WeBikes: A configurable, open-source add-on for simulating single-track vehicle dynamics in the Webots robot simulation software

Wenjia Li[1] , Paris Francis[1], Robert McClosky[1], Alexander Brown[2]
[1]*Integrative Engineering Program—Lafayette College, and*
[2]*Department of Mechanical Engineering—Lafayette College,*
Easton, PA
email: brownaa@lafayette.edu

Abstract

This paper introduces WeBikes, an open-source simulation element for the Webots robot simulation software that allows users to easily simulate the dynamics of bicycles and motorcycles. WeBikes allows users to configure the dimensional, inertial, and visual properties of a two-wheeled vehicle. Open-loop comparisons between the WeBikes vehicle's behavior and a canonical linear model of single-track vehicle dynamics show that WeBikes captures the self-stabilization behavior of two wheeled vehicles as predicted by the linear model. Additionally, the WeBikes add-on includes a simple virtual rider that stabilizes the vehicle using optimal linear feedback control. Comparisons between the closed-loop behavior of the WeBikes vehicle and the linear model show strong agreement, which indicates that under the direction of a virtual rider, the WeBikes vehicle can be used to simulate a vehicle-rider system navigating a roadway. Finally, a simple case study demonstrates how WeBikes can be used in a road safety simulation context. The case study explores how a motorcycle-rider system behaves when performing a lane change between lanes of disparate heights, illustrating how WeBikes can provide insight into the interaction between single-track vehicle behavior and geometric roadway design.

Keywords – Vulnerable Road Users, Single-Track Vehicles, Powered Two-Wheelers, Vehicle Dynamics, Dynamic Simulation, Motorcycle Dynamics, Bicycle Dynamics

1. Introduction

The Insurance Information Institute (III) reports that in 2021, there were 9,881,414 road registered motorcycles in the United States, and that motorcycles were involved in 5,932 traffic fatalities and 82,686 traffic injuries [19]. The number of road-going bicycles is more difficult to estimate, but the III reports an estimated 41,615 bicycle injuries and 966 fatalities [18]. Given these statistics, there's a visible need to improve single-track vehicle safety on public roads. One way to accomplish this is with high-fidelity simulations of single-track vehicle dynamics and their interactions with both the roadway itself and with other road users. For single-track vehicles specifically, simulation software options include the capable and popular "BikeSim," a commercial vehicle dynamics analysis and simulation software for two-wheeled vehicles [2,4,5,6,17]. BikeSim's uses in safety research range widely, from the development of technology to optimization of vehicle design [3] to analysis of rider comfort on varying road surfaces [16]. However, BikeSim's capability and versatility comes with a high monetary cost, and its closed-source, proprietary nature somewhat limits the customizability of vehicles and certain aspects of their environments. An open-source alternative with similar capability would allow better replicability of results and better access to bicycle and motorcycle safety simulations for researchers.

Webots [20] is a free, open-source, and multi-platform desktop application suitable for the simulation of vehicles including cars [7,8,10,12]. Users can modify nearly every aspect of a simulated vehicle, the behavior of other road-users, and properties of the environment. Additionally, users can equip simulated vehicles with a variety of fully customizable sensors [14]. Despite evidence that Webots is a capable tool for automobile simulation (e.g. [1,7]), studies that use Webots to simulate single-track vehicle dynamics are scarce.

While the official Webots software releases feature a standard basic two-wheeled vehicle for users to include in their simulations, this built-in model does not accurately reflect the complex dynamics of bicycles, motorcycles, or other Powered Two-Wheelers (PTWs). Developing models of PTWs from scratch is possible in the Webots software by manually connecting appropriate rigid bodes with joints, sensors and actuators. However, this is both time-consuming and complex, requiring a high degree of familiarity with the software. To address this limitation, we have developed the "WeBikes" add-on [21] that offers easy customization and accurately represents the dynamics of single-track vehicles.

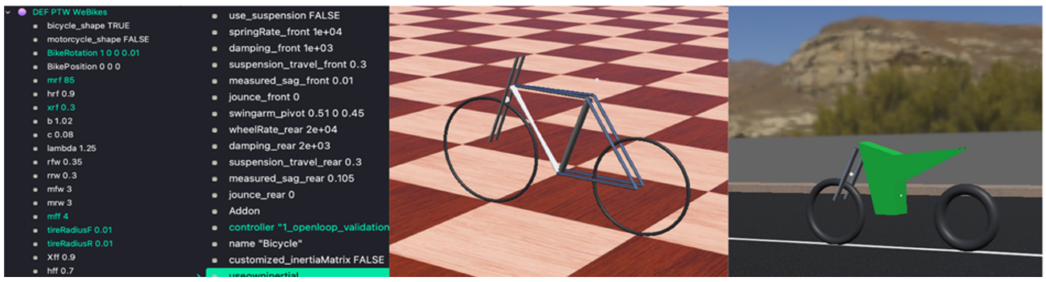
This add-on consists of a Webots "PROTO," or configurable drop-in simulation element, that can be modified to represent a bicycle or PTW. This PROTO is accompanied by a nominal "controller" element written in Python that is capable of stabilizing and steering the vehicle.

The remainder of this paper is organized as follows. First, it outlines the structure and capability of the WeBikes PROTO in Webots. It then examines how the dynamics of the PROTO configured as both a bicycle and a motorcycle compare with a canonical linear model of single-track vehicle dynamics from the literature. Simulations are performed both with and without control inputs from a virtual rider. Finally, to show its utility in a road safety context, we use the WeBikes PROTO to study motorcycle dynamics when traversing a lane change maneuver between two lanes of disparate pavement height. Specifically, we compare the vehicle's behavior when the uneven lanes are separated by a sharp pavement edge to its behavior when two paved, uneven lanes are separated by an angled transition between low and high pavement.

2. Methods

2.1. The WeBikes add-on for the Webots robot simulation software

The Webots software is a powerful simulation environment, but it does not, by itself, provide sufficient infrastructure for researchers to study the stability and/or dynamics of single-track vehicles. As mentioned in Section 1, Webots has a simple built-in "twoWheeler" vehicle that users can add into their simulation to *visually* replicate a scooter or motorcycle. However, its physics lack any resemblance to the complex dynamics involved in the balancing and steering of a bicycle or PTW. WeBikes fills this gap, making dynamic simulation of bicycle and PTW dynamics quick and easy. The add-on consists of a Webots PROTO, along with a rudimentary virtual rider that balances and steers the vehicle by applying torque at the handlebars. The combination of the configurable vehicle itself with a basic controller means that users can easily study both open and closed loop dynamics of a bicycle or PTW. Figure 1 shows the WeBikes vehicle and its associated adjustable parameters. In addition to dimensional and inertial properties, users also can choose a visual representation of either a motorcycle or bicycle. Users can enable or disable front and rear wheel suspension depending on the embodiment of the vehicle they wish to simulate.



```
DEF PTW WeBikes
  bicycle_shape TRUE
  motorcycle_shape FALSE
  BikeRotation 1 0 0 0.01
  BikePosition 0 0 0
  mrf 85
  hrf 0.9
  xrf 0.3
  b 1.02
  c 0.08
  lambda 1.25
  rfw 0.35
  rrw 0.3
  mfw 3
  mrw 3
  mff 4
  tireRadiusF 0.01
  tireRadiusR 0.01
  Xff 0.9
  hff 0.7
  use_suspension FALSE
  springRate_front 1e+04
  damping_front 1e+03
  suspension_travel_front 0.3
  measured_sag_front 0.01
  jounce_front 0
  swingarm_pivot 0.51 0 0.45
  wheelRate_rear 2e+04
  damping_rear 2e+03
  suspension_travel_rear 0.3
  measured_sag_rear 0.105
  jounce_rear 0
  Addon
  controller "1_openloop_validation
  name "Bicycle"
  customized_inertiaMatrix FALSE
  useowninertial
```

(a) (b) (c)

Figure 1: (a) Adjustable parameters for the WeBikes PROTO; (b) Example bicycle embodiment; (c) Example motorcycle embodiment

Basic parameters of the vehicle, such as the mass and Center of Gravity (C.G.) location of the front (steered) and rear (ridden) frames, are adjustable, with parameters corresponding to the distances and masses shown in Figure 2b. The vehicle's wheels are modeled with a torus shape, featuring an adjustable major and minor radius for each. By default, the front and rear frames of the vehicle are modeled as small spheres approximating point masses for the purposes of physics simulation, but the PROTO also allows users to override this default behavior and specify their own inertia tensors for both the front and rear frames. This was accomplished by writing the WeBikes PROTO as a "procedural PROTO," with embedded JavaScript code that reconfigures and rebuilds the vehicle based on user inputs in the Webots Graphical User Interface (GUI). For example, to set custom inertia tensors for one of the vehicle's rigid bodies, a user changes the parameter "customized_inertiaMatrix" in Figure 1a to "TRUE," and then provides the six unique values of the tensor. The nominal representation of the front and rear frames as "point masses" allows a user to create a relatively realistic representation of a real vehicle quickly using basic properties that are easily measured for any bicycle or PTW. For vehicles where more detailed inertial information is available, WeBikes allows a user to improve the model fidelity if necessary.

The visual appearance of the vehicle is currently configured with two options: as seen in Figure 1a, a user can change the option "bicycle_shape" to TRUE to represent the vehicle visually as a bicycle (Figure 1b). The dimensions of the bicycle frame are responsive and automatically scale to the dimensions of the vehicle selected by the user. Similarly, if the user selects "motorcycle_shape," a generic motorcycle visual embodiment is generated based on the user's selected dimensional properties (Figure 1c). These visual representations allow users to create environments where other vehicles or infrastructure systems can sense and react to a WeBikes vehicle using visual sensors like cameras or LIDAR sensors.

The WeBikes vehicle can also easily be configured with suspension, both front and rear. With the suspension enabled, the vehicle's configuration of joints and rigid bodies is built as shown in Figure 2a. In the front, it includes the option for a prismatic suspension functionally equivalent to the telescopic fork configuration common on bicycles and PTWs, shown by joint P1 in Figure 2a. This suspension has adjustable travel, spring and damping rates, along with an adjustable preload configurable by setting the "measured_sag_front" variable, which is easily measured on real vehicles. The rear suspension uses a torsional spring and damper on a massless rear swinging arm connecting the rear frame to the rear wheel. Users can specify linear spring and damping rates at the wheel, which is how they are measured on a rear vehicle, and the WeBikes PROTO converts these to effective torsional spring and damping rates at the swingarm pivot, Joint R2 in Figure 2a.

Like the front, suspension preload in the rear is adjustable by setting the "measured_sag_rear" parameter.

The WeBikes PROTO is additionally equipped with a Webots "motor" element at the steering head that can be used to steer and balance the vehicle by applying torque about the steering axis, labeled $\hat{e}_\lambda$ in Figure 2b (See Section 2.2.2), along with a motor at the rear wheel joint (R1) to allow the vehicle's controller in Webots to propel it forward. Finally, the WeBikes PROTO includes a Webots standard "AddOn" slot attached to the rear frame, which allows users to add whatever they like to the vehicle, including but not limited to sensors and/or more complex visual or physical models of riders.

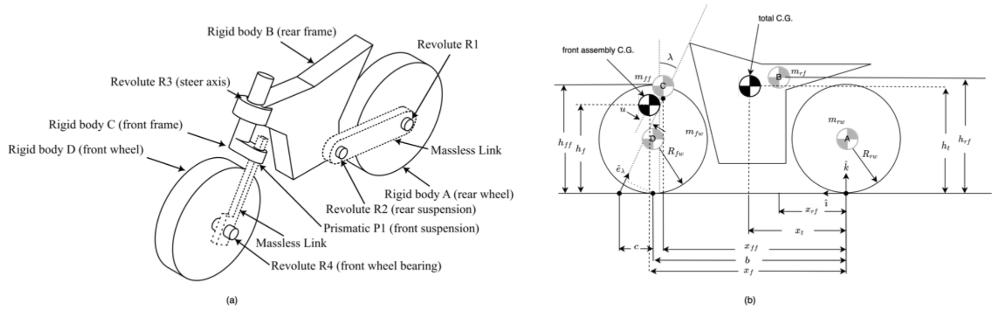

Rigid body B (rear frame)
Revolute R1
Revolute R3 (steer axis)
Rigid body C (front frame)
Rigid body D (front wheel)
Massless Link
Rigid body A (rear wheel)
Revolute R2 (rear suspension)
Prismatic P1 (front suspension)
Massless Link
Revolute R4 (front wheel bearing)
(a)
total C.G.
front assembly C.G.
(b)

Figure 2: (a) Joint and rigid body configuration for the WeBikes add-on; (b) Key user-adjustable mass and dimensional parameters of the WeBikes add-on

2.2. Comparison of WeBikes with a linear models of single-track vehicle dynamics with and without a rider

To understand how the WeBikes vehicle PROTO compares with mature models of PTW dynamics from the literature, we investigated how the WeBikes vehicle responds to initial conditions for constant-speed driving without any active input from a rider. We also investigated how adding a rider model based on linear control of the vehicle would impact these comparisons. While it would be unrealistic to expect the WeBikes vehicle to show behavior identical to a linear model of single-track vehicle dynamics, since Webots is a nonlinear simulation and its physics include more complex dynamics than a linear model is capable of, we were especially interested in determining whether the Webots physics engine was capable of accurately capturing the self-stabilization of a single-track vehicle built using the WeBikes PROTO. Additionally, we were interested in whether a linear model of its dynamics would be sufficient for designing a "virtual rider" controller capable of balancing and steering the vehicle. These two tests are described in more detail in the following subsections.

2.2.1. Representative low-order vehicle dynamic model

The vehicle model chosen as a benchmark for the WeBikes PROTO is based on the 4th order linear differential equation model summarized by Meijaard et al. In [15]. This model was chosen because it captures the self-stabilization behavior of single-track vehicles such as bicycles and PTWs by describing the energetic independence of the vehicle's front (steered) and rear (ridden) frames, and how the two interact with the vehicle's wheels. It was originally intended to describe the roll and steer dynamics of a bicycle with knife-edged tires that have no significant lateral or longitudinal slip. The model represents the dynamics of the four bodies A, B, C, and D as shown in Figure 2b when the vehicle is operating at or close to the "straight running" condition. For the comparisons in this paper, the model was re-derived using LaGrange's method to conform to the vehicle coordinate system as shown in Figure 2b. Additionally, bodies A, B, C, and D were assumed to be point masses for simplicity. Basic dimensional and inertial parameters for the model are presented in Table 1, along with parameter values for both the bicycle described in Meijaard et al. [15] and a small dual-purpose motorcycle. In Table 1, "M.O.I." stands for "moment of inertia" and "C.G." stands for Center of Gravity. Also note that this model assumes that the "rider" is part of the rear frame (body B) and is rigidly attached to the vehicle, so the mass and mass center location of the rear frame listed in Table 1 are inclusive of the passive mass of a rider. The WeBikes PROTO allows the user to attach a non-rigid (even active) multi-joint rider to the rear frame, but for the simulations of this paper, the rider remained passive and rigidly attached.

Table 1. Basic Vehicle Parameters

Symbol	Description	Value (bicycle)	Value (motorcycle)	Units
m_{ff}	Front frame mass	4	10	kg
m_{rf}	Rear frame mass	85	158	kg
m_{fw}	Front wheel mass	3	10	kg
m_{rw}	Rear wheel mass	3	13	kg
J_{yyf}	Front wheel spin M.O.I.	0.368	0.798	kg-m^2
J_{yyr}	Rear wheel spin M.O.I.	0.27	0.833	kg-m^2
x_{rf}	x-distance to rear frame C.G.	0.3	0.689	m
x_{ff}	x-distance to front frame C.G.	0.9	1.25	m
h_{rf}	Height of rear frame C.G.	0.9	0.519	m
h_{ff}	Height of front frame C.G.	0.7	0.735	m
R_{rw}	Radius of rear wheel	0.3	0.330	m
R_{fw}	Radius of front wheel	0.35	0.356	m
c	Trail (see Figure 1)	0.08	0.115	m
b	Wheelbase (see Figure 1)	1.02	1.45	m
λ	Rake angle (see Figure 1)	1.25	1.10	rad

In addition to the 15 basic model parameters listed and described in Table 1, several derived quantities are defined in Meijaard et al. [15] to allow the model's equations of motion to be written compactly. The expressions for these derived quantities, along with descriptions of each, are summarized in Table 2, assuming that the front and rear frames can be approximated as point masses. Additionally, the model assumes that only the spin ($\hat{\jmath}$) moment of inertia of the front and rear wheels is significant. As before, the model has been modified slightly from the form presented in [15] to represent the change from SAE to ISO vehicle-fixed coordinates, and to simplify the equations in representing the front and rear frames as point-masses. In Table 2, M.O.I stands for "moment of inertia," and P.O.I stands for "product of inertia."

Table 2. Derived Vehicle Parameters

symbol	Description	Expression
m_t	Total mass	$m_{ff} + m_{rf} + m_{fw} + m_{rw}$
m_f	Front assembly mass	$m_{fw} + m_{ff}$
m_r	Rear assembly mass	$m_{rw} + mrf$
x_t	x-distance to total C.G.	$(m_{fw}b + m_{rf}x_{rf} + m_{ff}x_{ff})/m_t$
x_f	x-distance to front assembly C.G.	$(m_{fw}b + m_{ff}x_f f)/m_f$
h_t	Height of total C.G.	$(m_{rw}R_r w + m_{fw}R_{fw} + m_{rf}h_{rf} + m_{ff}h_{ff})/m_t$
h_f	Height of front assembly C.G.	$(m_{fw}R_{fw} + m_{ff}h_{ff})/m_f$
u	Steer axis lever arm	$hf sin(\lambda) - (b + c - xf)cos(\lambda)$
T_{xx}	Total x M.O.I.	$m_{rf}h_{rf}^2 + m_{ff}h_{ff}^2 + m_{fw}{R_{fw}}^2 + m_{rw}R_{rw}^2$
F_{xx}	Front assembly x M.O.I.	$m_{ff}(h_{ff} - h_f)^2 + m_{fw}(R_{fw} - h_f)^2$
T_{zz}	Total z M.O.I.	$m_{rf}x_{rf}^2 + m_{ff}{x_{ff}}^2 + m_{fw}b^2$
F_{zz}	Front assembly z M.O.I.	$m_{ff}(x_{ff} - x_f)^2 + m_{fw}(b - x_f)^2$
T_{xz}	Total xz P.O.I.	$-m_{rf}x_{rf}h_{rf} - m_{ff}x_{ff}h_{ff} - m_{fw}bR_{fw}$
F_{xz}	Front assembly xz P.O.I.	$-m_{ff}(x_{ff} - x_f)(h_{ff} - h_f) - m_{fw}(b - x_f)(R_{fw} - h_f)$
$F_{\lambda\lambda}$	Front assembly steer M.O.I.	$m_f u^2 + F_{xx}sin(\lambda)^2 - 2F_{xz}sin(\lambda)\,cos(\lambda) + F_{zz}cos(\lambda)^2$
$F_{\lambda x}$	Front assembly projected P.O.I.	$-m_f u h_f - F_{xx}sin(\lambda) + F_{xz}cos(\lambda)$
$F_{\lambda z}$	Front assembly projected P.O.I.	$m_f u x_f - F_{xz}sin(\lambda) + F_{zz}\,cos(\lambda)$
f	Mechanical trail to wheelbase ratio	$c\,cos(\lambda)/b$
S_f	Front wheel specific angular momentum	J_{yyf}/R_{fw}
S_r	Rear wheel specific angular momentum	J_{yyr}/R_{rw}
S_t	Total specific angular momentum	$S_f + S_r$
S_u	Common static moment term	$m_f u + f m_t x_t$

The fourth-order, linear vehicle dynamic model can be written in Mass-Spring-Damper (MDK) form as shown in Equation 1:

$$M\ddot{\vec{q}} + D\dot{\vec{q}} + K\vec{q} = \overrightarrow{u_0}, \qquad (1)$$

where the generalized coordinate vector $\vec{q}$ is given by $\vec{q} = [\phi\ \delta\ ]^T$, with ϕ representing roll angle, or the rear frame (body B) rotation about the vehicle-fixed $\hat{\imath}$-axis, and δ representing steer angle, or the front frame (body C) rotation about the $\widehat{e_\lambda}$ axis. The input vector $\overrightarrow{u_0} = [T_\phi\ T_\delta\ ]^T$ consists of rider input torque about the roll axis and rider input or external disturbance torque about the steer axis. The terms in the M, D, and K matrices are given in Equation 2.

$$M = \begin{bmatrix} T_{xx} & F_{\lambda x} + fT_{xz} \\ F_{\lambda x} + fT_{xz} & F_{\lambda\lambda} + 2fF_{\lambda z} + f^2 T_{zz} \end{bmatrix},$$

$$D = \begin{bmatrix} 0 & -U\big(fS_t + S_f\cos(\lambda) - T_{xz}f/c + fm_t h_t\big) \\ U\big(fS_t + S_f\cos(\lambda)\big) & U\big(F_{\lambda z}\cos(\lambda)/b + f(S_u + T_{zz}f/c)\big) \end{bmatrix},$$

$$K = \begin{bmatrix} -gm_t h_t & gS_u - U^2\big((S_t - m_t h_t)f/c\big) \\ gS_u & -g\,S_u\sin(\lambda) + U^2\left(\big(S_u + S_f\sin(\lambda)\big)f/c\right) \end{bmatrix}, \quad (2)$$

with constituent terms as defined in Tables 1 and 2, along with the gravitational constant $g = 9.81 m/s^2$ and U representing the vehicle's (assumed constant) forward speed in m/s, the vehicle model is complete. This model can be converted to state space form for numerical simulation and for controller design by solving for $\ddot{\vec{q}}$, and defining a state vector that allows the system to be written in state space form as shown in Equation 3.

$$\frac{d\vec{x}}{dt} = A\vec{x} + B\overrightarrow{u_1}. \quad (3)$$

In Equation 3, the input vector $\overrightarrow{u_1} = T_\delta$, indicating that the model uses rider steer torque as its sole input. While nominally the model presented in Equation 2 is 4th order, the model is augmented with two auxiliary states in order allow the rider model developed in Section 2.2.2 to control lateral position and yaw angle. Therefore, the state vector was chosen as shown in Equation 4,

$$\vec{x} = \left[\phi\ \delta\ \dot{\phi}\ \dot{\delta}\ \psi\ y\right]^T \quad , \quad (4)$$

where the time derivative of ψ, or the vehicle's yaw rate, is found using the no-tire-slip condition applied to the vehicle's steering behavior, and the vehicle's lateral position (e.g. in a lane) is found by integrating lateral velocity assuming small yaw angles. The state space matrices for the resulting model are given in Equation 5.

$$A = \begin{bmatrix} \multicolumn{2}{c}{0_{2x2}} & \multicolumn{2}{c}{I_{2x2}} & \multicolumn{2}{c}{0_{2x2}} \\ \multicolumn{2}{c}{-M^{-1}K} & \multicolumn{2}{c}{-M^{-1}D} & \multicolumn{2}{c}{0_{2x2}} \\ 0 & \frac{U\cos\lambda}{b} & 0 & \frac{c\cos\lambda}{b} & \multicolumn{2}{c}{0_{1x2}} \\ 0 & 0 & 0 & 0 & U & 0 \end{bmatrix}, B = \begin{bmatrix} 0_{2x1} \\ M^{-1}[0 \quad 1]^T \\ 0_{2x1} \end{bmatrix} \quad (5)$$

The model form given by Equations 3-5 can be easily integrated numerically to produce model predictions for the linear model and can also be used for the virtual rider design outlined in Section 2.2.2.

2.2.2. Controlling the WeBikes Powered Two-Wheeler using a representative rider model

The WeBikes PROTO's behavior needs to be governed in the Webots environment using a script called a "controller." This controller influences any powered joints in the PROTO, including the rear drive motor, the motor attached to the steering column of the PTW, along with any motors included in the "AddOn" slots (See Section 2.1) attached to the front and/or rear frames. At the very least, to study a PTW's open-loop (hands-free) behavior, the drive motor must propel the vehicle forward. The "controller" script we wrote in Python for the WeBikes package implements a basic rider model that stabilizes a PTW and allows the user to specify a goal lateral lane position at a particular forward speed. More complex rider models are certainly implementable by the user, and this is one of the key advantages of using WeBikes over a commercial software: the rider model is infinitely customizable.

For the initial simulations described in Section 2.2.3, and the case study outlined in section 2.3, the rider model is a Linear Quadratic Regulator (LQR) synthesized using Python's "control" library. The LQR is a simple, mature, optimal linear full state feedback control law. In the case of the WeBikes vehicle, this controller uses the state vector $\vec{x}$ from Equation 4, and the state space model in equations 3-5 for synthesis. It provides a torque to the handlebar-mounted motor that is computed via Equation 6 where $\overrightarrow{x_d}$ is a vector of goal values for each state.

$$u_1 = K_{lqr}\vec{e} = K_{lqr}(\overrightarrow{x_d} - \vec{x}). \qquad (6)$$

The matrix of controller gains K_{lqr} is found using the Python control library's "lqr" command, which finds a gain vector K_{lqr} that minimizes a quadratic objective function J which penalizes both state error and control input effort when the system is under closed loop control. J is shown in Equation 7.

$$J = \int_0^\infty \left(\overrightarrow{x^\intercal} Q\vec{x} + \overrightarrow{u^\intercal} R\vec{u}\right) dt\,. \qquad (7)$$

In Equation 7, Q is a square matrix penalizing state error, and R is a scalar penalizing control effort. For all simulations in this paper, Q was a 6x6 identity matrix, and R was 0.1, although these values can be tuned to change how "relaxed" or "aggressive" the controller is. Once a matrix of gains K_{lqr} is computed, the controller can perform simple tasks such as lane changes and/or lane keeping at constant forward speeds.

2.2.3. Comparison Simulations

In developing WeBikes, it was important to understand whether the configuration of rigid bodies in Figure 2a, combined with the Webots physics engine, could capture the self-stabilization behavior of bicycles and PTWs, which is one of their most unique (and challenging) characteristics. PTWs are not stable at *all* speeds, but they are able to keep themselves upright and allow for "hands-free" riding for a range of speeds [13][15]. To assess how Webots compares with the predictions of the model developed in Section 2.2.1, we ran two simulations of open loop vehicle behavior, one for each of the vehicle configurations summarized in Table 1. For each, we compared both roll (ϕ) and steer (δ) angles between the linear model and Webots. Both simulations were free responses (representing hands-free riding) with a small initial roll angle to ensure that the comparison between the linear model, which was designed for small deviations from "straight running," and Webots, was fair, since the linear model assumes the vehicle is near the "straight running" condition. The bicycle configuration from Table 1 was simulated with a forward speed U=4.35m/s, chosen due to the linear model's prediction (via the eigenvalues of the A matrix in Equation 5) that this should be one of the vehicle's "most self-stable" speeds. The motorcycle was simulated with a forward speed U=5.96m/s, also chosen using the eigenvalues of the open-loop vehicle model summarized in Equation 5. Each simulation ran for 12 seconds, sufficient time for the linear model to reach steady state. For all simulations, the Webots world was set up with a coefficient of friction between tire and ground of 1.0, and a basic simulation timestep of 1 millisecond. Note that while the Webots simulations in this paper were all performed using a simple "coulomb friction" model for tire forces, which ignores lateral and longitudinal tire slip, more advanced tire models are possible to implement in Webots by modifying the simulation world's friction parameters, including its "force-dependent slip" parameter. The linear model used for comparison with Webots was simulated using Runge-Kutta integration via the "control" library in Python, also using a timestep of 1 millisecond.

The open-loop simulations described above were designed to answer the question of whether Webots is a viable option for the simulation of bicycle and PTW dynamics. However, road safety-adjacent simulations of PTW and bicycle dynamics will often also require the simulation of how a *rider* interacts with the vehicle to steer it and keep it stable outside of its self-stable range of forward speeds. To this end, we equipped the WeBikes PROTO with the rudimentary lane-keeping controller described in Section 2.2.2. Then, to answer the question of whether this "rider" model (designed using linear control theory) could stabilize the nonlinear WeBikes vehicle, and to determine how well the behavior of this closed loop system in Webots could be predicted by linear control theory, we ran 2 simulations of closed-loop behavior using the WeBikes vehicle in Webots. Each simulation corresponded with one of the vehicle configurations described in Table 1. These simulations compared roll angle, steer angle, and vehicle lane position between the *closed loop* linear model (inclusive of the rider) and Webots while each vehicle performed a lane change at one representative speed. Because the simulations represented lane change maneuvers, both were step responses in desired lane position. Each was performed with zero initial roll and steer. The bicycle's lane change was performed at a forward speed U=4m/s, and the motorcycle's lane change was performed at a forward speed of U=15.57m/s. The simulations both ran for 12 seconds, allowing plenty of time for the closed loop systems to reach steady state. The step in desired lane position was set to 4 meters for both vehicles, as shown in Figure 3.

2.3. Pavement Transition Case Study

2.3.1. Simulation setup

To assess WeBikes's utility in a road safety context, we conducted a small case study where a motorcycle changed lanes as described in Section 2.2.3, but on a road with lanes that vary by 3" in height. This was intended to simulate a lane change on a road where one of the two lanes of travel has either been milled or repaved. Often, roads with uneven lanes are marked with signs for motorcycles to "use caution" because of the abrupt change in lane height. Hypothesizing that perhaps a more gradual change in lane height would result in a safer transition between lanes, we ran blocks of simulations with both an abrupt transition and a "Safety Edge" style, 30° beveled transition between the higher and lower lane. A "Safety edge" is an angled road pavement treatment for vertical road edges. Its purpose is to make returning to paved road from non-paved surfaces safer, and its use is supported by the literature. For example, Lyon, Persaud, and Donnell [11] found that the Safety edge was effective in improving two-lane rural road safety for cars. However, studies that assess the effects of the "safety edge" on motorcycle crashes specifically are absent from the literature, as are studies investigating whether such an edge may make lane changes on uneven lanes safer for motorcycles. Because their dynamics are so different from cars, studying the behavior of motorcycles traversing uneven pavement edges is valuable. If adding angled transitions between uneven lanes could improve motorcycle safety, it could expand the Safety edge's capability and justify its use on roadways between lanes of different heights. Webots with the WeBikes add-on could be a great fit for this type of study, since creating custom road geometry is relatively easy using the Webots interface.

To analyze the effects of implementing an angled transition between two uneven paved lanes on an urban road for motorcycles, we performed two sets of 32 simulations. In each simulation, the motorcycle described in Table 1 performed a 4-meter-wide lane change. The first set of simulations was performed at a forward velocity of 10m/s (22.3mph), and the second set was performed at 15.57 m/s (35mph). During each lane change, the vehicle encountered a 3-inch shift in pavement height during the lane change maneuver. During each simulation, this shift occurred at varying lateral (y-direction) offsets "d" of 0.25m to 4m from the beginning of the lane change in increments of 0.25m. Half of the simulations at each speed were performed on a road featuring an abrupt edge of 90° between the two lanes, and the other half used a beveled edge with an angle of 30° (similar to a Safety edge) between the two lanes. The simulation setup is shown in Figure 3.

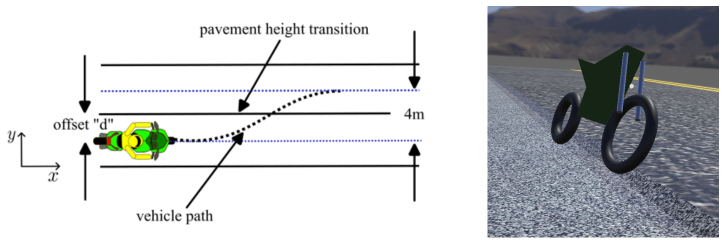


Figure 3: Case Study Simulation Setup

By comparing maximum rider steer torque, steer angle, and vehicle roll angles between trials, we were able to assess the effects of the beveled pavement edge on the rider's corrective steering effort and the motorcycle's motion when crossing the transition in pavement height. We hypothesized that some of these experiments might result in crashes, and that crashes were more likely to occur in simulations with a small or large lateral offset of the uneven edge. This is because at the beginning and end of a lane change, the motorcycle has a small "angle of attack" with respect to the pavement edge. Motorcycle riding manuals such as [9] instruct riders to cross uneven features with a large angle of attack to minimize the disturbance torque that the feature exerts on the steering system.

2.3.2. Limitations of the case study design

It's important to note that this study's design is too small to make sweeping, generalizable claims about highway design. A more extensive investigation of how Webots's simulation fidelity (especially with regards to tire friction and contact force simulation) impacts these results is necessary to make strong recommendations, and more variation in the simulation conditions (e.g. vehicle parameters, rider characteristics, and pavement edge characteristics) would also be required to improve the power of the simulation's results. However, this set of simulations does offer insights about how rider effort varies with the geometry of a pavement height, and it is able to show how WeBikes could be used for road safety simulation studies. In a more exhaustive study, rider corrective effort could be used as a surrogate metric for safety to help guide policy for pavement geometry across various vehicle configurations, pavement geometries, and rider models.

3. Results and Analysis

3.1. Comparing WeBikes to canonical linear models of single-track vehicle dynamics

Figures 3a and 3b show the results of the simulations described in Section 2.2.1. They compare roll and steer angles between the Webots simulation employing the WeBikes vehicle and the linear model for the bicycle (Figure 2a) and for the motorcycle (Figure 2b). As figures 4a and 4b show, the general trends in roll and steer match between Webots and the linear model, with more high-frequency, lightly damped oscillation in the more complex, nonlinear WeBikes vehicle. The WeBikes vehicle is self-stable as predicted by the linear model for both vehicle configurations, which shows that the physics engine in Webots *can* capture the critical self-stabilization behavior of bicycles and PTWs. As Figures 4c and 4d show, the roll, steer, and lane position match well between Webots and the linear model for both bicycle and motorcycle configurations under the direction of the virtual rider during a lane change.

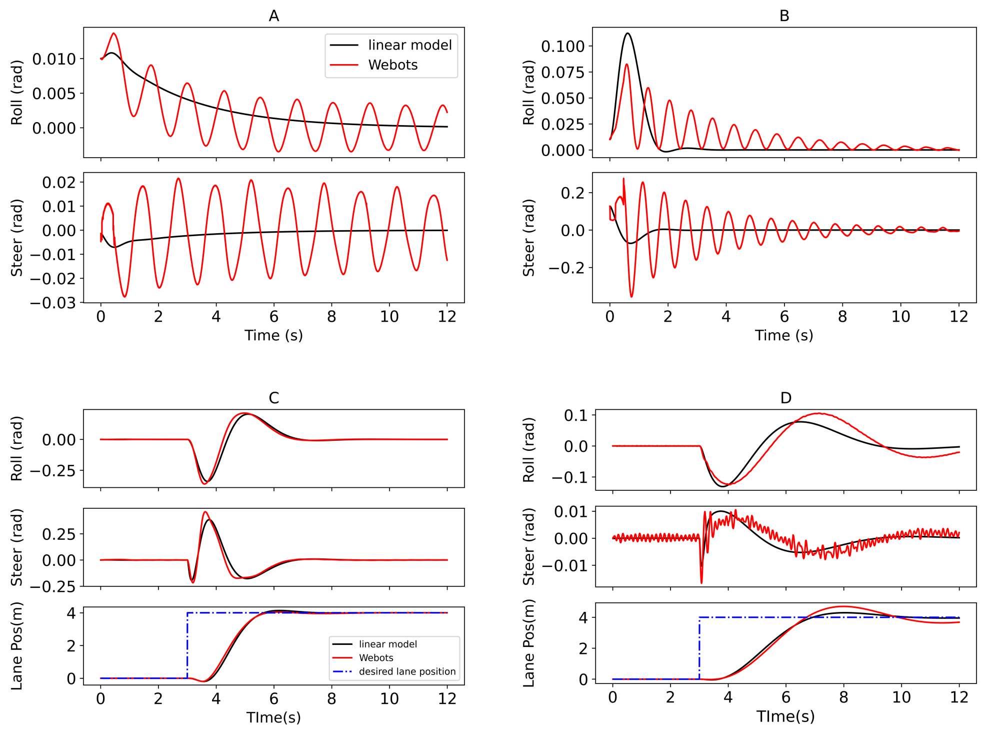


Figure 4: (a,b) open loop and closed loop (c,d) comparisons of Webots simulations vs. linear model for bicycle (a,c) and motorcycle (b,d).

3.2. Pavement Transition Case Study

Figures 5 and 6 show results from the case study described in section 2.3 for speeds of 10 m/s and 15.7 m/s, respectively. Figures 5a and 6a show whether the vehicle successfully navigated the lane change for a particular simulation case, with “1” indicating success and “0” indicating a crash. Figures 5b-5d and Figures 6b-6d show, for each simulation, maximum absolute rider steer torque, vehicle steer angle, and vehicle roll angle respectively during the lane change.

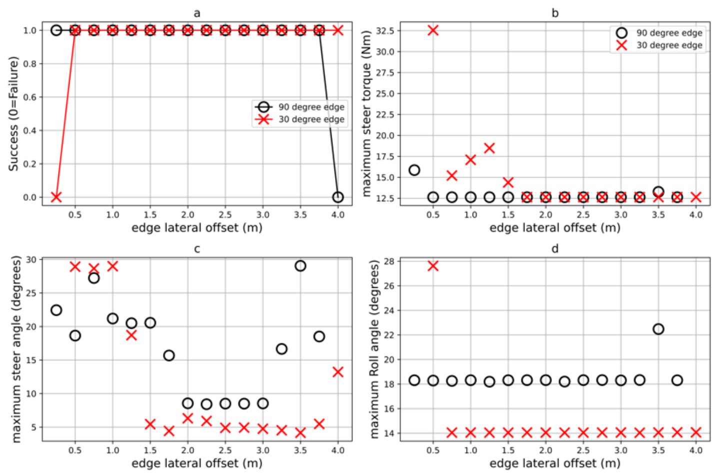


Figure 5: (a) Success of all 10 m/s simulations; (b-d) Comparisons of vehicle motion for lane changes with varying lateral offsets for pavement edge angles of 90 and 30 degrees

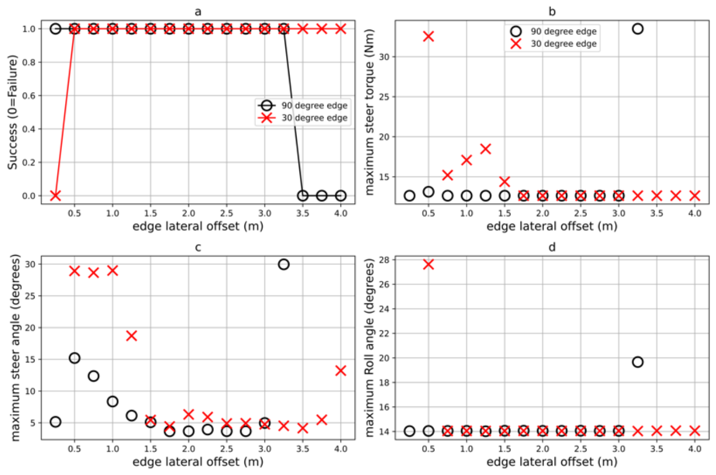


Figure 6: (a) Success of all 15.7 m/s simulations; (b-d) Comparisons of vehicle motion for lane changes with varying lateral offsets for pavement edge angles of 90 and 30 degrees

As hypothesized in Section 2.3, a successful lane change was more likely to occur when the uneven pavement edge was present near the middle of the lane change maneuver, or when the vehicle's "angle of attack" was large. The sharp 90° pavement edge seemed to be more difficult for the rider-vehicle system to navigate if it was encountered near the end of the lane change, while the beveled 30° edge was more difficult to navigate if it was at the beginning of the lane change. This may be because at the beginning of the maneuver, rider torque and steer angle are high even though the angle of attack is not, which means that the front *wheel* has a higher angle of attack at the beginning of a lane change, allowing it to ride over the sharp edge with less difficulty than at the end of the maneuver, when steering effort and angle are both low. Conversely, the beveled edge imparts a disturbance on the front wheel for a longer duration, by definition, which could explain why it poses a problem near the beginning of a lane change when tire friction demands are higher. A more exhaustive study would be required to make a definitive conclusion.

Surprisingly, both figures 5 and 6 show that while steer angle and torque are generally similar for the beveled edge and the sharp edge, and that trends in vehicle motion and rider effort are relatively consistent between the two speeds considered, the sharp edge seems to require slightly less maximum corrective torque and steer angle. This may be due to the longer duration of the disturbance imparted by the beveled edge. Does this mean that the sharp edge is "better?" It is difficult to say, especially given the limitations outlined in Section 2.3.2. Anecdotally, videos from each battery of simulations indicate that the sharp pavement edge sometimes caused the vehicle's rear wheel to leave the ground. Although this did not always cause the virtual rider of Section 2.2.2 to crash, it's difficult to interpret how sensitive a human rider might be to this effect, and difficult to know how different virtual riders with varying objective functions J (Equation 7) would react without a more extensive simulation treatment.

In summary, however, the results presented in Figures 5 and 6 align with expectations for simulation results based on mature rider training recommendations [9] and indicate that Webots's physics engine is able to produce results that show fine-grained variation in vehicle dynamics for small changes in road surface geometry and vehicle speed.

4. Discussion

The results presented in Section 3 support the utility of Webots/WeBikes as a tool for simulating bicycle and PTW dynamics, both for studying vehicle and rider behavior and for studying how the vehicle-rider system interacts with a roadway in a road safety context. Figures 4a and 4b show that although Webots captures more complex dynamics than the canonical linear model described in Equation 5, Webots equipped with the WeBikes PROTO can capture both vehicles' self-stabilization behavior. It is not possible to know how well WeBikes matches actual motorcycle or bicycle dynamics without data from instrumented test vehicles, but the mismatch between the linear model and the Webots simulation is manifested mostly in higher frequency, lightly damped oscillations. The slower dynamics of the WeBikes vehicle and the linear model are similar. This type of disagreement is consistent with the fact that Webots captures high-fidelity, nonlinear physics that are not present in the simpler model.

Figures 4c and 4d show that the vehicle model developed in Section 2.2.1 is sufficiently descriptive of the WeBikes vehicle's dynamics that a simple rider model based on optimal linear control theory (Section 2.2.2) can control the WeBikes vehicle predictably for basic lane-keeping and lane change maneuvers. This means that as presented, the WeBikes add-on is suitable for a wide range of simulation studies that deal with both vehicle dynamics and road safety. Note as well that the addition of more sophisticated virtual riders, possibly including more biomechanically correct steering and leaning action, are both possible and welcome additions if a particular research question requires them.

The road safety case study's results, which are summarized in Figures 5 and 6, present a mix of expected and surprising results. While the crashes that occurred for both large and small edge offsets follow expectations, the apparent (although slight) advantage of a sharp pavement transition based on vehicle motion and rider effort do not follow expectations based on the recommendations in [13] for cars. Intuitively, it seems that an angled transition between disparate pavement heights would be favorable, especially given the success of the Safety edge in reducing road re-entry crashes for four-wheeled vehicles. It clear based on the simulation results in Figures

5 and 6 that a sharp edge seems to disrupt the lane change slightly less than the beveled edge. To be sure, a more exhaustive and detailed investigation of this phenomenon is warranted to see how well this result generalizes for different road friction, different vehicle configurations, and different riders. Many of those parameters are easy to tune with the infrastructure presented in this paper. For example, a rider with "stronger" steering characteristics is easy to generate by lowering the value of R in Equation 7. However, in general, the results presented in Figure 4 could be attributed to the fact that because a motorcycle's stability is inextricably linked to the torque applied to and angle of the front frame, the beveled edge influences the rider's corrective torque for a longer duration than the sharp edge. This is *especially* true for the shallow angles of attack that occur with both very small and very large offsets of the pavement transition from the start of the lane change maneuver. In any case, the results of the case study do illustrate the ease with which a large batch of simulations can be constructed that vary rider, vehicle, and/or road parameters to answer questions related to road safety for single-track vehicles.

5. Conclusions

The WeBikes add-on for the Webots simulation software has the potential to grow into a viable open-source alternative to commercial PTW simulation software. Our analysis of how the WeBikes vehicle's dynamics compare to a mature linear model of the single-track vehicle dynamics and the rider-vehicle system demonstrates that WeBikes captures essential stabilization behavior and rider-vehicle interactions that are necessary to accurately model two-wheeled vehicle dynamics. Our road safety case study illustrates how WeBikes can be used to answer important questions about road safety quickly, easily, and without the prohibitive costs of a commercial simulation package.

The configurable nature of the WeBikes package, coupled with the integration of the already powerful Webots simulation software, creates new opportunities for research related to road safety. The open-source nature of WeBikes not only ensures that this technology will always be free for end users, but additionally allows for a constant improvement to the software as it is adopted by more research groups. Because the WeBikes add-on is open-source, it is easily extensible by users to include higher-fidelity tire models, more standard sets of vehicle configurations, and more realistic rider models. Additionally, opportunities for innovative applications of the WeBikes package exist in areas like machine learning, with the add-on providing a flexible testbed for data-driven modeling and control approaches for motorcycles, and of course for data-driven highway design or policy studies. For safety research, WeBikes' ability to simulate single-track vehicle dynamics empowers investigators to rigorously evaluate design factors like road geometry and obstacle avoidance in risk-free virtual environments.

By lowering the barriers to entry for the simulation of single-track vehicle dynamics, WeBikes has the potential to accelerate progress across multiple disciplines focused on reducing injuries and fatalities among vulnerable road users like motorcyclists and bicyclists. Due to the unlimited customizability and the robust capabilities of the underlying Webots physics engine and simulation environment, WeBikes is well-positioned to become a widely adopted open-source platform for single-track vehicle research.

References

1. S. Abilkassov et al., "Facilitating Autonomous Vehicle Research and Development Using Robot Simulators on the Example of a KAMAZ NEO Truck," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-8, doi: 10.1109/ITSC45102.2020.9294716.
2. O. Alrazouk, A. Chellali, L. Nehaoua, and H. Arioui, "Vision-based approach for estimating lateral dynamics of Powered Two-Wheeled Vehicles," in 2023 American Control Conference (ACC), San Diego, CA, USA: IEEE, May 2023, pp. 999–1005. doi: 10.23919/ACC55779.2023.10155807.
3. Y. Bello et al., "Two wheels electric vehicle modelling: Parameters sensitivity analysis," in 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France: IEEE, Apr. 2019, pp. 279–284. doi: 10.1109/codit.2019.8820319.
4. P.-M. Damon, H. Dabladji, D. Ichalal, L. Nehaoua, and H. Arioui, "Estimation of lateral motorcycle dynamics and rider action with Luenberger observer," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil: IEEE, Nov. 2016, pp. 2392–2397. doi: 10.1109/ITSC.2016.7795941.
5. P.-M. Damon, D. Ichalal, and H. Arioui, "Steering and Lateral Motorcycle Dynamics Estimation: Validation of Luenberger LPV Observer Approach," IEEE Trans. Intell. Veh., vol. 4, no. 2, pp. 277–286, Jun. 2019, doi: 10.1109/TIV.2019.2904384.
6. M. Fouka, L. Nehaoua, H. Arioui, and S. Mammar, "Interconnected Observers for a Powered Two-Wheeled Vehicles: Both Lateral and Longitudinal Dynamics Estimation," in International Conference on Networking, Sensing and Control (ICNSC 2019), in Proc. of the International Conference on Networking, Sensing and Control (ICNSC 2019). Banff, Canada, May 2019, pp. 163–168. doi: 10.1109/ICNSC.2019.8743290.
7. M. Franchi et al., "Webots.HPC: A Parallel Simulation Pipeline for Autonomous Vehicles," in Practice and Experience in Advanced Research Computing, Boston MA USA: ACM, Jul. 2022, pp. 1–4. doi: 10.1145/3491418.3535133.
8. S. Gowtham, R. Praveen, P.S. Charan, M. Parthiban, N. Seenu and R. K. Chetty, "Simulation of Autonomous Multifunctional Mobile Robot using Machine Vision," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 135-149, doi:10.1109/ICACCS51430.2021.9441946.
9. D. L. Hough, Proficient Motorcycling: The Ultimate Guide to Riding Well. Bow Tie Press, 2000.
10. V. Křivánek, V. Starý, and Y. Bergeon, "Optical Sensor Placement Optimization for Unmanned Ground Vehicles by the Simulation," in 2023 International Conference on Military Technologies (ICMT), Brno, Czech Republic: IEEE, May 2023, pp. 1–7. doi: 10.1109/ICMT58149.2023.10171309.
11. C. Lyon, B. Persaud, and E. Donnell, "Safety Evaluation of the SafetyEdge Treatment for Pavement Edge Drop-Offs on Two-Lane Rural Roads," Transp. Res. Rec., vol. 2672, no. 30, pp. 1–8, Dec. 2018, doi: 10.1177/0361198118758054.
12. X. Ma, C. Mu, X. Wang, and J. Chen, "Projective Geometry Model for Lane Departure Warning System in Webots," in 2019 5th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China: IEEE, Apr. 2019, pp. 689–695. doi: 10.1109/ICCAR.2019.8813404.
13. 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," Proc. R. Soc. Math. Phys. Eng. Sci., vol. 463, no. 2084, pp. 1955–1982, Jun. 2007, doi: 10.1098/rspa.2007.1857.
14. O. Michel, "Cyberbotics Ltd. WebotsTM: Professional Mobile Robot Simulation," Int. J. Adv. Robot. Syst., vol. 1, no. 1, p. 5, Mar. 2004, doi: 10.5772/5618.
15. A. L. Schwab, J. P. Meijaard, and J. M. Papadopoulos, "Benchmark results on the linearized equations of motion of an uncontrolled bicycle," J. Mech. Sci. Technol., vol. 19, no. S1, pp. 292–304, Jan. 2005, doi: 10.1007/BF02916147.

16. L. Yang, L. Tao, W. Hai, L. Rui, P. Jianzhong, and F. Zhibin, “Safety and Comfort Evaluation of High-Speed Cycling on Urban Asphalt Pavement,” Transp. Res. Rec., p. 03611981231208909, Dec. 2023, doi: 10.1177/03611981231208909.
17. “BikeSim Overview.” Accessed: May 29, 2024. [Online]. Available: https://www.carsim.com/products/bikesim/ .
18. Bicycle Deaths,” Injury Facts. Accessed: May 29, 2024. [Online]. Available: https://injuryfacts.nsc.org/home-and-community/safety-topics/bicycle-deaths/
19. “Facts + Statistics: Motorcycle crashes | III.” Accessed: May 29, 2024. [Online]. Available: https://www.iii.org/fact-statistic/facts-statistics-motorcycle-crashes
20. "Webots: Open-source Mobile Robot Simulation Software" Accessed: January 23, 2024 [online]. Available: https://www.cyberbotics.com
21. “WeBikes: an open-source add-on for the Webots robot simulation software” Accessed: November 28, 2024 [online]. Available: https://github.com/Alexanderallenbrown/webikes

AdvancesinTransportationStudies
an international journal