Supplementary Materials for

Cobble motion characterisation with smart sensors through laboratory experiments for ground-based landslide monitoring

Alessandro Sgarabotto*, Irene Manzella, Kyle Roskilly, Miles J. Clark, Georgie L. Bennett, Chunbo Luo, Aldina M. A. Franco

*Corresponding author: a.sgarabotto@bham.ac.uk
Paper DOI: 10.59236/geomorphica.v1i1.42

Table of Contents:

Supplementary Text for Detailed Methods

Sensor calibration

The measurement model for an accelerometer reads as (Frosio et al., 2009)

$$\boldsymbol{a}_g = M(\boldsymbol{a} - \boldsymbol{b}_a) \tag{1}$$

where $\boldsymbol{a}_g$ is the calibrated acceleration in unit of g, M is the error matrix, $\boldsymbol{a}$ is the acceleration after the offset and $\boldsymbol{b}_a$ is the bias. In the error matrix M, diagonal elements are the scale factors along the three axes, whereas the extra diagonal elements are due to both axes misalignment and crosstalk effects. Assuming M symmetrical, there are 8 unknown parameters in eq. (1), e.g. 5 unknowns in M and 3 unknowns in $\boldsymbol{b}_a$. These parameters are computed through nonlinear optimization so that, in static conditions, the intensity of the acceleration vector equals the g acceleration. Collecting data in N random static positions (N>30), a least squares method allows computing the unknown parameters Frosio et al. (2008). The acceleration data used for the calibration procedure are represented in Figure S1.

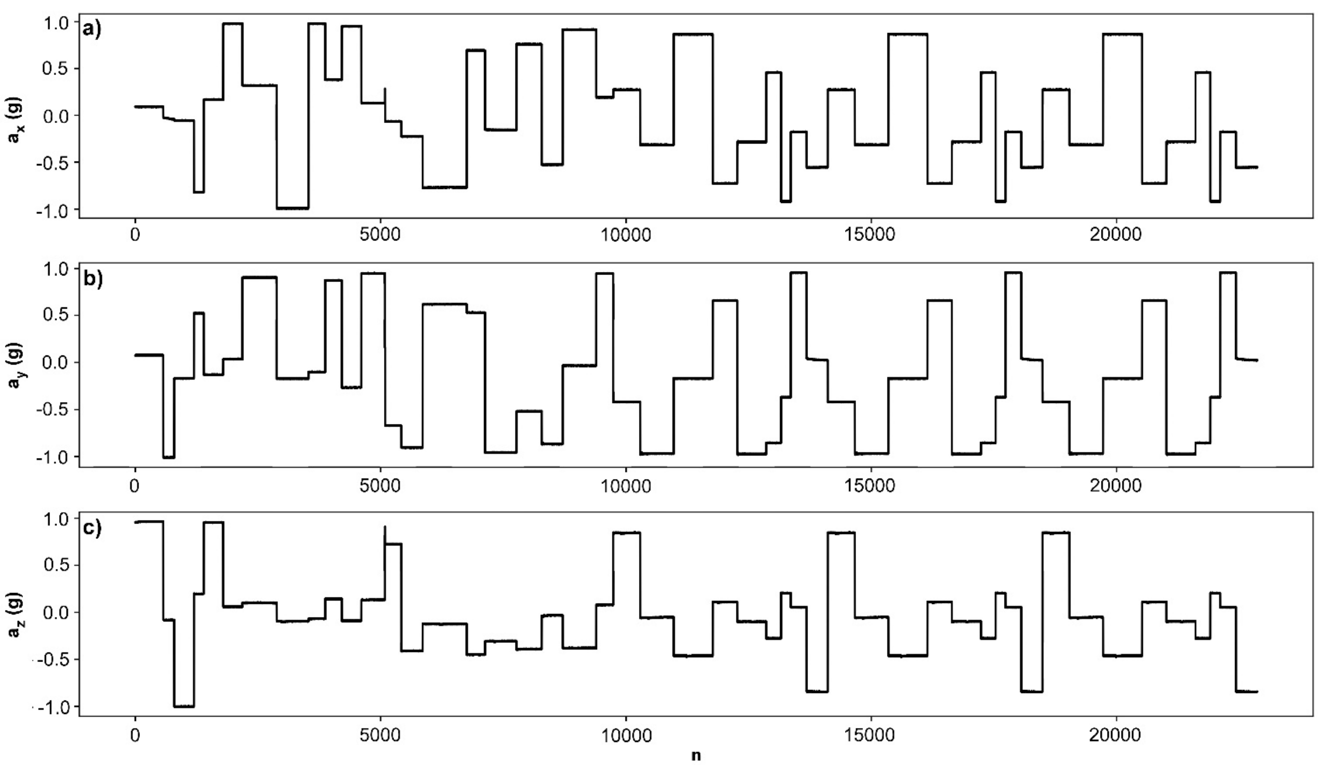

Figure S1. Raw acceleration recordings in static positions used for accelerometer calibration.

Under steady conditions, the angular velocity is expected to be null. Thus, each of the three axes of the gyro should read 0 °/s when the gyroscope is in a still position. For this reason, the simplest calibration of a gyroscope consists of calculating the offset for each axis. The offsets are measured by recording the signal when the gyroscope is in a static position and then averaging the signal recorded in each direction. Without considering additional noise sources, the measurement error model for a gyroscope applied in all three directions read as (Glueck et al., 2013)

$$\dot{\boldsymbol{\varphi}} = \dot{\boldsymbol{\varphi}}_m - \dot{\boldsymbol{\varphi}}_O \tag{2}$$

where $\dot{\boldsymbol{\varphi}}$ is the calibrated angular velocity, $\dot{\boldsymbol{\varphi}}_m$ is the gyro reading, $\dot{\boldsymbol{\varphi}}_O$ is the gyroscope offset. The gyroscope data used for the calibration procedure are represented in Figure S2.

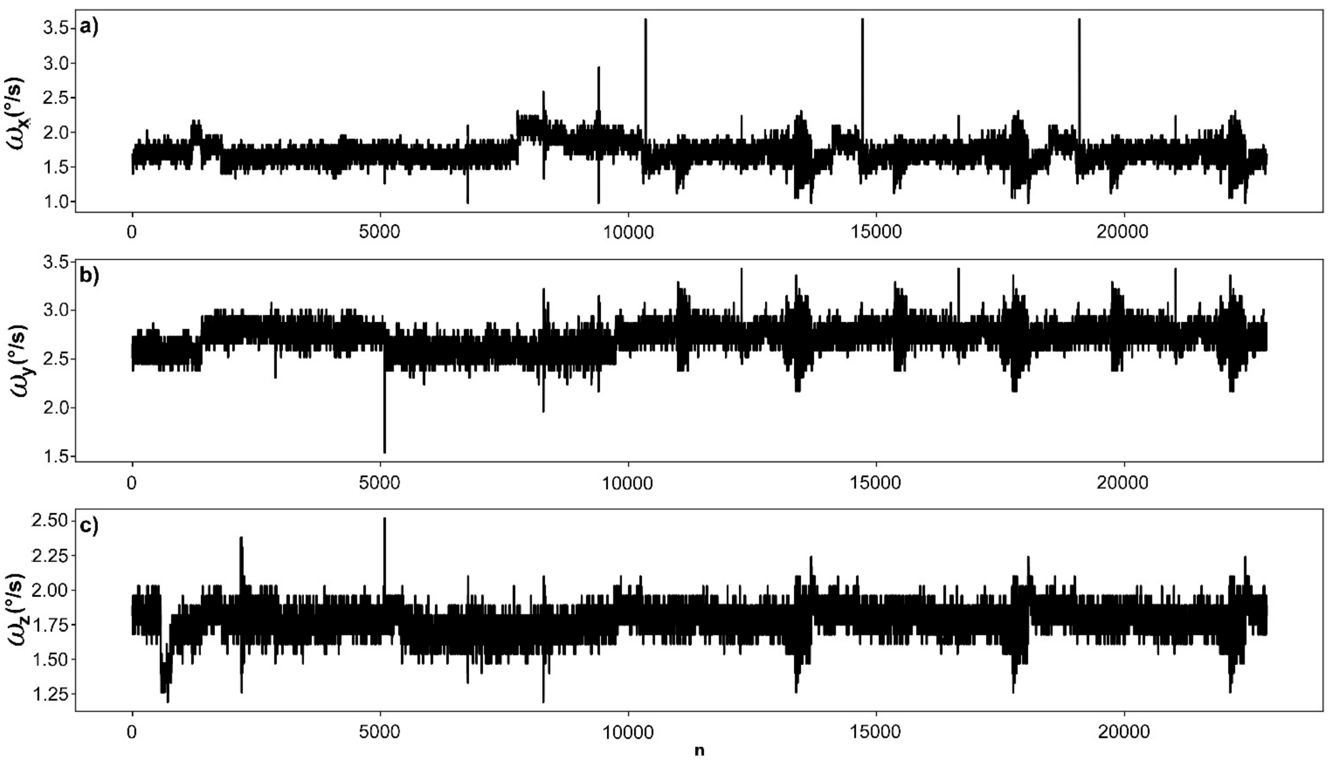

Figure S2. Raw angular velocity recordings in static positions used for gyroscope calibration.

The measurement error for a magnetometer read as (Dewhirst et al., 2016)

$$\boldsymbol{mag}_m = S\mathcal{N}(A_{si}\,\boldsymbol{mag} - \boldsymbol{b}_{hi}) + \boldsymbol{b}_m \tag{3}$$

where $\boldsymbol{mag_m}$ is the magnetic field measured by sensors, S is scale factor to generate outputs that fall within our desired range, $\mathcal{N}$ is the misalignment error, $\boldsymbol{b}_m$ is the bias error (offset), $\boldsymbol{b}_{hi}$ is hard iron errors due to the presence of permanent magnets and residual magnetism, A_{si} is soft iron errors due to the presence of material that influences or distorts a magnetic field, but does not necessarily generate a magnetic field itself and $\boldsymbol{mag}$ is the actual magnetic field. Hence, the calibrated magnetic field can be computed as

$$\boldsymbol{mag} = A_m^{-1}\left(\boldsymbol{mag}_m - \overline{\boldsymbol{b}}\right) \tag{4}$$

where $A_m = \mathcal{SN} A_{si}$ and $\overline{\boldsymbol{b}} = \mathcal{SN}\boldsymbol{b}_{hi} + \boldsymbol{b}_m$. The calibration procedure aims at taking the points over an elliptic surface, finding the offset, and recasting them over a spherical surface. The calibration procedure computes the arrays A_m^{-1} and $\overline{\boldsymbol{b}}$ by using the orthogonality property of the magnetic field vector (Dewhirst et al., 2016). The magnetometer data used for the calibration procedure are represented in Figure S3.

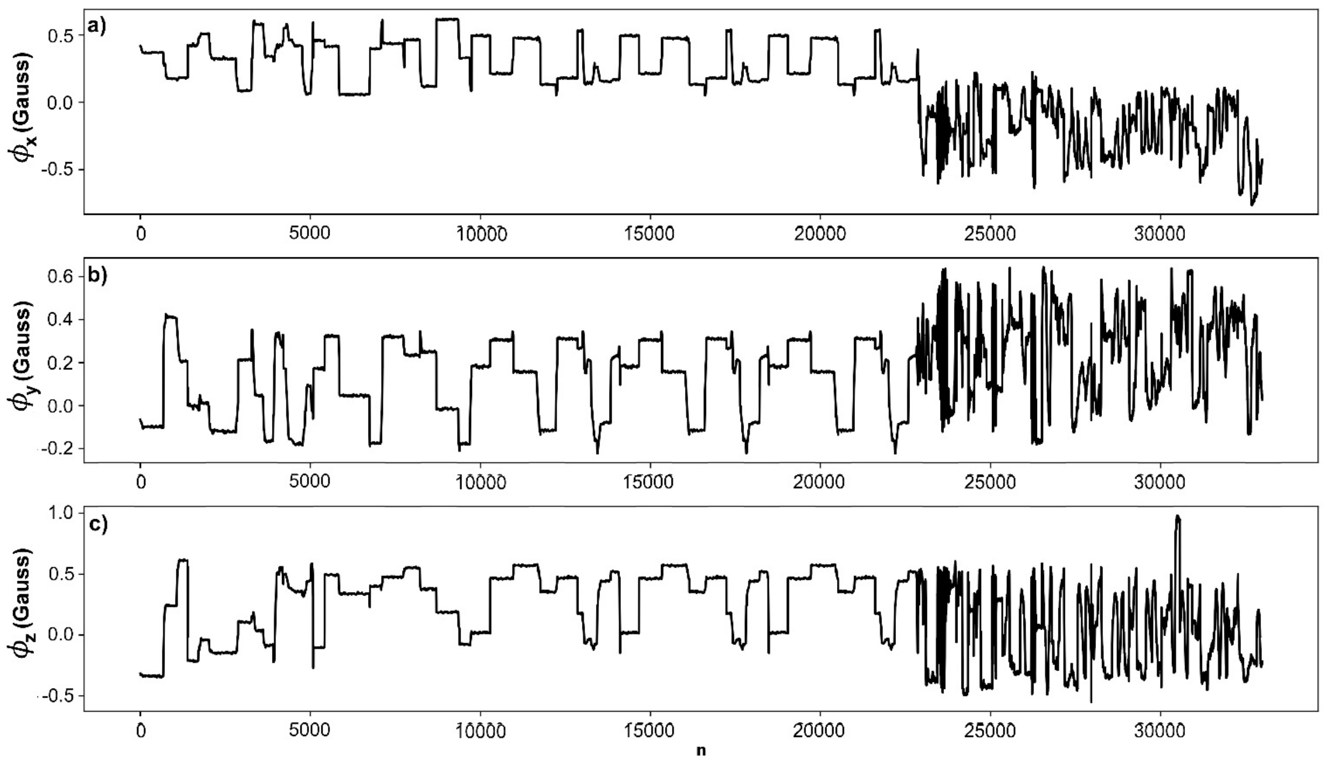

Figure S3. Raw magnetic field recordings used for magnetometer calibration.

<u>Camera tracking</u>

Different techniques have been tested to detect and track objects over time (e.g., Soleimanitaleb et al., 2019). Classical approaches for object detection consist of subtracting the tracked object from the background when the minimum number of pixels representing the object is exceeded (Soeleman et al., 2012; Stolle et al., 2016). In the last 10 years, advanced techniques have been developed to detect objects with specific shapes (e.g., Liu et al., 2020). Data-driven techniques are usually not affected by the light condition and scale of the object to track. Hence, when a sample of pictures of the object of interest is available, object detection can leverage the information retained in the set of images without any background subtraction such as in the deep-learning algorithm YOLOv5 (Redmon et al., 2016).

A set of operations was followed to use this tracking algorithm. First, different training images of the object were collected (i.e. 390 images). Second, in each training image, the object was framed by drawing a bounding box and labelling it with the class name. Third, to protect the prediction of the detection algorithm from overfitting issues, cross validation was carried out by splitting the set of images into 70% for training, 20% for validating and 10% for testing. Then, the YOLOv5 Detector was trained and tested over different samples of the data. Camera calibration was performed by reducing lens distortion (checkerboard method, Zhang, 2000) and rectifying the perspective distortion (Bradski, 2000). Specifically, given four corners of the region of interest (namely, the titling table boards), the perspective transformation re-projected them into an undistorted reference system where the frame width and frame height were specified to cover the region of interest. The aspect ratio of the frame (given in pixels) had to match the aspect ratio of the board on which cobble motion occurred. This procedure made it possible to calculate the position of the centre of the bounding box and use it as a proxy of the position of the centre of mass for the cobble. The vertical elevation was inferred from the coordinates of the centre of mass of the cobble under the assumption that the cobble always had a point of contact on the tilting table.

Kalman filter

The Kalman filter is a recursive algorithm that provides an estimate of the state of the system given some measurements. The unknown variables are usually noisy and define the state of the system $\boldsymbol{x}$. Conversely, the measurements $\boldsymbol{z}$ are reliable temporal observations feeding the algorithm to improve the estimate of the state of the system (Dewhirst et al., 2016; Kim and Bang, 2019). The model describes the evolution of the state from k to $k-1$ as (Kim and Bang, 2019)

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + B\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1} \tag{1}$$

where A is the transition matrix applied to the previous state $k-1$, B is the control-input matrix applied to the control vector $\boldsymbol{u}_{k-1}$ and $\boldsymbol{w}_{k-1}$ is the process noise vector. In the Kalman filter process model, the noise is assumed to be a zero-mean Gaussian variable with Q as covariance, i.e. $\boldsymbol{w} \approx \mathcal{N}(0, Q)$. In the present application, the state of the system at the step k is defined as

$$\boldsymbol{x}_k = [x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]^T \tag{2}$$

where x, y, z are the position components, v_x, v_y, v_z are the velocity components and a_x, a_y, a_z are the acceleration components. The transition matrix A and the control-input matrix B are defined as

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 1/2\Delta t^2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 1/2\Delta t^2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 1/2\Delta t^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} ; \qquad B = [0]_{9x6} \tag{3}$$

where Δt is the time step, that in the present application is equal to 0.017 s. The state-evolution equation is coupled with the measurement model describing the relation between the measurement $\boldsymbol{z}_k$ and the state $\boldsymbol{x}_k$ (Kim and Bang, 2019)

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \boldsymbol{v}_{k-1} \tag{4}$$

where H is the measurement matrix and $\boldsymbol{v}_k$ the noise of the measurements assumed to be a zero-mean gaussian variable with R as covariance, i.e. $\boldsymbol{v} \approx \mathcal{N}(0, P)$. In the present application, the measurement vector $\boldsymbol{z}$ stores the camera-based positions x_m, y_m, z_m and the IMU-based linear accelerations $a_{x,m}$, $a_{y,m}$, $a_{z,m}$ and at the step k reads as

$$\boldsymbol{z}_k = [x_m, y_m, z_m, 0,0,0, a_{x,m}, a_{y,m}, a_{z,m}]^T \tag{5}$$

The measurement matrix H is defined as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The Kalman filter algorithm consists of two stages, prediction, and update. The state is predicted from a former state and then is updated using the measurements available for the process investigated. The iterative process starts from the initial uncertainty regarding the state and the measurements. The initial measurement noise is 10^{-14} m for position and 10^{-8} m/s^2 for acceleration in the case of rolling experiments and 10^{-6} m/s^2 in the case of sliding experiments.

Limitations

The sensor-based motion reconstruction via Kalman filter is based on some assumptions. First, the cobble is assumed to keep a point of contact with the experimental table (Figure 1). Second, the sensor embedded in the cobble is assumed to remain fixed during cobble motion. The present methodology shows some limitations. When the table inclination is small (<45°), the point of contact is ensured. However, for higher inclinations, the cobble starts losing contact with the board and thus the vertical elevation in camera-derived positions, that is computed assuming a point of contact with the board, becomes more approximated. Furthermore, the cotton pad buffer in the sensor enclosure does not ensure a fixed sensor installation position within the cobble increasing the degree of approximation of the motion reconstruction via Kalman filter. This notwithstanding, the variation of the kinetic energy reasonably explains the magnitude of the movement for changes in slope and mode of movement (Figure 7). The camera and sensor fusion are thus able to provide a lumped representation of the motion magnitude and determine the mode of movement.

Comparison with analytical frameworks

We compared the results of the data fusion algorithm with some simple conceptual analytical models, namely the rigid motion over a tilted plane and pendulum oscillations around a frictionless pivot (Figure S4).

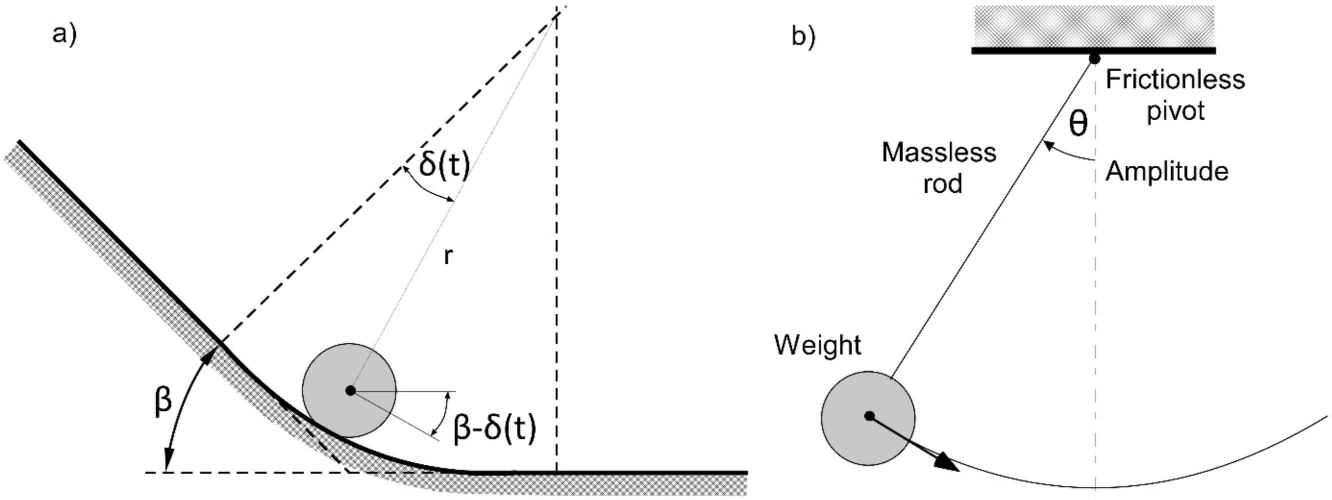

Figure S4. Simple conceptual frameworks to assess the robustness of the sensors data. (a) Rigid body motion over a tilt plane where β is the tilt plane slope, Φ the dynamic friction angle, and $\boldsymbol{r}$ is the radius of curvature. (b) Pendulum oscillations around a frictionless pivot where $\boldsymbol{L}$ is the length of massless string and θ is the angle the weight of the pendulum forms with the vertical.

Taking for example the experiments with an incline of 30°, the trajectories of cobbles travelling down the slope are represented in Figure 1. For the rolling experiments, the cobble average trajectory is smooth showing a larger variability on the slope (Figure S5a). Then, when the cobble is on the horizontal board, the trajectory has a low variation from the average showing some wiggles due to the uneven cobble surface that makes the rolling motion irregular (Figure S5). Conversely, in sliding experiments, the cobble trajectory is smoother along the whole path (Figure S5b). Similarly to the rolling case, the variability of the cobble path is smaller on the horizontal board and is slightly larger on the slope. In both examples, the standard motion equation describes a path that matches the experimental trajectories with an acceptable degree of approximation.

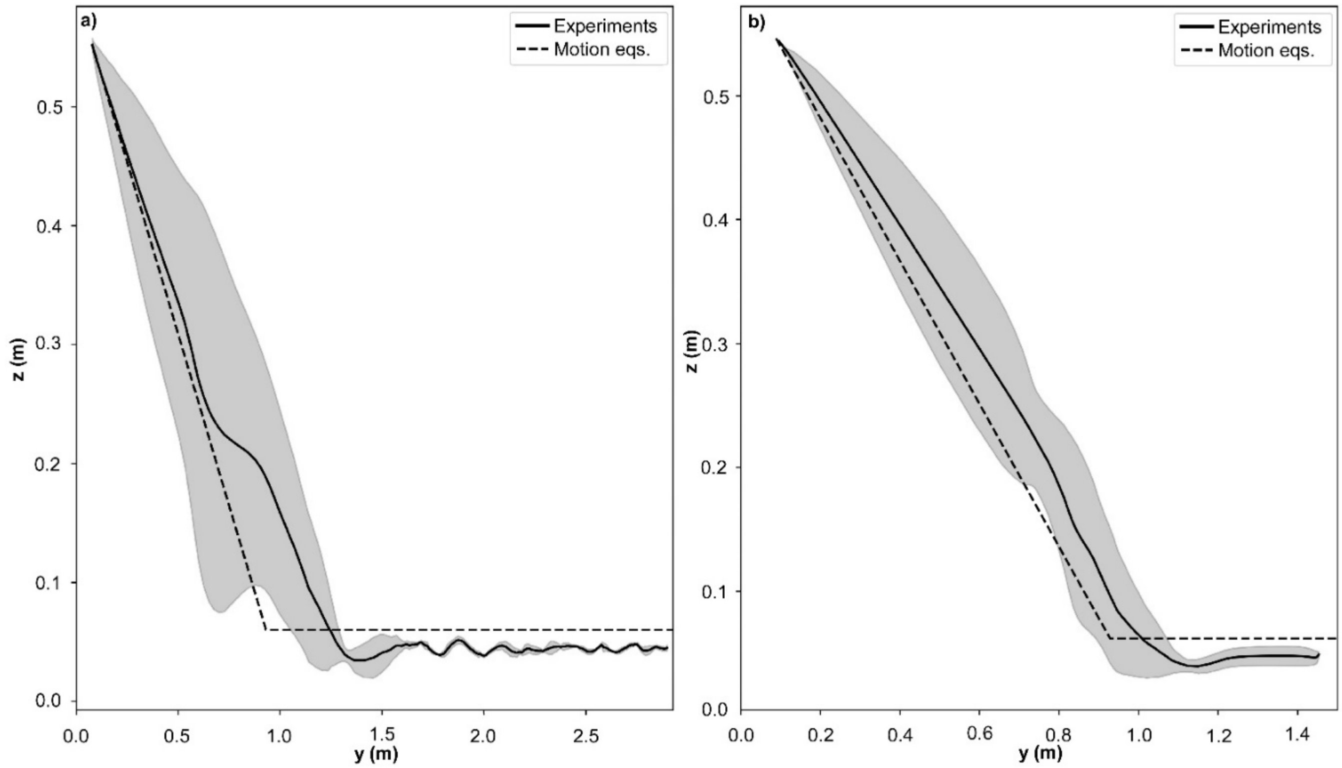

Figure S5. Comparison between the experiments and the standard motion equation describing the trajectories of the cobble for a slope at 30˚ in the vertical plane z-y. (a) Rolling experiments where the motion equation response is obtained for a friction angle Φ = 7.5˚. (b) Sliding experiments where motion equation response is obtained for a friction angle Φ = 2.5˚. Solid lines represent the average behaviour during the experiments at the same slope, whereas the grey areas show the standard deviation of the variable considered.

The pendulum trajectory is represented in Figure S6a. Overall, the experimental trajectories match the motion equation results reasonably well when the pendulum length and the initial angle are set at 2.05 m and 19˚, respectively. The experimental pitch angle is compared to the motion equation angular displacement in Figure S6b. The motion equation prediction overestimates the average pitch angle, and the experimental oscillations do not show a periodicity similar to the pendulum conceptual model.

However, the experimental pitch angle ranges between -17.5˚ and 13.5˚ which is in good agreement with the motion equation angular displacement ranging between -19˚ and 19˚.

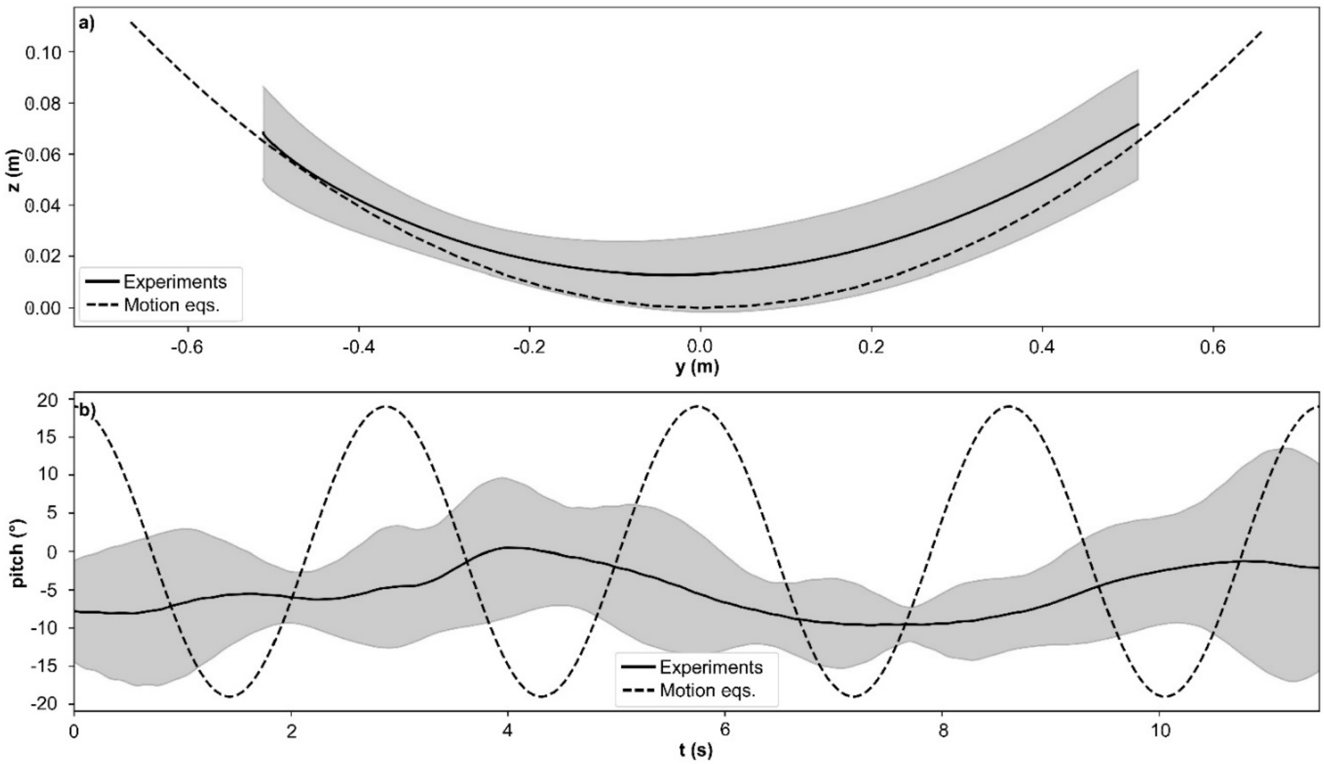

Figure S6. Comparison between the experiments and standard motion equation. (a) Trajectories in the vertical plane z-y. (b) The angle the weight of the pendulum forms to the vertical. Motion equation response is obtained for pendulum length L = 2.05 m and initial oscillation angle θ_0= 19˚.

Reference

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 120, 122-125, 2000.

Dewhirst, O. P., Evans, H. K., Roskilly, K., Harvey, R. J., Hubel, T. Y. and Wilson, A. M. Improving the accuracy of estimates of animal path and travel distance using GPS drift-corrected dead reckoning. *Ecology and Evolution*, 6, 17, 6210-6222, 2016. doi: 10.1002/ece3.2359

Frosio, I., Pedersini, F. and Borghese, N. A. Autocalibration of MEMS accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 58, 6, 2034-2041, 2009. doi: 10.1109/TIM.2008.2006137

Glueck, M., Oshinubi, D. and Manoli, Y. Automatic realtime offset calibration of gyroscopes. *IEEE Sensors Applications Symposium Proceedings,* Galveston, TX, USA, 214-218, 2013. doi:10.1109/SAS.2013.6493589

Kim, Y. and Bang, H. Introduction to Kalman filter and its applications. In: F. Govaers (editor), Kalman Filter, chapter 2, 1-17. *InTechOpen,* 2019. doi: 10.5772/intechopen.80600

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, 128(2), 261–318, 2020. doi:10.1007/s11263-019-01247-4

Redmon, J., Santosh, D., Girshick, R., & Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *ArXiv (Cornell University)*, 1-10, 2016. doi:10.48550/arxiv.1506.02640

Soeleman, M. A., Hariadi, M., and Purnomo, M. H. Adaptive threshold for background subtraction in moving object detection using Fuzzy C-Means clustering. *TENCON 2012 IEEE Region 10 Conference*, Cebu, Philippines, 1-5, 2012. doi:10.1109/TENCON.2012.6412265

Soleimanitaleb, Z., Keyvanrad, M. A., and Jafari, A. Object Tracking Methods: A Review. *9th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 282-288, 2019. doi: 10.1109/ICCKE48569.2019.8964761.

Stolle, J., Nistor, I., and Goseberg, N. Optical Tracking of Floating Shipping Containers in a High-Velocity Flow. *Coastal Engineering Journal*, 58(2), 1650005-1-1650005–29, 2016. doi: 10.1142/s0578563416500054

Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334, 2000. doi: 10.1109/34.88871.