iCSC International Cycling Safety Conference

6th Annual International Cycling Safety Conference
21–22 September 2017, Davis, California, USA

Description of a model based bicycle simulator

O. Lee, G. Dialynas, J. C. F. de Winter, R. Happee, A. L. Schwab

Department of BioMechanical Engineering
Delft University of Technology
Mekelweg 2, NL-2628 CD Delft, The Netherlands
e-mail: {o.z.lee, g.dialynas, j.c.f.dewinter, r.happee, a.l.schwab}@tudelft.nl

Keywords: bicycle dynamics, haptics, simulators.

1 Introduction

Recent years have seen an increase in cycling as a transport mode in urban centers. This has spurred an interest in the use of bicycle simulators to study cyclist behavior [1, 2, 3, 4]. However, few implement a model based approach that couples the bicycle roll and steer in a realistic manner [5]. Balancing is a key task in cycling and we aimed to develop a simulator that allows us to study the effect of balance on the rider's higher level cognitive decisions.

2 Bicycle model and state estimation

The simulator implements the Whipple bicycle model using the benchmark parameters [6], as it is a simple model that has been experimentally verified [7]. The Whipple model uses linearized equations of motion to describe the bicycle stability in terms of roll angle ϕ and steer angle δ. These equations can be expressed in state space form parameterized by forward speed v

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(v)\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{1}$$

where $\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\psi} & \dot{\phi} & \dot{\delta} & \ddot{\phi} & \ddot{\delta}\end{bmatrix}^T$, $\boldsymbol{x} = \begin{bmatrix}\psi & \phi & \delta & \dot{\phi} & \dot{\delta}\end{bmatrix}^T$, $\boldsymbol{u} = \begin{bmatrix}T_\phi & T_\delta\end{bmatrix}^T$, and expressions for $\boldsymbol{A}(v)$ and $\boldsymbol{B}$ can be obtained from [6]. Note that our formulation extends the state by adding yaw angle ψ because yaw rate $\dot{\psi}$ can be expressed linearly in terms of the state $\boldsymbol{x}$. While roll torque T_ϕ is an input to the system, it is not measured and cannot be applied by the rider in our simulator since experiments have shown steer control to be the dominant rider control action [8]. Because the simulator has a fixed base, ϕ and ψ do not physically exist and cannot be measured with sensors. We treat them as unmeasured states and a Kalman filter is used to obtain an estimate.

3 Simulator description

An image of the fixed base simulator is shown in Figure 1. The physical system consists of a steering shaft, the rear half of a bicycle frame, and rollers. Reach and stack can be adjusted for a given rider as the

This work is licensed under the Creative Commons Attribution 4.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.



Figure 1: Delft fixed base bicycle simulator with model based haptic feedback torque

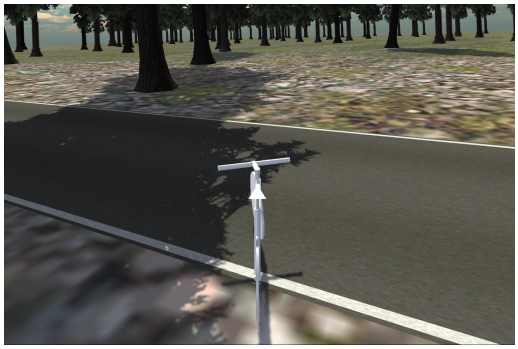


Figure 2: Screenshot of the bicycle simulator visual environment rendered by Unity

steering shaft is telescopic and connected to the frame using prismatic joints. The steering shaft incorporates a high resolution angular encoder, an inline torque sensor, and an electric motor connected in a direct drive configuration. A lower resolution angular encoder measures rear wheel position and is used to estimate v.

The simulation of the bicycle model and state estimator are implemented in a microcontroller that runs an update loop at a rate of 1 kHz. Because the bicycle itself is visualized in our simulator, and the expressions for the rear wheel position coordinates x_p, y_p are nonlinear, explicit integration is used to obtain an extended state $\boldsymbol{x}_e = \begin{bmatrix} x_p & y_p & \psi & \phi & \delta & \dot{\phi} & \dot{\delta} \end{bmatrix}^T$.

A pose object $(x_p, y_p, \theta, \psi, \phi, \delta)$ describing the configuration of the virtual bicycle is transmitted to the visual environment at a rate of 120 Hz. The pose values are set using the most recently computed $\boldsymbol{x}_e$, except the rear frame pitch angle θ, which is computed using the Newton-Raphson method just prior to transmission.

The Unity game engine is used to render the virtual bicycle in the visual environment. The engine supports output to a standard monitor, multiple monitors, or a head mounted display (e.g. Oculus Rift). An example environment is shown in Figure 2.

4 Virtual dynamics

The simulator has a fixed base and as stated in Section 2, roll angle ϕ is purely virtual, as are x_p, y_p, θ, and ψ, although the latter four do not affect the bicycle steer dynamics. All these pure virtual states are presented to the rider only through visualization. However, steer angle δ is both virtual and physical. It is displayed in the visual environment and realized by the handlebars in the physical environment and must be synchronized between the two environments. This is handled using admittance control to impose the steer dynamics of the virtual bicycle on the physical steering assembly.

We specify that a term represents a physical value with the subscript postfix $_\pi$ and a virtual value with $_\rho$, when explicitness is necessary. For virtual terms, we also optionally use the superscript $^-$ to denote the value during the simulation loop before a sensor and Kalman update and the superscript $^+$ to denote the

This work is licensed under the Creative Commons Attribution 4.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

value after the sensor and Kalman update.

The physical torque applied by a rider must be virtualized to the steer torque $T_{\delta,\rho}$ used as input to the Whipple model as the inertia of the bicycle handlebars, bicycle stem, and steering shaft above the torque sensor will affect the torque sensor measurement. We use the term *upper assembly* to denote these components as a group.

Assuming that friction can be neglected, the dynamics of the upper assembly are described by

$$T_{\delta,\rho} + T_{S,\rho} = I_{\delta_u}\ddot{\delta}_\rho \tag{2}$$

where I_{δ_u} is the moment of inertia of the upper assembly about the steer axis, $T_{S,\rho}$ is the sensor torque measurement which we take to be equal to $T_{S,\pi}$, $T_{\delta,\rho}$ is the rider applied steer torque, and $\ddot{\delta}_\rho$ is the steer acceleration.

The calculation of virtual steer torque $T_{\delta,\rho}$ occurs at the beginning of the simulation loop allowing us to use $\hat{\boldsymbol{x}}_\rho$ and v updated in the previous iteration, denoted with $^-$. Using Equation 2 and the Whipple model to obtain $\ddot{\delta}$, we calculate $T_{\delta,\rho}$ as

$$T_{\delta,\rho} = I_{\delta_u,\rho}\ddot{\delta}_\rho^- - T_{S,\pi} \tag{3}$$

where

$$\ddot{\delta}_\rho^- = \begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix}\boldsymbol{A}(v_\rho^-)\hat{\boldsymbol{x}}_\rho^- \tag{4}$$

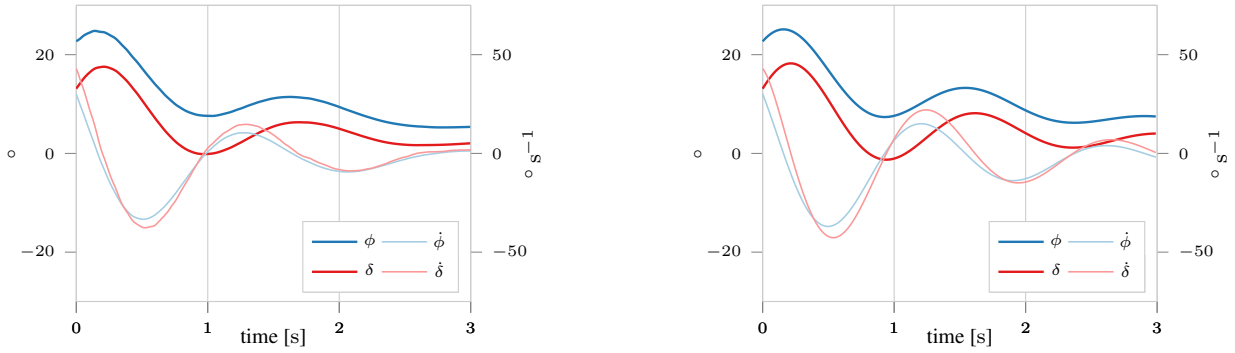
With steer torque calculated in Equation 3, we obtain the input vector $\boldsymbol{u}_\rho = \begin{bmatrix}T_{\phi,\rho} & T_{\delta,\rho}\end{bmatrix}^T$. In the current implementation, we set $T_{\phi,\rho} = 0$ because we do not simulate any external roll torques. An update of the Kalman filter returns an updated state estimate $\hat{\boldsymbol{x}}_\rho^+$, which contains the desired velocity of the steering assembly $\dot{\delta}_\rho^+$.

$$\begin{aligned}\hat{\boldsymbol{x}}_\rho^{+|-} &= \boldsymbol{A}_d(v_\rho^+)\hat{\boldsymbol{x}}_\rho^- + \boldsymbol{B}_d\boldsymbol{u}_\rho \\ \hat{\boldsymbol{x}}_\rho^+ &= \hat{\boldsymbol{x}}_\rho^{+|-} + \boldsymbol{K}\left(\begin{bmatrix}\psi_\rho^- \\ \delta_\pi\end{bmatrix} - \boldsymbol{C}\hat{\boldsymbol{x}}_\rho^{+|-}\right)\end{aligned} \tag{5}$$

where $\hat{\boldsymbol{x}}_\rho^{+|-}$ is the a priori state estimate, $\boldsymbol{A}_d, \boldsymbol{B}_d$ are the discretized state and input matrices that can be obtained from Equation 1, $\boldsymbol{K}$ is the optimal Kalman gain, $\boldsymbol{C}$ simply returns the ψ and δ terms of $\boldsymbol{x}$, and we simply use the last calculated value as a measurement for ψ because there is no physical sensor associated with this coordinate. Admittance control is used as we impose $\dot{\delta}_\rho^+$ by commanding the motor in velocity mode; the motor drive determines the feedback torque to apply given the velocity reference.

The implementation of the equations in this section is presented as pseudocode in Algorithm 1 and is also available online [9].

This work is licensed under the Creative Commons Attribution 4.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.



(a) Bicycle simulator

(b) Whipple simulation

Figure 3: Time series plots of the Whipple model state, excluding ψ, for the bicycle simulator (3a) and a simulation of the Whipple model (3b). The model velocity is set to $v = 5.0\,\mathrm{m\,s^{-1}}$ which lies in the stable speed range. Initial state is set to $\boldsymbol{x}_0 = \begin{bmatrix} 22.72^\circ & 13.15^\circ & 29.94\,^\circ\,\mathrm{s^{-1}} & 42.86\,^\circ\,\mathrm{s^{-1}} \end{bmatrix}$ and $T_\delta = 0 \;\forall\; t$. Both 3a and 3b show that all states converge to zero.

Algorithm 1 Simulation time step iteration

1: $v_\rho^- \leftarrow v_\rho^+$
2: $\hat{\boldsymbol{x}}_\rho^- \leftarrow \hat{\boldsymbol{x}}_\rho^+$
3: measure $T_{S,\pi}, \delta_\pi, v_\pi$
4: compute $T_{\delta,\rho}$ using $T_{S,\pi}, v_\rho^-, \hat{\boldsymbol{x}}_\rho^-$ ▷ Equation 3 and Equation 4
5: $\boldsymbol{u}_\rho \leftarrow \begin{bmatrix} 0 & T_{\delta,\rho} \end{bmatrix}^T$
6: $v_\rho^+ \leftarrow v_\pi$
7: compute $\hat{\boldsymbol{x}}_\rho^+, \dot{\delta}_\rho^+$ using $v_\rho^+, \hat{\boldsymbol{x}}_\rho^-, \boldsymbol{u}_\rho, \delta_\pi$ ▷ Equation 5
8: apply $T_{M,\pi}$
9: apply $\dot{\delta}_\rho^+$

5 Preliminary results

The bicycle simulator is still under development. However, initial tests show behavior subjectively resembles the Whipple model. The results of a test are shown in Figure 3a for the bicycle simulator running at a constant $v = 5.0\,\mathrm{m/s}$ with $T_\delta = 0$ and an arbitrarily chosen initial condition $\boldsymbol{x}_0$. A simulation of the Whipple model with the same $v, T_\delta, \boldsymbol{x}_0$ is shown in Figure 3b and we observe similar damped oscillating behavior in both cases. Next steps involve studying balance behavior for a small group of participants given the observed realistic steering dynamics, and therefore, realistic handlebar haptic feedback.

This work is licensed under the Creative Commons Attribution 4.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

6 Acknowledgements

We gratefully acknowledge the European Commission for their support of the Marie Curie Initial Training Network (ITN) project Nr. 608092 MOTORIST (Motorcycle Rider Integrated Safety), www.motorist-ptw.eu.

REFERENCES

[1] S. Caro and S. Bernardi, "The role of various sensory cues in self-speed perception: a bicycle riding simulator preliminary study," in Driving Simulation Conference, 2015, pp. 63–66.

[2] R. Herpers, W. Heiden, M. Kutz, D. Scherfgen, U. Hartmann, J. Bongartz, and O. Schulzyk, "FIVIS bicycle simulator: an immersive game platform for physical activities," in Proceedings of the 2008 Conference on Future Play: Research, Play, Share. ACM, 2008, pp. 244–247. [Online]. Available: https://doi.org/10.1145/1496984.1497035

[3] S. O'Hern, J. Oxley, and M. Stevenson, "Validation of a bicycle simulator for road safety research," Accident Analysis & Prevention, vol. 100, pp. 53–58, 2017. [Online]. Available: https://doi.org/10.1016/j.aap.2017.01.002

[4] J. M. Plumert, J. K. Kearney, and J. F. Cremer, "Children's perception of gap affordances: Bicycling across traffic-filled intersections in an immersive virtual environment," Child development, vol. 75, no. 4, pp. 1243–1253, 2004. [Online]. Available: https://doi.org/10.1111/j.1467-8624.2004.00736.x

[5] S. Yin and Y. Yin, "Implementation of the interactive bicycle simulator with its functional subsystems," Journal of computing and information science in engineering, vol. 7, no. 2, pp. 160–166, 2007. [Online]. Available: https://doi.org/10.1115/1.2720885

[6] J. P. Meijaard, J. M. Papadopoulos, A. Ruina, and A. L. Schwab, "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review," in Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 463, no. 2084. The Royal Society, 2007, pp. 1955–1982. [Online]. Available: https://doi.org/10.1098/rspa.2007.1857

[7] J. D. G. Kooijman, A. L. Schwab, and J. P. Meijaard, "Experimental validation of a model of an uncontrolled bicycle," Multibody System Dynamics, vol. 19, no. 1-2, pp. 115–132, 2008. [Online]. Available: https://doi.org/10.1007/s11044-007-9050-x

[8] J. K. Moore, J. D. G. Kooijman, A. L. Schwab, and M. Hubbard, "Rider motion identification during normal bicycling by means of principal component analysis," Multibody System Dynamics, vol. 25, no. 2, pp. 225–244, 2011. [Online]. Available: https://doi.org/10.1007/s11044-010-9225-8

[9] O. Lee and M. M. van Gelderen, "oliverlee/phobos: v0.1," Mar. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.399233

This work is licensed under the Creative Commons Attribution 4.0 Unported License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.