Periodica Polytechnica Mechanical Engineering, 64(2), pp. 120–127, 2020

Comparison of Pixel-based Position Input and Direct Acceleration Input for Virtual Stick Balancing Tests

Balazs A. Kovacs[1,2*], Tamas Insperger[1,2]

[1] Department of Applied Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, 5 Műegyetem rkp., H-1111 Budapest, Hungary
[2] MTA-BME Lendület Human Balancing Research Group, 5 Műegyetem rkp., H-1111 Budapest, Hungary
[*] Corresponding author, e-mail: balazs.kovacs@mm.bme.hu

Received: 30 November 2018, Accepted: 29 March 2020, Published online: 24 April 2020

Abstract
A virtual stick balancing environment is developed using a computer mouse as input device. The development process is presented both on the hardware and software level. Two possible concepts are suggested to obtain the acceleration of the input device: discrete differentiation of the cursor position measured in pixels on the screen and by direct measurements via an Inertial Measurement Unit (IMU). The comparison of the inputs is carried out with test measurements using a crank mechanism. The measured signals are compared to the prescribed motion of the mechanism and it is shown that the IMU-based input signal fits better to the prescribed motion than the pixel-based input signal. The pixel-based input can also be applied after additional filtering, but this presents an extra computational delay in the feedback loop.

Keywords
human balance, human-computer interaction, virtual environment, inverted pendulum, reaction delay

1 Introduction

Balancing an object about an unstable equilibrium or around an unstable path is a natural everyday activity for humans. Standing still on two legs, running, hopping or simply holding an object still can be mention as trivial examples. All these processes are maintained by a complex feedback mechanism operated by the Central Nervous System (CNS). The nature and the characteristics of this feedback process is still a subject of debates. Investigation of different human balancing tasks, such as simple quiet standing [1–5] standing on pinned or rolling balance boards [6, 7], stick balancing on the fingertip or on a Ping-Pong racket [8–11], may help in identifying and in understanding the underlying control mechanism. Experimental investigation of these balancing tasks requires motion capture system, e.g., IMU sensors fixed on the balanced object or a camera system, which records the spatial position of markers fixed on the balanced object. These systems are expensive, typically not mobile and requires considerable time to calibrate. Alternatively, virtual balancing tasks, which involve an interaction between a human and a computer, are also often used to analyze human behavior during balancing [8, 12–16]. Advantages of virtual balancing tasks over actual balancing task are that they do not require motion capture systems, they are mobile, and they make possible to manipulate some key parameters of the system. For instance, the dynamics of the system (e.g., the dimension or the order of the system) can easily be adjusted, the feedback delay can artificially be increased, or blank out tests can easily be implemented [13, 14]. Disadvantages of virtual balancing tasks are the extra delay between the actuation and the visualization and the finite resolution of the computer screen. Establishing and implementing a virtual environment sets several technical problems related to the visualization of the system. The size of the screen, the refresh rate and the resolution are usually key factors during virtual balancing tests. The optimal visualization setup depends on the balancing task and strongly affects the execution of the task. The other major component of the virtual environment is the interface between the subject and the computer. Many options are available for assessing implementing the action of the subject. The different input devices have different features, such input lag, resolution and communication speed. These preferences of a virtual

Cite this article as: Kovacs, B. A., Insperger, T. "Comparison of Pixel-based Position Input and Direct Acceleration Input for Virtual Stick Balancing Tests", Periodica Polytechnica Mechanical Engineering, 64(2), pp. 120–127, 2020. https://doi.org/10.3311/PPme.13507

environment must be carefully analyzed during the design and the development of the virtual environment.

During balancing tasks, the CNS uses the information received from different sensory organs. In case of standing still, visual, vestibular, proprioceptive and mechanoreceptive feedback are all available to support the control mechanism by the CNS. In contrast, in case of stick balancing on the fingertip, mostly visual information is used, which might be amended by feedback from the mechanoreceptors at the fingertip. For stick balancing on Ping-Pong racket, however, there is no direct contact between the subject's skin and the stick, therefore the only sensory information is from the visual perception. Clearly, in case of virtual balancing tasks, only visual feedback is available. A key feature of all balancing tasks is that corrective actions to a perturbation are made with a reaction delay. The nature of the reaction delay is an important factor in the mathematical models. Whether it can be modeled as a point delay or a distributed delay? Whether it is varying in time regularly with some well-defined sampling frequency or rather in a stochastic manner? Although experiments show a large variety of reaction delays, in most of the mathematical models in the literature a fixed-point delay is used.

A key feature of stick balancing tasks is the critical length of the stick. If the stick is shorter than the critical length, then human subjects cannot balance the stick, since it falls faster than the subject could react. In case of a delayed proportional-derivative controller, the critical length can be derived after a stabilizability analysis as $L_{crit} = \frac{3g}{4}\tau^2$, where g = 9.81 m/s^2 is the gravitational acceleration and τ is the reaction delay. Thus, the critical length is increasing with the reaction delay. The nature of the relation between the critical length and the reaction delay is in the focus of many research papers [3, 9, 11, 16].

In this paper, a virtual stick balancing environment is designed and implemented. An inverted pendulum is balanced on a computer screen using a computer mouse as manipulator. In the view of Newtonian dynamics, the acceleration of the mouse is used to determine the motion of the virtual stick. The main challenges in this task are

1. accurate measurement of the mouse acceleration,
2. the reduction of the extra computational delay; and
3. proper visualization of the position of the balanced stick on the screen.

The structure of the paper is as follows. First, the virtual environment is described and the differences between two input concepts (i.e., pixel-based acceleration and IMU-based acceleration) are explained. Then, in Section 3, the calibration of the input device is presented. The paper is concluded in Section 4.

2 The virtual stick balancing environment

The structure of the virtual stick balancing system shown in Fig. 1 consists of three elements:

1. the underlying mechanical model of the balancing task;
2. the software, which simulates and visualizes the motion of the inverted pendulum; and
3. the hardware, which creates the interface between the virtual environment and the human operator.

2.1 The underlying mathematical model

The implemented virtual balancing task is based on the two-degree-of-freedom model of a linearly driven planar pendulum-cart system shown in Fig. 2. This is a simplified model of human stick balancing at the fingertip while it still captures the main characteristics of the control processes. The general coordinates are the displacement ξ of the cart and the angular position ϕ of the pendulum.

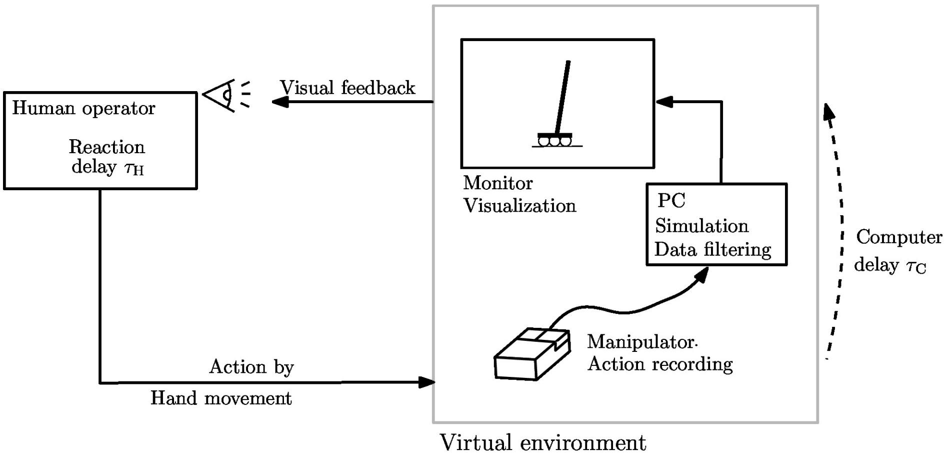

Fig. 1 Schematic model of the virtual stick balancing system

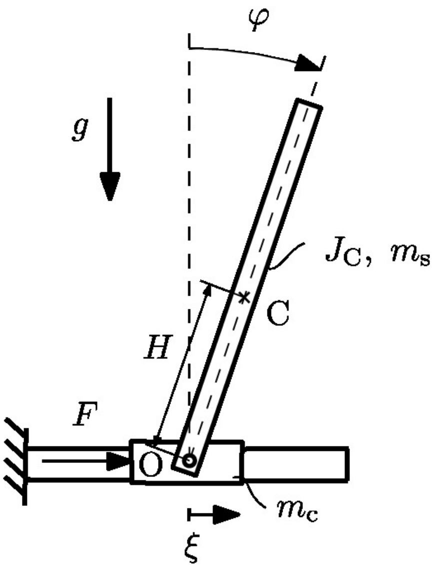

Fig. 2 The mechanical model: a horizontally driven planar inverted pendulum

The governing equations can be written in the form

$$\begin{bmatrix} m_c+m_s & Hm_s\cos\varphi \\ Hm_s\cos\varphi & H^2m_s+J_C \end{bmatrix}\begin{bmatrix}\ddot{\xi}\\ \ddot{\varphi}\end{bmatrix} - \begin{bmatrix} Hm_s\dot{\varphi}^2\sin\varphi \\ Hm_s g\sin\varphi - k\dot{\varphi}\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}F, \qquad (1)$$

where m_c and m_s are the mass of the stick and the cart, respectively, H is the distance between the pivot point O and the center C of mass of the rod, J_C is the second moment of inertia of the rod with respect to the normal axis through point C, k is a damping factor.

If we assume that the cart is moving according to some prescribed function $\xi(t)$, then the degrees of freedom of the system reduces to one. The corresponding governing equation reads

$$\left(H^2m_s+J_C\right)\ddot{\varphi}+k\dot{\varphi}-Hm_sg\sin\varphi=-Hm_s\ddot{\xi}\cos\varphi. \qquad (2)$$

Here, ξ is not a general coordinate anymore but an input signal, whose second derivative (i.e., the acceleration of the cart) shows up in the equation as a forcing term. This equation can easily be implemented in a simulation environment. Equation (2) can be transformed into state space form, where the state variable vector is $x=\begin{bmatrix}\varphi & \dot{\varphi}\end{bmatrix}^T$. The first-order representation of the governing equation is

$$\dot{\boldsymbol{x}}=\boldsymbol{f}\left(\boldsymbol{x},u\right)=\begin{bmatrix}\dot{\varphi}\\ b_1\left(s\sin\varphi-u\cos\varphi\right)-b_2\dot{\varphi}\end{bmatrix}, \qquad (3)$$

where

$$b_1=\frac{Hm_s}{H^2m_s+J_C},\quad b_2=\frac{k}{H^2m_s+J_C},$$

and $u=\ddot{\xi}$ is the input signal, which is provided by the test subjects using the computer mouse as a manipulator. The movement of the computer mouse (position, velocity and acceleration) is to be used to generate the input u.

Due to the human reaction time delay and the finite processing time of the computer, there is an unavoidable feedback delay, which, without loss of generality, is modeled as an overall input delay. It is assumed that the human subjects' action is a function of the delayed state variables such that $u(t)=f\left(\varphi(t-\tau),\dot{\varphi}(t-\tau)\right)$. Here τ is an equivalent overall delay, which is the sum of the human reaction time τ_H and the computer delay τ_C. The human reaction time τ_H is the time between the visual perception of the motion of the stick on the computer screen and the onset of the corrective hand (computer mouse) motion. The computer delay τ_C is the time between the onset of the mouse motion and its appearance on the computer screen.

2.2 The software

The software for the virtual stick balancing task was developed in JAVA environment. Three Graphical User Interfaces (GUIs) were created, one for the specification of the input parameters for simulation, one for the visualization of the pendulum-cart system during simulation and one for plotting the time history of the input signals. The balancing task was modeled using the mathematical model of the inverted pendulum presented in Section 2.1. The employed governing equation was Eq. (3). A fourth order Runge-Kutta method was used with an adaptive time step such that the simulation was running in real time. During the visualization of the pendulum-cart system, the screen refresh rate was set to 60 FPS, therefore the frequency of the simulation was also set to 60 Hz in order to be synchronized to the screen refresh rate. Thus, the time step for the simulation was $\Delta t = 16.67$ ms. Visualization of the position of the inverted pendulum on the computer screen presented another limitation due to the size of the screen (see Section. 2.3). The vertical dimension of the pendulum was scaled such that the whole stick was displayed on the computer screen. The displacements in the horizontal direction were not scaled, which implies that the horizontal deviation between the top and the bottom of the pendulum on the screen was equal to the real deviation of the underling mechanical model.

The selection of the input signals used for the simulation, which resembles actual stick balancing for the operator, is not straightforward. The governing Eq. (2) contains the horizontal acceleration of the cart $\ddot{\xi}$, which is equal to the acceleration of the mouse moved by the subjects.

In order to visualize the instantaneous position ξ of the cart on the screen, the exact position of the mouse is also required. Thus, both the position and the acceleration of the mouse should be measured for the realization of the virtual balancing task. The position of the mouse can be determined based on the number of pixels the mouse cursor passes over on the screen. This gives a position signal with a resolution that matches to screen's pixel resolution. Note that the cursor nonlinearity must be turned off in the pointer precision settings of the operating system. The acceleration signal was determined in two different methods: by discrete differentiation of the position measured in pixels (*pixel-based acceleration*) and by direct measurement via an Inertial Measurement Unit (*IMU-based acceleration*).

2.2.1 Pixel-based acceleration

Since acceleration is the second derivative of the position, acceleration of the mouse can be obtained by two consecutive discrete differentiation of the pixel-based position.

This scheme results in a very noisy acceleration signal due to the finite number of pixels and the finite time step. Therefore, the pixel-based acceleration signal has to be filtered before feeding back in the simulation. Here, a simple re-sampling filter was used that evaluates the position values only at multiple time steps. Using this concept, the filtered acceleration signal can be computed easily from the pixel-based position in the form

$$a_i = K \frac{x_i - 2x_{i-N_f} + x_{i-2N_f}}{\Delta t^2}, \tag{4}$$

where x_i is the position of the mouse measured in pixels at the i^{th} time step, $N_f \in Z^+$ is the filtering parameter and K is a gain parameter given in m/pixel. Gain K depends on the monitor size and monitor resolution and should therefore be calibrated for actual measurements (see Section 3). Application of this filter results in a much smoother signal but, at the same time, introduces an additional artificial delay $\tau_{filter} = N_f \Delta t$. The optimal value for the filtering parameter was experimentally found to be $N_f = 3$ during preliminary tests, which gives $\tau_{filter} \cong 50$ ms. Note that other types of filtering options are also possible, but here we reduce our analysis the above one for the sake of simplicity.

2.2.2 IMU-based acceleration

Another possibility is to measure directly the acceleration of the computer mouse. For this purpose, an Arduino mini microcontroller and a three-axis accelerometer were used. The controller reads the raw data of the sensor and converts it to acceleration in SI unit with sampling frequency 100 Hz and then sends the data through a serial port with baud-rate of 115200 to the PC. There are no further data processing in the controller, therefore the resulted acceleration signal becomes available for the simulations without significant processing delay. However, this scheme is sensitive to static error caused by the alignment error of the accelerometer versus the vertical direction.

In order to produce an accurate acceleration input for the simulation, the pixel-based acceleration and the IMU-based one was combined using a Kalman filter. In the implementation of the Kalman filter, a prediction model is used for the motion of the mouse, which can be written in the form

$$\boldsymbol{y}_j = \boldsymbol{F}\boldsymbol{y}_{j-1} + \bar{\boldsymbol{w}}_j, \tag{5}$$

$$\boldsymbol{z}_j = \boldsymbol{H}\boldsymbol{y}_j + \boldsymbol{v}_j, \tag{6}$$

where $\boldsymbol{y}_j = \begin{bmatrix} x_j & v_j & a_j \end{bmatrix}^T$ is the vector of model variables, $\boldsymbol{z}_j = \begin{bmatrix} \xi_j^{meas} & \ddot{\xi}_j^{meas} \end{bmatrix}^T$ is the vector of measured variables and the model and the measurement matrices are written as

$$\boldsymbol{F} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The stochastic variables of the model are assumed in the form $\bar{\boldsymbol{w}}_j = \begin{bmatrix} \frac{1}{2}\Delta t^2 & \Delta t & 1 \end{bmatrix}^T w_j$, where $w_j \sim N(0, \sigma_K)$ is the stochastic variable of the implemented model with zero expected value and standard deviation σ_K, while $\boldsymbol{v}_j \sim \begin{bmatrix} N(0, \sigma_p) & N(0, \sigma_a) \end{bmatrix}^T$ is the stochastic variables of the measurement signals with zero expected value and standard deviations σ_p and σ_a, respectively. This is a predictor-corrector method, which produces a filtered signal of the position, velocity and acceleration with minimal delay. For the implementation of the Kalman filter, the values of σ_K, σ_p, σ_a should be determined. The values of σ_p and σ_a were determined based on the screen's pixel resolution and the resolution of the accelerometer, respectively, while the value of σ_K was determined empirically.

The following balancing tasks were implemented in the software:

- *Normal virtual balancing*
  The operator must balance an inverted pendulum of a given length on the screen for certain time duration.
- *Random perturbation*
  Perturbation is applied at random time instances during normal virtual stick balancing. This feature

can be used to analyze responses to unexpected perturbations.

- *Blankout test*
  During normal virtual stick balancing, the screen is blanked out at random time instant for certain duration (typically for less than 500 ms). This feature can be used to analyze reaction time delay of the human operators.

2.3 The hardware

The hardware consists of a processing unit (e.g., PC or notebook), a monitor as output interface and a modified computer mouse equipped with an accelerometer as input interface.

The acceleration of the computer mouse was measured by a MEMS sensor and an Arduino mini microcontroller, which reads the raw data of the sensor and transfers it to the PC. The selected sensor was a MMA845x IMU three-axis accelerometer. In the software of the Arduino board, the reading sampling frequency was set to 60 Hz, which was synchronized to the sampling of the simulation. The Ardunio-MEMS unit and the base of the optical computer mouse was combined in the same plastic case as shown in Fig. 3. This manipulator unit sends both the IMU-based and the pixel-based position with the same sampling frequency to the computer. Since the digital effects and the additional computational delays change the qualitative behavior of the balancing task, minimization of the computer delay τ_C is a crucial task.

The overall computer delay is affected by the response time ${}_{response}$ of the screen, the delay $\tau_{sampling}$ caused by the sampling effect, and the filtering delay τ_{filter}. Thus,

$$\tau_C = \tau_{response} + \tau_{sampling} + \tau_{filter}. \tag{7}$$

The acceleration of the mouse and its effect on the pendulum is performed within a single sampling period of length Δt = 16.7 ms. This sampling effect introduces an extra delay varying linearly between 0 and Δt with average of $\tau_{sampling} = {}^{\Delta t}\!/_{2} = 8.3\ \text{ms}$ [17].

As shown in Section 2.2, the filtering delay is $\tau_{filter} \cong 50\ \text{ms}$ for N_f = 3. When the signal of the accelerometer is used either directly or combined with pixel-based position via the Kalman filter, then $\tau_{filter} \cong 0\ \text{ms}$.

The response time of the screen was measured for different computer and screen configuration setups using a light sensor system (see Fig. 4), which detects a change in the screen synchronized to the mouse input. The different configurations and the measured response times are collected in Table 1. Five different processing units (five notebooks) were tested, one Lenovo X260, one Dell Vostro, one HP Elitebook 840 and two HP Probook 430 indicated by numbers (1) and (2) in Table 1. Each processing unit were measured with its own built-in display and with a LG24BK550Y monitor connected via either VGA or HDMI or both (where available). The processing unit HP Probook 430 (2) was also measured with a TV screen. It was observed that the shortest response time is produced by the HP Elitebook and the HP Probooks with their own built in display. However, during the balancing trials, large screens are preferable over built-in displays, since the motion of the visualized pendulum is more realistic in large screens. The response time is much larger when the processing unit is connected to the TV screen. Therefore, the selected configuration for the balancing task is the HP Probook 430 connected to the LG24BK550Y monitor via VGA cable. The corresponding overall computer delay is $\tau_C \cong 64\ \text{ms}$.

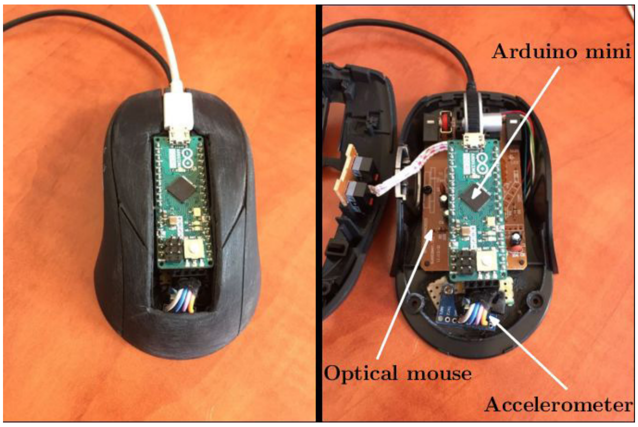

Fig. 3 The modified computer mouse equipped with an IMU

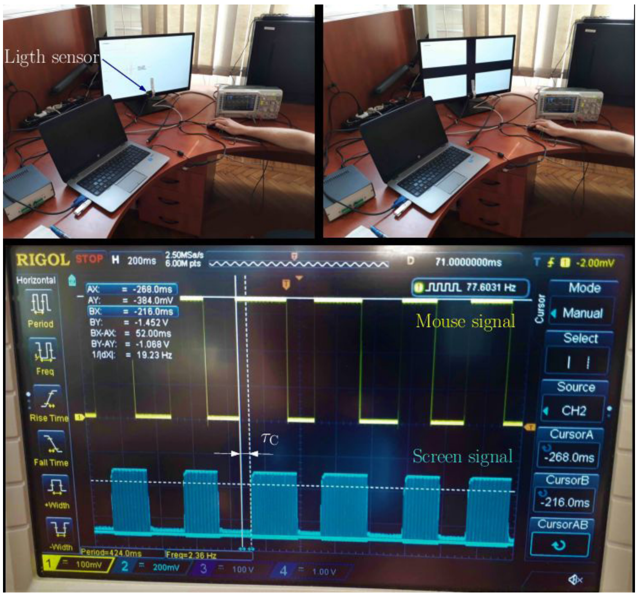

Fig. 4 Monitor response time measurement setup

Table 1 Computer delays for different computer-monitor configurations. In order to compare the same type of processing units, two HP Probook 430 were tested indicated by (1) and (2).

Processing unit	Monitor	$t_{response}$ [ms] (mean±std)
Lenovo X260	built-in display	69.6±6.7
Lenovo X260	HDMI+LG24BK550Y	101.2±10.9
Dell-Vostro	built-in display	76.4±10.2
Dell-Vostro	VGA+LG24BK550Y	80±6.9
HP Elitebook 840	built-in display	46.8±6.9
HP Elitebook 840	VGA+LG24BK550Y	49.6±5.7
HP Elitebook 840	HDMI+LG24BK550Y	58.8±7.8
HP Probook 430 (1)	built-in display	45.6±4.5
HP Probook 430 (1)	VGA+LG24BK550Y	49.2±3.1
HP Probook 430 (1)	HDMI+LG24BK550Y	58.0±10.5
HP Probook 430 (2)	built-in display	51.2±3.0
HP Probook 430 (2)	VGA+LG24BK550Y	55.6±4.9
HP Probook 430 (2)	HDMI+LG24BK550Y	58.4±9.2

3 Comparison of the different inputs

In order to calibrate the gain K Eq. (4), a crank mechanism was constructed, which produces a well-defined periodic motion as output. The mechanism can also be used to compare pixel-based acceleration and IMU-based acceleration with the exact motion. The sketch of the calibration setup can be seen in Fig. 5. The mechanism was driven by a standard 12V DC motor with constant rotational speed. The speed of the shaft was measured by an incremental encoder. The computer mouse (with the acceleration sensor) was attached to the end point of the mechanism and a linear motion was generated with known velocity and acceleration profile. The displacement and the acceleration of the end point can be expressed explicitly as function of time in the form

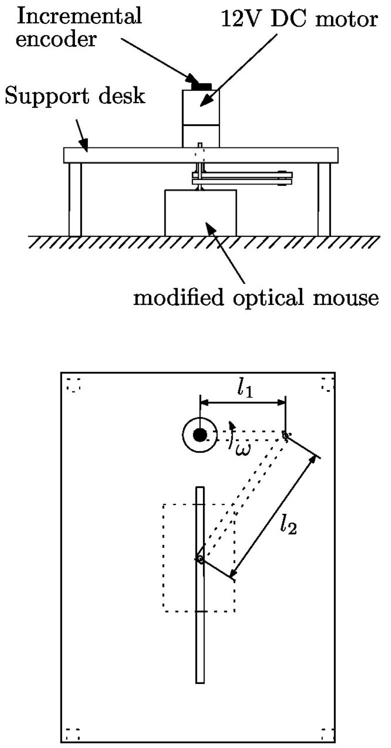

Fig. 5 Sketch of the calibration mechanism

$$r_\xi(t) = l_1 \cos(\omega t) + \sqrt{l_2^2 - l_1^2 \sin^2(\omega t)}, \tag{8}$$

$$a_\xi(t) = \ddot{r}_\xi(t), \tag{9}$$

respectively.

The measured acceleration signals can then directly be compared to the computed acceleration $a_\xi(t)$. Measurements were carried out with different rotational speed and both the pixel-based and the IMU-based acceleration signals were recorded. The gain factor was $K = {}^{l_{mainp}}\!/_{s_{screen}} = 8.85*10^{-5}$ m/pixel, where l_{mainp} is the length of the manipulation in meters, and s_{screen} is the size of the screen in pixels. This gain was determined by minimizing the error between the pixel-based acceleration and the computed acceleration $a_\xi(t)$. (Note that the same gain factor is obtained when the manipulation length of the mouse is converted to screen pixels.) An example for the time series of the two different measurements and the exact acceleration profile can be seen in Fig. 6. Both the pixel-based and the IMU-based signals are affected by noise. This is caused mainly by the friction and the small tolerance play in the calibration mechanism. In order to remove this noise, a Savitzky-Golay filter was applied on the measured signals (see bottom panel in Fig. 6). Here, the delay between the pixel-based acceleration and the IMU-based acceleration can clearly be seen, which is equal to the filtering delay $\tau_{filter} \cong 50$ ms.

In order to compare the different inputs, we calculated the RMS error between the measured and the computed signals over 2^{11} = 2048 sampling points (a time duration of appr. 34 s). In Table 2, the calculated cumulative errors were collected. It is concluded that both inputs are good enough to make realistic measurements in virtual stick balancing. The application of pixel-based signal is more convenient and straightforward, because of the commercial availability of computer mouse.

This approach, on the other hand, presents an additional delay ($\tau_{filter} \cong 50$ ms) due to signal filtering. The application of the IMU-based acceleration sensor can reduce the computer delay in the virtual balancing task, since no filtering is required, while the resulted cumulative

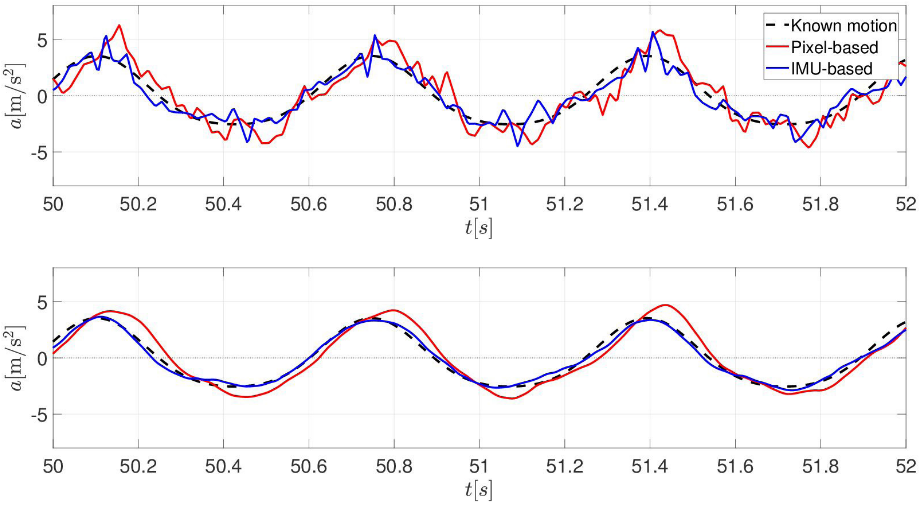

Fig. 6 Comparison of the raw input signals (top) and the data after filtering using 5th-order Savitzky-Golay filter (bottom)

Table 2 Cumulative errors of the different inputs

Input	$\mathrm{RMS}_{\mathrm{error}}$ [m/s^2]
Pixel-based	0.7075
IMU-based	0.3553

error is about the half of that of the pixel-based approach. Therefore, in this analysis the computer mouse equipped with the IMU is used for the virtual balancing tests.

4 Conclusion

An environment for virtual stick balancing tests was designed, constructed and validated by measurements. The human operator must balance a pendulum-cart system on the screen using the computer mouse as manipulator. The acceleration of the computer mouse is determined in two different ways: using double discrete differences of the pixel-based position of the cursor on the screen; and using an IMU fixed on the computer mouse. Preliminary investigation shows that IMU-based approach is more precise while it is not affected by extra filtering delays. On the other hand, preliminary balancing trials showed that subjects were able to complete the balancing tasks using both the pixel-based and the IMU-based approach despite the extra delay in the pixel-based approach. This suggests that due to its simplicity, the pixel-based approach can be used to initial balancing trials. As a plan, a microcontroller with a simplified display unit is to be developed to further reduce the artificial computer delays.

Acknowledgement

The research reported in this paper was supported by ÚNKP-2017-2-1 New National Excellence Program of the Ministry of Human Capacities and the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of Biotechnology research area of Budapest University of Technology and Economics (BME FIKP-BIO). The authors thank Richard Wohlfart for the cooperation in the response measurement tests, and the voluntary subjects, who performed the virtual balancing tests.

References

[1] Asai, Y., Tateyama, S., Nomura, T. "Learning an Intermittent Control Strategy for Postural Balancing Using an EMG-Based Human-Computer Interface", PLoS ONE, 8(5), Article number: e62956, 2013.
https://doi.org/10.1371/journal.pone.0062956

[2] Hwang, S., Agada, P., Kiemel, T., Jeka, J. J. "Identification of the Unstable Human Postural Control System", Frontiers in Systems Neuroscience, 10, Article number: 22, 2016.
https://doi.org/10.3389/fnsys.2016.00022

[3] Maurer, C., Peterka, R. J. "A New Interpretation of Spontaneous Sway Measures Based on a Simple Model of Human Postural Control", Journal of Neurophysiology, 93(1), pp. 189–200, 2005.
https://doi.org/10.1152/jn.00221.2004

[4] Milton, J., Insperger, T. "Acting together, destabilizing influences can stabilize human balance", Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 377(2153), Article number: 20180126, 2019.
https://doi.org/10.1098/rsta.2018.0126

[5] Suzuki, Y., Nomura, T., Casadio, M., Morasso, P. "Intermittent control with ankle, hip, and mixed strategies during quiet standing: A theoretical proposal based on a double inverted pendulum model", Journal of Theoretical Biology, 310, pp. 55–79, 2012. https://doi.org/10.1016/j.jtbi.2012.06.019

[6] Chagdes, J. R., Rietdyk, S., Haddad, J. M., Zelaznik, H. N., Cinelli, M. E., Denomme, L. T., Powers, K. C., Raman, A. "Limit cycle oscillations in standing human posture", Journal of Biomechanics, 49(7), pp. 1170–1179, 2016. https://doi.org/10.1016/j.jbiomech.2016.03.005

[7] Molnar, Cs. A., Zelei, A., Insperger, T. "Estimation of Human Reaction Time Delay During Balancing on Balance Board", In: 2017 13th IASTED International Conference on Biomedical Engineering (BioMed), Innsbruck, Austria, 2017, pp. 176–180. https://doi.org/10.2316/P.2017.852-048

[8] Cabrera, J. L., Bormann, R., Eurich, C., Ohira, T., Milton, J. "State-dependent Noise and Human Balance Control", Fluctuation and Noise Letters, 4(1), pp. L107–L117, 2004. https://doi.org/10.1142/S0219477504001719

[9] Milton, J., Meyer, R., Zhvanetsky, M., Ridge, S., Insperger, T. "Control at stability's edge minimizes energetic costs: Expert stick balancing", Journal of the Royal Society Interface, 13(119), Article number: 20160212, 2016. https://doi.org/10.1098/rsif.2016.0212

[10] Suzuki, S., Harashima, F., Furuta, K. "Human Control Law and Brain Activity of Voluntary Motion by Utilizing a Balancing Task with an Inverted Pendulum", Advances in Human-Computer Interaction, 2010, Article number: 215825, 2010. https://doi.org/10.1155/2010/215825

[11] Yoshikawa, N., Suzuki, Y., Kiyono, K., Nomura, T. "Intermittent Feedback Control Strategy for Stabilizing Inverted Pendulum on Manually Controlled Cart as Analogy to Human Stick Balancing", Frontiers in Computational Neuroscience, 10, Article number: 39, 2016. https://doi.org/10.3389/fncom.2016.00034

[12] Bazzi, S., Ebert, J., Hogan, N., Sternad, D. "Stability and predictability in human control of complex objects", Chaos, 28(10), Article number: 103103, 2018. https://doi.org/10.1063/1.5042090

[13] Mehta, B., Schaal, S. "Forward Models in Visuomotor Control", Journal of Neurophysiology, 88(2), pp. 942–953, 2002. https://doi.org/10.1152/jn.2002.88.2.942

[14] Milton, J. G., Fuerte, A., Bélair, C., Lippai, J., Kamimura, A., Ohira, T. "Delayed pursuit-escape task as a model for virtual stick balancing", Nonlinear Theory and Its Applications, IEICE, 4(2), pp. 129–137, 2013. https://doi.org/10.1587/nolta.4.129

[15] Suzuki, T., Lubashevsky, I., Zgonnikov, A. "Complexity of human response delay in intermittent control: The case of virtual stick balancing", [q-bio.NC] Neurons and Cognition, arXiv:1808.05002v2, Cornell University, Ithaca, NY, USA, 2018. [online] Available at: https://arxiv.org/abs/1808.05002 [Accessed: 23 July 2019]

[16] Zgonnikov, A., Lubashevsky, I., Kanemoto, S., Miyazawa, T., Suzuki, T. "To react or not to react? Intrinsic stochasticity of human control in virtual stick balancing", Journal of The Royal Society Interface, 11(99), Article number: 20140636, 2014. https://doi.org/10.1098/rsif.2014.063

[17] Insperger, T., Milton, J., Stepan, G. "Semidiscretization for Time-Delayed Neural Balance Control", SIAM Journal on Applied Dynamical Systems, 14(3), pp. 1258–1277, 2015. https://doi.org/10.1137/140975632