system, ultimately enhancing the effectiveness of our active control strategy for vibration reduction in buildings. To make this feasible, it will be necessary to explore advanced actuation systems that can meet the required speed and precision specifications and refine our control algorithm to handle the complexities presented by real seismic wave data.

4.4 Ethical considerations

4.4.1 Ethics

1. we have carefully read the relevant code of ethics of IEEE [3] and must ensure that MEMS will not violate the rules. All product-related data must be true, which is related to the privacy of users. Invading the privacy of others is strictly prohibited.
2. The feedback controller relies on data from sensors, which must be protected from unauthorized access or misuse. The data collected from the MEMS-based accelerometer must be securely transmitted and stored to prevent any compromise of privacy or security. The controller must also be designed with robust security features to prevent unauthorized access to the system.
3. It is essential to ensure that the feedback controller is transparent and accountable in its operations. This includes ensuring that the controller's actions are clearly visible, understandable, and auditable. There should be a mechanism in place to monitor the performance of the controller, identify any issues that arise, and take corrective action as necessary.
4. The development of the feedback controller must also consider the environmental impact of the system. The materials used in the MEMS-based accelerometer and AMD control system should be environmentally friendly, and the system's power consumption should be minimized. The system's life cycle should also be considered, including its end-of-life disposal.

4.4.2 Safety

1. the reliability and safety of the MEMS-based accelerometer and the AMD control system are of utmost importance. The controller must be designed to operate within safe limits, and the MEMS-based accelerometer must be accurately calibrated and tested for reliability. Failure to ensure reliability and safety can lead to catastrophic consequences, including loss of life and property damage.
2. the production of products needs a long time of welding work. We need to follow the laboratory

rules to improve the accuracy of products and avoid scalding of operators.
3. MEMS technology needs to be controlled by electricity. We should be careful about electric shock accidents caused by touching the wire connecting the sensor during the experiment.

References

[1] B. F. Spencer Jr., S. J. Dyke, and H. S. Deoskar, "Benchmark problems in structural control: Part I-active mass driver system," Earthquake Engng. Struct. Dyn., vol. 27, pp. 1127-1139, 1998. [Accessed: 24- Mar- 2023]

[2] Active Vibration Control in Three-Story Building. Available at https://www.mathworks.com/

Appendix

Table System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
A flexible wooden bamboo structure	Flexible enough	Y
Available shaker and senser with NI acquisition system	Meeting the functional requirements	Y
Practicable actuator	Mechanical functions: Enough power and mass Control software: Simulink & Arduino	Y
System Effectiveness	Reduce the vibration at least 10%	Y
Feedback Control	Closed loop control	Y
Cart Protection	Limitation of cart's position and velocity	Y