

# Shakebot: A Low-cost, Open-source Robotic Shake Table for Earthquake Research and Education

Zhiang Chen, Devin Keating, Yash Shethwala, Aravind Adhith Pandian Saravanakumaran, Ramon Arrowsmith, Albert Kottke, Christine Wittich, Jnaneshwar Das

**Abstract**—Shake tables provide a critical tool for simulating earthquake events and testing the response of structures to seismic forces. However, existing shake tables are either expensive or proprietary. This paper presents the design and implementation of a low-cost, open-source shake table named *Shakebot* for earthquake engineering research and education, built using Robot Operating System (ROS) and robotic concepts. The Shakebot adapts affordable and high-accuracy components from 3D printers, particularly a closed-loop stepper motor for actuation and a toothed belt for transmission. The stepper motor enables the bed to reach a maximum horizontal acceleration of  $11.8 \text{ m/s}^2$  ( $1.2 \text{ g}$ ), and velocity of  $0.5 \text{ m/s}$ , with a  $2 \text{ kg}$  specimen. The Shakebot is equipped with an accelerometer and a high frame-rate camera for bed motion estimation. The low cost and easy use make the Shakebot accessible to a wide range of users, including students, educators, and researchers in low-resource settings. An important application of the Shakebot is to examine the dynamics of precariously balanced rocks (PBRs), which are negative indicators of earthquakes in nature. Our earlier research built a virtual shake robot in simulation for the PBR study. The Shakebot provides an approach to validate the simulation through physical experiments. The ROS-based perception and motion software facilitates the code transition from our virtual shake robot to the physical Shakebot. The reuse of the control programs ensures that the implemented ground motions are consistent for both the simulation and physical experiments, which is critical to validate our simulation experiments.

## I. INTRODUCTION

For decades, researchers have used shake tables to simulate the effects of earthquakes on structures, systems, and materials, which provides a critical approach to earthquake research and education [1, 2]. Simulated earthquake conditions can help civil engineers identify potential weaknesses in the design and suggest ways to improve the structure's resilience [3, 4]. In a controlled laboratory setting, shake tables can also be used to study the behavior of fragile geological features, which aids seismologists in finding upper-bound constraints on ground motions [5]. By simulating earthquakes in the classroom, students can better understand the forces involved and the factors that influence the behavior of structures [6], inspiring the next generation of engineers and seismologists.

An important application of shake tables is to examine the dynamics of precariously balanced rocks (PBRs), which are free-standing rocks balanced on, but not attached to sub-horizontal pedestals [7, 8, 9]. Because an earthquake ground motion of sufficient amplitude and duration may topple PBRs, the fragile configurations of PBRs provide upper-bound constraints on ground motions, refining probabilistic

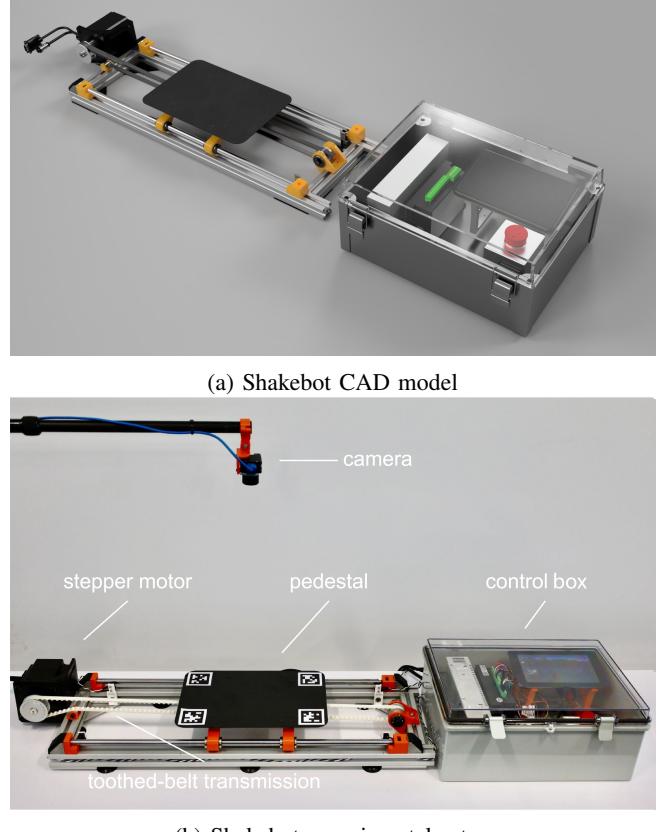


Fig. 1: Shakebot CAD model and actual experimental setup. Stepper motor actuates pedestal through toothed belt and pulley transmission. Control box includes power supply, stepper motor driver, Raspberry Pi, emergency push button, and touchscreen. For scale, the control box has a dimension of  $390 \times 290 \times 160 \text{ mm}$ .

seismic hazard analysis [10, 11]. The dynamics of the PBR overturning process are non-linear [12, 5]. Purvance et al. [5] used a shake table to study the PBR overturning dynamics. However, difficulties can arise when using conventional shake tables to repeat overturning experiments. For example, when analyzing PBR fragility, shake experiments would involve accurate and repeated positioning of a large boulder on the shake table to record the overturning responses given ground motions. These boulders are often so heavy that the use of a crane or forklift is needed, and repositioning them precisely is difficult (e.g., [13]), complicating experiment repeatability.

TABLE I: Shake tables and their specifications.

	Open source	Cost (USD)	Payload (kg)	Vel (m/s)	Acc (g)	Software	Programmable	Transmission
Quanser [14]	×	30k	7.5	0.4	2.5	NA	✓	ball screw
CUSHAKE [15]	×	45k	3500	0.01	1	DEPSIM	✓	threaded rod
Kelvin [16]	×	1.3k	NA	NA	NA	NA	✗	reciprocating arm
SARSAR [17]	✓	1.8k	200 (0.4g)	0.35	1.5	NA	✓	ball screw
Shao and Enyart [18]	×	NA	228	NA	4	NA	✓	hydraulic actuators
Kinay [19]	✓	NA	80	0.25	2	Scilab	✓	ball screw
QuakeLogic [20]	×	15k	50	0.5	1	TESTLAB	✓	ball screw
Shakebot (ours)	✓	1.3k	2	0.5	1.2	ROS	✓	toothed belt

Note: Shakebot cost does not include camera or logistics. The cost of SARSAR is estimated based on available items in the United States.

The Specifications of payload, velocity, and acceleration are the maximum values. SARSAR has 200 kg maximum payload at 0.4 g acceleration.

In an effort to reduce variability, explore a broad parameter space of ground motions, and automate PBR overturning analyses, our previous study developed a virtual shake robot in simulation [21]. The virtual shake robot utilized core technologies in robotics, such as Robot Operating System (ROS), Gazebo simulation toolbox, and Bullet physics engine. Although the virtual shake robot allows us to conduct overturning experiments in simulation, the results need to be validated through physical experiments. To complete this validation, we designed and built a low-cost, small-scale shake table, called *Shakebot* (Fig. 1).

The Shakebot provides a reverse method for simulation validation. Physics engines have been used for robotic development. However, when robotic systems are transitioned from simulations to the real world, simulation gaps may cause critical failures (e.g., in the arena of reinforcement learning [22, 23]). When using a physics engine for scientific studies, we also need to address the simulation gap to calibrate and scale-up the simulation experiments. Using the Shakebot, we test the overturning dynamics of down-scaled, 3D-printed PBRs. We build the same PBR models in simulation and validate the overturning dynamics using our virtual shake robot. Following this approach, we are able to compare the overturning patterns obtained from the Shakebot and the virtual shake robot, further allowing us to quantify the uncertainty in simulation. The Shakebot serves as a validation tool for simulation studies, aiding in learning more about the real-time dynamics of the PBRs and potentially calibrating physics engine parameters to reduce the simulation gaps.

The Shakebot demonstrates interdisciplinary research in the use of automated robotic systems for geoscience studies, and enhances the impact of robotics in the natural sciences. Besides its contribution to seismology, the key contributions of this study include:

- Shakebot is the first tabletop shake table developed based on robotic concepts (perception and motion systems) and tools (ROS). Because of using ROS, the control systems are reused from our previous virtual shake robot, ensuring that the ground motions (the motion of the bed) are consistent in physical and simulation experiments. This consistency is vital for simulation validation.
- The hardware mechanism is simplified by design. The

Shakebot adopts a closed-loop stepper motor with a toothed belt and pulley transmission mechanism. Most of the mechanical parts and components are off-the-shelf, and only a few parts are 3D-printed. The simplified design lowers costs and expedites prototyping. Additionally, for safety anticipation, we employ a double-emergency mechanism to allow system override in case of malfunction.

- Compared with traditional tabletop shake tables that only use accelerometers, our Shakebot also leverages a top-down view camera to estimate ground motion. The perception system provides an option of fusing accelerations and camera-based displacements for ground velocity estimation.
- We have open-sourced our hardware and software to promote robotic applications in seismic research and education: [https://github.com/DREAMS-lab/asu\\_shake\\_table](https://github.com/DREAMS-lab/asu_shake_table). Despite focusing on PBR dynamics in this study, the Shakebot provides an affordable and accessible platform for other earthquake research and education.

The paper is structured as follows. In Sec. II, we discuss commercial and academic shake tables available for civil engineering and seismology, and highlight a few of their capabilities and limitations. From Sec. III to Sec. VI, we describe the mechanical design, the perception system, the motion system, and the calibration system. In Sec. VII, we discuss the results of our experiments, followed by our conclusions and future directions in Sec. VIII.

## II. RELATED WORK

Full- or large-scale shake tables and tabletop shake tables have different applications for research and education. Full-scale shake tables are designed to simulate earthquakes on full-scale buildings, bridges, and other large structures [24]. Full-scale shake tables, which usually incorporate acceleration and displacement sensors for ground motion estimations, are expensive to build and maintain. The high cost limits their availability in research and education. Full-scale shake tables may also involve safety considerations in operation and damage prevention.

Tabletop shake tables, on the other hand, are typically used to test smaller models and structure components. Because of their low cost, these tables are also useful in educational settings to teach students about the behavior of structures

under earthquake conditions. TABLE I compares existing tabletop shake tables and the Shakebot. Existing tabletop shake tables are comparatively expensive [14, 15, 20] or only allow a fixed ground motion pattern [16]. None of the existing tabletop shake tables have open-sourced their hardware. Another limitation of existing shake tables is that they solely employ accelerometers to measure ground motions.

### III. SYSTEM DESCRIPTION

In this section, we discuss the mechanical design, motor selection strategy, and safety mechanism. The system is designed to satisfy the ground motion requirements—maximum dynamic acceleration of  $11.8 \text{ m/s}^2$  ( $1.2 \text{ g}$ ), maximum velocity of  $0.5 \text{ m/s}$ , and maximum displacement of  $0.45 \text{ m}$ , with maximum specimen (e.g., PBR) payload mass of  $2 \text{ kg}$ .

#### A. Hardware Mechanism

The two main components of the Shakebot are a chassis and a control box, as shown in Fig. 1. The chassis is constructed of an extruded aluminum T-slot frame that supports two linear shafts ( $625 \text{ mm}$  length for each shaft). A carriage is mounted on the linear shafts with ball bearings, which reduce the friction between the linear shafts and the carriage. A flatbed (pedestal) is attached to the carriage to hold specimens. A stepper motor (i.e., NEMA 34) actuates the carriage through a toothed belt and pulley transmission. The stepper motor driver (CL86T), contained in the control box, drives the stepper motor. The control box also houses the power supply (S-350-60), an emergency stop button, a touchscreen for the user interface, and a Raspberry Pi that handles trajectory generation and low-level system control.

#### B. Motor Selection

To select a stepper motor, we considered the stepper motor torque at its maximum angular velocity to satisfy the required ground motion acceleration and velocity. Because stepper motor torque decreases with velocity, if the torque at the maximum velocity satisfies the acceleration requirement, the stepper motor can also achieve so at any smaller velocities. We selected Nema 34HS31, a closed-loop stepper motor (with an encoder) with a torque output of  $1.56 \text{ Nm}$  at the maximum speed of  $1200 \text{ RPM}$ . The maximum translational force is calculated from the torque,

$$F = ma = \frac{\tau}{r} \quad (1)$$

where  $F$  is the translational force,  $r$  is the outside radius of the toothed pulley,  $m$  is the maximum payload mass (including carriage and specimen),  $a$  is the maximum translational acceleration, and  $\tau$  is the stepper motor torque. We select a toothed pulley with an outside radius of  $25.91 \text{ mm}$ . The translational force at the maximum speed is  $60.21 \text{ N}$ . Considering a maximum total payload mass  $m$  of  $4 \text{ kg}$  (including a  $2 \text{ kg}$  carriage and  $2 \text{ kg}$  specimen), the corresponding acceleration produced from the translational motion is  $15.1 \text{ m/s}^2$  ( $1.54 \text{ g}$ ). The ratio of the calculated

acceleration to the required acceleration is  $1.28$ , which is adequate empirically to compensate for the friction and energy losses in transmission. Additionally, at the maximum stepper motor speed of  $1200 \text{ RPM}$ , the translational speed can reach  $0.52 \text{ m/s}$ , which also satisfies our design requirement for PBR studies. Note that we used the outside radius on the specifications of the toothed pulley for the calculation here. The outside radius value is appropriate for the stepper motor selection. We conducted a calibration process that uses a camera and several fiducial markers to estimate this radius precisely for accurate control. The calibration process is discussed in Sec. VI-B.

The stepper motor driver reads signals from the encoder attached to the stepper motor, forming a closed-loop system to control the stepper motor. The microstepping technology in the stepper motor driver allows a higher step resolution ( $1.8^\circ/2000$  per step), which achieves smooth motions at low speeds (as low as  $0.001 \text{ m/s}$  in our implementation).

#### C. Safety Mechanism

The design incorporates two levels of safety features. The first level consists of a pair of limit emergency switches situated at either end of the carriage's range of motion, as shown in Fig. 2. These switches link to the emergency stop function on the stepper motor driver. Once either of these switches is triggered, the stepper motor is disabled. The second level of protection is supplied by an emergency push button in the control box, which disconnects the power supply from the stepper motor driver. The emergency push button can be manually activated by the Shakebot's operator. Fig. 2 shows a top-down view of the Shakebot.

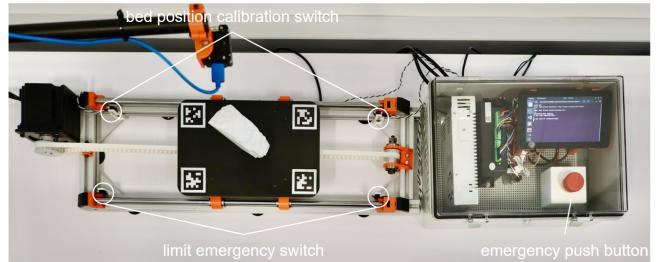


Fig. 2: Shakebot from top-down view illustrating double emergency mechanism.

### IV. PERCEPTION SYSTEM

The perception system estimates the status of the bed using a camera and an accelerometer. Four fiducial markers are affixed to the corners of the bed (Fig. 1b and Fig. 2). A top-down view camera (FLIR Chameleon®3 Color Camera CM3-U3-13Y3C-CS 1/2") captures the position and orientation of the bed using the ROS package *apriltag\_ros* [25, 26]. Additionally, an accelerometer (Wit Motion HWT905-TTL) attached underneath the bed measures acceleration (ground motion acceleration). We directly obtain ground motion displacement and acceleration from the fiducial marker detection and accelerometer, respectively. We additionally esti-

mate the ground motion velocities by fusing the information from the fiducial marker detection and accelerometer.

### A. Displacement Estimation

The camera estimates ground motion displacement by detecting the fiducial markers at each corner of the bed. The *apriltag\_ros* package provides the pose of each visible marker in the camera coordinates. When the bed moves from one position to another, we obtain the relative pose of a marker attached to the bed,

$$H_i^j = H_c^j \cdot H_i^c = [H_j^c]^{-1} \cdot H_i^c \quad (2)$$

where  $i$  and  $j$  indicate marker positions at two time stamps,  $c$  indicates the camera coordinates,  $H_a^b$  is the transformation matrix for position  $a$  with respect to position  $b$ .  $H_j^c$  and  $H_i^c$  are obtained using the *apriltag\_ros* package.

We extract the translation vector from  $H_i^j$  as the relative displacement of the fiducial marker at two positions. We estimate ground motion displacement by averaging the relative displacements from all visible fiducial marker detections. When a fiducial marker is obstructed by a specimen, the ground motion displacement estimation only relies upon the remaining visible fiducial markers. Note that the relative displacement from *apriltag\_ros* is inaccurate despite intrinsic camera calibration. To address this issue, perception calibration is performed (Sec. VI-A).

### B. Acceleration Estimation

The accelerometer installed beneath the bed measures the ground motion acceleration vector in three dimensions. However, the bed's motion is limited to one degree-of-freedom prismatic movement. To obtain the acceleration along the bed movement direction, we align the accelerometer's x-axis with the bed's direction of motion. However, the manual alignment is not perfect. We additionally calculate the absolute value (Euclidean distance) of the acceleration vector, because the absolute acceleration value is the same value of the acceleration vector along the bed movement direction. To eliminate high-frequency vibration, we apply a low-pass filter to the raw acceleration data.

### C. Velocity Estimation

We estimate ground motion velocity by fusing the derived velocities from the fiducial marker detection and accelerometer data. Fig. 3 illustrates the velocity fusion process. During a shake experiment, the timestamped ground motion displacements and accelerations are recorded. After the shake experiment, we implement numerical derivation to obtain velocity  $v_d(t)$  from the ground motion displacements  $d(t)$ . From the ground motion accelerations  $a(t)$ , we conduct numerical integration to obtain velocity  $v_a(t)$ .

From the recordings of fiducial markers and accelerometer, we obtain two sets of derived velocities:  $\mathbf{v}_d = \{v_d(t_1), \dots, v_d(t_M)\}$  and  $\mathbf{v}_a = \{v_a(t_1), \dots, v_a(t_N)\}$ . We phrase the ground motion velocity estimation as a regression problem with an objective function that minimizes the square of

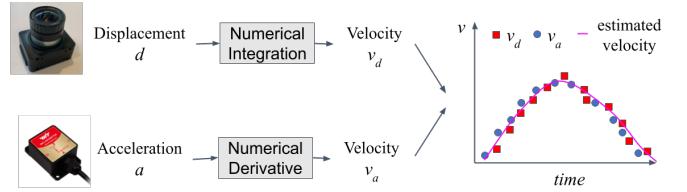


Fig. 3: Velocity fusion workflow. Estimated velocity is regression of velocities from displacement and acceleration.

errors between estimated and derived velocities,

$$\arg \min_{\theta} \|\hat{v}(t; \theta) - \mathbf{v}_s(t)\| \quad (3)$$

where  $\hat{v}(t; \theta)$  is a parametric ground motion velocity function, and  $\mathbf{v}_s = \mathbf{v}_d \cup \mathbf{v}_a$  is a set of velocities derived from measurements. In this ground motion velocity estimation setting,  $\{v_d(t)\}$  and  $\{v_a(t)\}$  need not be synchronized, simplifying the electronic hardware configuration of our Shakebot. The fiducial marker detection and accelerometer have different frame rates. The parametric regression function  $\hat{v}(t; \theta)$  has many options. We use a 6<sup>rd</sup> order polynomial function for demonstration in Sec. VII.

## V. MOTION SYSTEM

The Shakebot has a hierarchical motion system and a user interface (UI) to facilitate shake experiments. The hierarchical motion system provides two options to emulate ground motions: single-pulse cosine displacement motion and realistic ground motion from seismogram acceleration recordings. The UI assists users in repeating the shake experiment where single-pulse cosine displacement motions are deployed with different ground motion parameters.

### A. Motor Control

The Shakebot motion is enabled by a hierarchical motion planner in the Raspberry Pi, which outputs PWM signals to the closed-loop stepper motor driver. Fig. 4 shows the hierarchical motion system. The first ground motion option is a single-pulse cosine displacement,

$$d(t) = A - A \cos(2\pi ft), \quad (4)$$

where  $d(t)$  is the ground displacement function,  $A$  is the amplitude,  $f$  is the frequency, and  $t \in [0, 1/f]$  is time.  $A$  and  $f$  are derived from ground motion parameters,

$$f = \frac{1}{2\pi\kappa} \quad (5)$$

$$A = \frac{\alpha g}{4\pi^2 f^2} \quad (6)$$

where  $\alpha$  is the peak ground acceleration (PGA),  $\kappa$  is the ratio of peak ground velocity to peak ground acceleration (PGV/PGA), and  $g$  is the gravitational acceleration. As shown in Fig. 4, the motion interpreter converts (PGV, PGA) to  $(A, f)$  using Eqs. 5-6. With  $(A, f)$ , the trajectory planner

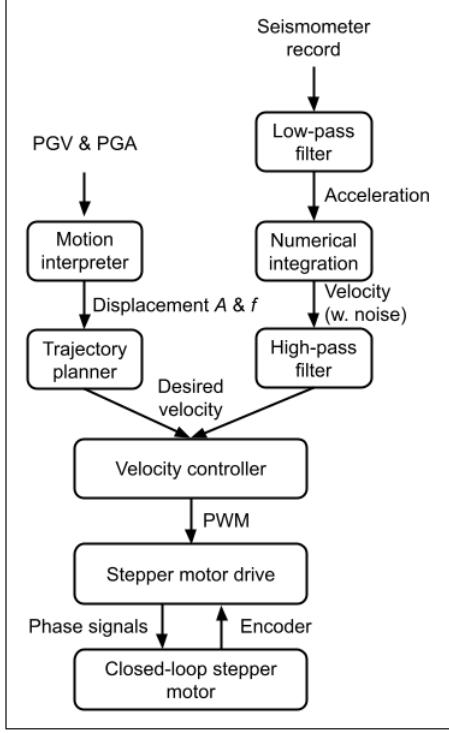


Fig. 4: Motion system supports two types of ground motions: single-pulse cosine displacement and realistic ground motion from seismogram acceleration recordings. Single-pulse cosine displacement is defined by peak ground velocity (PGV) and peak ground acceleration (PGA).

takes derivative of the cosine displacement function (Eq. 4) to obtain ground velocity function,

$$v(t) = 2\pi Af \sin(2\pi ft) \quad (7)$$

where  $v(t)$  is the ground velocity function, and  $t \in [0, 1/f]$  is time. From the ground velocity function, we uniformly sample velocity points  $\{v\}$  as the input of the velocity controller. The sampling frequency is a user-defined parameter (usually 200 Hz). Note that the velocity commands  $\{v\}$  are desired translational velocities of the bed. The velocity controller converts the translation velocity to the angular velocity of the stepper motor,

$$\omega(t) = \frac{v(t)}{r} \quad (8)$$

where  $\omega(t)$  is angular velocity for the stepper motor, and  $r$  is the radius of the toothed pulley. The angular velocity commands are further converted to PWM signals for the input of the stepper motor driver. The stepper motor driver, stepper motor, and encoder attached to the stepper motor form a closed-loop control to execute the PWM signals.

Besides the cosine ground displacement function, the Shakebot also supports ground motion from observed seismograms. As shown in Fig. 4, we first implement a low-pass filter to remove the high-frequency noise in the raw acceleration data. Numerical integration computes velocities from the accelerations. Then we have a high-pass filter to

remove the low-frequency noise in the velocities, because the low-frequency noise in velocities may cause accumulated displacement errors. The output of the high-pass filter is a set of velocities  $\{v\}$ , which have the same format as the output from the trajectory planner. Because both the trajectory planner and high-pass filter had the same output format, we share a velocity controller to process the desired velocity commands.

### B. Control User Interface

The UI is a portal through which users conduct shake experiments to study PBR overturning responses. Once initialized, the UI prompts the user to place a PBR on the bed. Then the user needs to input (PGV/PGA, PGA), which is converted to (PGV, PGA). The motion module (Fig. 4) executes a single-pulse cosine displacement ground motion based on the (PGV, PGA). After the ground motion is completed, the UI prompts the user to input the overturning response of the PBR (i.e., being toppled or remaining balanced). The overturning responses with ground motion parameters and perception estimates are saved in a CSV file. Based on the user's input, the UI exits or continues to conduct a new overturning experiment.

## VI. SYSTEM CALIBRATION

The Shakebot needs to be calibrated before conducting the shake experiments. We first calibrate the camera intrinsics and fiducial marker detection, and this calibrated perception system is used to facilitate velocity controller calibration.

### A. Perception System Calibration

The perception system requires intrinsic camera calibration and fiducial marker detection scaling. We used the *camera\_calibration* ROS package [27] to obtain the intrinsic camera parameters including scale factor, focal length, pixel dimension, and distortion. Despite the intrinsic camera calibration, the translation estimation from *apriltag\_ros* was found to be inaccurate in our implementation. To calibrate this fiducial marker detection process, we estimate a multiplicative factor between the fiducial marker displacement and manually measured displacement,

$$D = \sigma d \quad (9)$$

where  $D$  is the manually measured displacement,  $\sigma$  is the multiplication factor, and  $d$  is the displacement from fiducial marker detection.

To better estimate  $\sigma$ , we have an overdetermined system by sampling a number of displacement points along the linear shaft  $\{d\}$ ,

$$\mathbf{D} = \sigma \mathbf{d} \quad (10)$$

where  $\mathbf{D} = [D_1, D_2, \dots, D_N]^T$  is a vector of manually measured displacements, and  $\mathbf{d} = [d_1, d_2, \dots, d_N]^T$  is a vector of fiducial marker displacements. The Moore–Penrose inverse is applied to solve the overdetermined system,

$$\sigma = \mathbf{D} \mathbf{d}^T [\mathbf{d} \mathbf{d}^T]^{-1} \quad (11)$$

We utilize the limit switches at the ends of the linear shaft (i.e., bed position calibration switches shown in Fig. 2) to facilitate the measurement of bed displacement,  $D$ . To begin the perception system calibration, the bed moves to the left end until the left bed position calibration switch is triggered. At the same time, the fiducial marker detection at this position is recorded. The bed then moves to the right end, triggering the right calibration switch, and the fiducial marker detection is recorded again. We input the manually measured length between the two calibration switches. During the perception calibration, we have a position controller that rotates the motor shaft by one angular step given one PWM signal. As the bed moves from one end to the other, we count and record the motor steps, from which we calculate the bed displacement per angular step (translational step resolution). Then, the bed randomly moves along the linear shaft to sample data points, such as step counts (from the left calibration switch) and fiducial marker detections. Using the step counts and translation step resolution, we calculate the actual bed displacements. The samples of the actual bed displacements and fiducial marker displacements are used in Eq. 11 to estimate  $\sigma$ .

### B. Velocity Controller Calibration

The velocity controller calibration aims to reduce the uncertainties from the toothed belt transmission. Because of the closed-loop stepper motor, we assume that the error between the actual angular velocity and the desired angular velocity of the stepper motor is diminished. Given a desired translational velocity of the bed, Eq. 8 calculates the corresponding desired angular velocity. However, the toothed pulley radius  $r$  is close but not equal to the outside radius of the pulley. The uncertainty in  $r$  comes from factors such as the tightness of the toothed belt, manufacturing imprecision, and tooth geometry. We correct for such uncertainty using a multiplicative factor,

$$\omega'(t) = \frac{\gamma v(t)}{r} \quad (12)$$

where  $\gamma$  is a constant that captures the first-order systematic uncertainty in the toothed belt transmission. The velocity controller calibration needs to estimate a  $\gamma$  that minimizes the error between the desired and actual translational velocities. Because  $\gamma$  is a factor that calibrates velocity, we denote  $\gamma v(t)$  as calibrated velocity. Instead of directly measuring translational velocity in real time, we conduct half-cosine displacement motions (i.e.,  $t \in [0, 1/(2f)]$  in Eq. 4) and estimate  $\gamma$  from displacement differences,

$$D' = \gamma D \quad (13)$$

where  $D$  is the desired half-cosine displacement, and  $D'$  is the actual half-cosine displacement measured from the fiducial marker detection. Similarly, we randomly sample the half-cosine displacement motions and use the Moore–Penrose inverse to estimate  $\gamma$ . Using displacement difference to estimate velocity difference has the advantage of easy implementation, because directly measuring translation velocity is challenging.

We briefly demonstrate that  $\gamma$  estimated from the displacements is the same multiplicative factor for velocity calibration. Based on Eq. 13, we have calibrated displacement function,

$$D'(t) = \gamma A - \gamma A \cos(2\pi ft). \quad (14)$$

Note that  $\gamma$  here is estimated from the displacement differences described above. We take the derivative of  $D'(t)$  and obtain the calibrated velocity function,

$$v'(t) = \dot{D}'(t) = 2\pi\gamma Af \sin(2\pi ft). \quad (15)$$

We substitute Eq. 7 into Eq. 15,

$$v'(t) = \gamma v(t). \quad (16)$$

Therefore,  $\gamma$  estimated from displacements is also the factor that calibrates velocity.

## VII. EXPERIMENTS AND DISCUSSION

We conducted experiments to demonstrate an application of the Shakebot in PBR fragility studies and to test its performance in validating simulation. These experiments focused on a PBR at the Double Rock site in coastal Central California [11]. The PBR, as shown in Fig. 5, was mapped by an unpiloted aerial vehicle and constructed using structure from motion and Poisson reconstruction algorithms [21].

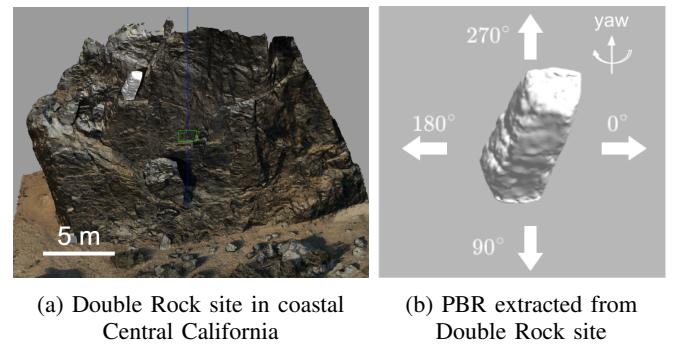


Fig. 5: Precariously Balanced Rock (PBR) at Double Rock site in coastal Central California. The yaw orientation indicates the relative direction of single-pulse cosine displacement motion.

The first experiment demonstrates the Shakebot application in studying the fragility anisotropy of PBRs. We down-scaled the Double Rock PBR from 151.0 cm to 12.0 cm in height and 3D-printed it with PLA material. The resulting PBR, denoted as 3D-printed PBR(a), had a total mass of 105g. Because of the asymmetric geometry of 3D-printed PBR(a), the overturning responses were expected to vary with different ground motion directions. To investigate this fragility anisotropy idea, we placed the 3D-printed PBR(a) with two initial orientations (yaw angles of 0° and 270°, as shown in Fig. 5) on the bed. Using the UI, we obtained the response diagrams from a set of ground motions, as shown in Fig. 6. The response diagrams indicated that 3D-printed PBR(a) oriented at a yaw angle of 0° was more fragile than at yaw angle of 270°. This result is consistent with previous

studies showing that PBRs with smaller minimal contact angles along the motion direction are more fragile [5, 28]. Additionally, the resulting boundary curves closely resembled the simulation results of the Double Rock PBR with original shape [21].

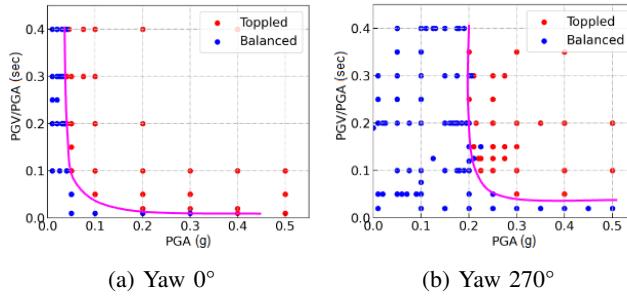


Fig. 6: Overturning response diagrams of the Double Rock PBR from orientations of yaw  $0^\circ$  and yaw  $270^\circ$ . The red and blue dots represent overturning responses of being toppled and balanced after single-pulse cosine ground motions, respectively. The curves indicate the boundaries of the overturning responses.

In the second experiment, we used the Shakebot to evaluate the overturning simulation. The Double Rock PBR was down-scaled to 12.8 cm in height and 3D-printed with the PETG material. The resulting PBR, denoted as 3D-printed PBR(b), had a total mass of 404 g. We imported the 3D model of the down-scaled PBR to Gazebo and conducted overturning experiments using the virtual shake robot [21]. To examine the simulation results, we used the Shakebot to overturn 3D-printed PBR(b). In both simulation and real-world experiments, the PBR had a yaw angle of  $0^\circ$ . As shown in Fig. 7, the overturning boundary curves from the real-world and simulation experiments are similar. Future work should explore the causes of their difference. For example, we observed that Gazebo reduced the precision of configuration parameters when passing those parameters to the Bullet physics engine. The data from real-world experiments will aid in calibrating and improving the simulation experiments.

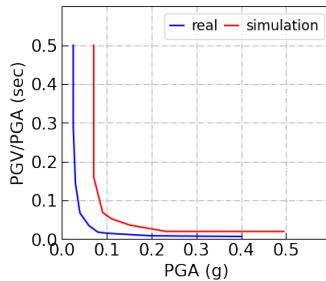


Fig. 7: Overturning boundary curves from Shakebot and virtual shake robot. Reprinted from [21]. Reprinted with permission.

Fig. 8 shows the estimation results of acceleration, displacement, and velocity from the input PGA of 0.98

$m/s^2$  and PGV/PGA of  $0.1\ s$ . According to the PGA and PGV/PGA, the desired PGV and peak ground displacement are  $0.1\ m/s$  and  $0.01\ m$ , respectively. The acceleration estimation and displacement estimation can be improved with better measurement equipment such as a better-quality accelerometer and motion capture system. Our velocity estimation method used the 6<sup>rd</sup> order polynomial function to fuse data from afford sensors, resulting in a PGV estimate of  $0.12\ m/s$ . The resulting velocity estimation is better than individual velocities derived from the accelerometer and fiducial marker detection. Future work should quantify uncertainties in displacement, velocity, and acceleration estimates. Investigations on other regression models (e.g., Gaussian process regression) and real-time state estimation may also improve the ground motion estimation.

Because validating ground motion estimation requires future investment in expensive measurement equipment, we assumed that the actual ground motions were the same as the desired ground motions during the overturning experiment. This assumption is reasonable given our hierarchical control system and calibration processes. However, future work should verify this assumption. We also anticipate that the PBR response status registration can be automated by an object detection system and PBR placement can be implemented by a robotic arm. By automating the overturning experiments, we can save on human labor and ensure accurate PBR placement for reliable results.

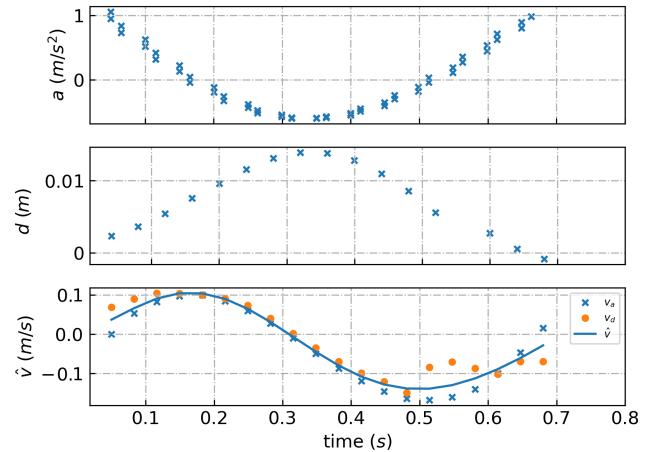


Fig. 8: Velocity estimation from regression of acceleration and displacement.

## VIII. CONCLUSION

We introduced a low-cost, open-source shake table named *Shakebot* for earthquake research and education. Leveraging concepts and tools from robotics, we simplified and advanced the hardware and software compared to the existing shake tables. We presented the technical details of the perception and motion systems, which are important for the open-source community to upgrade or customize the Shakebot. Experiments showed the applications of the Shakebot for PBR studies. Our work demonstrates an interdisciplinary

method of using robotics for natural sciences, providing a step towards robotics-enabled automated geoscience [21]. The Shakebot provides an affordable platform for education, outreach, and other ground motion experiments.

## ACKNOWLEDGEMENTS

This research was supported by the Pacific Gas and Electric Company and Southern California Earthquake Center (Contribution No. 21137). SCEC is funded by NSF Cooperative Agreement EAR-1600087 and USGS Cooperative Agreement G17AC00047. Thank you to Chris Madugo for his support and advice in the research.

## REFERENCES

- [1] L. S. Jacobsen. “Vibration research at Stanford university”. In: *Bulletin of the Seismological Society of America* 19.1 (1929), pp. 1–27.
- [2] X. Lu, G. Fu, W. Shi, and W. Lu. “Shake table model testing and its application”. In: *The Structural Design of Tall and Special Buildings* 17.1 (2008), pp. 181–201.
- [3] A. Bogdanovic, Z. Rakicevic, and E. Noroozinejad Farsangi. “Shake table tests and numerical investigation of a resilient damping device for seismic response control of building structures”. In: *Structural Control and Health Monitoring* 26.11 (2019), e2443.
- [4] Z. Rakicevic, A. Bogdanovic, E. N. Farsangi, and A. Sivandipour. “A hybrid seismic isolation system toward more resilient structures: Shaking table experiment and fragility analysis”. In: *Journal of Building Engineering* 38 (2021), p. 102194.
- [5] M. D. Purvance, A. Anooshehpoor, and J. N. Brune. “Free-standing block overturning fragilities: Numerical simulation and experimental validation”. In: *Earthquake Engineering & Structural Dynamics* 37.5 (2008), pp. 791–808.
- [6] A. Elgamal, M. Fraser, and F. McMurtin. “On-line educational shake table experiments”. In: *Journal of Professional Issues in Engineering Education and Practice* 131.1 (2005), pp. 41–49.
- [7] J. N. Brune. “Precariously balanced rocks and ground-motion maps for southern California”. In: *Bulletin of the Seismological Society of America* 86.1A (1996), pp. 43–54.
- [8] B. Shi, A. Anooshehpoor, Y. Zeng, and J. N. Brune. “Rocking and overturning of precariously balanced rocks by earthquakes”. In: *Bulletin of the Seismological Society of America* 86.5 (1996), pp. 1364–1371.
- [9] A. Anooshehpoor, J. N. Brune, and Y. Zeng. “Methodology for obtaining constraints on ground motion from precariously balanced rocks”. In: *Bulletin of the Seismological Society of America* 94.1 (2004), pp. 285–303.
- [10] J. G. Anderson, G. P. Biasi, and J. N. Brune. “Precarious rocks: Providing upper limits on past ground shaking from earthquakes”. In: *Earthquake hazard, risk and disasters*. Elsevier, 2014, pp. 377–403.
- [11] A. Rood, D. Rood, M. Stirling, C. Madugo, N. Abrahamson, K. Wilcken, T. Gonzalez, A. Kottke, A. Whittaker, W. Page, et al. “Earthquake hazard uncertainties improved using precariously balanced rocks”. In: *AGU Advances* 1.4 (2020), e2020AV000182.
- [12] C.-S. Yim, A. K. Chopra, and J. Penzien. “Rocking response of rigid blocks to earthquakes”. In: *Earthquake Engineering & Structural Dynamics* 8.6 (1980), pp. 565–587. ISSN: 1096-9845.
- [13] M. K. Saifullah and C. E. Wittich. “Seismic response of two freestanding statue-pedestal systems during the 2014 South Napa Earthquake”. In: *Journal of Earthquake Engineering* 26.10 (2022), pp. 5086–5108.
- [14] Quanser. *Quanser Shake Table II*. URL: <https://www.quanser.com/products/shake-table-ii/>.
- [15] T. Baran, A. Tanrikulu, C. Dundar, and A. Tanrikulu. “CONSTRUCTION AND PERFORMANCE TEST OF A LOW-COST SHAKE TABLE”. In: *Experimental Techniques* 35.4 (2011), pp. 8–16.
- [16] K. Educational. *Kelvin advanced earthquake simulator*. URL: <https://kelvin.com/kelvin-e-q-machine-2021/>.
- [17] E. Damci and Ç. Şekerci. “Development of a Low-Cost Single-Axis Shake Table Based on Arduino”. In: *Experimental Techniques* 43.2 (2019), pp. 179–198.
- [18] X. Shao and G. Enyart. “Development of a versatile hybrid testing system for seismic experimentation”. In: *Experimental Techniques* 38.6 (2014), pp. 44–60.
- [19] G. Kinay. “Construction and control of a desktop earthquake simulator”. MA thesis. Izmir Institute of Technology, 2006.
- [20] QUAKELOGIC. *Testbox-shaketable*. URL: [https://quakelogic.net/actuators-shaketables/testbox\\_shaketable](https://quakelogic.net/actuators-shaketables/testbox_shaketable).
- [21] Z. Chen. “Automated Geoscience with Robotics and Machine Learning: A New Hammer of Rock Detection, Mapping, and Dynamics Analysis”. PhD thesis. Arizona State University, 2022.
- [22] W. Zhao, J. P. Queraltá, and T. Westerlund. “Sim-to-real transfer in deep reinforcement learning for robotics: a survey”. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 737–744.
- [23] Z. Ding and H. Dong. “Challenges of reinforcement learning”. In: *Deep Reinforcement Learning*. Springer, 2020, pp. 249–272.
- [24] S. Gavridou, J. W. Wallace, T. Nagae, T. Matsumori, K. Tahara, and K. Fukuyama. “Shake-table test of a full-scale 4-story precast concrete building. I: Overview and experimental results”. In: *Journal of Structural Engineering* 143.6 (2017), p. 04017034.
- [25] D. Malyuta. “Guidance, Navigation, Control and Mission Logic for Quadrotor Full-cycle Autonomy”. Master thesis. 4800 Oak Grove Drive, Pasadena, CA 91109, USA: Jet Propulsion Laboratory, Dec. 2017.
- [26] J. Wang and E. Olson. “AprilITag 2: Efficient and robust fiducial detection”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 4193–4198. ISBN: 978-1-5090-3762-9.
- [27] ROS.org. *camera\_calibration*. URL: [http://wiki.ros.org/camera\\_calibration](http://wiki.ros.org/camera_calibration).
- [28] D. E. Haddad, O. Zielke, J. R. Arrowsmith, M. D. Purvance, A. G. Haddad, and A. Landgraf. “Estimating Two-dimensional Static Stabilities and Geomorphic Settings of Precariously Balanced Rocks from Unconstrained Digital Photographs”. In: *Geosphere* 8.5 (2012), pp. 1042–1053.