



Namazu: Low-Cost Tunable Shaking Table for Vibration Experiments Under Generic Signals

J. Grashorn^{1,2} · M. Bittner^{1,2} · M. Banse³ · X. Chang⁴ · M. Beer^{1,2,5,6} · A. Fau^{2,4}

Received: 19 January 2024 / Accepted: 2 June 2024
© The Author(s) 2024

Abstract

This article presents Namazu, a low-cost tunable shaking table framework for uniaxial vibration experiments in engineering education and research. All components and corresponding assembly are detailed. The design is easy to use and requires minimum maintenance. Open-source software covering signal generation and microcontroller programming is proposed to prescribe the motion of the table. There is no restriction in the programming language used to interface with the table. Communication with the microcontroller is performed via a serial interface, which eliminates the need for additional software. Besides, any displacement signals, including random ones, can be chosen. Due to the open-source nature of the Namazu table, users can also implement custom methods for signal generation and modify the table hardware. Suggestions are given in the paper. Accuracy is analyzed through displacement measurements. In addition, the Shinozuka benchmark is proposed and applied to test the table accuracy in the frequency domain. The results show good consistency of the signals obtained with the setpoints. Thus, Namazu, including the shaking table and a software suite, offers a versatile, accessible, and accurate solution for vibration experiments.

Keywords Experimental · Vibration · Stochastic dynamics · Shaking table · Earthquake engineering · Spectral representation method

Introduction

Vibrations are omnipresent, be it in the computer caused by a fan, in the car due to the road and the engine, in the airplane because of a turbo-fan engine and wind, or on the moon and earth due to seismic activity. Structures or machinery undergo vibrations of different magnitudes and durations. A specific topic of vibration analysis in structural engineering is the design of earthquake-resistant structures. Experiments are conducted to test designs or gather insights into dynamical and material behavior and structural responses under specific loads. Among the most practical testing setups, shaking tables are of great interest. On shaking tables complete buildings, load-bearing components, or smaller specimens with a particular function undergo dynamic single- or multi-axial displacements in a controlled environment [1]. Therefore, many shaking tables use large-scale devices that reach length

scales of several meters [2–7]. Several applications also allow for smaller scales or scaled tests. They include tests for sensing methods [8–10], material property calibrations [11, 12], structural health monitoring (SHM) [13–16] or general response analysis [17–19]. All of these research topics underline the importance and necessity of experimental analyses to validate methods and test hypotheses in structural engineering. However, large-scale vibration experiments need to be prepared carefully, are often costly, require a tight schedule, and thus come with a limitation on the number of experiments that can be performed. All the presented applications do not account for a stochastic signal generation embedded in a generalized uncertainty quantification framework, which is of high advantage when designing earthquake-resistant structures.

Some low-cost, small-scale shaking tables have been recently developed to minimize costs and raise the practicality of dynamic experiments to a larger audience, which may include research teams as well as teachers. Among small-scale low-cost shaking tables, [20] have proposed a shaking table with a bipolar stepper motor for actuation and Arduino-based hardware for control. The mountable payload

Jan Grashorn, Marius Bittner and Marvin Banse contributed equally to this work.

Extended author information available on the last page of the article

and achievable acceleration are noteworthy. However, the controller code must be changed based on the desired input signal. It needs more versatility and robustness for general test procedures. In [21], the open-source Robotic Operating System (ROS) controls a small shaking table with a feedback loop consisting of accelerometers and a camera for position tracking of the table. Even though the usage of ROS ensures high accuracy by, e.g., offering standardized means to utilize a feedback loop in the control of the shaking table motion, ROS is not particularly suited to implement algorithms for structural analysis. These small-scale shaking tables lack a generalized interface to standard programming languages, such as Python or MATLAB, used to develop algorithms and perform simulations. Additionally, the cost of both tables is still comparatively high at over 1000 USD each [21].

We propose a small-scale tunable shaking table embedded in a framework called Namazu, which offers versatility in terms of input signals, cost, and hardware. Namazu is specifically tailored for a seamless connection to other scientific frameworks used for structural simulations or stochastic analyses. Any arbitrary loading scenario can be directly uploaded to the controller by any programming language that offers a serial connection to microcontrollers. Most currently used languages and frameworks support serial ports, thereby providing an easy-going integration of Namazu. Additionally, there is no restriction on the used operating system. The present implementation uses MATLAB [22] to prescribe time-displacement signals and to control the table; examples for Python are also provided.

Details about the hardware and software components of the shaking table design are listed. Off-the-shelf components, such as a bipolar stepper motor and driver, apply the motion to the table. The table is scalable in dimension and performance quantities, such as achievable maximum acceleration and maximum speed. This implementation makes the presented shaking table adaptable for specific user needs. Different hardware layouts are discussed with advantages and disadvantages toward the mentioned quantities, such as achieved acceleration, speed, and possible payload. With the presented shaking table layout it was opted for reasonable accuracy under minimal costs.

The accuracy of the table motion with respect to desired motions is verified with linear variable differential transformer (LVDT) measurements. Additionally, the so-called Shinozuka benchmark is introduced [23]. The latter represents a formalized procedure to analyze the transfer of specific frequency components of stochastic signals to the shaking table motion. It is based on the well-known Spectral Representation Method (SRM) [24].

In the research field of stochastic dynamics and more specifically risk analysis of stochastic dynamic systems, large-scale experiments are usually not regarded in the

early stage of algorithmic developments. Developing them in close connection with experimental validation even at the early stages of the studies is beneficial, including performing dynamic tests for stochastic signals. A tunable small-scale shaking table enables researchers to quickly develop vibration experiments and validate some methods since the hardware is generally less complex than larger-scale tables. The small-scale experiment fits on a tabletop and is fast to assemble and disassemble whenever needed. Additionally, safety measures are less restrictive when working with small-scale specimens, and there is less requirement for specialized operator training. Algorithms targeted to analyze vibrations, i.e., those used in SHM or response analyses, can be tested and enhanced faster before applying them to large-scale tests.

The design files of the table as well as the control software implementation of the framework in MATLAB and Python are publicly available on Github and Zenodo [25]. Further documentation for the whole project is found at <https://www.namazu.de/>.

The whole flowchart of the Namazu framework is summarized in Fig. 1. The techniques for generating harmonic and random signals are summarized in Section “[Prescribing Complex and Random Signals](#)”, prescribing complex and random signals is the main goal of the software. The post-processing required for the safe usage of the shaking table is also explained. Section “[Namazu Hardware](#)” details the shaking table hardware (frame, motor, motor driver, and microcontroller specifications) and explains the technical choices. Variants for personalizing the shaking table for specific budget or experimental requirements are also suggested and commented on. Section “[Control System](#)” gives the key features of the control system i.e., the signal generation in a microcontroller readable script, serial interface to the microcontroller, and the microcontroller firmware itself. Sections “[Prescribing Complex and Random Signals](#)”, “[Namazu Hardware](#)” & “[Control System](#)” make up the core part of the Namazu framework, the relation between these parts, highlighting the flow of consecutive parts and their functions. All terms shown in Fig. 1 are explained in the referenced sections. Experimental results based on LVDT measurements are presented in Section “[Proof of Concept: Validation of Table Accuracy](#)”. Several signals including the Shinozuka benchmark are employed for evaluating the accuracy of the shaking table.

Prescribing Complex and Random Signals

Generating complex signals, particularly random signals, is necessary for earthquake engineering and risk simulations. We adapt the corresponding signal generation techniques

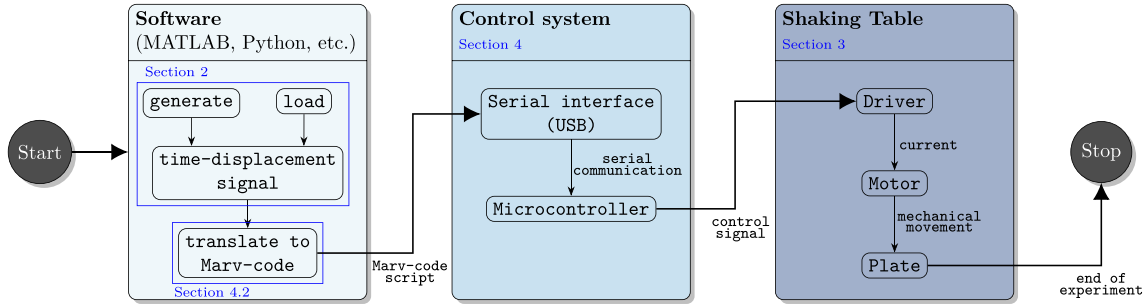


Fig. 1 Flowchart and design concept of the Namazu framework, from signal generation via the software over the control system to plate motion

[26–28] in order to preprocessing the data that will be then transferred to the shaking table (Section “Control System”). The goal is to prescribe complex and random motion to the table with a software including in Namazu. To seamlessly connect the data to the software architecture, signals will be discretized as a set of positions (or displacements considering the initial position is zero) for a set of equally discretized time instants. In this section, the continuous description of the signal as a function $x_g(t)$ is discussed for any time $t \in [0, T]$ with T the end time of the signal.

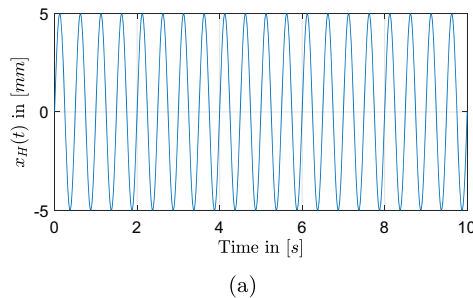
Generating Input Signals

Any arbitrary time-displacement signal could be of interest and used in the Namazu framework. Two classes of signals, namely harmonic and stationary stochastic signals, are presented hereafter.

Harmonic signals

First, harmonic signals are considered. They consist of the sum of frequency components and are closely related to Fourier series. Choosing a number n_Ω of frequencies, the position $x_H(t)$ is calculated as the finite sum

$$x_H(t) = \sum_{k=1}^{n_\Omega} \mathbb{A}_k \sin(2\pi \Omega_k t), \quad (1)$$



where $\Omega_i \in \mathbb{R}_{\geq 0}$ are the frequencies of interest and $\mathbb{A}_i \in \mathbb{R}_{\geq 0}$ the corresponding amplitude values with $i \in [1, n_\Omega]$. Thus, a fixed harmonic signal with a single frequency is depicted in Fig. 2(a), whereas a mixed harmonic signal with three superimposed frequency components is depicted in Fig. 2(b). These signals are deterministic.

Stationary stochastic signals

Another class of signals of interest in structural engineering consists of univariate stationary Gaussian stochastic processes. They can be described by the Spectral Representation Method (SRM) from a functional frequency content, the so-called Power Spectral Density (PSD) function. The SRM and its variations, such as the Stochastic Harmonic Function (SHF) representation [26], are commonly used to simulate artificial ground motion [27, 28].

Any PSD function can be used in the Namazu framework according to the user preferences. Among possible PSD functions, the Shinozuka and Deodatis frequency content [24] given by

$$S_{x_S}(\omega) = \frac{1}{4} \sigma^2 b^3 \omega^2 e^{-b|\omega|}, \quad -\infty < \omega < \infty, \quad (2)$$

is chosen. The constant σ corresponds to the standard deviation of the stochastic signal and b a parameter that models the correlation distance of the stochastic signals [23]. The val-

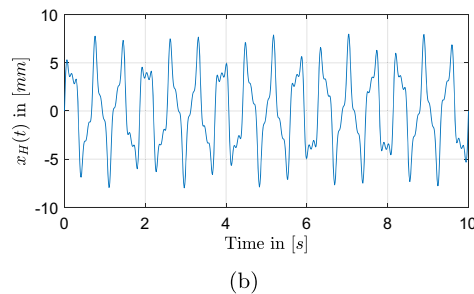


Fig. 2 Signals: (a) single harmonic signal with $n_\Omega = 1$, $\mathbb{A}_1 = 5$ mm, $\Omega_1 = 2$ Hz, (b) Mixed harmonic signal with $n_\Omega = 3$, $\mathbb{A} = \{5, 2, 1\}$ mm, $\Omega = \{1.6, 4.3, 7\}$ Hz

ues $\sigma = 1$ and $b = 1$ are chosen. From the reference PSD, a realization of the stochastic signal denoted $x_S(t)$ reads

$$x_S(t) = \mathbb{A}_s \sum_{n=0}^{N-1} \sqrt{4S_{x_S}(\omega_n)\Delta\omega} \cos(\omega_n t + \varphi_n). \quad (3)$$

The angular frequencies ω_n in rad/s are defined as $\omega_n = n\Delta\omega$ with $n \in [0, N - 1]$ and $\Delta\omega = \omega_u/N$. The angular frequency $\omega_u = 4\pi$ rad/s is the upper cut-off angular frequency, directly linked with the upper cut-off frequency as $f_u = \omega_u/2\pi$ beyond which the PSD function vanishes. The corresponding time discretization is $\Delta t = \pi/\omega_u$ in [s]. The number of terms N is a choice that governs the accuracy of the approximation. The phase angles φ_n are independent realizations of a uniform distribution between 0 and 2π [24]. The SRM offers two features for the Namazu framework. First a way of generating stochastic signals from arbitrary PSD functions related to a frequency space. Second, with this specific choice of PSD function as in equation (2) and the choice of the specific parameters e.g. the cut-off angular frequency, time discretization, etc. presented above, a suitable benchmark problem is constructed. The signal magnitude is scaled by the factor $\mathbb{A}_s \in \mathbb{R}_{\geq 0}$ to achieve the displacement amplitude desired by the user. For instance, to reach a maximum amplitude of ca. 18 mm, the scaling factor $\mathbb{A}_s = 3$ has been chosen. One realization, i.e. a stochastic signal for the Shinozuka benchmark generated via SRM is shown in Fig. 3.

The periodogram [29–31] is used to estimate the signal PSD based on the discrete Fourier transformation of the random signals (equation (3), Fig. 3), as detailed in Appendix A. The direct approximation of the signal PSD by an estimator like the periodogram is usually not possible, as the target PSD is not known exactly for natural processes [32].

The Shinozuka benchmark is of high interest because the approximation of the Shinozuka PSD can be used to assess the shaking table accuracy. Figure 4 shows the approximation of the Shinozuka PSD by the periodogram compared to the target PSD originally used to simulate the signal (equation

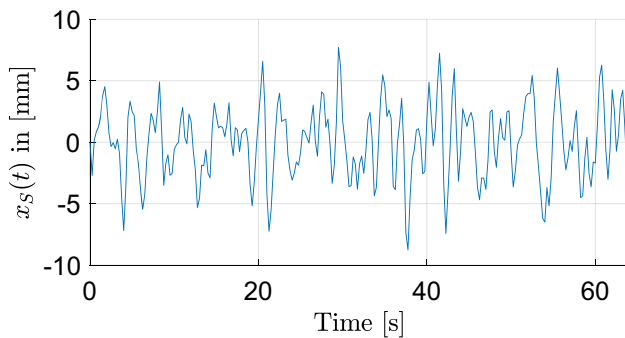


Fig. 3 One realization of a random signal with the Shinozuka PSD (equation (3)) for $N = 128$, $\omega_u = 4\pi$, $\Delta t = \pi/\omega_u$, $T = 64$ s, $\mathbb{A}_s = 3$

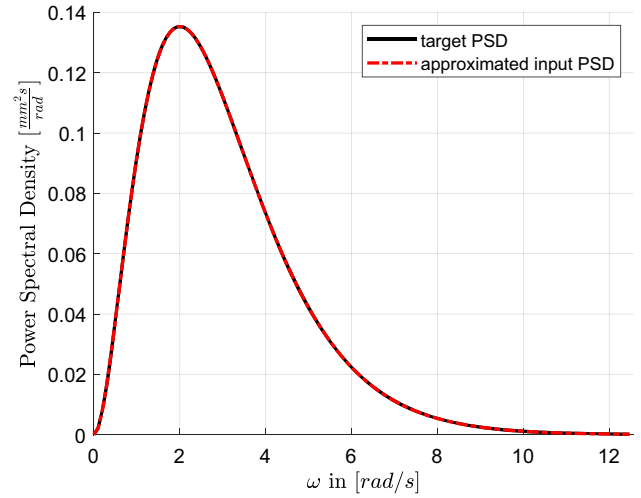


Fig. 4 Shinozuka benchmark: scaled periodogram of one arbitrary realization (equation (3)) and target PSD (equation (2))

(2)). All frequency components in the signal are preserved and thus the approximation is accurate. The accuracy of the table can thus be evaluated by simulating one (or more for variance estimation) realization with the Shinozuka PSD, measuring the displacement, and then comparing the measured PSD to the original one. For an accurate motion, both PSDs should coincide. Therefore, the shaking table motion can be validated for the prescribed range of frequencies and accurate artificial ground motions of the shaking table will be expected for other stochastic signals generated by the SRM with similar frequency ranges to the Shinozuka benchmark.

Post-processing Input Signals for the Shaking Table

Some generated signals may require large initial accelerations or velocities due to the underlying numerical procedures. As an example, the formulation in equation (1) has as a consequence that the initial velocity of the table is non-zero. However, at the start of each experiment, the table has no velocity. To circumvent these issues, post-processing procedures are introduced to ensure the safe usage of the shaking table.

First, for harmonic signals $x_H(t)$, artificial linear acceleration and deceleration phases based on a window function are implemented. The acceleration phase lasts from $t = 0$ to $t = t_u$ and the deceleration phase lasts from $t = t_d$ to $t = T$ such that the durations of acceleration and deceleration are t_u and $T_d = T - t_d$, respectively. Then, the harmonic signal $x_H(t)$ is adjusted from these two windows as

$$\hat{x}_H(t) = \begin{cases} \frac{t}{t_u} x_H, & t < t_u, \\ \left(1 - \frac{t - T_d}{T - T_d}\right) x_H, & t > T_d, \\ x_H, & \text{otherwise.} \end{cases} \quad (4)$$

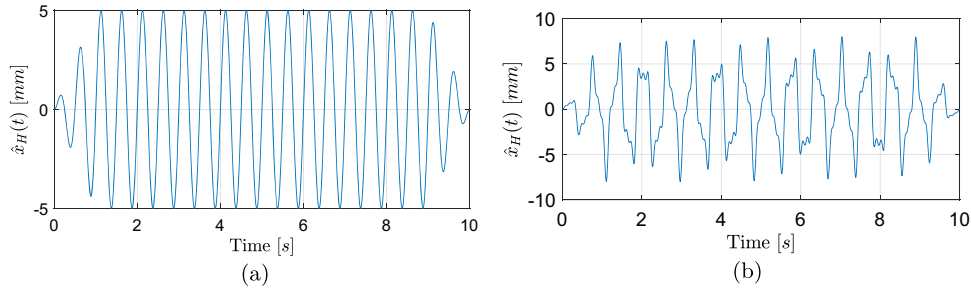


Fig. 5 Post-processing artificial linear acceleration and deceleration phases with $t_u = t_d = 1$ s on signals from Fig. 2: (a) single harmonic signal with $n_\Omega = 1$, $\mathbb{A}_1 = 5$ mm, $\Omega_1 = 2$ Hz (b) Mixed harmonic signal with $n_\Omega = 3$, $\mathbb{A} = \{5, 2, 1\}$ mm, $\Omega = \{1.6, 4.3, 7\}$ Hz

Considering $t_u = t_d = 1$ s, the single harmonic signal from Fig. 2(a) is corrected to have adequate acceleration and deceleration as depicted in Fig. 5. The maximum accelerations occurring are in Fig. 5(a), $\max(|\hat{x}_H(t)|) = 0.79$ m/s² and in Fig. 5(b), $\max(|\hat{x}_H(t)|) = 3.8$ m/s².

Second, the position for the first time instance must always be zero, whereas it may not be a priori the case for the stochastic signal. The correction is achieved by zero padding for all signals, adding a zero position for the time instance $t = 0$. Only one time instant is added and not an acceleration ramp because the frequency content of the signal has to be preserved. It results in $\hat{x}_S(t) = [0, x_S(t)]$ and the time discretization is adjusted to $t \in [0, T + \Delta t]$ (Fig. 6). Other post-processing procedures, such as different window functions for the ramp up and down, are possible due to the open-source nature of Namazu, but they are not implemented yet.

Namazu Hardware

A 3D rendering of the table is shown in Fig. 7. The basic hardware components of the shaking table are an aluminum frame (1), linear rails (2), linear bearings (3), the plate (4),

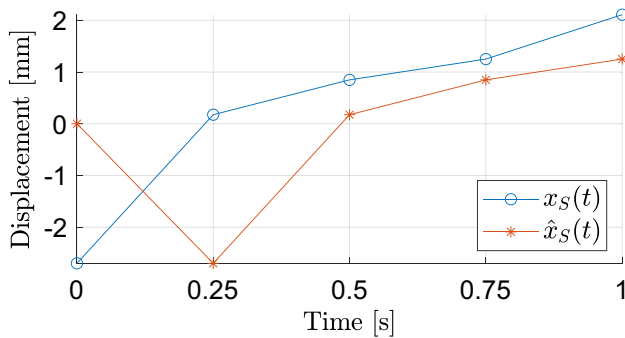


Fig. 6 Comparison of an unmodified SRM signal $x_S(t)$ and the zero-padded signal $\hat{x}_S(t)$

the motor (5) with its driver and the driving assembly, as well as the necessary electrical components. Each component is described in this section and some details on their functionality are given. Since different choices can be made for each part, some comments about possible variants are given.

Pictures of the bare table and the table with a 3D-printed specimen prepared for measurements with digital image correlation (DIC) [33] are shown in Fig. 8.

A technical drawing with dimensions is shown in Fig. 9. The mass of the table is 6.80 kg. Its footprint is about 400 mm \times 500 mm and it takes one person about one hour to fully assemble it. Table 2 gives a list of hardware components and the associated costs. The properties, as found in [21], obtained with Namazu, are given in Table 1.¹

Frame and Rails

The frame is made up of standard 30 mm \times 30 mm I-Type aluminum extrusions. They make assembly and disassembly of the table very practical and allow for flexibility in the design since hardware components can be mounted anywhere the user desires. The plate is designed with different mounting holes for a large variety of mock-up specimens. The plate is attached to the frame by two low-maintenance friction bearings and rails (igus drylin@h W). The frame has to be fixed to some object with high mass to avoid vibrations of the surroundings, e.g., it can be bolted to a rigid high-mass table. As shown in Table 1, the table is mounted on an anchor (experimental table, see also Fig. 8(b)) with a mass of 121 kg. The anchor mass must provide sufficient inertia to prevent unwanted vibrations. Therefore, it depends on the used hardware, payloads and accelerations (Table 2).

¹ The Namazu specifications can be adapted to the user choices by changing the used hardware.

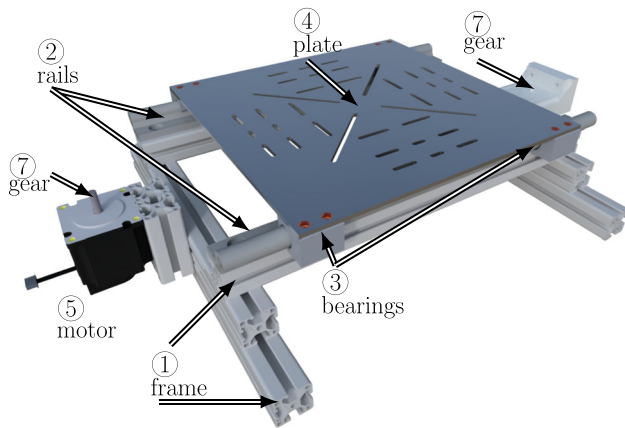


Fig. 7 3D rendering of the table CAD model

Motor and Driver

The displacement is applied to the plate by a bipolar stepper motor through a belt-gear system. Through a gearwheel, the motor provides linear motion to the belt, which in turn is clamped to the table, moving the table as the motor induces a rotation. The belt is displayed in Fig. 8(a). Bipolar stepper motors are made up of two sets of coils. By switching the current through the coils and, therefore, the polarity of the induced magnetic fields, a turn by one step is applied. The motor driver is a specialized electronic circuit, which handles the necessary operations for the electrical current output to achieve this careful switching. It acts as an interface between the controller and the motor.

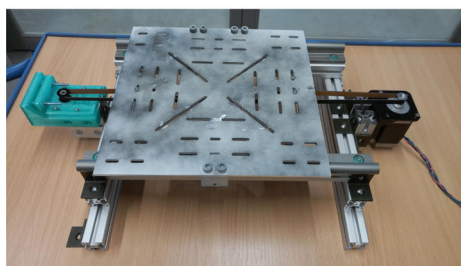
Under ideal circumstances, the position of a stepper motor is known at all times, and a feedback loop is not needed. The setup is thus cost-effective and overall simple. In addition, the driver is able to perform so-called microsteps which enable the motor to be set to positions that do not correspond to full steps. Microstepping is achieved by interpolating the coil current. Using microsteps enhances the resolution of the motor without changing the hardware components. It also increases the versatility of the setup.

An important characteristics of the table is the number of steps n_{disp} that the motor performs for a given displacement, often taken as the number of steps per millimeter. This characteristics depends on the motor, the microstep setting, and the hardware i.e., the driving components. It is calculated from the number of steps the motor performs per revolution and the displacement of the plate during one revolution of the motor

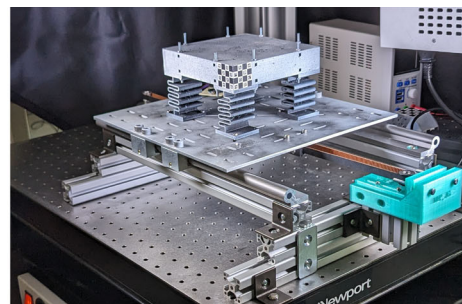
$$n_{disp} = \frac{\text{number of steps per motor revolution}}{\text{plate displacement per motor revolution}}. \quad (5)$$

The current setup uses a NEMA23 motor, where the coils are 1.8° apart. The DM542-driver is set to a microstep setting of 32 microsteps between two coils, thus 6,400 microsteps per revolution. This setup was chosen to match the accuracy of the LVDT measurements (see Section “[Proof of Concept: Validation of Table Accuracy](#)”). Further, we chose a GT2 belt and gear, which means that the gap between the teeth is 2 mm. The chosen gear has 20 teeth in total, so the belt moves by 40 mm per revolution. Therefore, the resolution of the Namazu setup with the features detailed before is $n_{disp} = 160$ steps per millimeter; one microstep corresponds to a displacement of $6.3 \mu\text{m}$. To reduce the effort for changes in microstep settings or hardware, this feature is directly set via a serial command to the motor without the need to reprogram the controller (see Section “[Microcontroller Code](#)”). Note that it is recommended to verify the correct settings with a simple test to ensure for safe operations.

For stepper motors, a trade-off has to be accepted between speed and available torque. With higher speeds the driver has less time to switch the coil current between steps and the inertia of the payload and the table assembly may perturb the control. If the motor torque is not adequate for the applied loads, it may result in loss of steps and therefore loss of accuracy. There are many different sizes of motors available with torque ratings ranging from as low as a few Ncm (e.g. NEMA 11) to large motors capable of achieving a few tens of Nm (e.g. NEMA 42). A similar comment can be made about the microstep setting. More microsteps per revolution decrease



(a)



(b)

Fig. 8 Pictures of the table, (a) Table without specimen, (b) Table with 3D-printed specimen prepared for DIC measurements [33]

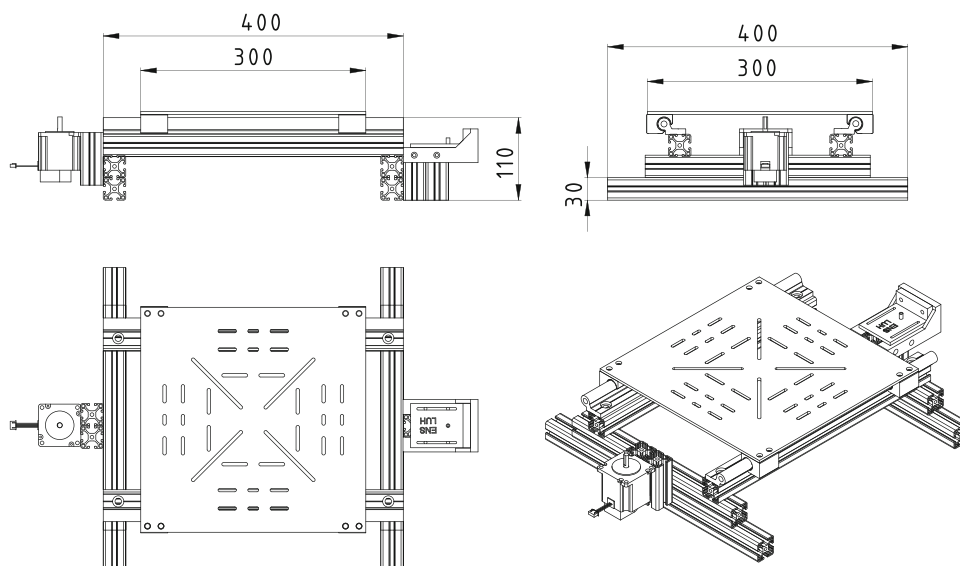


Fig. 9 Technical drawing of the table, dimensions in millimeters

the available torque and also slightly decrease the accuracy since the driver needs to interpolate the coil current between two full steps. Another possible solution is using planetary gearboxes with different ratios available for various motors. These gearboxes increase the resolution without the need for microsteps. The drawback is that the gearboxes usually introduce some backlash, the ratios are not adjustable and they increase the cost.

Microcontroller

The microcontroller is the core element of the Namazu control system. It is the interfacing element between the hardware (driver) and software. The microcontroller is responsible for setting the motor velocity and rotation direction as well as storing, starting, and stopping an experiment. A desired displacement signal (e.g. the ones introduced in Section “Generating Input Signals”) are uploaded to the controller via the serial interface. Once the experiment is started, the controller pulses the STEP-pin of the motor driver at a certain frequency that depends on the desired speed, which in turn is dependent on the desired displacement. The necessary calculations are performed by the microcontroller during the

experiment. For each STEP-pulse the motor turns by one step, i.e. changing the frequency of the pulse allows for the definition of arbitrary signals, which is the core functionality of Namazu.

The compatibility of the microcontroller with the free real-time operating system (FreeRTOS) and the Arduino framework is mandatory to use the provided code. Further, we recommend a high-resolution hardware timer for the microcontroller such that the motor is controlled with high precision. The chosen microcontroller is an ESP32 devkit module. A processor clock rate of 250 MHz enables for the motor velocity and direction of rotation up to 200 times per second. Moreover, the ESP32 module provides a second core that can handle interrupts and provide a control interface, without interfering with the motor update rate. Thus, the experiment is isolated from interfacing operations, which improves the repeatability of experiments.

Variants

The Namazu shaking table that is presented here comprises various pieces, which could be upgraded/downgraded independently depending on the user’s budget and research needs. This versatility makes the shaking table tunable. It can be personalized to fit desired performance or operational conditions. Some variants are gathered in Table 3. Note that variations in hardware make it necessary to recalculate the steps per millimeter setting for the motor, as this property depends on the physical properties of the table as well as the microstep settings in the driver. As an example, changing

Table 1 Namazu specifications

Tested payload:	2 kg
Maximum acceleration with payload:	2 g
Maximum speed with payload:	0.4 m/s
Mass of anchoring support:	121 kg

Table 2 Costs of shaking table hardware (2022–2023, Germany)

Nr.	Hardware	Description	Price
①	Frame	30x30 I-Type aluminium extrusions (4x 400 mm, 2x 300 mm, 4x 70 mm)	21.64 €
②	Linear rails	igus drylin W, WS16_0.4M	31.74€
③	Linear bearings	igus drylin W, WJ2000UME-01-16	41.48€
④	Plate	300x300x4 aluminium plate	42.69€
⑤	Motor	NEMA23 (ACT 23HS8430D8P1)	21.60€
⑥	Driver	ACT DM542	33.10€
⑦	Gears	GT-2, 20 teeth gear and idler	5.00€
⑧	Belt	GT-2, 1m	4.99€
⑨	Power supply	PEAKTECH 6227	123.17€
⑩	Microcontroller	NodeMCU ESP32-WROOM32	11.49€
-	Misc.	Attachment plates, screws, nuts, etc. (price is an estimate)	60.00€
		Total:	396.90€

The cables and tools used for assembly are not listed

the belt and gear assembly of the current setup to a 1610 ball screw assembly (1610 referring to a screw diameter of 16 mm and a screw lead of 10 mm) means that the table is moved by 10 mm per revolution of the motor. Together with the 200 steps per revolution of the motor, this leads to a resolution of the shaking table of n_{disp} equal to 20 steps per mm (see equation (5)), if no microstepping is used.

Most limitations on payload capacity, acceleration, and speed are only due to the hardware components. Tests have shown that the Namazu table with current components performs well with the physical properties reported in Table 1. The tested payload is 2 kg. This small payload fits some research projects. Besides, for teaching purposes, we find that using the table in combination with 3D-printed structures offers a versatile and low-cost solution to give students some experience with designing and testing parts. 3D-printed structures, when fabricated with materials such as polylactic acid (PLA) or thermoplastic polyurethane (TPU), are cheap and lightweight. Therefore, a low-torque motor already achieves satisfying performances, with low-cost hardware and simple handling of the Namazu framework. As a reference, the payload limitations on the SARSAR shaking table [20] are given as 200 kg capacity while also using a (more capable) stepper motor and a ball screw instead of the belt gear driving system.

The table can also be tuned to decrease the cost. A motor with a lower torque such as the NEMA17 motor could be employed. These motors are very common since they are standard parts used in 3D printers. There exists a wide variety of motor and driver combinations, thus we can not provide a comprehensive list.² Table 2 shows that the highest cost in

the setup comes from the adjustable power supply. It could be changed to a fixed voltage power supply to save cost.

Control System

The shaking table is equipped with a control system to apply any single-axis displacement to the plate as shown in Fig. 1. The software generates (Section “[Prescribing Complex and Random Signals](#)”) or loads a discretized time-displacement signal, translates it into Marv-code³ for the motor, and sends it to the microcontroller via the serial interface. The code is then saved on the microcontroller such that the experiment can be started or restarted at any point. The processing of Marv-code as well as the interface with the microcontroller and the microcontroller itself make up the control system.

From Signal to Motion

Starting from a (theoretically) continuous displacement signal $x(t)$ as described in equation (1) for harmonic signals and equation (3) for stochastic signals, there are two discretization layers needed for the motor control. The process is illustrated in Fig. 10, where the discretization process is greatly exaggerated for the sake of clarity. The continuous signal (solid blue line) is first discretized to obtain the digital signal (dashed red line), i.e. an array of size n_t consisting of numbers that specify the desired positions in a fixed interval Δt . The second discretization occurs in the microcontroller to determine the stepping rate of the motor. Programmatically,

² The costs highly depend on the availability of the components.

³ We refer to the motor-code as Marv-Code, in close resemblance to the name G-code and the third author’s name.

Table 3 Possible hardware variants

Original piece	Upgraded piece	Advantage of the upgraded piece
Belt & gear assembly	Trapezoidal screw	Higher accuracy and payload capabilities, more rigid construction due to rigid parts, no imprecisions due to belt slipping
	Ball screw	Even higher accuracy and payload capabilities, smoother operation than trapezoidal screws, no imprecisions due to belt slipping
Friction bearings	Roller bearings (SBR16UU etc.)	Higher payload capabilities
Frame	Pre-built linear positioning table	Higher manufacturing accuracy, multiple variants available
NEMA23 motor, DM542 driver	NEMA34 motor, DM860T driver	Higher torque, higher payload

the microcontroller is given a list of positions and the time discretization Δt . Therefore, the firmware linearly interpolates between the positions in time, where the distance between two positions determines the desired velocity and therefore the stepping rate. Note that the motor is only able to move one step at a time, which results in a piecewise constant position signal (dotted black line). However, choosing Δt small enough results in a good accuracy of the motion. We found that using $\Delta t = 5$ ms gives good results without pushing the limits of the microcontroller’s speed.

Microcontroller Code

The microcontroller code (called firmware) is the interfacing element between the software, an optional operator, and the motor. As shown in Fig. 11, the firmware provides two interfaces:

- **USB:** A serial interface, which processes readable commands and floating numbers in ASCII. An exemplary sequence of commands is shown in Algorithm 1.
- **PIN:** A set of electrical lines to trigger certain signals. These pins allow for the installation of an optional control panel with buttons to start, stop, and adjust the table, or optional limit switches to protect the hardware.

Algorithm 1 Example of Marv-code to move the table linearly to +10 mm, then -10 mm, then 0 mm with a speed of 10 mm/s.

```
# clear previous positions from memory
reset
# set steps per mm
set spmm 160
# set instructions per second
set rate 1
add 10 # add positions (in mm)
add 0
add -10
add 0
start # start the experiment
```

The commands and signals are processed by the *Interface* component of the firmware. There are three classes of commands to differentiate:

- Configuration, beginning with `set`
- Definition of the experiment, beginning with `add`
- Actions, like `start` or `stop`.

A full list of supported commands is listed in Table 4. A previous version of the motor control firmware used the programming language G-code, which is a standard already existing and widely used in 3D printers and CNC machines. Thus, G-code could be used to communicate with the motor.

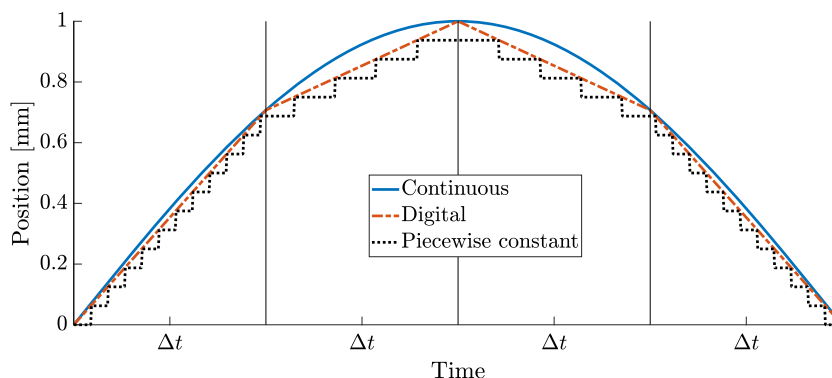


Fig. 10 From a continuous signal to the motor motion

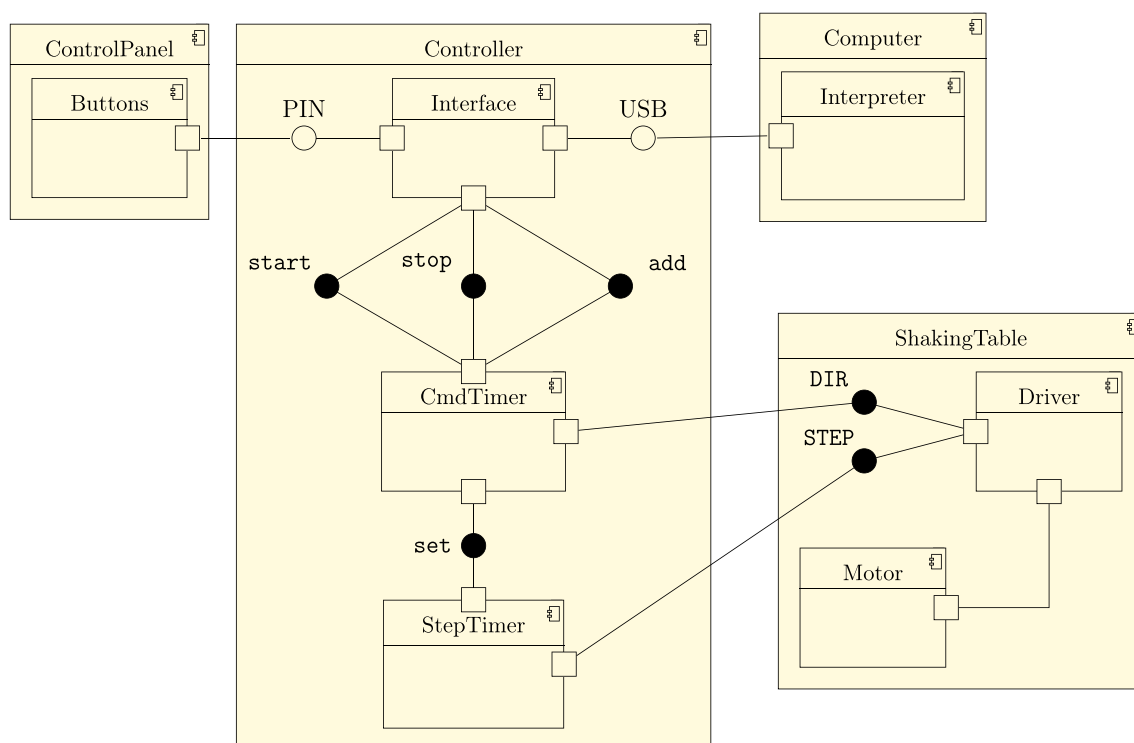


Fig. 11 Control system architecture

However, we found that there is some overhead (such as acceleration limits) in the implementation of G-code itself that is not needed for the shaking table but interferes with the accuracy of the timings. We therefore introduced Marv-code, which is a similar but slimmer approach. It provides the most important settings (such as steps per millimeter, the discretization length Δt) and a list of positions to the controller. It is possible to change the settings without the need to interfere with the controller firmware itself. This process is independent of the used programming languages and operating systems, as long as they can interface with a serial port (i.e. a USB port). The firmware is designed to work with any kind of stepper motor/driver combination that uses the same STEP/DIR interface (e.g., A4988, DRV8825,

TMC2208, TMC2209, DM542) and therefore also with a wide range of stepper motors.

The displacement of the table is applied by the motor through a sequence of discretized positions for a series of constant elapsed times between two consecutive positions (Δt) during the whole signal. The firmware uses two so-called timers to drive the motor correctly. The first timer, *CmdTimer*, is configured to trigger at a static rate during the experiment. This rate is received by the *Interface* (c.f. *rate* command in Table 4). The *Interface* sets up the *CmdTimer* and gets activated at a *start* command. Each time the *CmdTimer* triggers, it loads the next prescribed position of the experiment and calculates the velocity and direction of the motor necessary to reach it. Then the *CmdTimer* triggers

Table 4 Marv-code syntax

Syntax	Description
<code>start</code>	Starts motor motion with previously defined positions at the previously set rate.
<code>stop</code>	Immediately stops motor motion.
<code>info</code>	Writes out information about machine state to serial, OK: Info: state: READY, mode POSITION, command 25 of 25, rate: 20hz, spmm: 10.
<code>set rate nnn.nnn</code>	sets the amount of positions to handle in one second.
<code>set spmm</code>	Specifies how many steps the motor has to drive to move the plate one millimeter.
<code>add nnn.nnn</code>	Adds a position at the end of list.
<code>reset</code>	Resets all positions from memory.

again. The *CmdTimer* also sets the driver DIR pin according to the required direction. The calculated velocity multiplied by the step resolution (steps per millimeter) gives the rate of the so-called *StepTimer* to trigger. Thus, each time the *CmdTimer* processes the next position, it reconfigures the *StepTimer*. The *StepTimer* generates a rectangular signal and sets the STEP pin of the driver accordingly. Thus, Marv-code allows for the precise description of arbitrary, uniformly sampled signals directly via the serial interface. There is no need to re-upload the controller firmware for a new experiment. Furthermore, changes in the hardware setup are fully described by the *spmm* setting given in equation (5).

Microcontroller Interface

To upload the signal to the motor, it is translated into microcontroller-readable code via a simple interpreter that takes the pre-calculated time-displacement signal and generates a sequence of strings to be sent to the controller via serial. To ensure safe operating conditions, for each new experiment i.e., each time one loads new data, one resets the current experiment data and prescribes the steps-per-mm setting and the rate (see the *reset*, *set spmm* and *set rate* commands in Algorithm 1). Afterward, the interpreter writes the desired positions as a sequence of *add* instructions. Note that one does not need to prescribe the time since the signal is discretized uniformly. The rate is given by *rate n* i.e., the controller uses *n* positions per second.

The *start* command can be sent simultaneously or separately without sending all of the previous commands again, e.g. to make sure that the table is safe to operate before starting the experiment. The experiment can also be interrupted at any point using the *stop* command, or, after the experiment is finished, restarted by sending *start* again. All generated data such as input signal, Marv-code, and parameters can be stored on disk to re-run experiments that were previously generated. This operation is currently performed with

a custom class, which also makes handling of all settings and parameters easy.

Any programming language or framework that can use serial interfaces can communicate with the controller, which makes this approach very versatile. Already existing methods for signal generation can be employed without rewriting functions for different software packages or operating systems. A code base and interpreters for MATLAB and Python are provided [25], and the general functionality is the same if another software environment is chosen. It is expected that translating the existing implementation to other languages is straightforward.

Pre-/post-processing Code and Restrictions

Input harmonic and stationary random signals are generated with the algorithms introduced in Section “[Generating Input Signals](#)”. However, the input signals are not restricted to these algorithms. Any signal could be considered. For instance, one may generate non-stationary signals including frequency sweeps or increasing amplitude, or load pre-recorded data from vibration experiments or earthquakes. The only restriction is that the signal is discretized with a constant sampling rate (i.e. $\Delta t = \text{const.}$). Besides, the desired amplitudes and frequencies should not exceed safe operating conditions, especially when no limit switches are used. To avoid sudden acceleration spikes, it is advised to use the linear ramp up and ramp down shown in equation (4) whenever possible.

For the Shinozuka benchmark, the scaling factor needs to be considered. A change of the signal in the time domain also leads to a change in the frequency domain. This point is further discussed in Section “[Test 3: Shinozuka Benchmark](#)”.

Proof of Concept: Validation of Table Accuracy

The accuracy of the table was tested during a measurement campaign at the Laboratoire de Mécanique Paris-Saclay

Table 5 Experimental campaign for table validation

	Prescribed displacement Amplitude	Signal	Measurement Quantity	Method	Uncertainty	Structure
Calibration						
Test 0	±20 mm	Triangular	Displacement	LVDT	4 μm	No
Harmonic signal						
Test 1-1	±5 mm	Single-frequency	Displacement	LVDT	4 μm	No
Test 1-2	±7.5 mm	Mixed-frequency	Displacement	LVDT	4 μm	No
Stochastic signal						
Test 2	±2 mm	Stochastic	Displacement	LVDT	4 μm	No
Shinozuka benchmark (10 tests)						
Test 3	±10 mm	Stochastic	Displacement	LVDT	4 μm	No

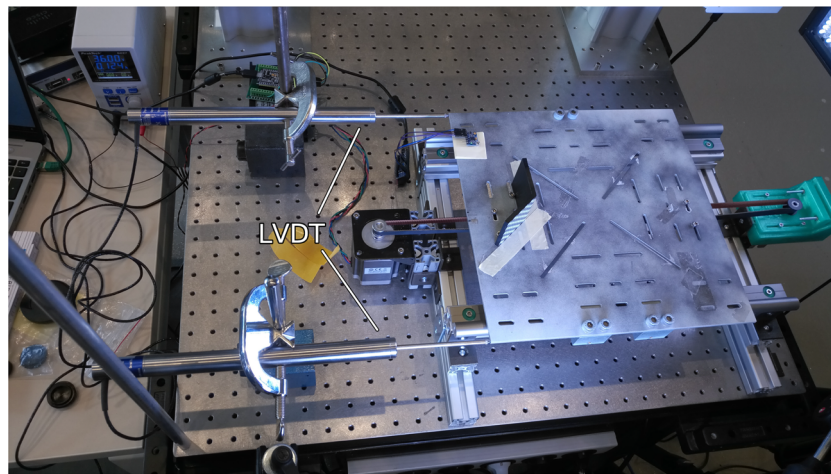


Fig. 12 Table set up for LVDT measurements

(LMPS) at ENS Paris-Saclay. The goal is to evaluate if the measured displacements, accelerations, and frequencies correspond with the prescribed ones. The different tests of the campaign are summarized in Table 5. Their complexity varies, ranging from simple displacement tests to the Shinozuka benchmark, where the focus is put on error statistics both in measured displacement and frequency.

For all measurements, two LVDTs (see Fig. 12) are utilized to measure the displacement of the plate. The LVDTs are attached to the plate at two corners to also measure the rotation of the plate. Data are acquired at a rate of 4.8 kHz, then downsampled to 200 Hz to match the motor sampling rate. No other filtering is performed.

Test 0: Calibration Test

The table was tested during Test 0 with a triangular displacement with a constant speed equal to 8 mm/s. Results are

shown in Fig. 13. In the top graph, the red line represents the LVDT-measured displacement whereas the blue line represents the desired displacement. The absolute displacement error is depicted in the bottom figure. The displacements measured by the LVDTs are very close to the prescribed ones. The mean absolute error and maximum error over the 20 seconds are about 0.03 mm and 0.11 mm, respectively. It is observed that the error fluctuates, the peaks occur during changes of direction, which indicate the presence of backlash in the system. Yet, the error does not accumulate over time and the plate does not drift far from the prescribed displacement.

Test 1: Harmonic Loading

To check that the table achieves the desired frequencies and to test the displacement accuracy with higher speeds we conduct two tests without specimens using harmonic signals with a single frequency and mixed frequency content. The pre-

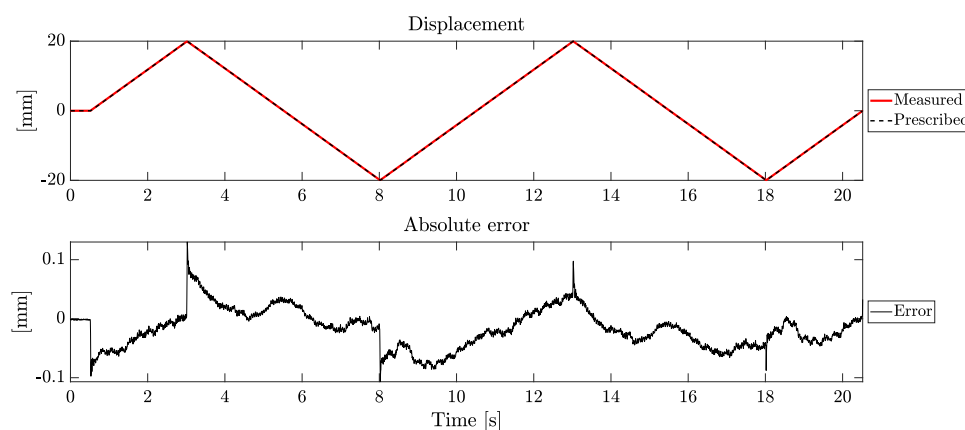


Fig. 13 Measured displacement and absolute displacement error for triangular loading, Test 0

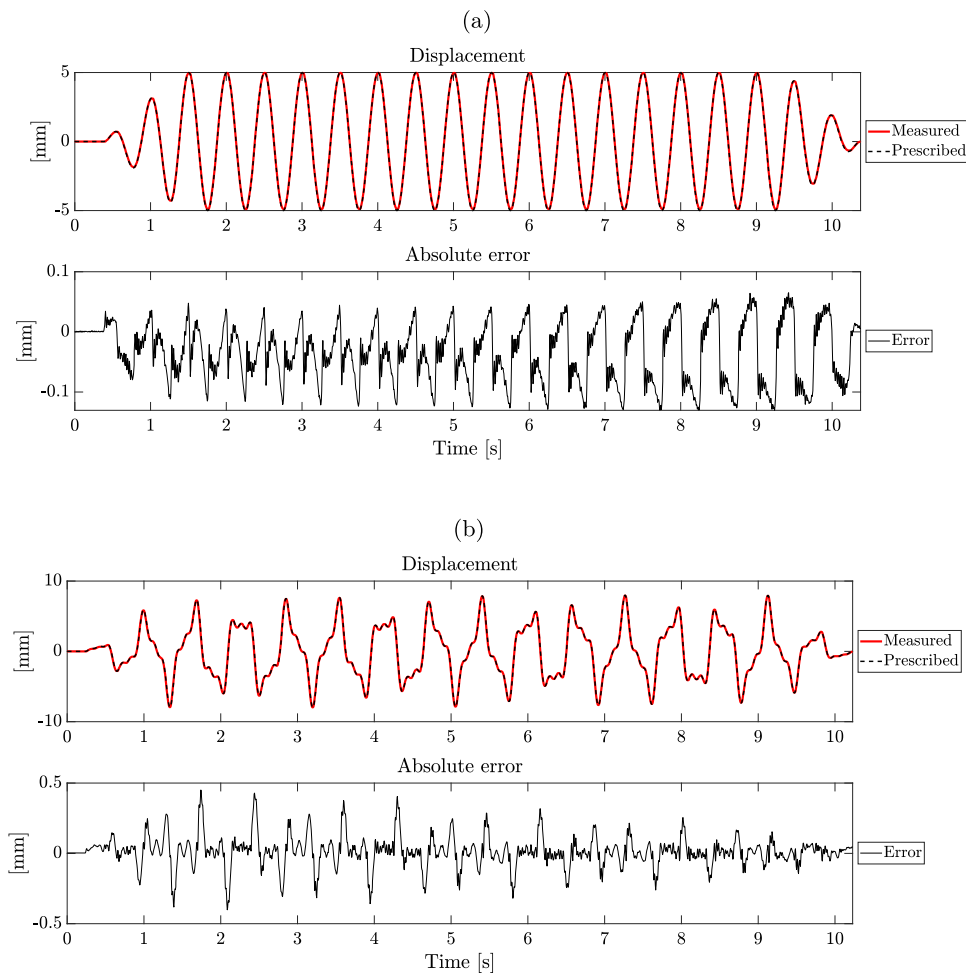


Fig. 14 Measured displacements and absolute displacement errors for harmonic loading, (a) Test 1-1: single 2 Hz frequency, (b) Test 1-2: mixed frequency

scribed loads are a harmonic load with a frequency of 2 Hz and an amplitude of 5 mm for Test 1-1 (cf. Fig. 2(a)), and a harmonic with mixed frequencies of 1.6, 4.3, and 7 Hz with corresponding amplitudes of 5, 2, and 1 mm respectively for Test 1-2 (cf. Fig. 2(b)). The input signal is sampled at 200 Hz which corresponds to the aforementioned Δt of 5 ms (Section “From Signal to Motion”). The resulting displacements and

the absolute errors for both tests are shown in Fig. 14. The behavior is similar for both tests. The overall performance of the table is satisfactory, as the measured displacements follow closely the prescribed ones. The backlash that was observed in Test 0 is also present. However, its magnitude is very low. The same error occurs periodically for each cycle. Moreover, the measured displacement never deviates signif-

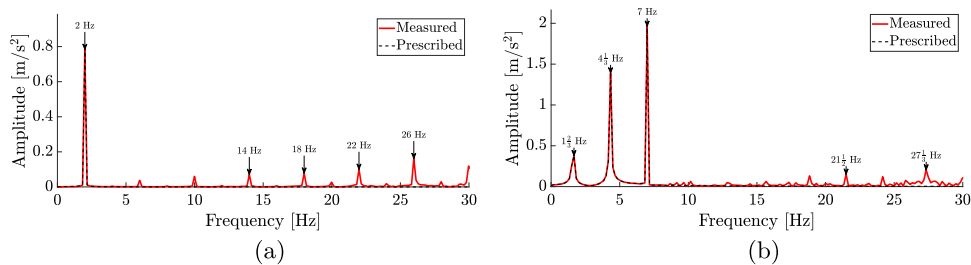


Fig. 15 Measured acceleration spectra for harmonic loading, (a) Test 1-1: single 2 Hz frequency, (b) Test 1-2: mixed frequency

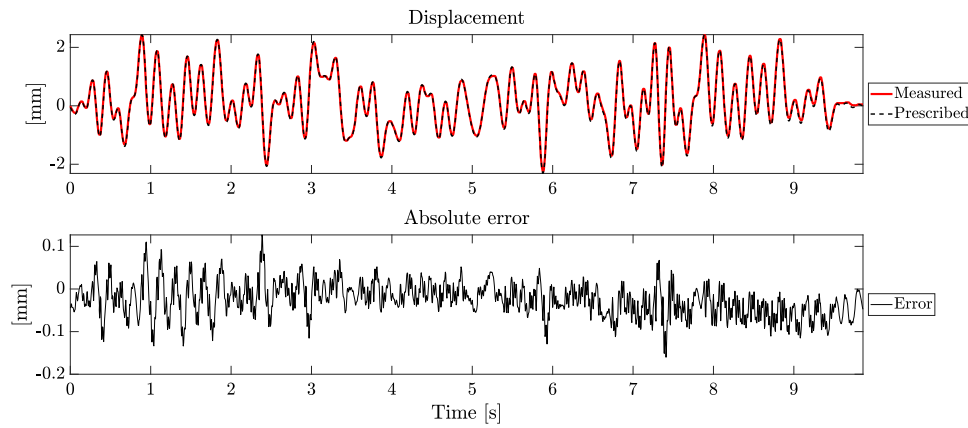


Fig. 16 Measured displacement and displacement error for random loading, Test 2

icantly from the prescribed signal, there is no visible phase shift.

Figure 15 shows the prescribed and measured acceleration frequency spectrum for both loadings in the stationary region of the test. The frequency resolution is $\frac{1}{6}$ Hz. The diagrams show frequencies up to this accuracy. The acceleration frequency spectra are obtained by differentiating the displacement twice in time and then applying the fast Fourier transform (FFT). Here, we apply the acceleration spectrum rather than position to emphasize the high-frequency uncertainties. Due to the differentiation in time, the discrepancy is amplified. In addition, the FFT diagrams show data from 0 to 30 Hz, which is a good compromise between the frequencies most relevant to earthquake engineering and structural mechanics, and those achievable with the table. Excitations at higher frequencies may be present due to the setup natural frequencies or non-linear effects. These effects depend on the table, its support, or the driving mechanism. Excitations in the 45-55 Hz region are observed in the present setup for some tests. For example, Fig. 15(a) shows excitations at multiples of the 2 Hz input frequency, indicating some non-linear effects. However, the highest peaks show that the measured frequencies match the desired frequencies precisely. Noise levels in the relevant frequency ranges are low enough to be handled by filters.

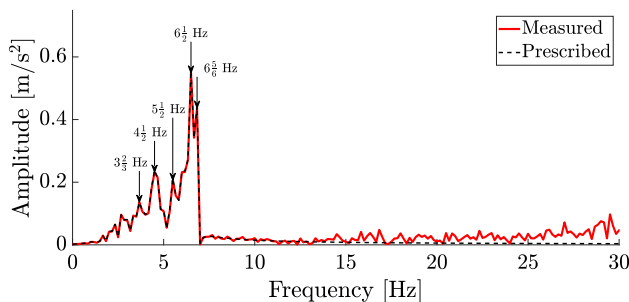


Fig. 17 Acceleration spectra for random loading, Test 2

Test 2: Random Signal

To test a more complicated signal, a randomly generated signal with a customized PSD function is defined as

$$S_{f \leq 7 \text{ Hz}}(f) = \begin{cases} 0.01 & 0 \leq f \leq 7 \text{ Hz}, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

A single draw is obtained from SRM. Results for the measured displacement and the corresponding error are shown in Fig. 16. The error due to backlash is less pronounced than in Tests 1-1 and 1-2 since the displacement amplitude is lower than in the previous tests. The mean absolute error is 37 μm , which corresponds to 1.5% of the maximum displacement, and the maximum error is 0.16 mm i.e., 6% of the maximum displacement. Furthermore, any error is quickly recovered. The table does not deviate from the prescribed signal and the error does not accumulate over time, which could be a risk for open-loop systems. In addition, no phase lag is visible.

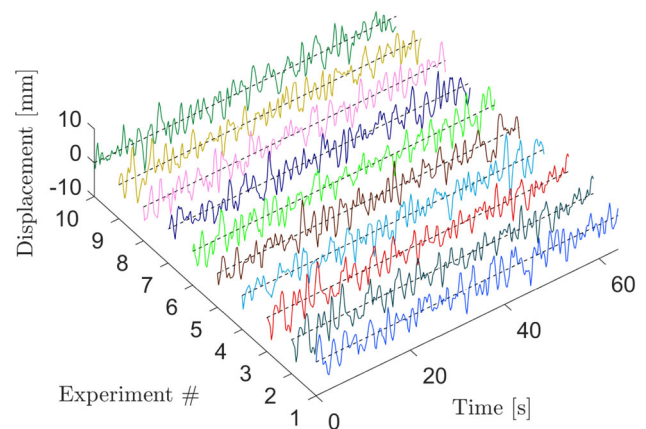


Fig. 18 Ten realizations of the random signal with the Shinozuka PSD and $N = 128$, $\omega_u = 4\pi$, $T = 64$ s, $\Delta t = \pi/\omega_u$, $A_s = 3$

Figure 17 shows the prescribed and measured acceleration frequency spectrum. The sharp decrease in the PSD function above 7 Hz due to the shape of the prescribed PSD (cf. equation (6)) is visible. The PSD function in equation (6) is given in terms of displacement. Therefore the resulting target PSD function has for prefactor the square of the frequency. As a further note, although the frequency content is not random in this particular PSD function, the underlying stochastic process is just an approximation by SRM. The FFT of the selected signal of the SRM only gives an estimation of the underlying PSD function. This remark also motivated the development of the Shinozuka benchmark, as it makes it possible to ensure a more predictable outcome of the PSD estimation. Discrepancy at higher frequencies is again present but generally low.

As a general remark, structures exhibit the most relevant responses in the first few modes. Therefore, the Namazu operating range is concentrated on low frequencies reproduced with a good agreement. Depending on the natural frequencies of the structure of interest, a closer examination of the achievable frequency range may be necessary.

Test 3: Shinozuka Benchmark

Test 3 aims at validating the accuracy of the table motion with desired stochastic signals in the frequency domain with the Shinozuka benchmark (Section “Stationary stochastic signals”). The benchmark consists of ten different displacement signals, each from one realization of the PSD in equation (2). Each signal is called an individual experiment and num-

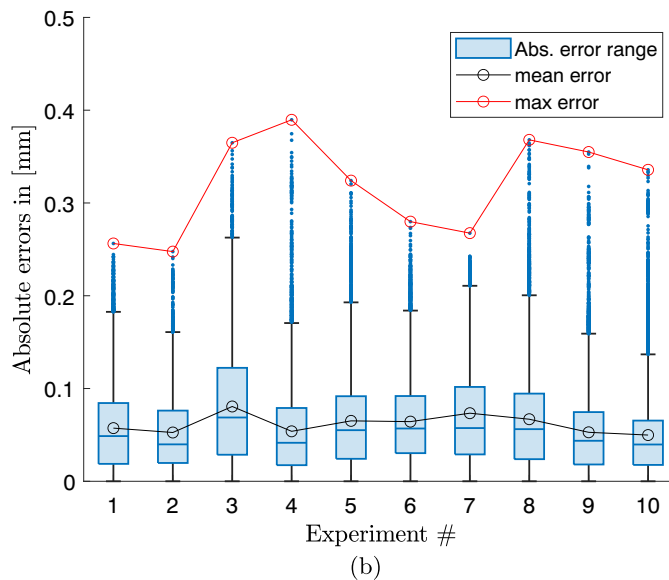
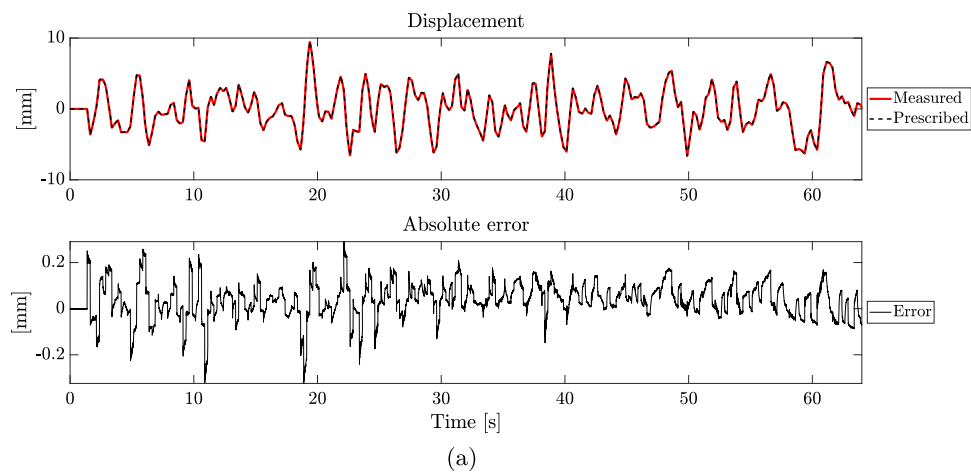


Fig. 19 Measured displacement and absolute displacement error for the Shinozuka loading, Test 3, (a) Experiment #4, (b) statistics of the instantaneous displacement error for the ten experiments

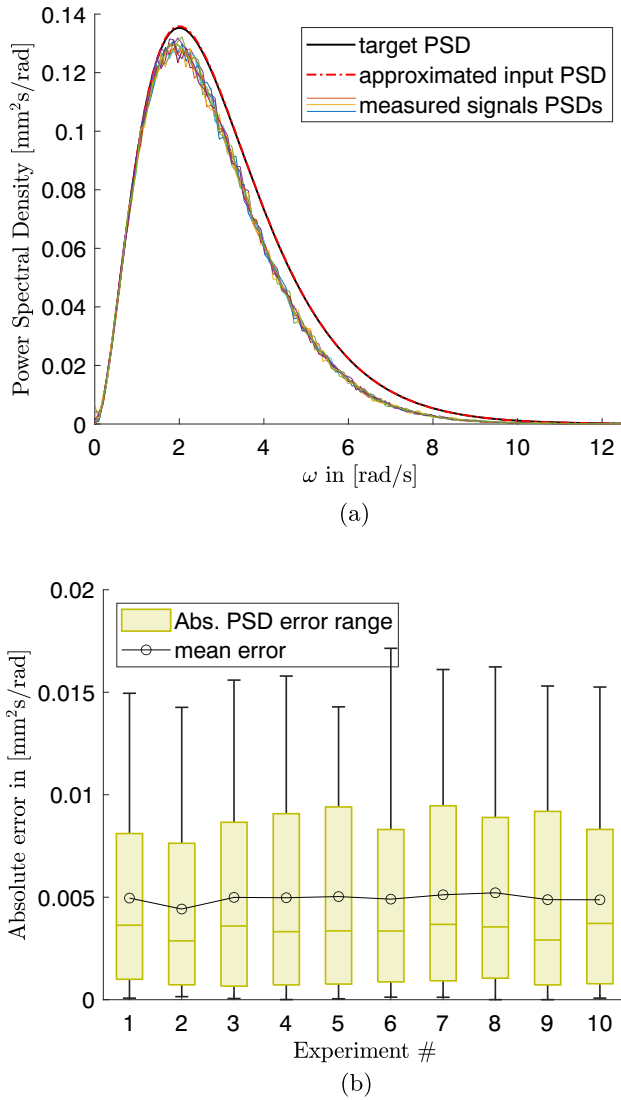


Fig. 20 Target and input PSD functions compared to PSD estimations of the measured displacements in experiments #1 to #10, (a) Direct PSD comparisons. (b) Statistics of the PSD errors per frequency range for the ten experiments

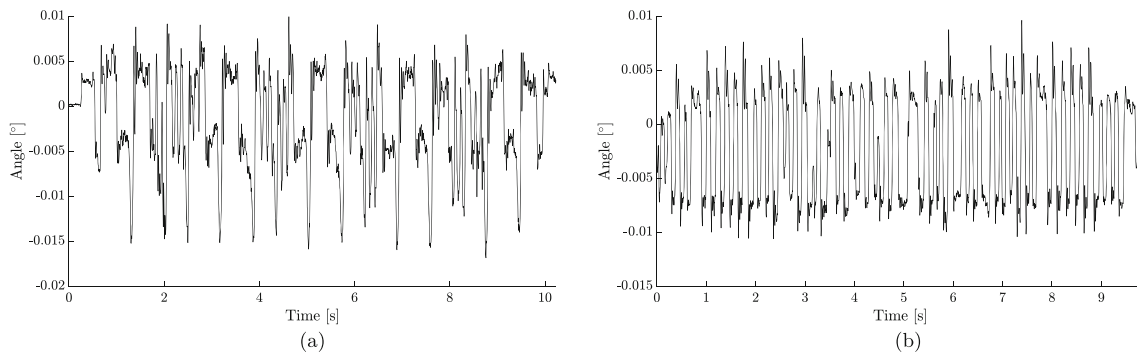


Fig. 21 Measured angle as given by the difference between the two LVDT measures for the harmonic and random loading, (a) Test 1-2: mixed frequency loading, (b) Test 2: $S_{\omega \leq 7 \text{ Hz}}$

bered from #1 to #10. One experiment therefore consists of a specific input signal and the corresponding LVDT measurements of the plate motion. As mentioned before, the exact PSD is theoretically obtained by the periodogram if the table motion during one experiment is accurate. Deviations in the measurements directly indicate inaccuracies in the table. The ten signal realizations depicted in Fig. 18 are generated from equation (3) and applied to the plate. The maximum displacement for the ten experiments is about 10 mm. The signals have been scaled by the factor $\mathbb{A}_s = 3$ to increase the true displacements. A Monte Carlo simulation of 10^6 signals utilizing SRM with $\mathbb{A}_s = 3$ resulted in a maximum absolute amplitude of 18 mm, which was chosen to have a reasonable range of the plate displacements.

Scaling of the signal in the time domain to achieve larger displacements results in larger values of the estimated PSD function. To perform the Shinozuka benchmark, such that one can compare how well the target PSD is reproduced, and before applying the periodogram equation (7), the measured displacements need to be divided by the factor \mathbb{A}_s . This operation has been done for the input signal in Fig. 4 to match the targeted PSD.

The first step of the benchmark focuses on the LVDT displacement measurements of the plate (Fig. 19). The results of experiment #4 for the measured displacement compared to the input signal of the plate motion is plotted in Fig. 19(a). The statistics of the absolute displacement errors of each time instance for the ten signals is shown in Fig. 19(b). These data are represented by standard box-whisker plots. Information on the boxplot parameters is provided in Appendix B.

From this sample set, experiment #4 exhibits the largest maximum absolute error of $390 \mu\text{m}$, whereas experiment #3 exhibits the largest mean absolute error of $81 \mu\text{m}$. The mean error over the whole set of experiments is $62 \mu\text{m}$. This observation shows that the shaking table provides the targeted absolute displacements in an acceptable range of error.

In the second step of this analysis, the PSD function estimations i.e. the signal frequency contents are analyzed in Fig. 20. The PSD functions, estimated from the measured displacements, for the ten experiments, are given in Fig. 20(a). A deviation is observable compared with the target PSD (Fig. 4) for all the experiments for angular frequencies greater than 1.8 rad/s. Some energy is dissipated by the shaking table due to friction, other sources of inaccuracies could stem from the material behavior, belt flexure, or bearing clearances for instance. These phenomena result in a lower power for all frequencies and can be of significant influence, especially for higher angular frequencies, $\omega > 9$ rad/s. It may happen that these frequencies do not exist in the table motion because the estimated PSD values of the measurements are converging faster to zero than the target PSD.

The statistical analysis of the errors over the ten experiments is shown in Fig. 20(b). Both mean and maximum errors do not fluctuate significantly between the different realizations of the Shinozuka benchmark. They are within acceptable ranges. Overall, the estimated PSD functions are in good accordance with this target.

Summary and Discussion

In all tests, the table has been able to closely follow the prescribed signal and with no phase lag. The largest errors have been observed during changes in direction, indicating backlash is present in the belt gear assembly. From the Shinozuka benchmark, it is concluded that frequencies above a certain threshold are damped more than others. However, the desired PSD is matched with a good accuracy.

Overall, the errors are mostly due to mechanical phenomena, e.g. friction, belt flexing, misalignment of the rails, and bearing clearances, and not due to the controller software. We also note that it is not required to perform another calibration but just setting the resolution of the table in steps per unit displacement (equation (5)). As shown in Fig. 21, some rotation of the plate due to bearing clearances is observed for the mixed harmonic (Test 1-2) and the PSD (Test 2) tests, which contributes to the difference in displacement. However, the total rotation remains small.

Conclusion

Namazu, a low-cost shaking-table framework for dynamic and stochastic dynamic analysis, is presented. All hardware parts are off-the-shelf components, making the table construction accessible to many teachers and researchers. Any desired time-displacement signal (harmonic, random excitation, seismic ground motion, and other vibrations) can be prescribed. The open-source control software is specially tailored for generating and applying stochastic signals. Some

elements of the shaking table, such as the motor or the driving system, can be changed depending on the user's needs without any software change since a standard interface for bipolar stepper motors was used. Moreover, table usage is possible through any programming environment that can interface with serial ports since the controller code has been developed from the ground up.

The shaking table is designed for testing 3D-printed mock-ups. Due to their low weight, it is possible to use motors with low torques so that this low-cost shaking table can be used for teaching and research purposes. We have assessed the accuracy of the table motion with respect to the desired acceleration signals through various loadings LVDT measurements. Experiments with a high accuracy in terms of time and frequency have been performed for various signals, including the Shinozuka benchmark.

It is planned to use Namazu for the verification of reliability analysis algorithms, predicting and analyzing failure mechanisms of specific materials or small-scale load-bearing components.

A Periodogram

The periodogram [30] is an estimator to determine PSD functions from stochastic signals. It is defined as the squared absolute value of the discrete Fourier transform of a time signal $X(t)$

$$\hat{S}_X(\omega_k) = \lim_{T \rightarrow \infty} \frac{\Delta t^2}{T} \left| \sum_{t=0}^{T-1} X(t) e^{-2\pi i k t / T} \right|^2, \quad (7)$$

where T is the total number of discretized signal points and Δt the time increment. The time instant in the record t is just an index and k is the angular frequency discretization with $\omega_k = \frac{2\pi k}{T}$.

B Boxplot Parameters

Figures 19(b) and 20(b) comprise boxplots. The horizontal line inside the box corresponds to the median of the analyzed samples. The top and bottom edges of each box represent the upper and lower quartiles i.e. 0.75 and 0.25 quantiles of the data, respectively. The distance between the top and bottom edges is the interquartile range denoted IQR .

The outliers are values that are more than $1.5 \cdot IQR$ away from the top or bottom of the box. The whiskers are lines that extend above and below each box. The top whisker connects the upper quartile to the non-outlier maximum (the maximum data value that is not an outlier), and vice versa for the lower whisker.

Acknowledgements Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for the GRK2657 (grant reference number 433082294) is greatly appreciated. Furthermore, we thank the Faculty of Civil Engineering and Geodetic Science of the Leibniz University Hannover for funding the construction of the table. We also want to thank our colleagues from the LMPS for allowing us to set up the table in their labs and for providing us with their expertise. Further, we acknowledge the contributions of the Institute for Risk and Reliability (IRZ) and the Institute of Mechanics and Computational Mechanics (IBNM) from the Leibniz University Hannover, where the first two authors had the chance to pursue this project. Special thanks to Daniel Ho from IBNM, as well as Xavier Pinelli and Rémi Legroux from LMPS, for providing technical support and thoughtful insights into the construction.

Author Contributions **Jan Grashorn**: Conceptualization, data curation, formal analysis, investigation, methodology, project administration, software, validation, visualization, writing - original draft; **Marius Bittner**: Conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, software, validation, visualization, writing - original draft; **Marvin Banse**: Conceptualization, data curation, methodology, software, validation, visualization, writing - original draft; **Xuyang Chang**: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing - original draft; **Michael Beer**: Funding acquisition, resources, supervision, writing - review and editing; **Amelie Fau**: Funding acquisition, project administration, resources, supervision, writing - review and editing.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest statement On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Horiuchi T, Ohsaki M, Kurata M, Ramirez JA, Yamashita T, Kajiwara K (2022) Contributions of e-defense shaking table to earthquake engineering and its future. *J Disaster Res* 17(6):985–999. <https://doi.org/10.20965/jdr.2022.p0985>
- Nakashima M, Nagae T, Enokida R, Kajiwara K (2018) Experiences, accomplishments, lessons, and challenges of e-defense—tests using world's largest shaking table. *Japan Archit Rev* 1(1):4–17
- Tao L, Ding P, Shi C, Wu X, Wu S, Li S (2019) Shaking table test on seismic response characteristics of prefabricated subway station structure. *Tunn Undergr Space Technol* 91:102994
- Zhou L, Wang X, Ye A (2019) Shake table test on transverse steel damper seismic system for long span cable-stayed bridges. *Eng Struct* 179:106–119
- Chang C-Y, Huang C-W (2020) Non-contact measurement of inter-story drift in three-layer rc structure under seismic vibration using digital image correlation. *Mech Syst Signal Process* 136:106500
- Bai Y, Li Y, Tang Z, Bittner M, Broggi M, Beer M (2021) Seismic collapse fragility of low-rise steel moment frames with mass irregularity based on shaking table test. *Bull Earthq Eng* 19:2457–2482. <https://doi.org/10.1007/s10518-021-01076-2>
- Wani ZR, Tantray M, Farsangi EN (2022) In-plane measurements using a novel streamed digital image correlation for shake table test of steel structures controlled with mr dampers. *Eng Struct* 256:113998
- Yu Y, Han R, Zhao X, Mao X, Hu W, Jiao D, Li M, Ou J (2015) Initial validation of mobile-structural health monitoring method using smartphones. *Int J Distrib Sens Netw* 11(2):274391
- Bedon C, Bergamo E, Izzi M, Noè S (2018) Prototyping and validation of mems accelerometers for structural health monitoring—the case study of the pietratagliata cable-stayed bridge. *J Sens Actuator Netw* 7(3):30
- Bianchini N, Mendes N, Calderini C, Candeias P, Rossi M, Lourenco P (2022) Seismic response of a small-scale masonry cross vault: experimental investigation by performing quasi-static and shake table tests. <https://link.springer.com/article/10.1007/s10518-021-01280-0>
- Jing H, Chen H, Yang J, Li P (2022) Shaking table tests on a small-scale steel cylindrical silo model in different filling conditions. In: *Structures*, vol 37. Elsevier, pp 698–708
- Park MJ, Cheon G, Alemayehu RW, Ju YK (2022) Seismic performance of f3d free-form structures using small-scale shaking table tests. *Materials* 15(8):2868
- Chen J, Chen X, Liu W et al (2014) Complete inverse method using ant colony optimization algorithm for structural parameters and excitation identification from output only measurements. *Math Probl Eng* 2014
- Wang D, Xiang W, Zeng P, Zhu H (2015) Damage identification in shear-type structures using a proper orthogonal decomposition approach. *J Sound Vib* 355:135–149
- Oh BK, Kim D, Park HS (2017) Modal response-based visual system identification and model updating methods for building structures. *Comput-Aided Civ Infrastruct Eng* 32(1):34–56
- Singh T, Sehgal S, Prakash C, Dixit S (2022) Real-time structural health monitoring and damage identification using frequency response functions along with finite element model updating technique. *Sensors* 22(12):4546
- Adam C (2001) Dynamics of elastic–plastic shear frames with secondary structures: shake table and numerical studies. *Earthq Eng Struct Dyn* 30(2):257–277
- Bianchini N, Mendes N, Calderini C, Candeias PX, Rossi M, Lourenço PB (2022) Seismic response of a small-scale masonry groin vault: experimental investigation by performing quasi-static and shake table tests. *Bull Earthq Eng* 20(3):1739–1765
- Zhong C, Christopoulos C (2023) Scaled shaking table testing of higher-mode effects on the seismic response of tall and slender structures. *Earthq Eng Struct Dyn* 52(3):549–570
- Damcı E, Şekerci Ç (2019) Development of a low-cost single-axis shake table based on arduino. *Exp Tech* 43(2):179–198
- Chen Z, Keating D, Shethwala Y, Saravanakumaran AAP, Arrow-smith R, Kottke A, Wittich C, Das J (2023) Shakebot: a low-cost, open-source robotic shake table for earthquake research and education

22. Inc. TM (2023) MATLAB Version: 9.14.0.2254940 (R2023a) Update 2. <https://www.mathworks.com>
23. Shinozuka M, Deodatis G (1988) Response variability of stochastic finite element systems. *J Eng Mech* 114(3):499–519
24. Shinozuka M, Deodatis G (1991) Simulation of stochastic processes by spectral representation. *Appl Mech Rev* 44(4):191–204. <https://doi.org/10.1115/1.3119501>
25. Grashorn J, Bittner M, Banse M (2024) NamazuST/Namazu: Pre-release. <https://zenodo.org/doi/10.5281/zenodo.10533795>
26. Chen J, Kong F, Peng Y (2017) A stochastic harmonic function representation for non-stationary stochastic processes. *Mech Syst Signal Process* 96:31–44. <https://doi.org/10.1016/j.ymsp.2017.03.048>
27. Lyu M-Z, Chen J-B, Shen J-X (2023) Refined probabilistic response and seismic reliability evaluation of high-rise reinforced concrete structures via physically driven dimension-reduced probability density evolution equation. *Acta Mech*, 1619–6937. <https://doi.org/10.1007/s00707-023-03666-4>
28. Liu S, Peng L, Liu J, Zhao S, Jiang Z (2022) Spectral representation-based efficient simulation method for fully non-stationary spatially varying ground motions. *Soil Dyn Earthq Eng* 161:107436. <https://doi.org/10.1016/j.soildyn.2022.107436>
29. Li J, Chen J (2010) *Stochastic dynamics of structures*, 1st edn. John Wiley & Sons, Ltd (Asia), United States. <https://www.wiley.com/en-us/Stochastic+Dynamics+of+Structures-p-9780470824252>
30. Newland DE (2012) *An introduction to random vibrations, spectral & wavelet analysis*. Dover Publication, United States
31. Behrendt M, Kitahara M, Kitahara T, Comerford L, Beer M (2022) Data-driven reliability assessment of dynamic structures based on power spectrum classification. *Eng Struct* 268:114648. <https://doi.org/10.1016/j.engstruct.2022.114648>
32. Huang Z, Xu Y-L (2021) A multi-taper s-transform method for spectral estimation of stationary processes. *IEEE Trans Signal Process* 69:1452–1467. <https://doi.org/10.1109/TSP.2021.3057488>
33. Dufour J-E, Beaubier B, Hild F, Roux S (2015) Cad-based displacement measurements with stereo-dic. *Exp Mech* 55(9):1657–1668

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

J. Grashorn^{1,2}  · M. Bittner^{1,2}  · M. Banse³ · X. Chang⁴ · M. Beer^{1,2,5,6} · A. Fau^{2,4}

✉ M. Bittner
bittner@irz.uni-hannover.de

J. Grashorn
grashorn@irz.uni-hannover.de

M. Banse
marvin.banse@uol.de

X. Chang
xuyang.chang@ens-paris-saclay.fr

M. Beer
beer@irz.uni-hannover.de

A. Fau
amelie.fau@ens-paris-saclay.fr

¹ Institute for Risk and Reliability, Leibniz University Hannover, Callinstr. 34, 30167 Hannover, Lower Saxony, Germany-mail: {grashorn,bittner,beer}@irz.uni-hannover.de

² International Research Training Group 2657 (IRTG 2657), Leibniz University Hannover, Appelstr. 11a, 30167 Hannover, Lower Saxony, Germany-mail: amelie.fau@ens-paris-saclay.fr

³ Department for Computer Science, Carl von Ossietzky University, Ammerländer Heerstr. 114-118, 26111 Oldenburg, Lower Saxony, Germany-mail: marvin.banse@uol.de

⁴ LMPS-Laboratoire de Mécanique Paris-Saclay, Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, 4 Avenue des Sciences, Gif-sur-Yvette 91191, Île-de-France, France-mail: xuyang.chang@ens-paris-saclay.fr

⁵ Institute for Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZL, Liverpool 610101, England, UK

⁶ International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Siping road 1239, Shanghai 200092, China