Development of a Student-Centric Cyber-Physical System (SCPS): An Android App for Interactive Learning of Structural Analysis and Dynamics

Arvind Sastha Kumar[1]; Kevin Han[2]; Youngjib Ham[3]; and Hyungchul Yoon[4]

[1]Dept. of Computer Science, North Carolina State Univ., Raleigh, NC.
Email: akumar34@ncsu.edu
[2]Dept. of Civil, Construction, and Environmental Engineering, North Carolina State Univ., Raleigh, NC. Email: kevin_han@ncsu.edu
[3]Dept. of Construction Science, Texas A&M Univ., College Station, TX.
Email: yham@tamu.edu
[4]School of Civil Engineering, Chungbuk National Univ., Cheongju, Korea.
Email: hyoon@g.cbnu.ac.kr

ABSTRACT

Although there have been many attempts to increase student engagement and interaction in class through virtual reality and augmented reality (VR/AR), structural engineering has remained as one of the disciplines that lacks interactive learning. There was an earlier attempt by the authors to improve student learning in structural analysis and dynamics through a student-centric cyber-physical system (SCPS) that reacts to student movement and outcomes structural behavior. This system used mobile devices as sensors and sent data to be processed by a server. As a result, it suffered from the inherent limitation of Transmission Control Protocol/Internet Protocol (TCP/IP)—lags and instability. Therefore, this paper presents the development of a new SCPS tool with a new system architecture that is much more stable and provide better user experience. This SCPS captures and uses student movement (i.e., walking, turning, and jumping) as point loads on a virtual bridge. As students move around, structural behavior (i.e., deflection) is simulated and visualized in real time.

INTRODUCTION

Although there have been many attempts to increase student engagement and interaction in class through VR/AR (Bressler and Bodzin 2013), structural engineering has remained as one of the disciplines that lacks interactive learning. The authors (Yoon et al. 2019) have attempted to improve student engagement and interaction through a SCPS that captures and reacts to student movement and provides game-like visualization. This SCPS turns students into point loads on a virtual bridge and simulates and visualizes structural behavior in real time as students move around. A mobile app was developed to record student movement (i.e., walking, turning, and jumping) and stream them via TCP/IP connection to a server that simulates structural behavior and visualize the bridge movement and student movement.

The objective was to use this SCPS as a learning tool that would help students build a “sense” or “intuition” for how structural components (i.e., simply supported beam, cantilever, etc.) behaves under different loading conditions. By visualizing how a structure reacts/behaves in real time as students (point loads) move, they can learn how their movement affects the behavior of the structure. This learning tool, however, suffered from constant lagging and connection instability due to the use of TCP/IP for connecting a mobile device with MATLAB. To overcome these limitations, this paper presents new Android-base mobile app development with a new system architecture and configuration. This new SCPS provides much better stability and user experience. This paper focuses on presenting the development process and the tool itself.

SYSTEM OVERVIEW

The overview of the proposed system is shown in Figure 1. (Yoon et al. 2019) developed a similar system, however, it suffers from lagging and instability issues due to the inherent remote execution of structural analysis utilizing TCP/IP protocol for communication. The proposed system is developed as a gaming application in android that students can interact, without any TCP/IP communication, to understand structural responses due to natural movement of a dynamic load. The mobile application builds on accelerometer sensor to detect the movement of a player. The movement detection module classifies the accelerometer signal as either step or jump. The structural dynamics analysis module is based on structural models like simply supported beam. It also calculates the dynamic live load based on mass of the player that is collected in prior and processed acceleration values obtained from movement detection module. The visualization module updates dynamic movements of the structure on the mobile screen in real-time based on a player's movements. The fact that the system is developed as a gaming application allows students to use this system and observe the changes happening to a structure in real-time in an interactive as well as immersive fashion thereby enhancing their engagement in class. The remainder of this section will explain each module of proposed system in detail including the algorithms governing the respective modules along with open challenges and possible directions for enhancements. Experiments were conducted and corresponding results described in the paper were obtained using Google Pixel 2 smartphone.

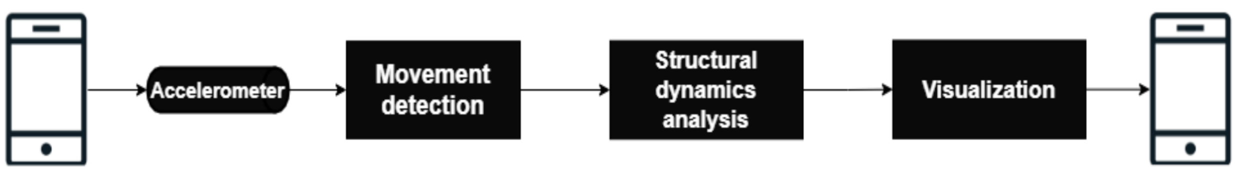

Figure 1. System overview

Movement Detection Module

The movement detection module is for tracking player's movements in a virtual space where they constantly apply live load on the structure. Since we focus on the finite element analysis of 1-D problem, we need to track movements pertaining to 1-D problem. Likewise, the movement detection module recognizes steps and jumps made by the players. The details are described in following sections.

Processing Raw Sensor Data

Players use a smartphone to interact with the proposed system. Hence, processing raw sensor data coming out of accelerometer sensor becomes a critical part of the system. Conceptually, a smartphone identifies the acceleration (a_t) applied on it by measuring the forces that are applied onto the sensor itself (Motion Sensors 2021). For this reason, when the phone is placed on a table (not accelerating) the sensor gives out a magnitude of a = 9.8m/s^2. Hence, it becomes necessary to remove environmental factors such as gravity (g).

$$a_t = \sqrt{a^2{}_{x,t} + a^2{}_{y,t} + a^2{}_{z,t}} - g \tag{1}$$

Figure 2 shows the difference between raw accelerometer readings with and without gravitation components.

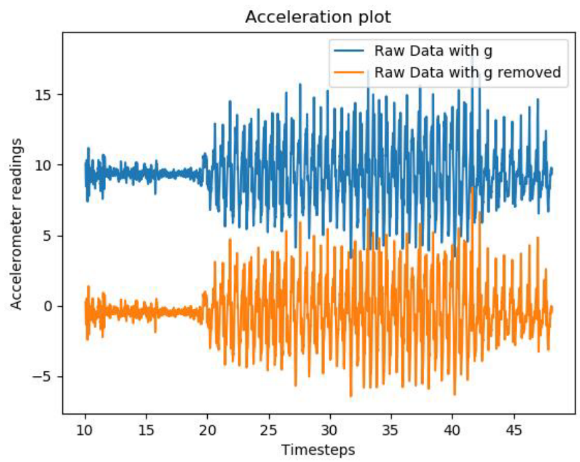

Figure 2. Accelerometer data with & without force of gravity

Similar to a data preprocessing technique suggested by (Yao et al. 2020), we use a 4-th order low-pass digital butter-worth filter with a cut-off frequency of 0.2π to 1) remove interference and noise 2) smoothen the signal output. Figure 3 shows resulting filtered output of raw accelerometer readings. Figure 4 presents screenshots of movement detection module taken from the developed mobile application.

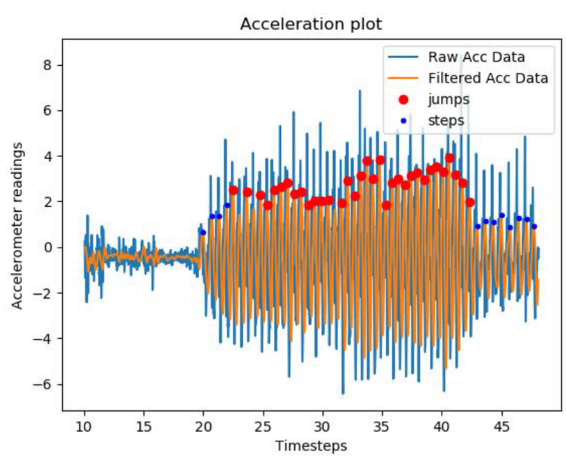

Figure 3. Output of 4th order butter-worth filter with jumps & steps

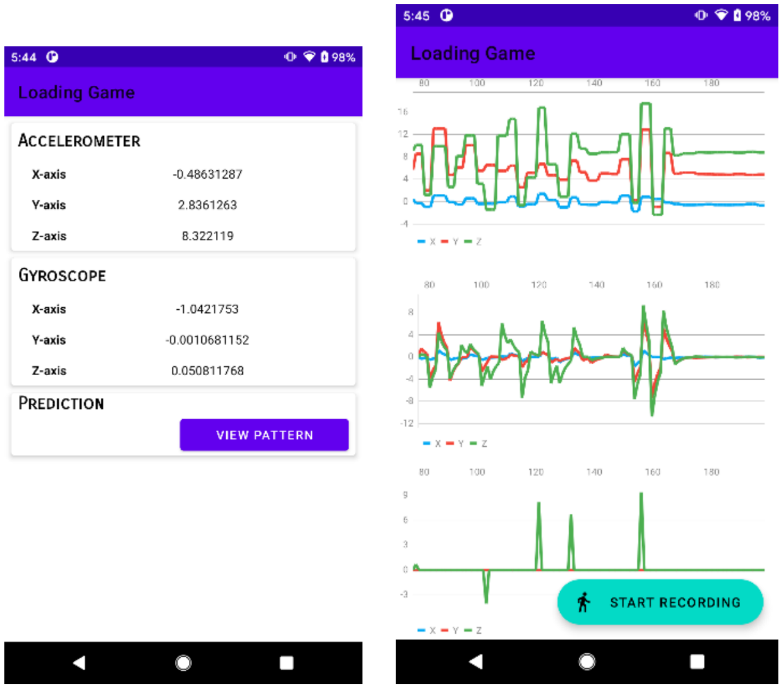

Figure 4. (left) Accelerometer readings; raw accelerometer data (right top) accelerometer data with gravity component removed & filtered (right middle) peak picking performed on filtered accelerometer data (right bottom)

Movement Detection Algorithm

In general, there are variety of algorithms to capture activities that can be categorized into time-domain, frequency-domain, and feature clustering approaches. The time-domain approach has variety of methods including peak-estimation (Ying et al. 2007) (Chen et al. 2015), threshold-based (Alzantot et al. 2012), zero-crossing (Jayalath et al. 2013), dynamic time warping (Li and Yang 2013), auto-correlation (Kappi et al. 2001) (Ailisto et al. 2005). While thresholding comes with its own shortcomings of choosing an optimal threshold (Kang et al. 2018) in uncontrolled settings, (Yoon et al. 2019) proposed an empirical method for activity detection based on thresholding that performed well in a controlled setting. We build on a similar movement detection technique as shown in figure 5. We employ a minimum difference threshold of 1 between positive & negative peak for step detection and a minimum threshold of 4 for jump detection. Since jump activity produces residual false-positive steps for short period, we force the movement detection algorithm to pause for about 0.5 seconds after each jump activity detection to reduce misclassifications. The algorithm was evaluated using 3 different players' data for jump and step detection. Each player's accuracy for step detection varied from 95.34% to 98%, while jump detection accuracy varied from 94.89% to 97.45%.

Structural Dynamics Analysis Module

Once player's movements are determined, we create and apply live load on a pre-defined structure based on a configurable mass and exerted acceleration. The overview of the structural dynamic analysis module is as shown in Figure 6.

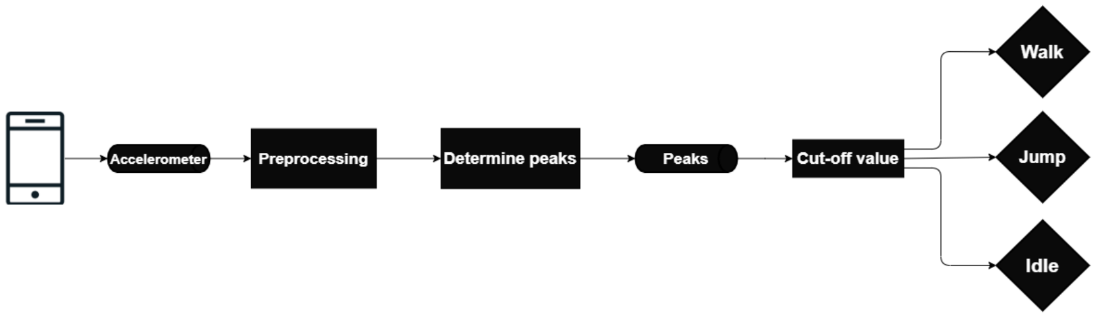

Figure 5. Movement Detection Algorithm

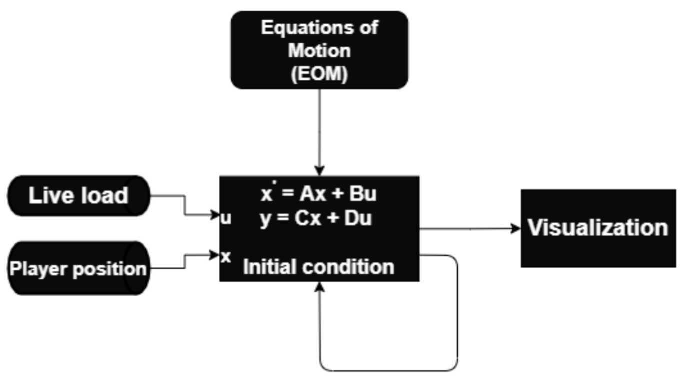

Figure 6. Structural dynamics analysis

We assume the number of modes to be 5 for our experiment. We constraint all our evaluations to the structure of a simply supported beam. Hence, the shape functions $\psi_1(x),\ \psi_2(x), \psi_3(x), \psi_4(x),\ \psi_5(x)$ are appropriately selected (in case of simply selected beam $\psi_n(x) = \sin(\frac{2nx}{L})$). The mode shape matrix ψ is then determined by combining all individual mode shapes. Figure 7 shows i) possible configurations that can be done to structure and dynamic load ii) & iii) structural dynamics solver's results.

The equations of motion can be represented as shown equation (2),

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = G \tag{2}$$

where M is the mass matrix, C is the damping matrix, K is stiffness matrix, and G is the force matrix. K and M are determined using following equations (3) and (4)

$$K = \int_0^L EI\psi''^T\, \psi'' dx \tag{3}$$

$$M = \int_0^L \rho A\psi\, \psi^T dx \tag{4}$$

where E is modulus of elasticity of the beam, I is moment of inertia, ρ is density, and A is area of sections of given structure. Damping ratio ξ is assumed to be 0.2. We employ java's *common* *math library* for necessary computation. Based on above two matrices, we can estimate natural frequency ω_n and the mode shape ϕ using equation (5). Modal matrices are calculated using mode shapes after which damping matrix C is calculated using equation (6).

$$[\omega_n, \phi] = eig(K, M) \tag{5}$$

$$M_r = \psi^T M \psi$$

$$K_r = \psi^T K \psi$$

$$C_r(i,i) = 2\xi\sqrt{K_r(i,i)\, M_r(i,i)}$$

$$C = \psi^{-T} C \psi^{-1} \tag{6}$$

The constructed models are represented as state space models using (7). We select displacement, velocity, and accelerometer to be the output of the system. Now the system matrices are represented using (8), (9).

$$A_s = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}, B_s = \begin{bmatrix} 0 \\ -G \end{bmatrix}, C_s = \begin{bmatrix} I & 0 \\ 0 & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}, D_s = \begin{bmatrix} 0 \\ 0 \\ -G \end{bmatrix} \tag{7}$$

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = A_s \begin{bmatrix} q \\ \ddot{q} \end{bmatrix} + B_s G \tag{8}$$

$$y = \begin{bmatrix} q \\ \dot{q} \\ \ddot{q} \end{bmatrix} = C_s \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + D_s G \tag{9}$$

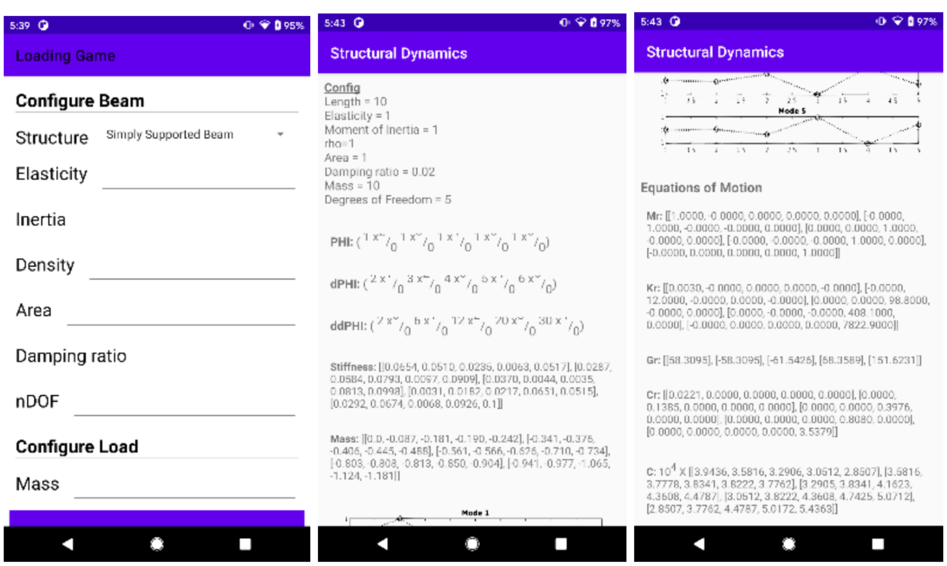

Figure 7. (left) Configurable parameters of system, (center) (right) structural dynamics module output

We solve the state space model and obtain y as a vector of displacement, velocity, and acceleration. The displacement of the structure v at any position x is given by equation (10).

$$v = q\,\psi(x) \tag{10}$$

Visualization

One of significant differences compared to the previous approach proposed by (Yoon et. al 2019) is that rendering of structural response happens in real-time in the mobile device. Once we obtain the displacement values at each x of given structure from the structural dynamic analysis module (i.e., a simply supported beam), we refresh the screen by plotting the new positions of each nodes on the screen using android's OpenGL (Android OpenGL 2021) graphics library. Figure 8 shows the visual rendering of a structure in the form of a bridge and a dynamic moving load as a small black mass.

Figure 8. Visualization of structural responses with black mass acting as live load

CONCLUSIONS

This paper presents a new SCPS app that can capture student movement on a virtual bridge and simulate and visualize structural behavior as students move around. This tool is designed to be an interactive tool that will help students build a "sense" or "intuition" for the behavior of a bridge (i.e., simply supported beam, trusses, cantilever, etc.). A future work will focus on assessing effectiveness of this tool in learning structural engineering concepts.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2021384. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

Ailisto, H. J., and Makela, S. M. Identifying people from gait pattern with accelerometers. In *Proceedings of SPIE—The International Society for Optical Engineering*, Orlando, FL, USA; SPIE: San Jose, CA, USA, 2005; Volume 5779, pp. 7–14.

Alzantot, M., and Youssef, M. "UPTIME: Ubiquitous pedestrian tracking using mobile phones." *2012 IEEE Wireless Communications and Networking Conference (WCNC)* (2012): 3204-3209.

Android openGL: https://developer.android.com/guide/topics/graphics/opengl.html; accessed: April 2021.

Bressler, D., and Bodzin, A. (2013). "A mixed methods assessment of students' flow experiences during a mobile augmented reality science game." *Journal of Computer Assisted Learning*, 29(6), 505-517.

Chen, G. L., Fei, L. I., and Zhang, Y. Z. Pedometer method based on adaptive peak detection algorithm. *J. Chin. Inert. Technol.* 2015, 23, 315–321.

Jayalath, S., and Abhayasinghe, N. A gyroscopic data based pedometer algorithm. In *Proceedings of the International Conference on Computer Science & Education*, Colombo, Sri Lanka, 26–28 April 2013; pp. 551–555.

Kang, X., Huang, B., and Qi, G. A Novel Walking Detection and Step Counting Algorithm Using Unconstrained Smartphones. *Sensors* 2018, 18, 297.

Kappi, J., Syrjarinne, J., and Saarinen, J. MEMS-IMU based pedestrian navigator for handheld devices. In *Proceedings of the 14th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS*, Salt Lake City, UT, USA, 11–14 September 2001. 24.

Li, H., and Yang, L. "Accurate and fast dynamic time warping," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2013, pp. 133–144.

Mannini, A., and Sabatini, A. M. Accelerometry-Based Classification of Human Activities Using Markov Modeling. *Comput. Intell. Neurosci.* 2011, 2011, 647858. 39.

Pirttikangas, S., Fujinami, K., and Nakajima, T. Feature Selection and Activity Recognition from Wearable Sensors. In *Proceedings of the International Symposium on Ubiquitious Computing Systems*, Seoul, Korea, 11–13 October 2006; pp. 516–527.

Motion Sensors: Android developer guide https://developer.android.com/guide/topics/sensors/sensors_motion; accessed: April 2021.

Yao, Y., Pan, L., Fen, W., Xu, X., Liang, X., and Xu, X. "A Robust Step Detection and Stride Length Estimation for Pedestrian Dead Reckoning Using a Smartphone," in *IEEE Sensors Journal*, vol. 20, no. 17, pp. 9685-9697, 1 Sept.1, 2020.

Ying, H., Silex, C., Schnitzer, A., Leonhardt, S., and Schiek, M. Automatic Step Detection in the Accelerometer Signal. In *Proceedings of the 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, Aachen Germany, 26–28 March 2007; Springer: Berlin/Heidelberg, Germany, 2007; Volume 13, pp. 80–85.

Yoon, H., Han, K., and Ham, Y. A Framework of Human-Motion Based Structural Dynamics Simulation Using Mobile Devices. *Sensors (Basel)*. 2019;19(15):3258. Published 2019 Jul 24.