BICYCLING SIMULATOR CALIBRATION: SPEED AND STEERING LATENCY

Technical Transfer Deliverable

by

David S. Hurwitz, Ph.D.
Dylan Horne
Hisham Jashami
Masoud Ghodrat Abadi, Ph.D.
Oregon State University

Sponsorship
Pacific Northwest Transportation Consortium (PacTrans)

for
Pacific Northwest Transportation Consortium (PacTrans)
USDOT University Transportation Center for Federal Region 10
University of Washington
More Hall 112, Box 352700
Seattle, WA 98195-2700

In cooperation with US Department of Transportation-Research and Innovative Technology Administration (RITA)

ABSTRACT
Bicycling simulation allows for the low-risk experimental study of human factors within transportation environments. A cyclist pedals on a stationary bike trainer, which is instrumented to detect the speed of the wheel and the steering angle of the bicycle. This paper proposes a speed calibration procedure to increase the validity of the simulator results, by using an independent bicycle computer for comparing the simulator speed. The speed ratio, defined as the simulator speed divided by the bike computer speed, approaches one when the simulator is properly calibrated. The effect of tire pressure was analyzed by examining the speed ratio for various tire pressures. The optimal tire pressure was selected as the one that provided a speed ratio closest to one when all other factors were held constant. In the final calibration, a gain factor was used to modify the simulator speed calculation that was embedded in the simulator's bicycle dynamics model. Following calibration, the final simulation speed was within 99.5% of the bicycle computer speed, indicating that the physical speed of the wheel was accurately modeled in the simulation environment. The calibration procedure uses general equations and techniques that can be applied to other bicycling simulators to calibrate speed measurements and improve the consistency of experimental data worldwide.

In the field of driving and bicycling simulation, simulator sickness has been shown to have a negative impact on driver performance. High latency, where the amount of time between operator input and the response of the visual field are mismatched, is correlated with higher rates of simulator sickness. The Oregon State University Bicycling Simulator was used to develop a framework for evaluating visual latency experienced in a bicycling simulator. A cyclist biked along a bike lane, sharply steered away from an obstacle, then countersteered to return to the bike lane. A relative measure of the steering angle was estimated with video data and was compared to the absolute measurement provided by the simulator. A cross-correlation technique identified a consistent 6 time step lag between the two measurements that represents the sampling rate of the steering cradle. During steady state steering, the delta steering angle approaches zero. Finally, the mean steering latency was found to be 115 milliseconds, with a median around 69 ms. The study provides framework for transportation researchers to measure steering latency which could be used to minimize the mismatch between the user's control of the system and the response of the visual simulation.

Keywords: Bicycling Simulator, Calibration, Speed

INTRODUCTION

Bicycling simulation allows for the careful examination of bicyclist behaviors and interactions with various elements in the built environment in a controlled experimental setting. Novel or existing infrastructure can be analyzed to determine the effectiveness of traffic control devices. Interactions between conflicting modes of travel can be evaluated with surrogate safety measures to understand crash risk. The controlled and repeatable nature of human-in-the-loop simulator experimentation provides a means to develop explanatory mechanisms for transportation user behavior, which is difficult to extract from naturalistic experiments (*1*). The virtual reality environment significantly reduces the risk for participants, who can be exposed to risky scenarios while avoiding potential harm (*2*).

Our ability to extrapolate conclusions from simulation studies to real-world practice requires that the simulation and real-world performances be matched. Thus, calibration, measurement accuracy, and validation must be given careful attention. Simulated environments may not yet be able to emulate every nuance of real-world experiences, but they are sufficient to create environments where users respond in similar ways as they do in the real world (*1*). Thus, these simulations include relative validity – meaning that users respond in the same direction as in the real world – but do not include absolute validity – meaning that the simulation response is not yet identical to the real-world response in both magnitude and direction (*2*). In fact, reducing some of the variability that is experienced in the real world contributes to the power of simulation in controlled experiments, as almost all environmental factors are administered. However, due to the limited number of bicycle simulators worldwide, results from such simulators have been considered less rigorous than similar results from the comparatively more mature field of driving simulation (*3*).

Latency is the amount of time between a stimuli and its corresponding response in a system. In the field of driving simulation, latency or transport delay measures the amount of time between the operators input and the response of the visual field (7). Figure 1A demonstrates the steering stimulus and simulator response relationship or steering latency described here. Latency is often measured in milliseconds (ms), and is dependent on the system's hardware and software configuration (7). Functionally the visual latency should be low enough that the operator is unable to notice the delay, otherwise they will have to use corrective maneuvers to settle on their desired heading (7).

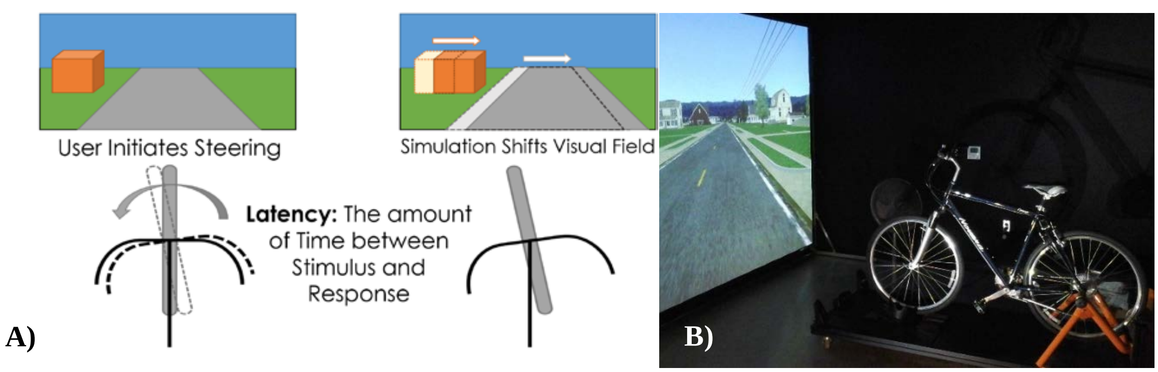

FIGURE 1 A) Visual latency in respect to a bicycle simulator, B) the Oregon State bicycling simulator

Previous studies have shown that high latency is correlated with higher rates of simulator sickness (7, 5). Simulator sickness is a type of motion sickness, commonly associated with queue conflict theory (5). During simulation the visual and vestibular (the inner ear that contributes to balance and orientation) systems are mismatched, resulting in dysphoria (5). For example, during a simulated turning movement, the visual system perceives that the body is moving through the curve, but the vestibular system perceives that the body is stationary. With high latency, the mismatch between the user's control of the system and the response of the simulation increases, resulting in higher simulation sickness rates (5).

The use of head mounted displays (HMD) can allow for a more immersive simulation experience, as the user can look in any direction and see a virtual environment, but also increases the perception of latency and in turn simulation sickness (5). With HMD, the visual field that is displayed is based on the movement of the users head, resulting in another level of computational complexity which increases latency. Predictive compensation methods use neural networks to estimate head movements, allowing the system to pre-calculate future visual fields, decreasing latency. A previous study found latency values for HMD's between 40 ms and 280 ms did not result in significant increases simulator sickness (5).

National Advanced Driving Simulator (NADS-1) researchers at the University of Iowa developed a method for evaluating the latency between an operators input and the response of the visual field (7). Their method used an image generator to display either an all-white or all black screen based on an input signal in the vehicle dynamics package, and used a photo-diode to measure the frame by frame changes between the white and black screen. The mean system latency was found to be 57ms, with a variance of 16ms. This was an improvement from a previous evaluation of the latency at 80ms, after a series of upgrades to LED projects and image generator software.

The Oregon State University bicycling simulator was the simulator evaluated in this calibration effort as shown in Figure 1B. The system uses a projection screen to display the simulated environment, surround sound speakers, a stationary bike trainer to convert the pedal energy into an input velocity that has been calibrated (Horne et al., 2018), and a front wheel steering cradle to generate the steering angle. The bicycling simulator software uses SimCreator (Realtime Technologies Inc.) to generate the virtual environment, and SimObserver to record data from the simulation simultaneously with three video feeds. The platform is adjustable, so bicycles of various sizes can be used with the simulator. The following sections evaluate the visual latency of this bicycling simulator.

The simulator uses SimCreator (Realtime Technologies Inc.) as the simulation software package, which manages the vehicle dynamics and visual field. For the bicycle simulation, the vehicle dynamics are modified to create a vehicle that has the characteristics of a bicycle. The user's vehicle has a narrower, shorter wheelbase and reduced speed to emulate the performance of a bicycle. The simulator has two input devices: a cradle for the front wheel to capture the steering angle, and an instrumented stationary bike trainer to capture the speed of the bicycle as shown in Figure 2. The visual field is projected on a screen in front of the cyclist, and a surround sound system provides audio. The platform is adjustable for various bicycle sizes, including a children's bike.

FIGURE 2 Bicycle computer used to collect the physical speed of the wheel.

METHODOLOGY

Speed Calibration

Calibrating the wheel speed input between real-world and virtual bicycling will increase the validity of the bicycling simulator. Wheel speed was calibrated through an independent bike computer as shown in Figure 2, which calculated the physical speed of the wheel based on the wheel size and a spinning magnet attached to a spoke (Figure 3). The physical speed of the wheel was transferred through the bike trainer onto a rotational sensor, which the bicycle simulator used to calculate the simulation speed.

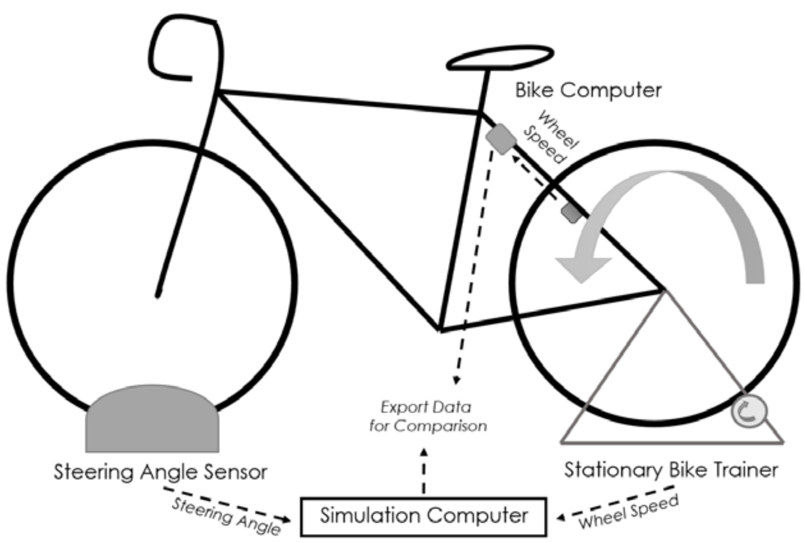

FIGURE 3 Wheel speed calibration diagram.

Speed data from the bike computer were exported and compared to the speed data recorded by the simulation computer (Figure 4). Two factors, gain and tire pressure, were investigated to understand the operational interface between the bicycle and the simulator. The gain factor, embedded in the vehicle dynamics package in the simulation software, was used to calculate the wheel speed based on the angular speed from the rotation sensor input. The gain factor can be defined by the operator and used to calibrate the simulated speed measurement.

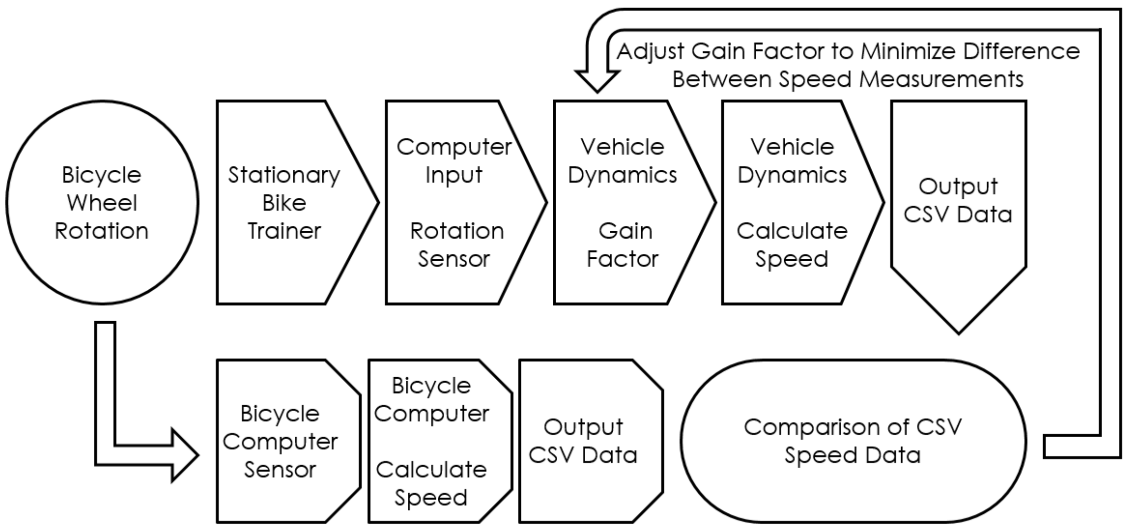

FIGURE 4 Flowchart of the comparison of two speed measurements.

The interface between the tire and the bike trainer effected the transfer of motion from the wheel and the input device to the computer. The tire pressure and tightness of the bike trainer on the tire are related to the amount of friction or rolling resistance. If the rolling resistance is too low, the tire will slip past the trainer, especially if the cyclist is exerting high torque. Previous research on bicycle tire pressure has found an inconsistent relationship between tire pressure and rolling resistance, with tire diameter being a better indicator of rolling resistance (*4*).

Various tire pressures between 40 and 60 pounds per square inch (psi) were tested with one cyclist using consistent gearing to determine the effect of tire pressure on speed measurements. The manufacturer’s recommended tire pressure for the bicycle was 50–60 psi. Higher pressure was expected to reduce slip, decreasing the variance of the simulation measurement. Equation 1 shows how the tire pressure factor and gain factor influence the simulated speed measurement.

$$Speed_{Sim} = Speed_{Wheel} * Tire\ Pressure\ Factor * Gain\ Factor \qquad (1)$$

where $Speed_{Sim}$ is the observed wheel speed calculated from the simulation; $Speed_{Wheel}$ is the actual speed of the physical wheel; *Tire Pressure Factor* accounts for losses due to the tire/bike trainer interface; and *Gain Factor* is a variable in the simulation for speed calibration.

Equation 2 shows the relationship between the bike computer and the wheel speed, where $Speed_{Bike\ Computer}$ is the speed recorded by the bike computer.

$$Speed_{Bike\ Computer} \approx Speed_{Wheel} \tag{2}$$

The goal was to minimize the difference between speed observations. Speed measurements of the bike computer and the simulator were converted to common units (mph), and the delta speed was calculated as the difference between the simulation speed and the bike computer measured speed, as shown in Equation 3.

$$\Delta\ Speed = \ Speed_{Sim} - Speed_{Bike\ Computer} \tag{3}$$

To account for variations in the magnitudes of speed measurements, a ratio of the two speed measurements was used (Equation 4). The bike computer speed, which directly represents the wheel speed, was used as the denominator. Ratio values greater than one indicate overestimates of bike speed, whereas values less than one indicate underestimates. Calibration occurs as the speed ratio approaches one.

$$Speed\ Ratio = \frac{Speed_{Sim}}{Speed_{Bike\ Computer}} \tag{4}$$

The simulation computer collects data during the whole simulation, but the bike computer only collects data while the rear bicycle wheel is moving. At the startup and termination of the simulation, several seconds of data are collected that do not have corresponding bike computer data. The time series for the simulation was trimmed to make it equal to the time series of the bike computer (Equation 5). A similar process was used to trim the end of simulation data by removing all zero-speed data recorded during simulation shut down.

$$Sim\ Time_{Modified} = Sim\ Time_{Original} - Time\ Step\ of\ First\ Non\ Zero\ Speed \tag{5}$$

The time intervals must be set to the same frequency. Instead of aggregating the higher-frequency simulation data, the bike computer data were disaggregated to match the higher frequency of the simulation computer. Each of the time intervals was rounded to the tenths place, and then the corresponding bike computer speed was matched to each time step. Under this framework, each bike computer speed corresponded to around 850 simulation speed measurements. This disparity in data resolution resulted in error, especially following large changes in speed. For example, when a participant stopped for a traffic signal in the simulation, the bike computer data were slow to respond to the change in speed, and then again slow to respond to the acceleration on the green indication. A potential improvement would be to use a geometric estimate of the higher resolution speeds based on the current and next bicycle computer speed measurements, as shown in Equation 6.

$$Speed_{Est} = m(x - x_0) + y_0 = \frac{Speed_{n+1} - Speed_n}{Time_{n+1} - Time_n} * (Time_{Est} - Time_n) + Speed_n \qquad (6)$$

where $Speed_{Est}$ is the estimated speed in the high-resolution interval; $Speed_n$ is the previous speed measurement (from the bike computer); $Speed_{n+1}$ is the next speed measurement; $Time_{Est}$ is the high-resolution interval; $Time_n$ is the previously measured interval (from the bike computer); and $Time_{n+1}$ is the next measured interval.

Inversely, the simulation data could be aggregated by using a moving average or a Kalman filter, to remove much of the noise and decrease the computational effort. These methods create a local average based on nearby data points, reducing data variability due to short-term fluctuations, while maintaining the general trend of the data. A moving average formula is shown in Equation 7.

$$\overline{Speed_{MA}} = \overline{Speed}_{Previous\ MA} + \frac{Speed_{Current\ Interval}}{n} - \frac{Speed_{Previous\ Interval}}{n} \qquad (7)$$

where $\overline{Speed_{MA}}$ is the simple moving average speed estimate; $Speed_{Current\ Interval}$ is the simulation speed measurement at current time; and n is the number of intervals included in the average.

Visual Latency
This method for observing and benchmarking the visual latency of a bicycling simulator is based on a comparison between video data and exported simulator data. Three cameras were installed in the bicycling simulator room capturing the displayed visual field, the handlebar angle, and the user. Figure 5 shows the A) visual field and B) handlebar angle video. This video feed was simultaneously recorded with output data from the simulator. The simulator input variable of interest is the steering angle. This measure is in degrees and shown in the top right of Figure 5A, with the red label Angle. Negative values correspond to steering left, and positive to right.

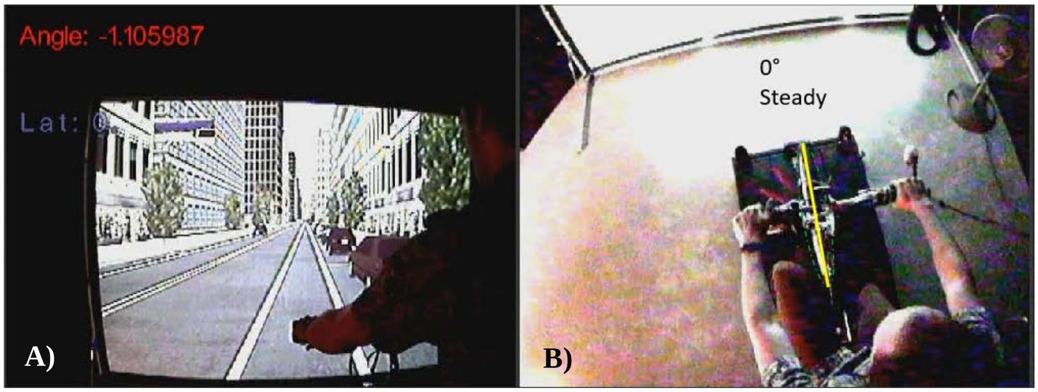

FIGURE 5 A) the visual field, simulator steering angle B) the relative steering angle

For this experiment, angles were marked on the simulator platform with tape at 0, 22.5, 45 and 90 degrees. These marks can be seen in Figure 5B, on the left side of the front wheel in the handlebar angle camera. A cyclist first pedaled the bicycle with the front wheel centered (0 degrees) in the steering cradle maintaining a steady trajectory along the bike lane in the simulator. Avoiding a car door, the cyclist sharply turned the handlebars to the left (counterclockwise) to the first angle (22.5 degrees) and held that heading into the travel lane. The cyclist then counter-steered the bicycle back to the bike lane by turning right (clockwise). Upon returning to the bike lane, the cyclist returned to a steady 0-degree turn.

A relative measure of the steering angle was produced using the video data from Sim Observer. The individual frames of the steering movement were extracted to Powerpoint. A line shape was superimposed on the front wheel to establish a datum as shown in Figure 5. The slide was duplicated, and the next video frame was pasted into the slide, replacing the original frame, leaving the datum in place. The line shape was rotated to align to the current direction of the front wheel. The rotation angle was used as a relative measure of the steering angle, based on the video data. This process was repeated for each frame throughout the duration of the steering event.

The relative measure was compared to the absolute measurement provided by the simulator. A consistent delay or lag was expected between the measurements based on system computational time. The relative measure is rather simplistic, but is consistent throughout the experiment. The angle represented was not exactly the steering angle, as the camera was positioned in a way that causes some foreshortening. Also the extraction of this angle using Powerpoint superimposed of captured images has limited precision. Nonetheless, the relative measure allows for a comparison of the steering angle, which was necessary to understand the calibration the steering and its relationship to latency.

The time latency for a time series can be estimated for each steering angle as the difference in time between the relative measure and the simulation measure as shown in equation 8. This is shown in Figure 6 as dx: Time. This represents the amount of time for the simulator to match the observed relative steering angle. Calculating this for the whole steering angle range was difficult however, as the data was not continuous, and has large gaps between measurements (especially for the simulation measurement). The total area between the relative measure and simulation measure was the same whether it was calculated in respect to the x or y direction. A similar difference could also be calculated in respect to the steering angle shown in Figure 6 as dy: Steering Angle. The time series data makes this calculation much simpler, as a relative and simulation measurement were recorded for every time step. To address the positive and negative direction, the absolute value of this difference will be used. This value can then be related to the length of a time step, to determine a distribution of visual latency as shown in equation 9. These mathematical procedures could be applied to other bicycling simulators.

$$Time\ Latency = \ Response\ time\ step\ \ - Stimulus\ time\ step \tag{8}$$

$$Steering\ Latency = (Relative\ \ - Simulator\ steering\ angle) * Timestep \tag{9}$$

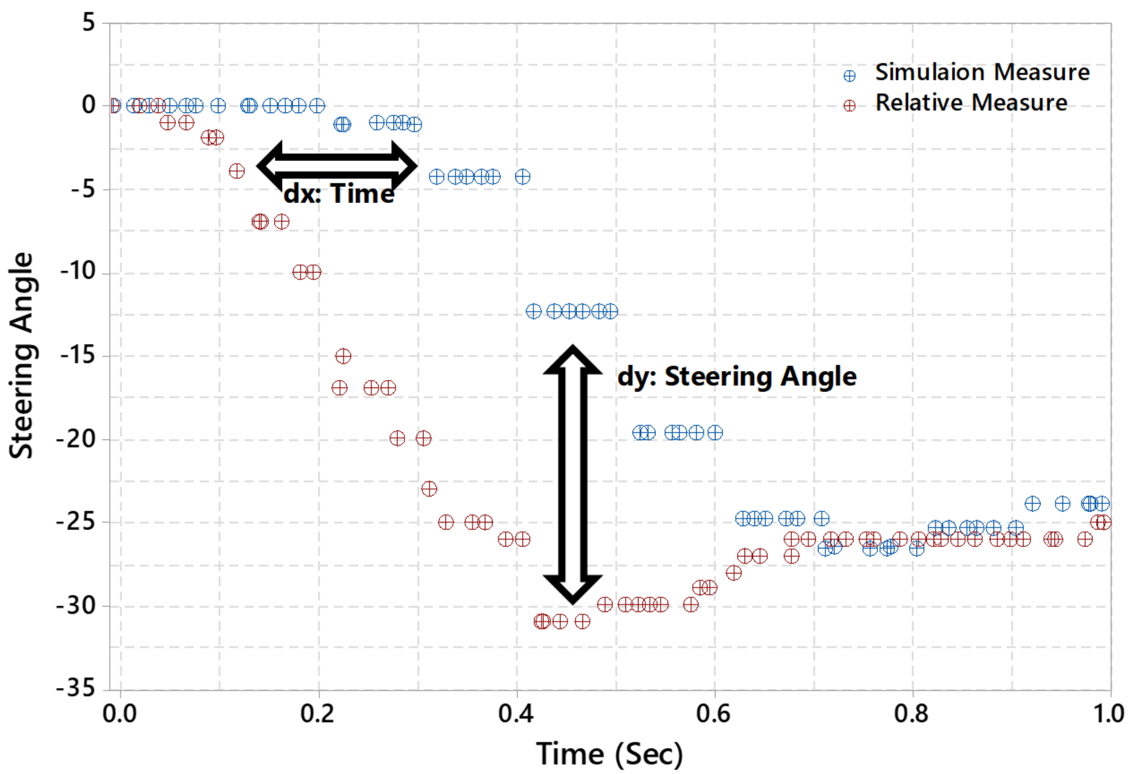

FIGURE 6 calculating the latency in respect to time or steering angle.

RESULTS

A graphical comparison of the two speed measurements was used to check the initial alignment of the data, as shown in Figure 7. Large variance between speeds followed large changes of speed due to the large difference in sampling rate, as evident around the 16,000th step when the participant stopped the bicycle. The much higher data resolution for the simulation contributed to some of the noise in the simulator speed. However, in general, the speeds followed the same trends. In Figure 7, the noise in the simulation speeds indicate that the system does occasionally over- or underestimate the speed for short durations. These events are typically very short, as the simulator records data at 85 Hertz.

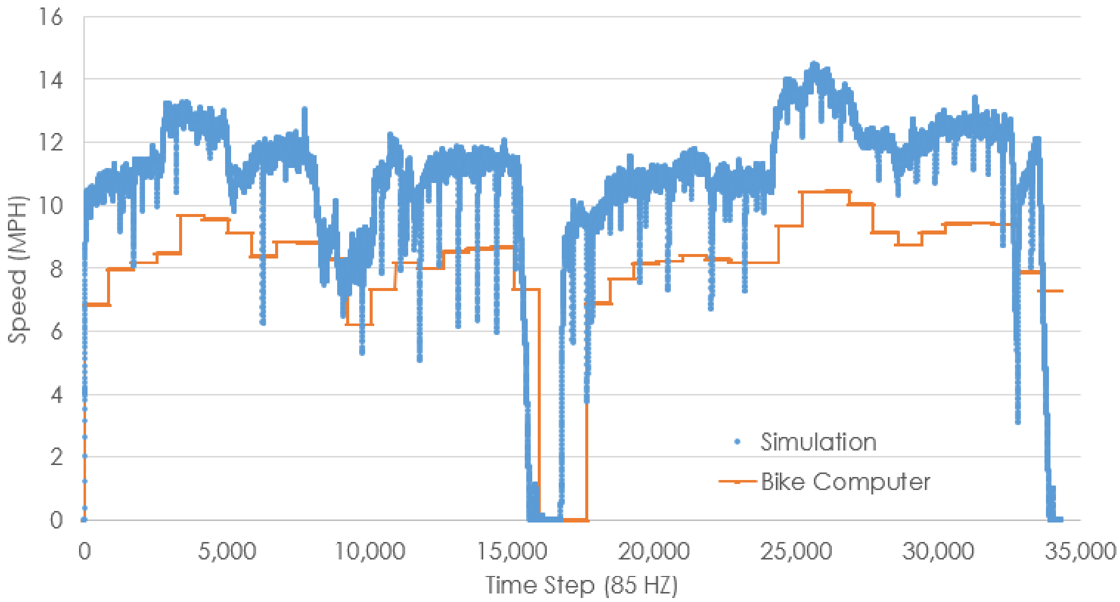

FIGURE 7 Comparison of speed data from typical participant

Tire Pressure
One participant rode the bicycle simulator while data on tire pressure were collected. In general, the ride was at least 5 minutes long using the same gearing for each run. The tire pressure of the rear tire was set at 40 psi and increased by 5 psi increments up to 60 psi. A digital tire pressure gauge was used to ensure accurate pressure measurements, as shown in Figure 8.

FIGURE 8 Digital tire pressure gauge during tire pressure analysis

Figure 9 shows the distribution of speed ratios for each tire pressure. A speed ratio of one indicates that the bike computer speed and the simulator speed were perfectly calibrated. The high variance for the 40 and 55 psi data was due to the low bike computer speeds during the startup and termination portions of the run. During these 10-second intervals, the bike computer speed was much lower than the simulation speed, resulting in large speed ratios. While the speed ratio means were similar, an ANOVA test indicated that they were statistically different (*F* value 1165.75, $P < 0.000$). A Tukey HSD test was used for multiple comparisons of the tire pressure, with all pairs except 45 and 50 psi ($P = 0.711$) being statistically different. This result indicates that tire pressure had a statistically significant influence on the speed ratio.

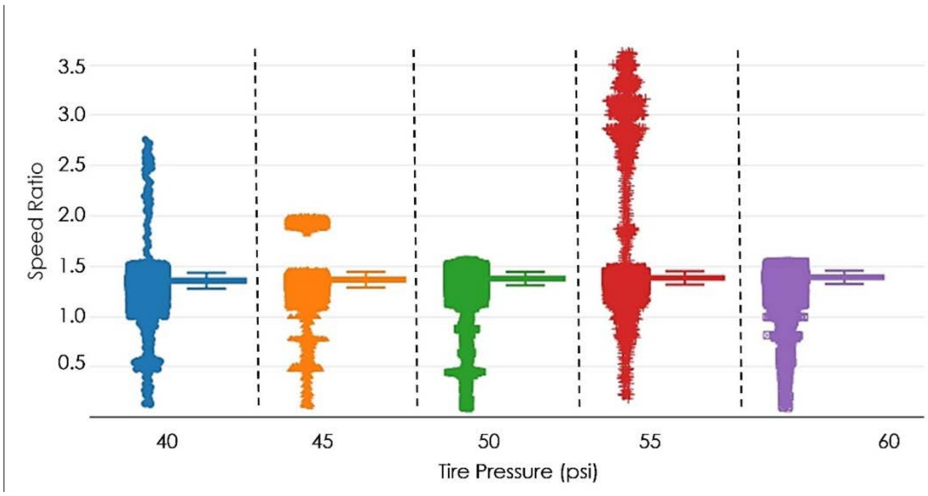

FIGURE 9 Distribution of speed ratios for each tire pressure

Descriptive statistics for the various tire pressure speed ratios are shown in Table 1. The speed ratios indicate the relative difference in measured speeds. For example, with a tire pressure of 40 psi and a bike computer speed of 10 miles per hour (mph), the simulator speed mean would be 13.49 mph. The lowest speed ratio corresponded to 40 psi, indicating that the simulator and bike computer were most calibrated at this tire pressure.

TABLE 1 Descriptive Statistics for Various Tire Pressure Speed Ratios

Tire		**Speed Ratio**			**95%CI**			
Pressure	**N**	**Mean**	**SD**	**SE**	**Lower**	**Upper**	**Min**	**Max**
40	35,375	**1.349**	0.0869	0.000462	1.349	1.350	0.1233	2.757
45	47,882	1.368	0.0990	0.000452	1.367	1.369	0.0975	1.998
50	39,332	1.369	0.0772	0.000389	1.368	1.370	0.0841	1.567
55	32,220	1.420	0.2854	0.001590	1.417	1.424	0.1846	3.626
60	34,964	1.380	0.0831	0.000444	1.379	1.381	0.0914	1.549
Total	189,773	1.376	0.1439	0.000330	1.375	1.376	0.0841	3.626

Gain Factor

Various gain factor settings were tested to analyze the effect of the gain factor on simulator speed. A single participant rode for 300 seconds at each setting, using the same gearing and tire pressure for all runs. The participant pedaled the bike to a steady-state speed before the simulation began to minimize variance during the startup period. The steady-state speed was maintained through the end of the simulation to reduce variance further.

Tire pressure was set to 60 psi during the gain factor analysis. This tire pressure is not the optimal tire pressure determined in the tire pressure experiment. Both data sets were collected before the data were analyzed, and it was incorrectly assumed that the higher tire pressure would reduce the slip the most. The factors were tested independently, however, and it is not expected that a significant interaction between the factors exist, as one is a physical relationship and the other is a software setting. The final calibration using 40 psi should be even more accurate than described here.

Table 2 shows the various gain factor levels and descriptive statistics for each of the runs. An ANOVA analysis indicated that the differences between gain factors was statistically significant (*F* value = 115845, $P < 0.000$). A Tukey HSD test indicated that each of the factors was significantly different except for the two 0.1 runs. The slightly different speed ratios with the same gain factor of 0.1 were evidence of random system variability. Setting the gain factor to zero reduced the simulation speed of the bicycle to zero, and setting the gain factor to one dramatically increased the simulation speed of the bike (8.19 times faster than real speed of the wheel). With a gain factor of one, the simulated bike would reach speeds of 65 mph, and then become unstable and crash with any steering input. Setting the gain factor to 0.1 produced the most calibrated results, with the simulated speed within 97.5% to 99.5% of the bike computer speed.

TABLE 2 Gain Factor Settings and Speed Ratio Descriptive Statistics for Each Simulation Run

Gain Factor	N	Speed Ratio Mean	SD	SE	95% CI Lower	95% CI Upper	Min	Max
0	23,487	-3.1E-7	3.44E-5	2.25E-7	-7.5E-7	1.29E-7	-0.003	0.002
0.1	22,556	**0.995**	0.140	0.00093	0.993	0.996	-6.88E-7	1.275
0.1	22,837	**0.975**	0.126	0.00084	0.973	0.976	-2.43E-8	1.252
Default*	22,147	1.585	0.194	0.00130	1.582	1.587	-8.12E-9	1.894
0.2	21,651	2.070	0.244	0.00166	2.066	2.073	-2.26E-8	2.391
1	23,487	8.190	3.191	0.02082	8.149	8.230	-9.454	11.792
Total	136,165	2.328	3.056	0.00828	2.311	2.344	-9.454	11.792

* Default gain factor is 0.15707963267949

Figure 10 graphically shows how the gain factor relates to the speed ratio from the data in Table 2. The X-axis shows the values that were tested during the sensitivity experiment, which should be interpreted as categorical variables. Theoretically, any speed ratio could be achieved by adjusting the gain factor; hence, a solid line was used between the data points. The target value was a speed ratio of one, as this was our indicator of good calibration. During the experiment, the gain factor was adjusted following the principles of Newton's Method, adjusting the value in an iterative fashion to approach our goal of a speed ratio of one. Additional steps could have been performed (0.095 or 0.15), but 99.5% was determined to be acceptable for demonstration purposes.

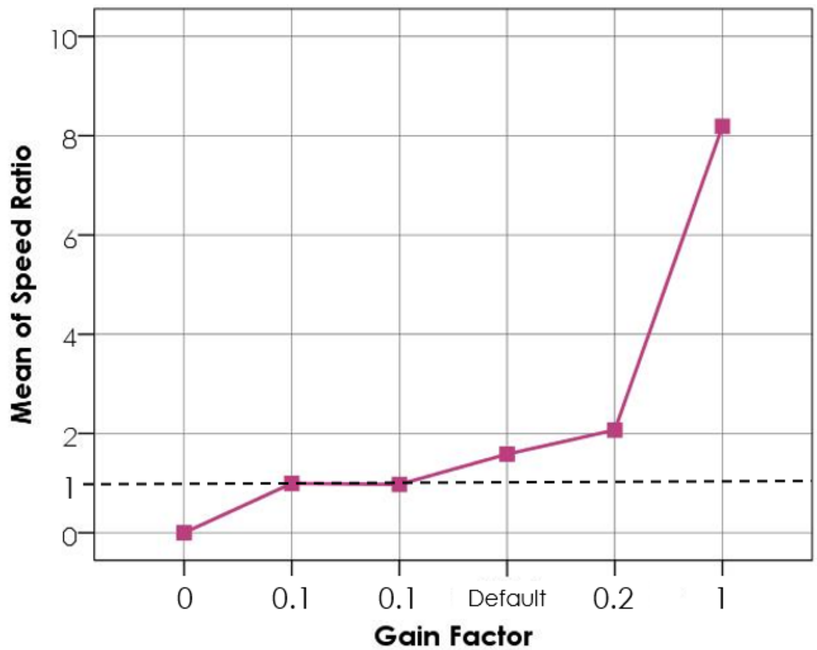

FIGURE 10 Speed ratios for each gain factor (speed ratio of one corresponds to proper calibration).

Visual Latency

The steering angle was observed for 283 time steps with or 6.4 seconds during a steering movement. The user rode in the bike lane with a steady state forward (0 degree) position, then sharply turned to the first marked 22.5 degree indication to avoid an obstacle. This position was held into the travel lane, then the user began a counter steering movement to return to the original position in the bike lane. Figure 11 shows the simulation reported steering angle measurement as well as the video relative measurement. In this graph, negative values represent turning to the left, and positive values are turning to the right. The steering latency can be interrupted as the difference between the two time series data sets (y axis). The relative measure has a slightly higher magnitude, compared to the simulation measurement, but this is likely due to the simplistic nature of the measurement. The steering latency difference is the largest during sharp steering movements (0 to 0.45 seconds). However, the difference between the measurements is minimized during steady state steering (0.92 to 1.38 seconds). The steering angle sampling rate can be observed in the stepwise behavior of the simulation measurement. The video data was recorded at 30 hz. From observing the time series data, the simulator records a steering angle every 6 time steps, or 67 ms. This sampling rate lag consistently contributes to the measured latency.

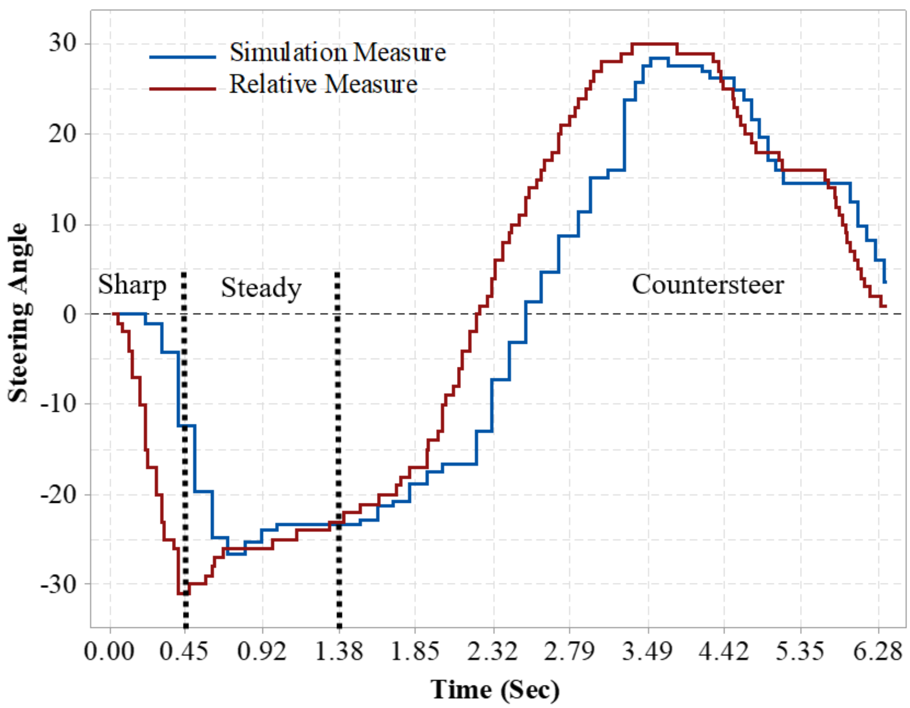

FIGURE 11 Comparison of simulation and relative steering angles

These values are then multiplied by the length of the time step (16.6 ms) to create a distribution of latency for the steering movement, as shown in Figure 12. This represents the steering latency in respect to steering angle, as shown in Equation 9. The mean latency is 115 ms, with the median around 69 ms.

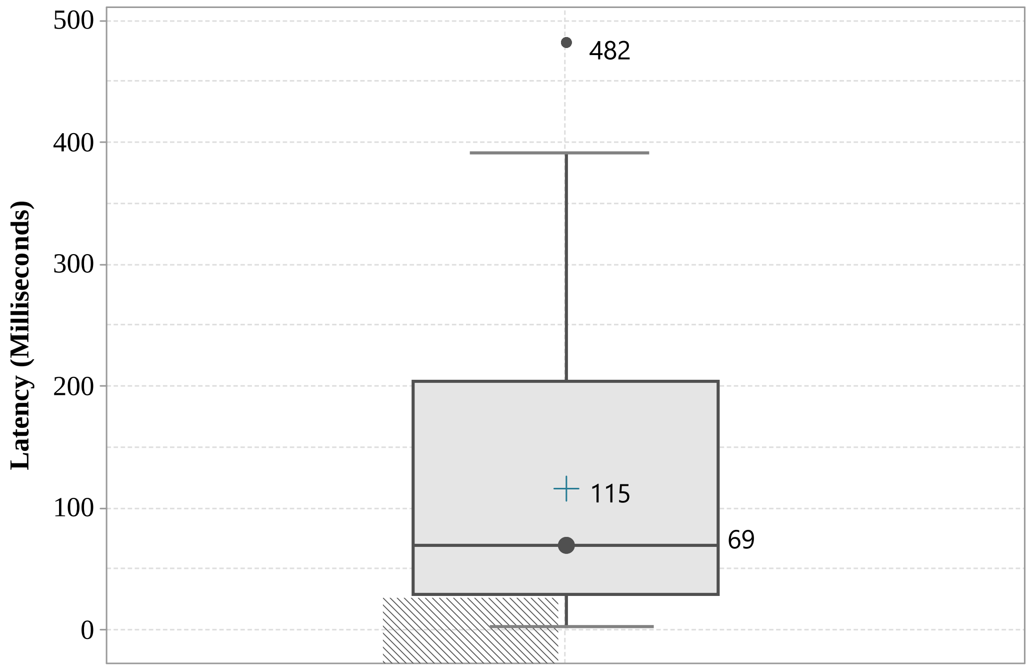

FIGURE 12 Box Plot of the distribution of steering latency values

A linear regression model was developed for the sharp steering section to evaluate the rate of steering. The based on the slope of the regression, each additional ms decreases the delta steering angle by an average of 0.7 degrees, or 11.6 degrees per time step (P-value < 0.001). Considering all 283 time steps, the average steering angles of the two measurements were not statistically different from each other using a T-test (P-value = 0.415) with 95% confidence interval (-4.66 to 1.92). The mean difference was very small (1.37 degree). This indicates that the simulator eventually records all steering movements despite the delay.

To better understand the relationship between simulation and relative measurements of steering angle, a Cross-Correlation Analysis (CCA) was conducted. Cross-correlation captures the similarities between two time series whether they related or not (6). Thus, it is helpful for identifying lag in output. Data were analyzed and visualized using Minitab software for Windows (version 18.1). As shown in Figure 13A, there is a highly positive autocorrelation between the two measurements during sharp steering phase (0 to 0.45 seconds). At twelve time steps the cross-correlation (0.59) is the greatest and is statistically significant. To transition from steady state to the sharp steering movement, the simulator needed 12 time steps to respond, and then it started updating. This suggests that the largest time latency is 12 time steps (12*16.6 ms). This is shown in Figure 13A as initial lag. However, on average, the simulation lag was 6 time steps (6*16.6 ms) to match the steering angle movement, as shown in Figure 13A. The 6 time steps interval is also is cross-correlated to multiples of 6 (time steps 6, 12, 18 and 24). This makes sense as the simulator samples the steering angle every 6 time steps, which contributes to the overall delay consistently. Finally, the autocorrelation graph shows that when the steering movement is steady for more than 6 time steps, the simulator output approaches the observed relative values. This indicates that steady steering allows the simulator to catch up on its sampling and settle into a consistent steering angle.

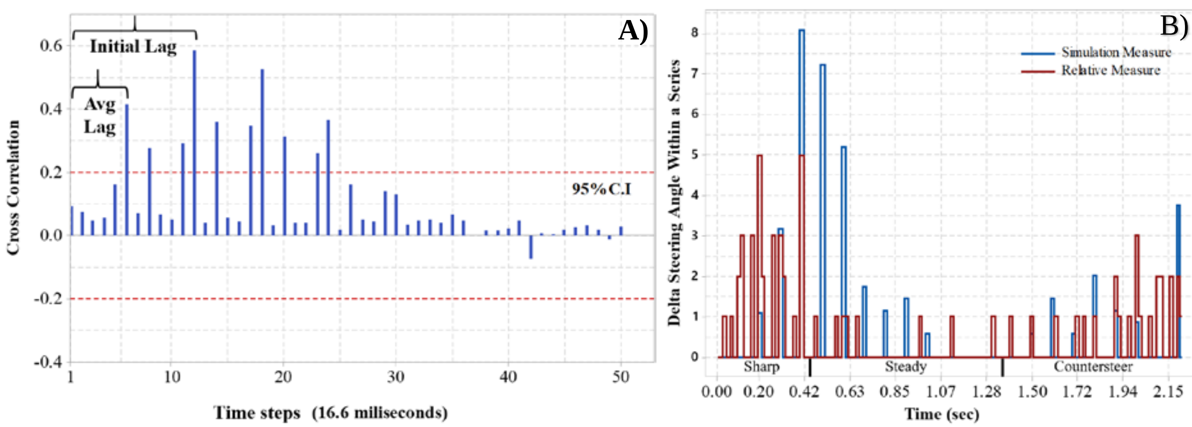

FIGURE 13 A) Cross-Correlation values for each time step B) time series delta Steering Angle values

The absolute value of the difference between the simulation measure and the relative measure is shown in Figure 13B for the sharp and steady steering. During sharp steering events the delta steering angle is large, and during steady state steering the delta steering angle approaches zero.

CONCLUSIONS

Bicycle simulator studies provide an experimental framework to evaluate novel and existing infrastructure and human factors while controlling for environmental factors and reducing risk to participants. Calibrating the inputs of the bicycle simulator improves the authenticity of the user experience. The calculated speed of the rear wheel was compared to an independent speed observation from a bike computer to minimize the difference between measurements. Calibrating the observed simulator speed and the actual speed of the bicycle wheel should make the simulation more representative of real cycling, thereby improving the user's experience and the applicability of the results.

The speed ratio, or the simulated speed divided by the bike computer speed, was used to evaluate the influences of tire pressure and gain factor. Various tire pressures were tested based on the manufacturer's recommended tire pressure range, with 40 psi having the most accurate and statistically significant speed ratio measurement. A gain factor of 0.1 brought the simulation to within 99.5% of the bike computer speed, indicating good calibration. The calibration could be further improved by additional refinement after testing tire pressures outside of the manufacturer's recommendations and additional gain factor settings.

The general procedures describe here can be applied to other bicycling simulators around the world. The use of a commercially available bike computer allows for the comparison of simulator speeds against an independent speed measurement. The calibration of speed measurements could increase the repeatability of experimental data across different simulators. The speed ratio framework enables discussion of the difference between the real speed of the bike and the simulated speed, which is especially important when validating simulator results to real-world experiences.

Latency is the amount of time between a stimulus and response in a system. For the OSU bicycling simulator, steering angle was used to develop a procedure for calculating steering latency. The difference between an observed relative measure and the simulation output of the steering angle is the foundation of the procedure. The mean steering latency was found to be 115 ms, with the median around 69 ms. These values are of a similar magnitude to other driving simulators based on existing literature (7, 5). It was expected that the steering latency would be relatively consistent throughout, but the results suggest that sharp large turning movements create more latency as compared to steady state operations. A consistent lag of 6 time steps was identified throughout the time series data using a cross-correlation method, which corresponds to the sampling rate of the steering cradle. During steady state steering the delta steering latency approaches zero. This methodology can be used to benchmark bicycle simulator steering latency, providing a performance measure for system upgrades, software improvements, or identifying high latency simulator sickness boundaries.

FUTURE RESEARCH

The experiment as described only explored the steady-state speed, to minimize speed variance. Evaluating acceleration or deceleration would be difficult using the current bike computer due to data resolution, as the system only records data every 10 seconds. Acceleration events are likely to be much shorter than this interval. Deceleration events are a potential performance measure during experimentation, as they reflect braking as a response to simulated conflicts. These events could be used as a measure of reaction time, specifically involving stopping situations. The calibration effort only focused on speed, but speed is a fundamental property of any human-in-the-loop simulator.

This research creates a standard procedure for bicycle simulator speed calibration. Applying this methodology to other bicycle simulations will help to improve fidelity of bicycling simulation in general, as speed measurements will have a common calibration procedure. The future of this research thread includes applying this procedure to other bicycle simulators, developing a procedure to analyze the calibration of steering input through observation of visual latency, and validation studies to match simulator performance to field experiments.

Future research for validation of the simulator will help to answer questions about the human perception of the simulation. The focus of this research was to calibrate the calculated speed of the simulator to a physical measurement of the speed of the wheel, rather than to calibrate the human perception of speed. The implication is that a calibrated simulator will better emulate the real-world experience. However, due to the relative validity of simulation research, the participant's perception of the simulation speed is arguably at least as important, if not more so. Due to the limits of what sensor information can be presented in a simulator environment, the simulator may seem much faster or slower to participants than the real-world experience. Therefore, a study of user perceptions of bicycling simulation should be performed with the research question, "Does the bicycle simulator match user expectations from riding a real bicycle?" This feedback mechanism should be used to validate the simulator to match user expectations.

This paper addressed the visual latency of a bicycling simulator based on the steering angle, but does not calibrate the steering angle. The bicycling simulator allows for only one degree of freedom for steering, rotating the handle bar. The actual steering of a real world bicycle is considerably more complicated however, as the bicycle and rider lean in the direction of steering. The real world steering relationship is nonlinear, with high steering angles potentially causing crashes.

To calibrate the steering angle the following procedure is proposed. Using an instrumented bicycle that can track the steering angle and the position of the bicycle, test a variety of riders at various speeds to determine the minimum turning radius of the bicycle. In an open environment, riders would attempt to minimize the radius of the bicycle turning 180 degrees. An average value of the minimum turning radius could be calculated for different speeds groups. A similar experiment could be developed in the bicycling simulator to test the minimum turning radius of the simulator. The maximum steering angle, and steering angle gain (both factors accounted for within the vehicle dynamics package) could be adjusted to match the bicycling simulator minimum turning radius to the observed experimental minimum for the instrumented bicycle.

REFERENCES

1. O'Hern, S., Oxley, J., and M. Stevenson. Validation of a Bicycle Simulator for Road Safety Research. *Accident Analysis & Prevention*, Vol. 100, 2017, pp. 53–58. http://dx.doi.org/10.1016/j.aap.2017.01.002
2. Kaptein, N., Theeuwes, J., and R. Van Der Horst. Driving Simulator Validity: Some Considerations. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1550, 1996, pp. 30–36.
3. Nehaoua, L., Arioui, H., and S. Mammar. Review on Single Track Vehicle and Motorcycle Simulators. *19th Mediterranean Conference on Control and Automation*, Corfu, Greece, 2011, pp. 940–945.
4. Akanda, F., Ahmad, D., Jamarie, O., Sulaiman, S., and A. Fashina. Empirical Determination of the Motion Resistance of Pneumatic Bicycle Wheels for On and Off-Road Performance. *African Journal of Agricultural Research*, Vol. 5, 2010, pp. 3322–3332.
5. Buker, T. J., Vincenzi, D. A., & Deaton, J. E. (2012). The effect of apparent latency on simulator sickness while using a see-through helmet-mounted display: Reducing apparent latency with predictive compensation. *Human factors*, *54*(2), 235-249.
6. Medenica, Z. (2012). Cross-correlation based performance measures for characterizing the influence of in-vehicle interfaces on driving and cognitive workload. Dissertation: University of New Hampshire
7. Omar, A., Heibrink, D., Schwarz, C., & G. Wagner (2016). Measuring the Latency of the NADS-1 Simulator. *Proceedings of IMAGE 2016 Conference*, Dayton, Ohio.