



## Development and Evaluation of an Economical Arduino-Based Uniaxial Shake Table for Earthquake and Wave Simulation

Mirza Dawood Baig<sup>1</sup>, Ahmed Murad Abdulrazzaq Saif<sup>1</sup>, Osinachi Mbah<sup>1</sup>, Umut Yildirim<sup>2</sup>,  
Görkem Ozankaya<sup>1</sup>, Qasim Zeeshan<sup>1\*</sup>

<sup>1</sup> Department of Mechanical Engineering, Eastern Mediterranean University, 99628 Famagusta, Turkey

<sup>2</sup> Department of Civil Engineering, Eastern Mediterranean University, 99628 Famagusta, Turkey

\* Correspondence: Qasim Zeeshan ([qasim.zeeshan@emu.edu.tr](mailto:qasim.zeeshan@emu.edu.tr))

Received: 04-09-2024

Revised: 05-13-2024

Accepted: 05-20-2024

**Citation:** M. D. Baig, A. M. A. Saif, O. Mbah, U. Yildirim, G. Ozankaya, and Q. Zeeshan, "Development and evaluation of an economical Arduino-based uniaxial shake table for earthquake and wave simulation," *J. Eng. Manag. Syst. Eng.*, vol. 3, no. 2, pp. 82–92, 2024. <https://doi.org/10.56578/jemse030203>.



© 2024 by the author(s). Published by Acadlore Publishing Services Limited, Hong Kong. This article is available for free download and can be reused and cited, provided that the original published version is credited, under the CC BY 4.0 license.

**Abstract:** In this study, an economical prototype of a uniaxial shake table, named the Eastern Mediterranean University (EMU) shake table, was developed using an Arduino platform for the simulation of sinusoidal waves and scaled earthquake data. The table incorporates a ball-screw mechanism actuated by a stepper motor. Simulations were conducted using sinusoidal signals and earthquake data for three distinct seismic events, recorded at discrete timestamps. The performance of the shake table was assessed by analyzing the discrepancies between the input signals and the table's outputs. In sinusoidal mode, a feedforward gain was computed to achieve the desired output amplitude values. Furthermore, a decreasing trend in the error between input and output acceleration values was observed. The table, without any payload, achieved an acceleration of 0.8 g at a frequency of 14.5 Hz and an amplitude of 1 mm. However, the effectiveness of earthquake simulations was constrained by the storage capacity of the Arduino Uno and the motor's performance capacity. Iterative methods were necessary for each earthquake simulation to determine the minimal timestep size that the motor could optimally handle. The methodology for simulating earthquakes was elaborated, identifying limitations and suggesting areas for future enhancement. The major constraints of the project were cost, time, and resource availability.

**Keywords:** Shake table; Arduino; Earthquake simulator; Seismic testing; SeismoMatch

### 1 Introduction

Due to the high mortality [1], and financial losses caused by earthquakes [2], it is essential to study the structural response under seismic loads to minimize catastrophes and rebuilding time, especially in the seismically risky regions, by ensuring that the response tests of structures confirm the expected performance based on the design methods [3]. In this regard, shake table testing has a wide scope and remains one of the most effective seismic model tests that helps examine foundation reactions, dynamic horizontal earth pressure, and soil-structure interaction. Shake tables (also known as earthquake simulators) consist primarily of a base upon which the specimen structure to be tested is installed. The motion of the base is then controlled by varying parameters such as displacement or accelerations in the x, y, or z directions (depending on the degrees of freedom (DOF)) based on the earthquake data [4]. Shake tables are categorized as small (smaller than 3 m × 3 m), medium (from 3 m × 3 m to 9 m × 9 m), and large (greater than 9 m × 9 m), which can include complex hydraulics and control systems and can incorporate motion in up to 6 DOF [5]. Despite its importance and necessity, shake table testing remains an expensive study due to the high cost of the equipment required. Even the smaller, one-axis shake tables cost around 30 thousand USD on average, while the options for open-source shake tables remain scarce [6].

The earliest efforts at constructing shake tables are attributed to the Japanese, followed by several researchers from the US and other parts of the world. The earliest designs incorporated uniaxial motion through purely mechanical means such as balancing wheels, springs, and reciprocating arm mechanisms. Since the 1960s, electric motors have become more widely used in shake table applications. Since the 2000s, most small-scale shake tables have utilized servo motors or hydraulic actuators depending on the required load capacity, while most large-scale shake tables utilize servo hydraulic actuators [7–11]. Optimizations in existing designs and the development of new designs

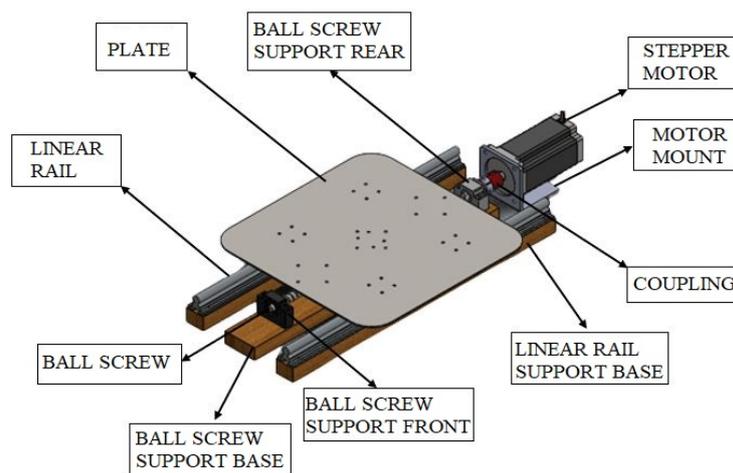
that could incorporate more DOF continued as the general objectives of shake table design remained to satisfy requirements, performance, system behavior, probabilistic reliability, hazard scenarios, and quality assurance [12].

Today, several different configurations exist depending on usage, and can include uniaxial, biaxial, and triaxial, as well as up to 6 DOF by incorporating pitch, yaw, and roll. Shake tables remain highly nonlinear devices, but with an increasing number of DOF, the complexity and cost rise [13]. Different actuation methods have been used in different kinds of shake tables, including electro-hydraulic, electromagnetic, and servo-hydraulic actuation [14]. Among other studies, a shake table has also been used to examine the behavior of rigid blocks and frames [4]. PID control systems have also been utilized in shake tables [8]. The literature discusses a wide range of applications for shake tables, including earthquake engineering, structural dynamics, and earthquake risk assessment. Shake table testing is also being used in nonstructural component testing, soil-structure interaction studies, and seismic retrofitting investigations [15].

A review of the literature showed that low-cost, open-source options for shake table testing remain scarce. SARSAR [16] is an open-source, low-cost uniaxial shake table based on an Arduino microcontroller. It utilizes a ball screw assembly, linear bearings, and a stepper motor. The SARSAR can reach a maximum speed of 0.35 m/s, an acceleration of 1.5 g, and a maximum payload of 200 kg. At 45,000 USD, CUSHAKE [17] is also an open-source uniaxial shake table that was built and tested in Turkey. The table can support a maximum payload of 1000 kg, utilizing a servo-driven AC motor, a rigid table, two rails, and four bearings. The Shakebot [6] is a simple setup based on Raspberry Pi and ROS. It utilizes 3D-printed components, a closed-loop stepper motor for actuation, and a tooth belt for transmission. The stepper motor enables the bed to reach a maximum horizontal acceleration of 1.2 g and a velocity of 0.5 m/s with a 2 kg specimen. TDG—Atom [18] is the most compact model being sold by TDG at 12,000 USD. It employs a servo motor in conjunction with a ball screw assembly, enabling it to linearly oscillate 50 kg of mass at 1g. Priced at 15,000 USD, the Quakelagic Test Box [19] is a biaxial shake table that utilizes a servo motor coupled with a ball screw assembly, capable of accelerating 100 kg of mass at g. The shake table built at the University of Engineering Technology, Pakistan [4] was custom-made and cost 3000 USD. It has a maximum payload capacity of 1220 kg and an operational frequency of 0–50 Hz. It utilizes a servomechanism that can produce an acceleration value of 1.9 g.

## 2 Design and Cost Analysis

The design of the EMU shake table was mostly inspired by the SARSAR [16], Quakelagic [19], and UET-CIU [4] shake table designs. The CAD isometric view of the prototype with the main components labeled is shown in Figure 1. The actuating mechanism for the EMU Shake Table prototype consisted of a ball screw, linear rails, and the ball screw nut. LiCHUAN-MC860H motor driver and LC86HS2801 stepper motor coupled with SFUR1610 ball screw were utilized for powering the system.



**Figure 1.** CAD isometric view of EMU shake table prototype

The prototype was to be tested under two operating modes: sinusoidal waves and scaled down earthquake simulation. The scaling down of earthquake data was done using the SeismoMatch 2024 software. Once the assembly was completed and the Arduino UNO controller was programmed, the prototype was ready to run. The schematic in Figure 2 illustrates the process by which the EMU—Shake Table functions. The bill of materials for the prototype has been shown in Table 1.

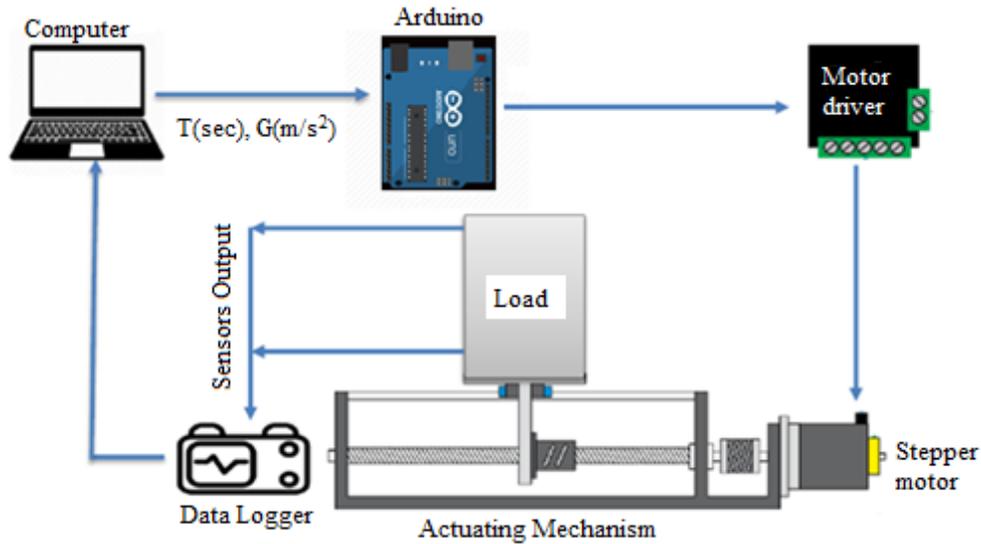


Figure 2. Functioning process of the EMU—shake table [4]

Table 1. Bill of materials

Item No.	Part Name	Model	Qty	Price (\$)	Shipping Cost (\$)	Total Cost (\$)
1	Ball screw	SFUR 1610	1	15	10	25
2	Ball screw nut	SFUR 1610	2	70	20	90
3	Ball screw nut housing	1610	1	10	14	24
4	Rear Ball screw support	BK12	1	6	9	15
5	Front Ball screw support	BF12	2	45	10	55
6	Linear Rails	SBR 16	5	20	5	25
7	Linear bearing	GS14	1	6	9	15
8	Coupling	Steel	1	10	N/A	10
9	Plate	Wood	2	10	N/A	20
10	Linear Rail Base Support	Wood	1	10	N/A	10
11	Ball Screw Base Support	015 – M4	16	5	N/A	5
12	Screws	015 – M5	42	5	N/A	5
13	Screws	015 – M6	10	5	N/A	5
14	Screws	Arduino	1	75	N/A	75
15	Arduino	MPU6050	1	2	5	7
16	Accelerometer	Limiting switch	2	1.5	N/A	3
17	Limiting switch	Stepper motor + Motor driver	1	134.62	N/A	135
18	Stepper motor + Motor driver	LC86HS2801 NEMA 34 (2 Phase Hybrid Motor) + MC860H	1	24	N/A	24
19	Step Down Transformer	50 V(AC)	1	3	2	5
20	Fuse	N/A	1	3	2	5
21	Machining cost	N/A	N/A	N/A	N/A	50
22	Cost of Operation and Testing					50
	<b>TOTAL</b>					<b>653</b>

### 3 Results and Discussion

#### 3.1 Tests and Results for Sinusoidal Waves

Initially, the assembly was tested to check for any mismatch between the input and the desired output values. To perform this test, the motor driver (MC860H) was switched to 400 steps/rev and full 6A current by turning on switches 4, 5, 6, 7, & 8, and by turning off switches 1, 2, & 3. The “move stepper” function within the AccelStepper library allows users to control the direction, distance, speed, and acceleration of the motor. Using this library, several acceleration values were fed at different amplitudes into the code (inputs) to check for the maximum

output acceleration values, the corresponding output amplitude values, and the maximum achievable frequencies. The acceleration values defined as inputs were in steps/s<sup>2</sup>, therefore the values in ‘g’ were calculated using this conversion factor:

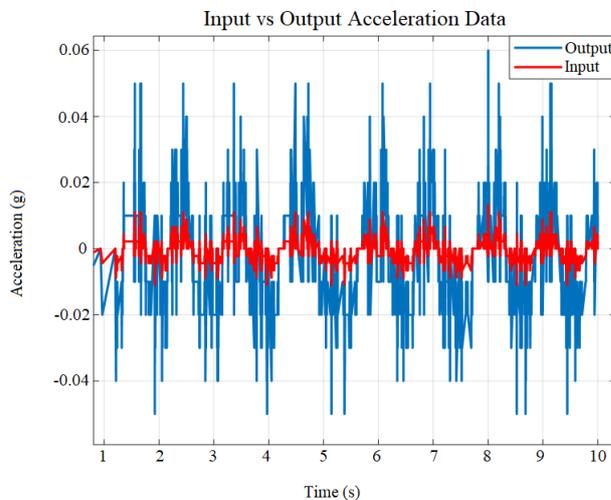
$$1 \frac{\text{step}}{\text{s}^2} = 2.7425 \times 10^{-6} \text{ g} \quad (1)$$

The desired input amplitude, in mm, was defined as steps in the code and compared to the output amplitude to obtain the error. To measure the output amplitude, two pens with a 0.5-mm-diameter tip were glued to the plate, and a wooden plank was placed perpendicular to the pens such that the pens sketched lines when the table oscillated back and forth. We then used a ruler to measure and record the lines. The acceleration was measured using an MPU6050 accelerometer (acceleration set at full scale range with a sensitivity factor of 16384), which was glued to the shake table surface and connected to a separate Arduino Uno. The experimental setup is shown in Figure 3.

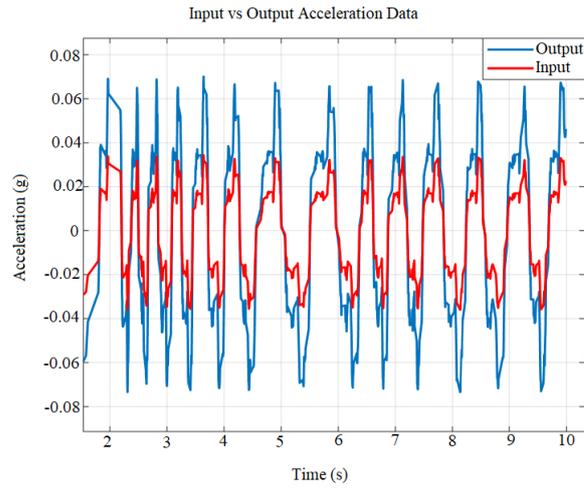


**Figure 3.** Preliminary experimental setup

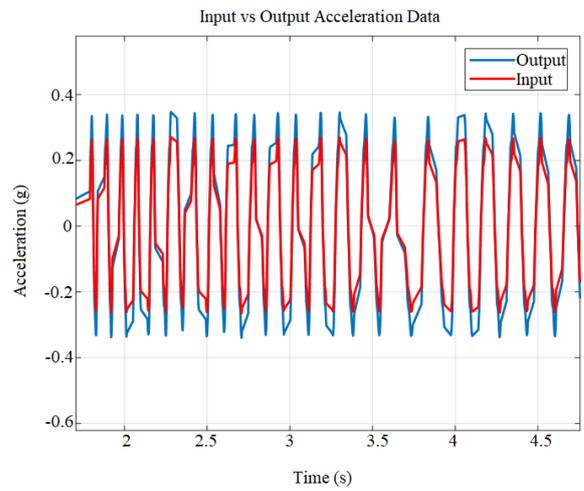
For a desired amplitude of 2mm, 80 steps were supplied to the stepper motor running on 400 steps/rev and full 6A current as set on the motor driver, and the output amplitude was 1mm. It was observed that the output amplitudes were less than expected; therefore, a feedforward gain of 2 was computed to correct the input amplitudes to obtain the desired output amplitudes. Furthermore, a significant overshoot was observed in the output acceleration values for small input acceleration values; however, the overshoot was observed to be significantly lower at higher acceleration input values. This can be observed in the graphs shown in Figure 4 as subgraph (a) of Figure 4 shows a significant overshoot with a ratio of input to output of 0.22; subgraph (b) of Figure 4 has a ratio of 0.49; subgraph (c) of Figure 4 has a ratio of 0.78; and subgraph (d) of Figure 4 has a ratio of 0.88 for an amplitude of 2mm. Figure 5 shows the corrected input amplitudes, input acceleration, and the ratios of input to output acceleration values.



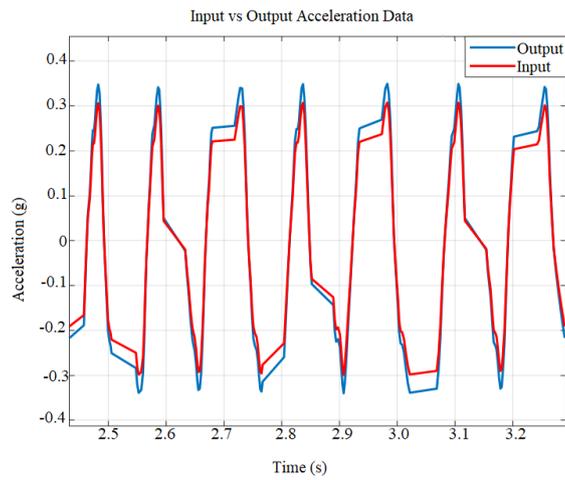
(a)



(b)



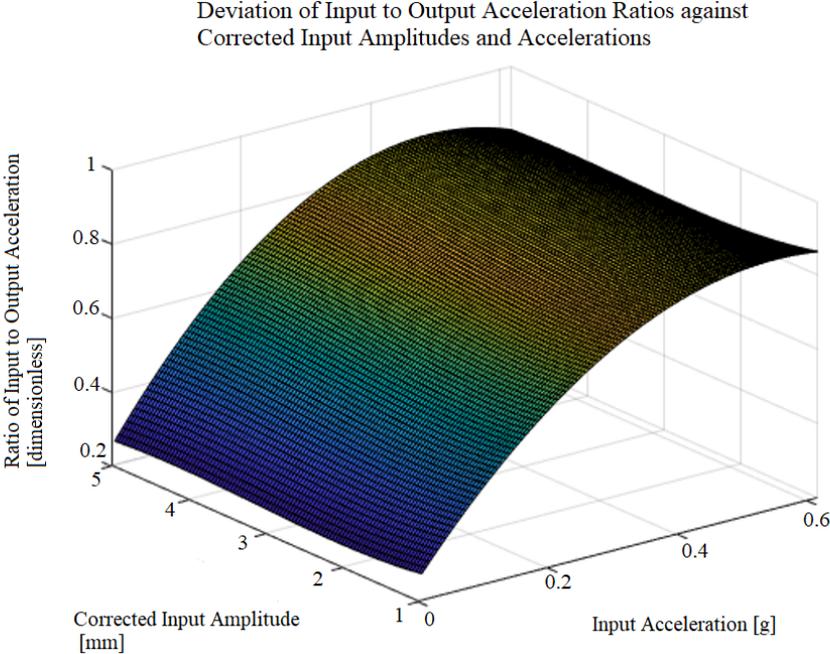
(c)



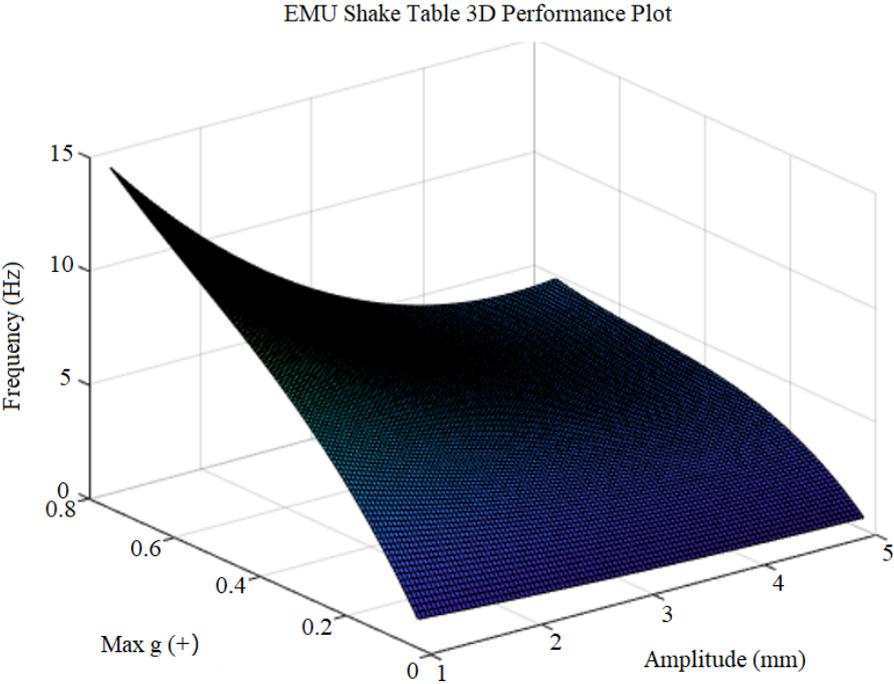
(d)

**Figure 4.** Input and output acceleration for amplitude of 2mm (80 steps): (a) acceleration at 4000 steps/s<sup>2</sup> (b) acceleration at 16000 steps/s<sup>2</sup> (c) acceleration at 128000 steps/s<sup>2</sup> (d) acceleration at 225000 steps/s<sup>2</sup>

The error between the input and output acceleration values continued to reduce as the input acceleration values went higher. This trend was observed to be true for all input amplitudes. The error was estimated to be 15 percent at high input accelerations. Based on the data from Figure 5, and assuming the magnitude of the acceleration values in positive and negative directions to be equal, a 3D surface was plotted using the Poly23 fit type in MATLAB, which helps visualize the overall performance of the EMU shake table in the sinusoidal mode as shown in Figure 6.



**Figure 5.** Ratio of input to output acceleration at different corrected input amplitudes and accelerations

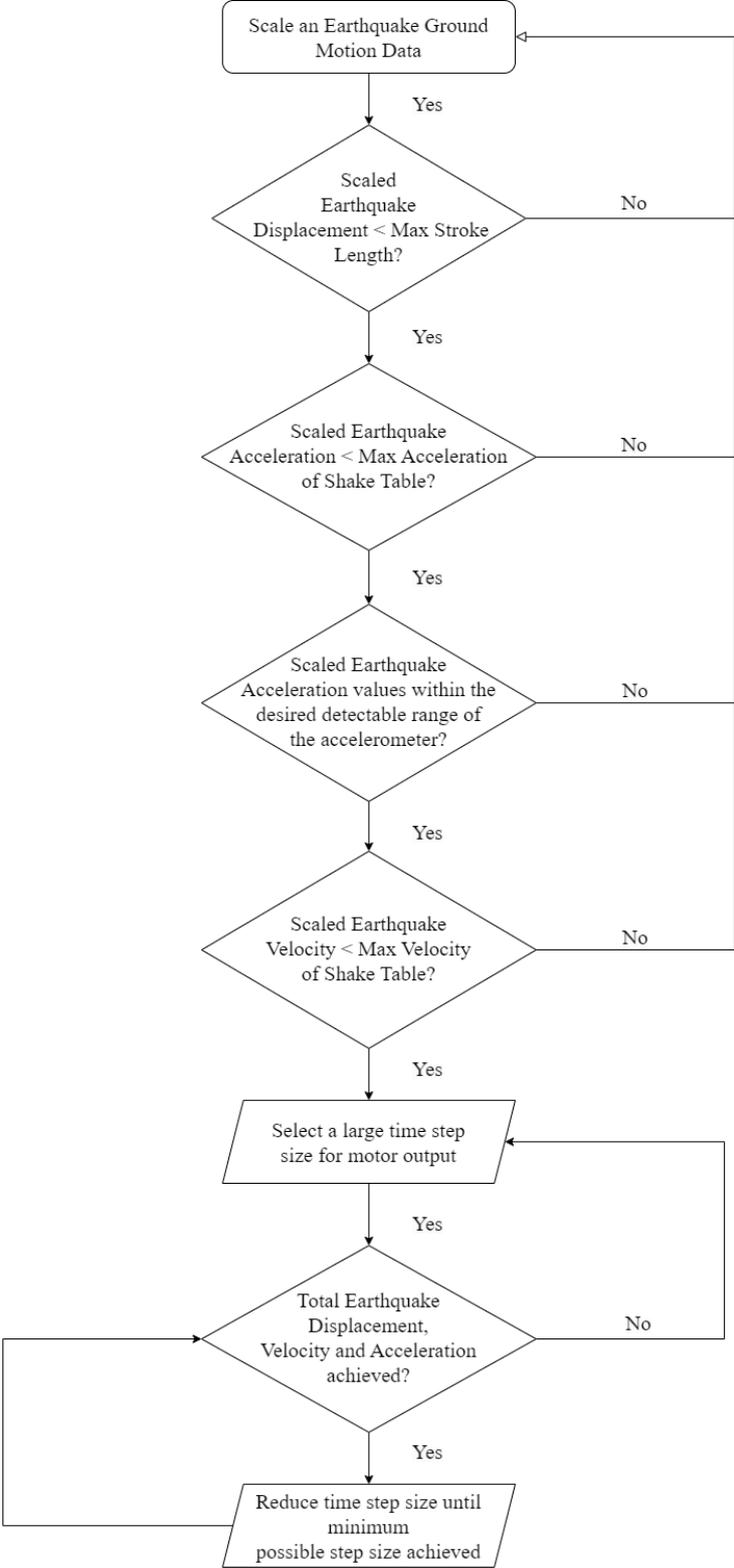


**Figure 6.** 3D Performance plot of EMU shake table

**3.2 Tests and Results for Earthquake Simulations**

To simulate an actual earthquake, it was crucial to obtain maximum performance of the motor with irregular and unpredictable displacement, velocity, and acceleration profiles.

**3.2.1 Methodology**



**Figure 7.** Process for earthquake simulation

To do this, the following methodology was employed:

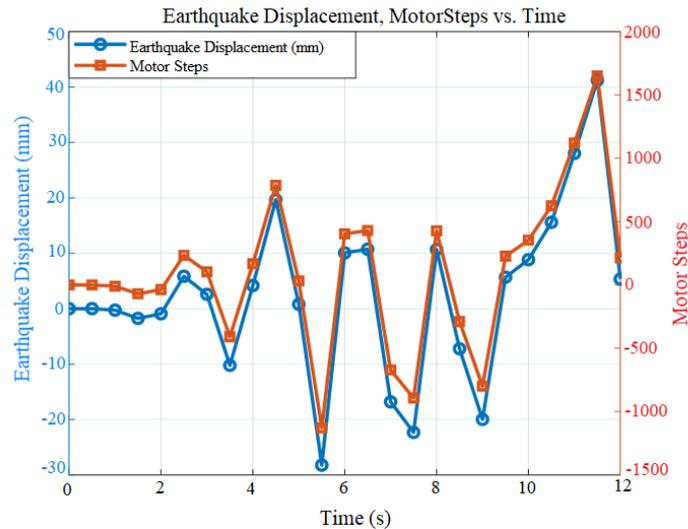
1. Earthquake data was obtained from SeismoMatch 2024 software, and scaled to ensure all displacement values were less than the maximum stroke length of the shake table.
2. With the selected earthquake profile, the maximum possible accelerations, and velocities that the motor was able to achieve over the selected overall time duration were checked iteratively:
  - a. A large number for velocity was fed into the code, and accelerations were altered until the values for the overall time taken were saturated.
  - b. Keeping the maximum acceleration constant, input velocities were altered until the values for the overall time taken saturated.
3. The smallest possible time step size at which the motor was able to completely perform the scaled earthquake displacements at the scaled earthquake velocities and accelerations was obtained iteratively. However, to account for latency, throughput, and external disturbances, a slightly larger time step size of 0.5 s was used.
4. To ensure some level of robustness, the Earthquake was scaled further down so that it did not exceed the motor's maximum acceleration and velocity capacities for the selected earthquake profile by a margin.
5. Scaled earthquake displacements were converted from cm to motor steps.
6. Scaled earthquake velocities were converted from  $\frac{\text{cm}}{\text{s}}$  to motor  $\frac{\text{steps}}{\text{s}}$ .
7. Scaled earthquake accelerations were converted from g to  $\frac{\text{steps}}{\text{s}^2}$ .
8. Time, displacement, velocity, and acceleration data were fed into the Arduino Code, and the output acceleration results were recorded using the MPU6050 module. Due to the limited data storage capacity of the Arduino board used, the earthquakes were simulated only for a total of 10 to 12 seconds.
9. The data obtained from the MPU6050 included noise, and the accelerometer could not detect acceleration values between the range of  $\pm 0.01$  g. Therefore, each output acceleration value was manually checked at the expected time stamp, after which the accelerogram data was smoothed.
10. RMSE was calculated using the relation:

$$RMSE = \sqrt{\frac{\sum (\text{Output Acceleration} - \text{Expected Acceleration})^2}{\text{Total Number of Time Stamps}}} \quad (2)$$

The flowchart shown in Figure 7 summarizes the process for simulating an earthquake.

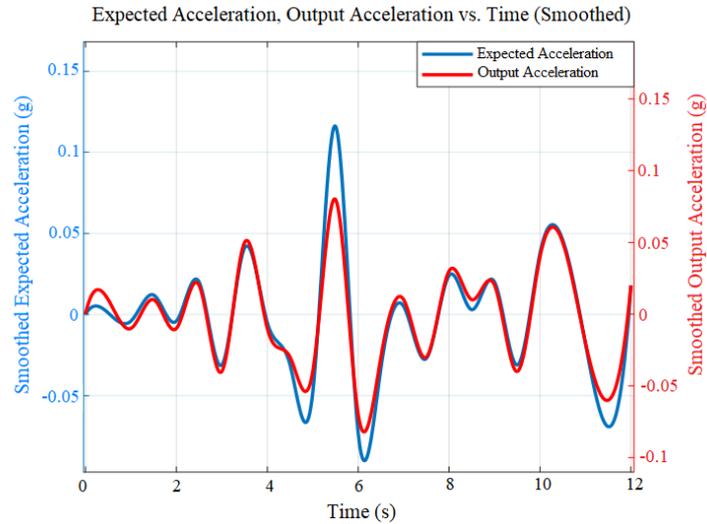
### 3.2.2 Results for scaled Kobe earthquake simulation

The selected earthquake was the Kobe 1995 earthquake at a scale factor of 0.45 and recorded at time intervals of 0.01s. The test initially failed due to the limited storage capacity of the Arduino UNO board. Therefore, the simulation was conducted at time intervals of 0.5s. The earthquake displacement converted to motor steps at different time stamps is presented in Figure 8. The displacements are not absolute, which is why the input steps share the same profile as the expected displacement.



**Figure 8.** Kobe earthquake displacements scaled at 0.45, and the equivalent input motor steps superimposed

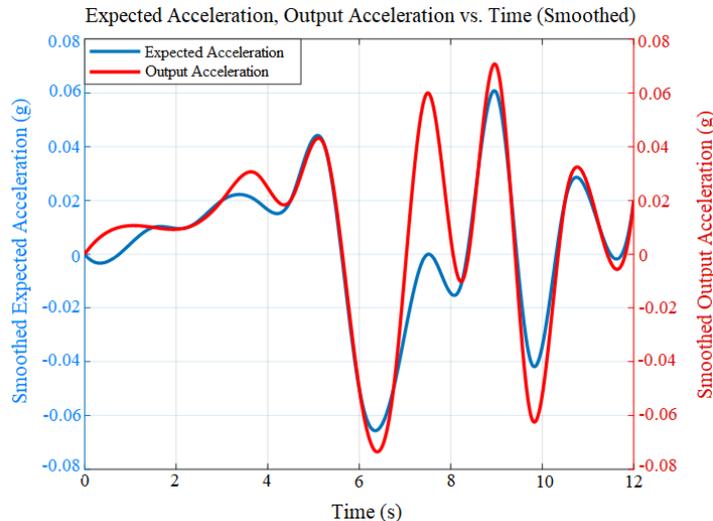
The actual and expected smoothed accelerograms have been shown in Figure 9. The significant mismatch between the expected and actual values at 0.5 s could be attributed to the accelerometer failing to detect accelerations in the range of  $\pm 0.01$  g.



**Figure 9.** Superimposed expected vs. output acceleration—time graphs, scaled Kobe earthquake

The intervals on which there is an undershoot for the actual acceleration values can be fixed by inputting a slightly higher acceleration value into the code until the expected acceleration meets the actual acceleration with an acceptable error margin. With the current results, an RMSE of 0.0091 was computed. The intervals on which there is an overshoot for the actual acceleration values need to be first checked with a more sensitive accelerometer, which would then discern if there is a need to alter the input accelerations to meet the desired values with acceptable error margins.

### 3.2.3 Results for Northridge earthquake simulation



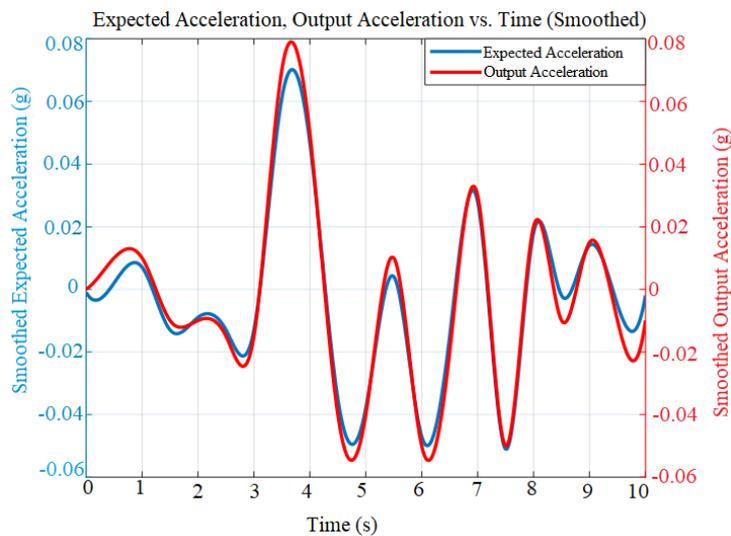
**Figure 10.** Superimposed expected vs. output acceleration—time graphs, scaled Northridge earthquake

The data for the Northridge earthquake was also obtained from SeismoMatch 2024, and then scaled down by a factor of 0.3. The raw data was then refined to ensure that the velocities were positive (negative) when the displacement curve was increasing (decreasing). Similarly, it was ensured that the accelerations were positive (negative) when the velocity curve was increasing (decreasing). Due to memory limitations, the data at a time interval of 0.75s was used, and the actual and expected smoothed accelerograms are shown in Figure 10. The RMSE of 0.017 was computed for the expected and actual accelerations.

### 3.2.4 Results for Friuli earthquake simulation

The same process was used in simulating the Friuli earthquake. The raw data was scaled down by a factor of 0.5. The raw data was then refined to ensure that the velocities were positive (negative) when the displacement curve was

increasing (decreasing). Similarly, it was ensured that the accelerations were positive (negative) when the velocity curve was increasing (decreasing). Due to memory limitations again, the data at a time interval of 0.5s was fed into the code to obtain the actual and expected smoothed accelerograms shown in Figure 11. RMSE of 0.0046 was obtained for the expected and actual acceleration.



**Figure 11.** Superimposed expected vs. output acceleration — time graphs, scaled Friuli earthquake

#### 4 Future Works and Conclusion

The project was successful in manufacturing a small-scale Arduino based shake table, proving the simplicity of simulating sinusoidal signals, however, simulating the scaled earthquake data was more complicated. One limitation was the low memory capacity of the Arduino board, which could not incorporate all the scaled earthquake data at smaller time stamps. The future works could be broadly categorized into three categories: design refinement, design optimization, and design modification.

1. Design refinement: different sensors could be added, and the values could be compared for more accuracy. One common problem that remained in the prototype was the misalignment of the motor shaft with the ball screw. This can be solved by incorporating a guiding mechanism for the motor instead of making manual adjustments before and after every operation. Using an Arduino with a higher storage capacity could allow simulating earthquakes with a smaller time step size. A motor with higher specifications should be used to prevent stalling.

2. Design optimization: Factors such as the distance between the linear bearings and its effects on the overall vibration of the plate, as well as the change in maximum stroke length due to the change in the distance between the linear bearings, can be studied to achieve an optimal value for the distance between the linear bearings.

3. Design modification: The prototype has the potential to be modified such that it could be mounted on another model rotated at 90 degrees to make the overall assembly a biaxial shake table. This could be Prototype 2. Upon completion, Prototype 2 could also go through design refinement, optimization, and modification processes until a triaxial shake table could be built.

4. The robustness of the system can be further analyzed.

#### Data Availability

The data used to support the research findings are available from the corresponding author upon request.

#### Conflicts of Interest

The authors declare no conflict of interest.

#### References

- [1] M. Naddaf, “Turkey–Syria earthquake: What scientists know,” *Nature*, vol. 614, no. 7948, pp. 398–399, 2023. <https://doi.org/10.1038/d41586-023-00364-y>
- [2] Federal Emergency Management Agency, “Hazardous Estimated Annualized Earthquake Losses for the United States: FEMA P-366,” [https://www.fema.gov/sites/default/files/documents/fema\\_p-366-hazardous-estimated-annualized-earthquake-losses-united-states.pdf](https://www.fema.gov/sites/default/files/documents/fema_p-366-hazardous-estimated-annualized-earthquake-losses-united-states.pdf), 2023.

- [3] K. Takanashi, "Recent achievements in substructuring on-line pseudodynamic tests at IIS," *Stab. Ductil. Steel Struct.*, pp. 141–152, 1998. <https://doi.org/10.1016/b978-008043320-2/50014-9>
- [4] A. Danish, N. Ahmad, and M. U. Salim, "Manufacturing and performance of an economical 1-D shake table," *Civ. Eng. J.*, vol. 5, no. 9, pp. 2019–2028, 2019.
- [5] P. D. Moncarz and H. Krawinkler, "Theory and application of experimental model analysis in earthquake engineering," *Rep. - Stanford Univ. John A. Blume Earthq. Eng. Cent.*, 1981.
- [6] Z. Chen, D. Keating, Y. Shethwala, A. A. P. Saravanakumaran, R. Arrowsmith, A. Kottke, C. Wittich, and J. Das, "Shakebot: A low-cost, open-source robotic shake table for earthquake research and education," *arXiv preprint*, 2022. <https://doi.org/10.48550/arXiv.2212.10763>
- [7] R. Wood, "Robert Mallet and John Milne—Earthquakes incorporated in Victorian Britain," *Earthq. Eng. Struct. Dyn.*, vol. 17, no. 1, pp. 107–142, 1988. <https://doi.org/10.1002/eqe.4290170103>
- [8] R. T. Severn, "The development of shaking tables-A historical note," *Earthq. Eng. Struct. Dyn.*, vol. 40, no. 2, pp. 195–213, 2011. <https://doi.org/10.1002/eqe.1015>
- [9] S. Street, "Historic developments in the evolution of consortium of universities for research in earthquake engineering," 1998.
- [10] R. Reitherman, "The effects of the 1906 earthquake in California on research and education," *Earthq. Spectra*, vol. 22, no. 2, 2006. <https://doi.org/10.1193/1.2187053>
- [11] R. T. Severn, "The contribution of shaking tables to early developments in earthquake engineering," *Geotech. Geol. Earthq. Eng.*, vol. 17, pp. 383–409, 2010. [https://doi.org/10.1007/978-90-481-9544-2\\_16](https://doi.org/10.1007/978-90-481-9544-2_16)
- [12] R. T. Duarte, M. Ritto-Corrêa, C. T. Vaz, and A. Campos-Costa, "Shaking table testing of structures," in *Earthquake Engineering, Tenth World Conference*, 1994, pp. 6837–6846.
- [13] O. Ozelcik, J. P. Conte, and J. E. Luco, "Comprehensive mechanics-based virtual model of NHERI@UCSD shake table—Uniaxial configuration and bare table condition," *Earthq. Eng. Struct. Dyn.*, vol. 50, no. 12, pp. 3288–3310, 2021. <https://doi.org/10.1002/eqe.3510>
- [14] N. A. I. E. Putra, R. Syam, I. Renreng, T. Harianto, and N. R. Wibowo, "The development of earthquake simulator," *EPI Int. J. Eng.*, vol. 4, no. 2, pp. 134–139, 2021. <https://doi.org/10.25042/epi-ije.082021.05>
- [15] T. T. Wang, O. L. A. Kwok, and F. S. Jeng, "Seismic response of tunnels revealed in two decades following the 1999 Chi-Chi earthquake (Mw 7.6) in Taiwan: A review," *Eng. Geol.*, vol. 287, no. 2, p. 106090, 2021. <https://doi.org/10.1016/j.enggeo.2021.106090>
- [16] E. Damc, "Development of a low-cost single-axis shake table based on arduino," *Exp. Tech.*, vol. 43, pp. 179–198, 2019. <https://doi.org/10.1002/et.21846>
- [17] T. Baran, A. K. Tanrikulu, C. Dundar, and A. H. Tanrikulu, "Construction and performance test of a low-cost shake table," *Exp. Tech.*, vol. 35, no. 4, pp. 8–16, 2011. <https://doi.org/10.1111/j.1747-1567.2010.00631.x>
- [18] T. D. Grubu, "Tdg-shaketable atom," <https://tdg.com.tr/en/products/desktop-shake-tables/tdg-shaketable-atom>.
- [19] QuakeLogic, "Bi-axial-tdg shake table," [https://www.quakelogic.net/\\_small-scale-shaketables/shaketable-bi-axial](https://www.quakelogic.net/_small-scale-shaketables/shaketable-bi-axial).