International Journal on Advanced Science Engineering Information Technology

Vol.10 (2020) No. 3
ISSN: 2088-5334

Design and Simulation of a Model Predictive Controller (MPC) for a Seismic Uniaxial Shake Table

Royce Val C. Malalis[a], Chyn Ira C. Crisostomo[a], Romel S. Saysay[a], Alexander C. Abad[b], Lessandro Estelito O. Garciano[c], Renann G. Baldovino[a,*]

[a] *Manufacturing Engineering and Management (MEM) Department, Gokongwei College of Engineering, De La Salle University, 2401 Taft Avenue, Malate, Manila, 1004 Metro Manila, Philippines*
Email: [*]renann.baldovino@dlsu.edu.ph

[b] *Electronics and Communications Engineering (ECE) Department, Gokongwei College of Engineering, De La Salle University, 2401 Taft Avenue, Malate, Manila, 1004 Metro Manila, Philippines*

[c] *Civil Engineering (CE) Department, Gokongwei College of Engineering, De La Salle University, 2401 Taft Avenue, Malate, Manila, 1004 Metro Manila, Philippines*

***Abstract*—Shake table is one of the apparatus that aids in researches to generate techniques, structural developments, and strategies to prevent, prepare, and minimize an earthquake's devastating effects. One important factor that should be considered in a shake table is the system dynamics due to control-structural interactions, which could either be linear or non-linear. To accurately model both has always been the challenge but becomes more plausible with the availability of faster hardware and computers and the continuous decrease in latency. Model Predictive Controller (MPC) is a type of controller extensively used in the industry that can be used on linear and non-linear systems. This study presents the design and simulation of an MPC for a uniaxial shake table intending to analyze the system's behavior and accuracy. MATLAB Simulink was utilized to handle the simulation analysis of the controller. Different MPC parameters such as sample time, prediction horizon, control horizon, and closed-loop performance were manipulated and adjusted to observe their effects on the output of the system. A signal that mimics the actual earthquake data was inputted into the controller, and the system's behavior and outputs were measured and presented through graphical representations. To determine the accuracy of the system's output, its relationship with the reference signal was compared. From the simulation produced, the system demonstrated high accuracy levels and could be adjusted depending on the set performance aggressiveness of the system.**

***Keywords*—predictive controller model; uniaxial shake table; MATLAB Simulink.**

I. INTRODUCTION

An earthquake, defined by the Philippine Institute of Volcanology and Seismology (PHIVOLCS), is a "weak to violent shaking of the ground produced by the sudden movement of rock materials below the earth's surface" [1]. Almost 90% of the world's earthquake occurred at a "horseshoe-shaped" region or border, located at the Pacific Ocean, called the Ring of Fire, or the Circum-Pacific Belt [2]. Earthquakes, however, are randomly occurring events and are independent of one another [3]. The Philippines is one of the countries that lie along this border. Every day, PHIVOLCS can record earthquakes, but most are at low intensities [4], so they are unlikely to be felt by people [5]. But occasionally, earthquakes of high intensities could occur. Recently, the country has been stunned by news reports, which alerts everyone for a 7.2 magnitude earthquake that could happen anytime—a devastating event that could destroy almost half of the country's buildings and could kill around 40,000 people [6].

Earthquakes with high intensities and/or magnitudes, in general, are devastating. Because of this, different private and government institutions worldwide are conducting research to generate techniques, structural developments, and strategies to prevent, prepare, and minimize, for such effects. One apparatus that aids in this kind of researches are shake tables.

Shake tables, literally, are a table or platform that is constructed to simulate an earthquake and study its effects on objects, structures, and people. An earthquake's performance could be verified by simulating the shaking of the ground [7]. One of the first applications for shake tables is for the improvement of building structures. Because of this, professionals and experts in this field were able to generate new standards, techniques, and tests, which ensures a building's integrity [8]. With an excellent building structure, it should be able to withstand earthquakes at absolute magnitudes. The next factors to consider then are the objects, machines, and people inside the building.

A good example of this is hospitals. Hospitals are full of equipment from oxygen tanks, storage cabinets, hospital beds, cutting tools, refrigerators, and many more, which are used to treat their patients. This equipment, however, are not all stationary or fixed around the hospital—prone to movement (sliding, falling, rolling, etc.). In an earthquake event, they could be considered potential threats that could endanger the lives of the patients. The importance of providing emphasis to the proper fixation and placement of such equipment around the hospital is highlighted during this scenario. This would require a study that analyzes the behavior of such equipment during an earthquake through an earthquake simulator capable of simulating earthquakes of different intensities, such as a shake table.

One important factor that should be considered in a shake table is the system dynamics due to control-structural interactions. A system's behavior could be linear or non-linear. Conventional controls commonly control linear systems. During an earthquake, the shaking of the ground and certain objects or people exhibits a nonlinear behavior. It would be hard for conventional controls to model a system as it becomes highly nonlinear [9].

Different controls such as PID controller, sliding mode control, minimal control synthesis, fuzzy-logic, and hybrids have been used to develop shake tables. Each controller provides its own share of advantages for a shake table's control system but suffers from disadvantages such as performance, complexity, and other undesired complications that either is unique or common to each controller.

Performance, accuracy, and adaptability are important factors that need to be considered when using or developing a control system. This research wanted to look for a controller that demonstrates potential along with these factors while still containing good features found on other controllers, which led to the usage of Model Predictive Control (MPC).

MPC has already been around and used in the industry for about 30 years now but is commonly used in linear systems due to its demanding computational process. This, however, with the availability of faster hardware and computers, and the continuous decrease in latency, even computationally demanding control systems capable of accurately modeling both linear and non-linear systems, such as MPC, becomes highly practical. Because of this, the study aims to design and simulate, through MATLAB Simulink, a model predictive controller (MPC) for a seismic uniaxial shake table.

II. Material and Method

A. Model Predictive Control

MPC is a "feedback control algorithm that uses a model to make predictions about future outputs of a process" [10]. As shown in Figure 1, inside an MPC are the plant model and optimizer. The optimizer is the one that provides optimal control strategies, which makes the predictive plant outputs closer to the desired objective or reference.

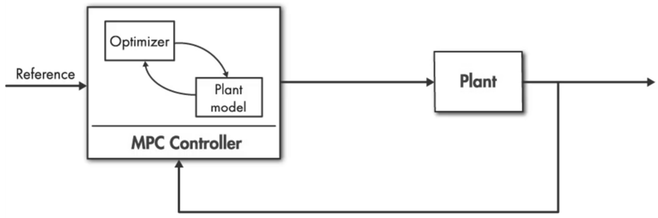

Fig. 1 A block model representation of MPC [11]

The prediction horizon ***p*** is the length of time or the number of times steps the controller could see into the future. The time steps are denoted by the k, k+1, k+2, etc. The green line in the graph (see Fig. 2) is the reference—the desired objective. As the MPC investigates the future for ***p*** time steps, it uses the model to simulate the future outputs or path of the system.

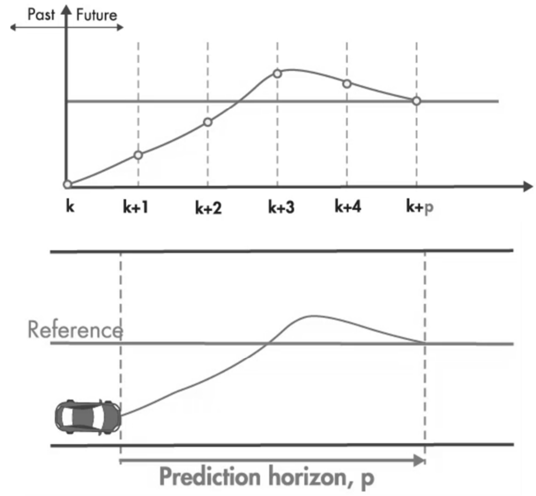

Fig. 2 Prediction horizon [11]

For the optimizer to ensure that the predicted path comes closer to the reference or desired objective, through its control strategies, it minimizes the error ***e*** between the predicted path of the system and the reference (see Fig. 3).

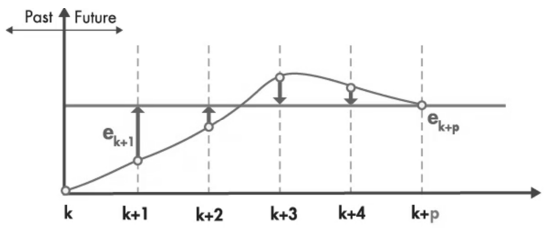

Fig. 3 The error between the reference and predicted path [11]

To summarize, at every time step, the MPC calculates the optimal solutions or paths or control strategy that would make the system's output come closer to the desired objective or reference, given a certain prediction horizon. However, going to the next time step, the prediction horizon also shifts, and the controller recalculates the optimal control strategy based on the future behavior of the system and reference that it could see and simulate. A more detailed explanation and examples of the concept of MPC can be viewed from MathWorks [12].

Since the 1980s, MPC has been used in the controls industry [10]. At first, due to its computationally demanding optimization process, it was more commonly used for linear systems [13]. However, with the computing power of microprocessors significantly increasing, its use has spread to other fields [10] and it is becoming feasible to run even in nonlinear systems [13].

Its major advantages lie on its multi-input multi-output (MIMO) control ability, constraints handling, and the capability to preview the future. MPC is a multivariable controller which can guide the outputs while considering the interactions of all the system's variables. Even though this feature has not been utilized yet in the simulation, upon the integration of the controls to the actual hardware, additional variables and constraints could arise, making good use of such features.

Talking about constraints, with MPC, any limitations on the values of the inputs and the outputs could easily be applied. With earthquakes being a nonlinear and inconsistent system, the capability of MPC to investigate the future and adjust its control strategy based on what it saw, improves its overall performance of the controller.

B. MPC Design

To properly design the MPC, it is important to identify the parameters, e.g., controller sample time, prediction and control horizons, weights, and constraints, that could affect the complexity of the controller's algorithm and its performance. These parameters are present when using MPC.

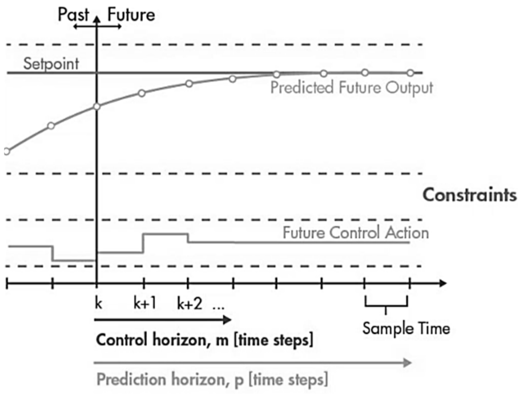

Fig. 4 A graphical representation of MPC [14]

The sample time is the rate at which the controller executes the control algorithm. A small sample time means the control system can react faster to changes, but its computational load will be very heavy. On the other hand, too big of a sample time makes the controller less reactive to changes and disturbances. It was recommended that the sample time should be around 1/10 to 1/20 of the open-loop system response rise time [15].

For the prediction horizon, it is recommended that it could cover 20 to 30 samples of the open-loop transient system response [15]. Too short of a prediction horizon makes the system to react late on changes while too long would generate a lot of wasted computations.

Meanwhile, the control horizon is the number of time steps where the controller could execute control actions. If the control horizon is too small, the computations would also be fewer, but it might also mean a suboptimal maneuver. On the contrary, too big of a control horizon value increases its computational complexity. The recommendation has it around 10% to 20% of the prediction horizon, with a minimum of 2-3 steps [15].

For the constraints, it is recommended that the outputs be set as soft, and the input and outputs, at the same time, should not be both sets as hard constraints [15]. It should be noted that the weights are relative to whatever is more important among the inputs and the outputs based on what is needed in the study or operation.

C. MPC Simulation

The simulation of the MPC was done through MATLAB Simulink. It is a simulation and model-based design environment offered by MATLAB to implement system models or designs without the need for coding them one by one. For this study, the generated simulation MPC block model is shown in the figure above (see Fig. 5).

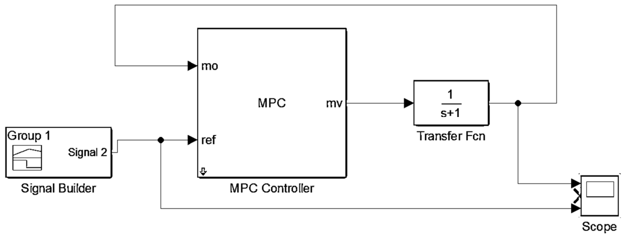

Fig. 5 Simulink MPC simulation block diagram

The model is composed of four parts, the signal builder, which generates the reference signals, the MPC controller, the plant transfer function, and the scope, which provided the graphical representations for the simulation.

In the actual data (see Fig. 6), the ground's acceleration (cm/s^2), in three different axes, namely east-to-west (EW), north-to-south (NS), and up-to-down (UD), were recorded at intervals of 0.01s. Since the study is limited only to a uniaxial shake table, only one of either the EW or NS data column could be used.

Time (sec)	EW (cm/s2	NS (cm/s2	UD (cm/s2)
0.01	0.416027	-0.5584	-12.7566
0.02	0.418071	-0.5622	-9.93244
0.03	0.445817	-0.5476	3.236374
0.04	0.448445	-0.54877	16.49718
0.05	0.439975	-0.52219	22.60728
0.06	0.455162	-0.53504	18.1032
0.07	0.457791	-0.5327	4.496596
0.08	0.434426	-0.57155	-5.59044
0.09	0.444064	-0.55373	-9.23062
0.1	0.45341	-0.56746	-2.72391
0.11	0.457207	-0.55198	-0.26509
0.12	0.45341	-0.54789	-7.36935
0.13	0.449321	-0.54672	-26.9958
0.14	0.456038	-0.55461	-47.4219
0.15	0.446693	-0.57038	-66.0087
0.16	0.443772	-0.57271	-73.0201
0.17	0.421868	-0.53562	-69.4859
0.18	0.415151	-0.51693	-61.6573
0.19	0.418363	-0.5292	-59.0525
0.2	0.415151	-0.53971	-65.0043
0.21	0.420116	-0.55724	-74.0122
0.22	0.409602	-0.52891	-75.0662
0.23	0.423036	-0.55169	-65.2417
0.24	0.412814	-0.51401	-46.3378
0.25	0.425665	-0.51956	-34.0571
0.26	0.423036	-0.5181	-38.2166

Fig. 6 Actual earthquake data in Surigao (PHIVOLCS data)

As of the current data set, however, it shows less varied data, which could still be used to show a more varied reference. The signal builder was used to generate a simple noise signal in a 2s period. In this case, a sample Gaussian noise was generated, as shown in Figure 7.

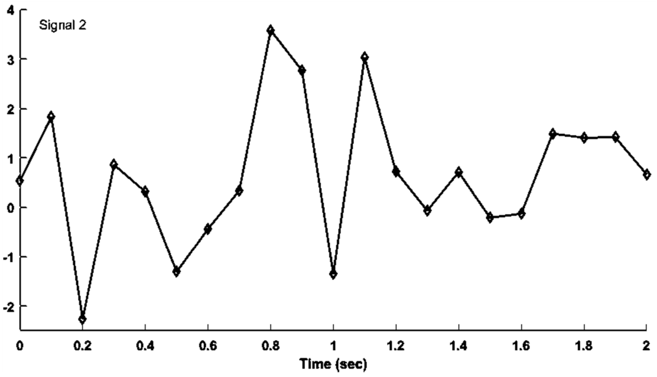

Fig. 7 Signal builder

Any earthquake data could be used for the system, but to highlight how effective the model is, a more random or nonlinear reference was applied. This is to demonstrate and used the concept of dynamics to at least mimic somehow the earthquake events.

As showcase in Figure 8 for the MPC designer block, both the parameters and the characteristics of the MPC were set. In this case, the sample time was set to 0.01s, the prediction horizon at 10, the control horizon at 2, and the performance of the MPC was balanced between robust and aggressive. Meanwhile, the state estimation was set to the middle.

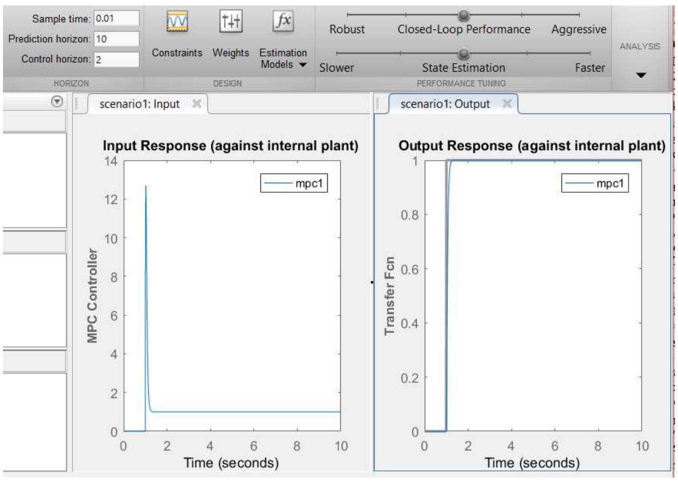

Fig. 8 MPC Designer

For the MATLAB simulation, the transfer function used in the experiment is *1/(s+1)*. Upon integration with the actual shake table, however, the transfer function of the shake table itself should be derived first and used. Through the MATLAB scope, as depicted in Figure 9, a graphical comparison between the reference signal and the output path of the system can be observed. In its current settings, the proposed system with the application of MPC was able to track the given reference signal, which is the actual earthquake data.

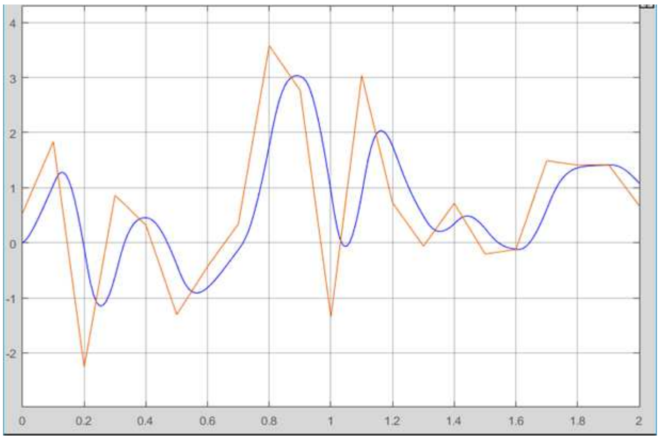

Fig. 9. Graphical comparison of the reference and the system's output.

III. Results And Discussion

A. Model Analysis

Different MPC parameters such as sample time, prediction horizon, control horizon, and closed-loop performance were manipulated and adjusted to observe their effects to the output of the system. This process could be similar or analogous to the tuning process conducted for a conventional PID control.

Based on the results, manipulating the prediction horizon and the control horizon did not show any significant changes to the system's output. This could be because the system used is not that complex. However, once the system gets more complicated, the two parameters will generate a significant impact on the system. On the other hand, manipulating the sample time and the MPC's close-loop performance gave significant changes to the system's output.

As observed in Figure 10, with the constant prediction horizon set at 10, the control horizon at 2, and the MPC performance is balanced between robust and aggressive, the sample time was varied into 0.001s, 0.01s, and 0.1s. It is very important to consider that the signal produced or generated by the signal builder is at an interval of 0.01s to mimic the actual earthquake data.

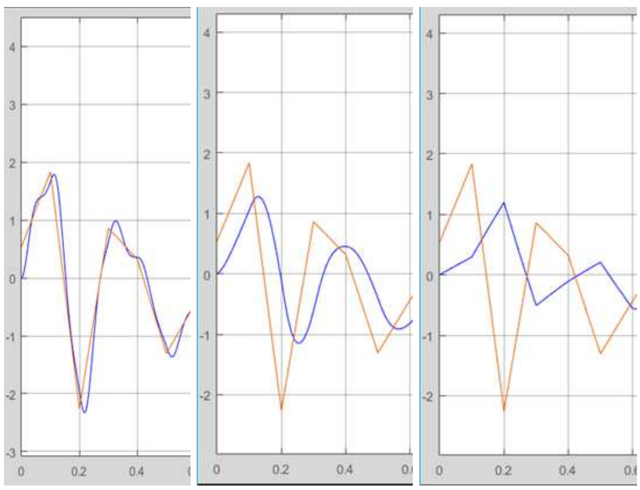

Fig. 10. Sample time (in s) at 0.001 (left), 0.01 (middle) and 0.1 (right)

Among the three different variations used for the sample time, setting the time at 0.001s, provided an output closest to the reference while the sample time at 0.1s, provided the farthest. For the 0.001s sample time, such setup requires a very fast computing power from the hardware. For the 0.1s sample time on the other hand, the controller had a more considerable sample time than the 0.01s signal interval of the reference. This caused a delay in the response of the system as it tries to close its path to the reference.

Another parameter that gave a significant change in the output path of the system is the closed-loop performance of the MPC. Maintaining the initial values of the other parameters, the MPC performance can be varied from robust, semi-robust, balanced, semi-aggressive, and aggressive. Figure 11 illustrates the generated system outputs as the MPC's close-loop performance was manipulated from robust to aggressive.

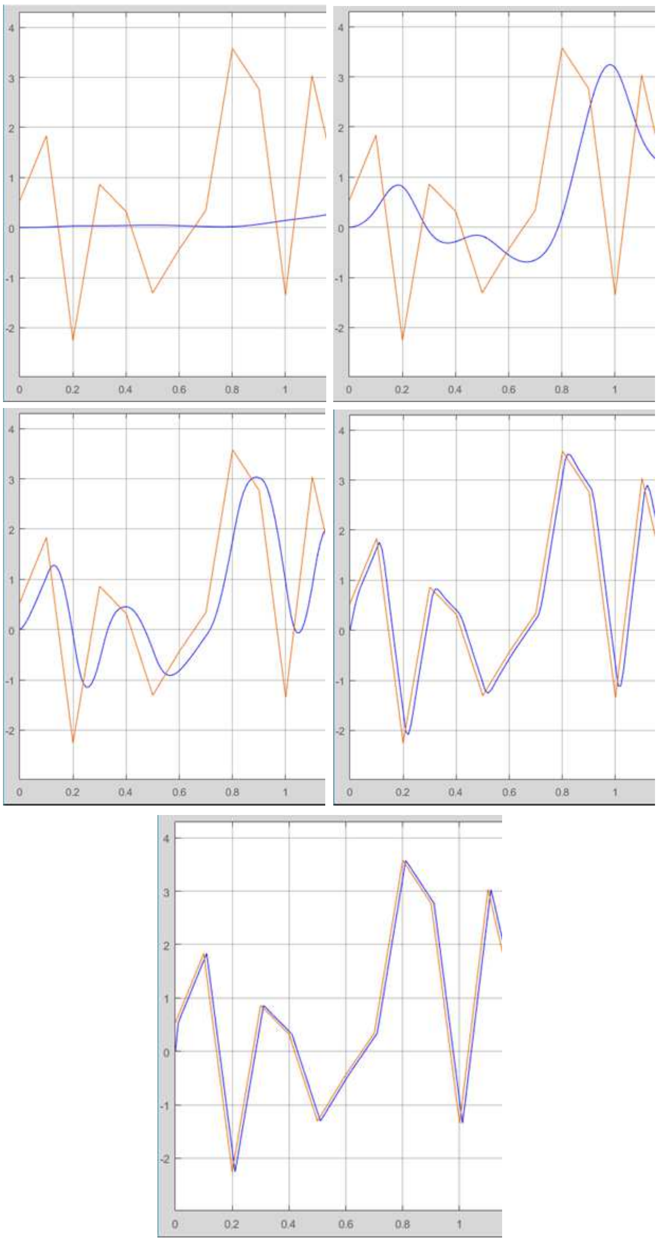

Fig. 11 Robust (upper left), semi-robust (upper right), balanced (middle left), semi-aggressive (middle right), and aggressive (bottommost)

As the system becomes more robust, the output of the system becomes less reactive, thus, having a path farther than the reference. On the contrary, as the system becomes more aggressive, the system becomes super reactive to a point where the output becomes nearly similar to the reference. This suggests that having an aggressive system would minimize the error between the reference and the system, the most. However, as the performance of the controller becomes more aggressive, it would require faster computational power from the hardware. Thus, the maximum controller performance is limited by the hardware being used.

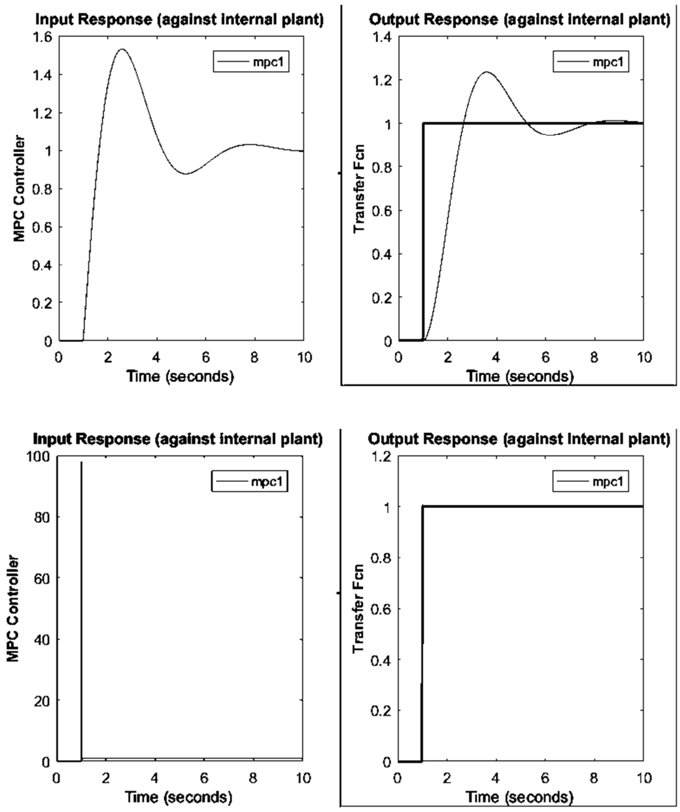

Fig. 12 Input and Output response. Robust (left) and Aggressive (Right)

Furthermore, the performance is also limited by the input and output response, which could go to very high values, unattainable by the current system, when an aggressive performance is set.

IV. Conclusion

In conclusion, the design and simulation of a model predictive controller for a uniaxial shake table was demonstrated successfully. The controller can show high levels of accuracy, depending on the set performance aggressiveness, when trying to follow the reference signal. This is favorable because it signifies that the system error between the actual intensity of the earthquake and the generated intensity based on the output of the system is small or minimized.

To further the study, the researchers have the following recommendations:

- The simulation results or MPC simulation be compared to other controllers by applying all to varying systems and compare the results, while maintaining similar hardware, software, and environment, for each controller, when testing on a similar system.

- The MPC be applied to a shake table and test its overall applicability and performance.
- The MPC is subject to different scenarios or systems, considering different variations of its manipulated variables to identify its limitations and optimal performance.

Acknowledgment

The authors would like to extend and express their sincerest gratitude and appreciation to the University Research Coordination Office (URCO) of the De La Salle University (DLSU) for the research funding and the Engineering Research and Development for Technology (ERDT) of the Department of Science and Technology (DOST) for the dissemination support.

References

[1] PHIVOLCS, "Introduction to Earthquake," DOST, 2019. [Online]. Available: https://www.phivolcs.dost.gov.ph/index.php/earthquake/introduction-to-earthquake.

[2] USGS, "Ring of Fire," 2019. [Online]. Available: https://earthquake.usgs.gov/learn/glossary/?term=Ring%20of%20Fire.

[3] K. Hickok, "Tons of Major Quakes Have Rattled the World Recently. Does That Mean Anything?," Live Science, 2018. [Online]. Available: https://www.livescience.com/63412-california-big-quake.html.

[4] PHIVOLCS, "Earthquake Information," DOST, 2019. [Online]. Available: https://www.phivolcs.dost.gov.ph/index.php/earthquake/earthquake-information3.

[5] PHIVOLCS, "PHIVOLCS Earthquake Intensity Scale (PEIS)," DOST, 2019. [Online]. Available: https://www.phivolcs.dost.gov.ph/index.php/earthquake/earthquake-intensity-scale.

[6] Rappler, "MAP: Strongest earthquakes in the Philippines," Rappler, 2015. [Online]. Available: https://www.rappler.com/science-nature/33807-map-strongest-earthquakes-in-ph.

[7] R. Baldovino and E. Dadios, "Design and Development of a Fuzzy-PLC for an Earthquake Simulator / Shake Table," 7th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM), pp. 1 - 6, 2014.

[8] ASTM, "New Building & Construction Standards," ASTM International, 2019. [Online]. Available: https://www.astm.org/industry/new-building-and-construction-standards.html.

[9] R. Langari, "Past, present and future of fuzzy control: a case for application of fuzzy logic in hierarchical control," Fuzzy Information Processing Society, 18th International Conference of the North American Source, pp. 760-765, 1999.

[10] M. Ulusoy, "Understanding Model Predictive Control, Part 1: Why Use MPC?," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/videos/understanding-model-predictive-control-part-1-why-use-mpc--1526484715269.html.

[11] M. Ulusoy, "Understanding Model Predictive Control, Part 2: What Is MPC?," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/videos/understanding-model-predictive-control-part-2-what-is-mpc--1528106359076.html.

[12] MathWorks, "Understanding Model Predictive Control," MathWorks, Inc., [Online]. Available: https://www.mathworks.com/videos/series/understanding-model-predictive-control.html.

[13] S. Brunton, "Model Predictive Control," Youtube, 2018. [Online]. Available: https://www.youtube.com/watch?v=YwodGM2eoy4&t=600s.

[14] M. Ulusoy, "Understanding Model Predictive Control, Part 6: How to Design an MPC Controller with Simulink and Model Predictive Control Toolbox," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/videos/understanding-model-predictive-control-part-6-how-to-design-an-mpc-controller-with-simulink-and-model-predictive-control-toolbox-1535693503432.html.

[15] M. Ulusoy, "Understanding Model Predictive Control, Part 3: MPC Design Parameters," MathWorks, 2019. [Online]. Available: https://www.mathworks.com/videos/understanding-model-predictive-control-part-3-mpc-design-parameters-1530607670393.html.