Two-Wheel Self-Balancing of a Four-Wheeled Vehicle

DAVID ARNDT, JAMES E. BOBROW, STEVEN PETERS,
KARL IAGNEMMA, and STEVEN DUBOWSKY

Cars and trucks are susceptible to accidents due to rollover. In the United States in 2005, 21.1% of a total of 54,718 deaths in vehicle crashes were caused by rollover [1]. Significant research has therefore been devoted to detecting and preventing rollover through active control. Numerous approaches attempt to detect or predict wheel liftoff using onboard sensing and a combination of automatic steering and braking to keep the wheels on the ground [2]–[4]. Rather than focusing on how to keep the wheels on the ground, it is also useful to understand how to control a vehicle while two wheels are in the air. This understanding may enable the design of control laws that can safely return the vehicle to the ground after inadvertent tip-up.

Similar to the Acrobat swing-up problem [5], vehicle roll control can be decomposed into a tip-up phase and a balance phase. During tip-up, energy is pumped into the vehicle roll angle, whereas, during balance, the vehicle is stabilized with its center of gravity above the wheels. It is desirable to develop a model that can be used for both phases of the motion, as well as a control law that can stabilize the vehicle in the balance phase. In fact, movie stunt car drivers are known for their ability to drive a car while balanced on two lateral wheels. A Guinness World Record for the most cars simultaneously driven on two lateral wheels was set in July of 2008, and a video of this record-breaking event can be viewed on YouTube [6].

Creating a self-balancing vehicle requires analysis of the vehicle dynamics and must be amenable to embedded control-system implementation. The vehicle balance control problem is complicated by the fact that the equations of motion are nonlinear and have nonholonomic rolling constraints. Related nonlinear control problems include balancing bicycles and motorcycles [7]–[9]. In [7] it is shown that a bicycle model with rear-wheel steering has a nonminimum phase zero whose location varies linearly with velocity. The linearized analysis developed in this article reveals a similar zero. The bicycle and nonholonomic motorcycle models developed in [8]and [9] separate the control problem into tracking control for the motion of the center of gravity (cg) and equilibrium balance control for the roll angle. In this article we focus on stabilization of the roll angle.

Several vehicle models for rollover stability are developed in [10]–[13]. In [10], a nonlinear 14-degree-of-freedom (14DOF) model that includes 6 DOFs for the vehicle body and 2 DOFs for each of the four suspensions and wheels is developed and compared to a simpler 8DOF model that does not include wheel suspension. The 14DOF model captures the dominant system dynamics but is difficult to use for control design since a nondifferentiable tire contact model is used. The roll control system developed for trucks in [11] incorporates two hydraulic servoactuators on an antisway bar mounted to each axle. The actuators improve stability by reducing the roll angle during cornering. In [12], a simple 3DOF model is used to develop a numerical index to indicate impending rollover. As shown in [13], scaled-down remote-controlled (RC) vehicles can be used to analyze rollover stability and test prototype electronic stability control laws.

In this article, we present a control law that uses roll angle and rate feedback to stabilize roll angle. This control law is tested on an RC vehicle modified for microprocessor control. The vehicle model used for the analysis has 11 DOFs and is simulated in the SimMechanics simulation environment of Matlab. We then develop the tire contact model, which is a spring-damper system modified to be continuous and piecewise differentiable. The contact model used in this article is not as sophisticated as the "magic formula" model of [14] but is amenable to control analysis while capturing the dominant tire-to-road force characteristics. Given the nonlinear vehicle and contact models, we linearize the equations about the balance equilibrium to investigate the vehicle stability properties. A root locus analysis then demonstrates that the roll angle of the vehicle can be stabilized with a control law that accounts for the nonminimum-phase zero in the linearized dynamics. Finally, the simulated and experimental results demonstrate the behavior of the nonlinear dynamic system.

EXPERIMENTAL HARDWARE

Figure 1 shows the remote-controlled vehicle used to test the balance control algorithm. The truck, which is manufactured by HPI Racing [15], is modified for microprocessor control. One of the modifications needed is an improved steering

Digital Object Identifier 10.1109/MCS.2010.939941
Date of publication: 16 March 2011

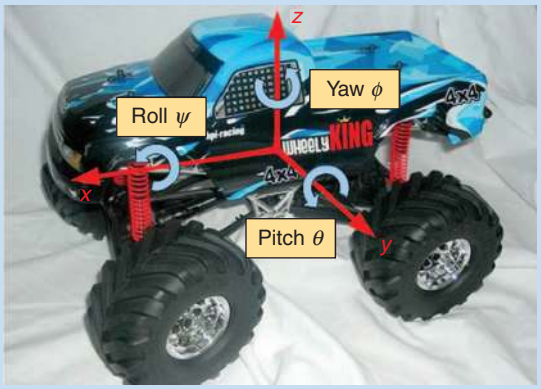

FIGURE 1 Remote-controlled vehicle. The origin of the coordinate frame shown is located at the center of gravity of the vehicle. The order of the Euler-angle rotations for representing the orientation of the vehicle is yaw, pitch, and roll.

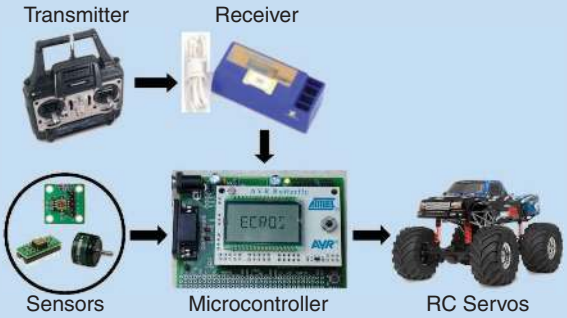

FIGURE 3 The main hardware used for the experiments. The roll angle and roll rate are measured by the microcontroller. Changes in the roll-angle setpoint are transmitted from the user to the microcontroller to achieve steering control for obstacle avoidance. A vehicle velocity command for the drive motor is also transmitted to the microcontroller. The arrows indicate the direction of information flow.

servo system. The original steering servo has backlash due to loose tolerances in the steering linkage and does not provide sufficient torque to hold a desired steering angle. The stock servo is therefore replaced with a Hitec HSR-5995TG titanium-gear high-torque servo as shown in Figure 2 along with a modified linkage to reduce backlash. In addition, spacers are placed under the coil suspension springs to reduce the displacement and increase the spring preload.

The main sensory and electronic components used to implement and test the control law are shown in Figure 3. The RC transmitter sends pulse code modulation (PCM) signals for the throttle and steering to the receiver. The receiver sends the signals to an Atmel AVR Butterfly microcontroller, which uses interrupts to detect the pulses and calculate the pulse times for each signal. The AVR Butterfly receives analog signals from a potentiometer, gyro, and accelerometer and converts them to digital signals. The roll angle is measured using a hinged linkage that rolls along the ground next to the vehicle while connected to the chassis through the potentiometer. The angular rate is measured directly with the gyro. The AVR Butterfly then uses these signals to calculate the control and convert it to a PCM value for use by the steering servo to balance the vehicle.

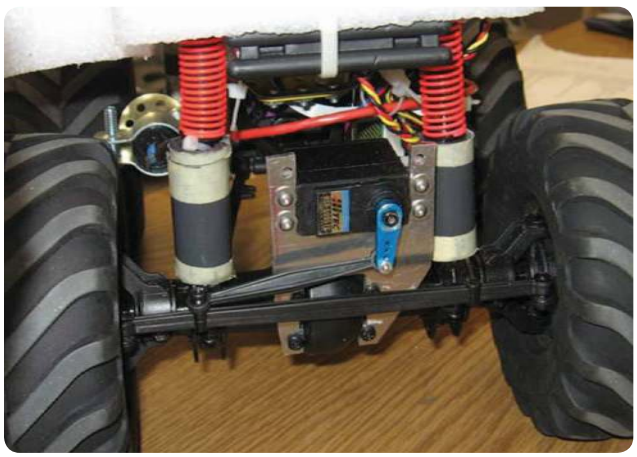

FIGURE 2 Modified high-torque steering servo. The original steering servo has significant compliance and backlash in the drive linkage, which can create instability in the closed-loop system response. A stiffer steering servo mount and a higher performance servo are used to eliminate this problem.

NONLINEAR DYNAMIC VEHICLE MODEL

The vehicle is modeled as a rigid body with three translational and three rotational DOFs located at the vehicle center of mass. In addition, a rotational DOF is used for each wheel, and a rotational DOF is used for the front steering angle, which is assumed to be the same for both front wheels. Since the total number of DOFs of the model is 11, a 22-dimensional state space is required for simulation and stability analysis. Defining the 11 generalized coordinates as $q \in \mathbb{R}^{11}$, the nonlinear equations of motion have the form

$$M(q)\ddot{q} + C(q, \dot{q}) + N(q) = J_1^T(q)F_{c1} + J_2^T(q)F_{c2},$$

where $M(q)$, $C(q, \dot{q})$, $N(q)$ are the mass, Coriolis, and gravity terms, respectively, from Lagrange's equations, and $J_1^T(q)F_{c1} + J_2^T(q)F_{c2}$ is the generalized force from the two wheels in contact with the ground. The Jacobian relationships $J_1^T(q)$, $J_2^T(q)$ define how the contact forces F_{c1}, F_{c2} influence the motion of each generalized coordinate.

The Mathworks SimMechanics toolbox is used to model and simulate the dynamics of the vehicle as well as to compute the Jacobians. Figure 4 shows the top two layers of the SimMechanics model definition. The mass properties used to represent the vehicle body and wheels are provided in Table 1. The experimental vehicle used for this research has a suspension system that adds additional DOFs to the

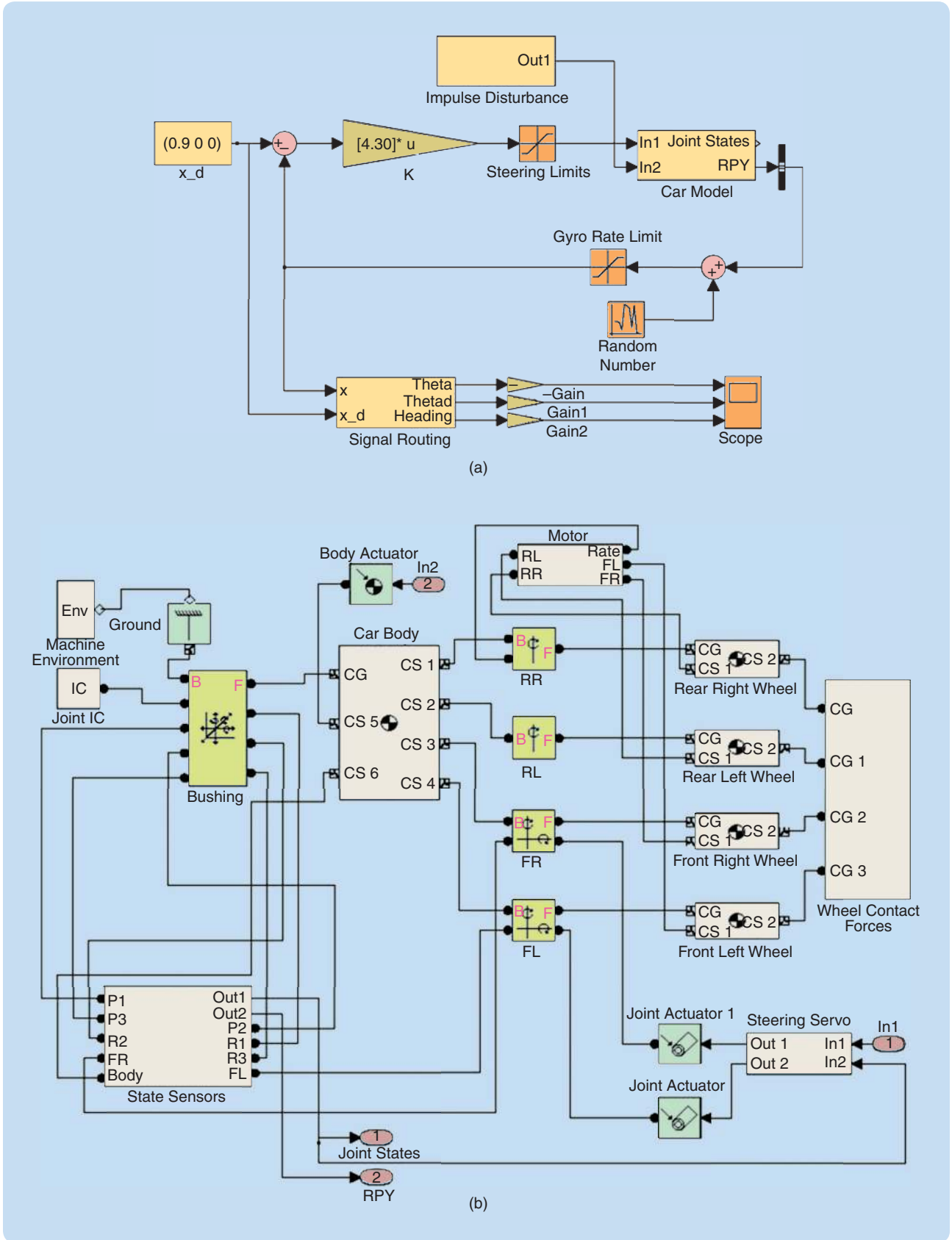

FIGURE 4 Simulink SimMechanics block diagrams. (a) defines the control law, steering saturation, impulsive disturbance, and additive noise and (b) defines the vehicle model.

Creating a self-balancing vehicle requires analysis of the vehicle dynamics and must be amenable to embedded control-system implementation.

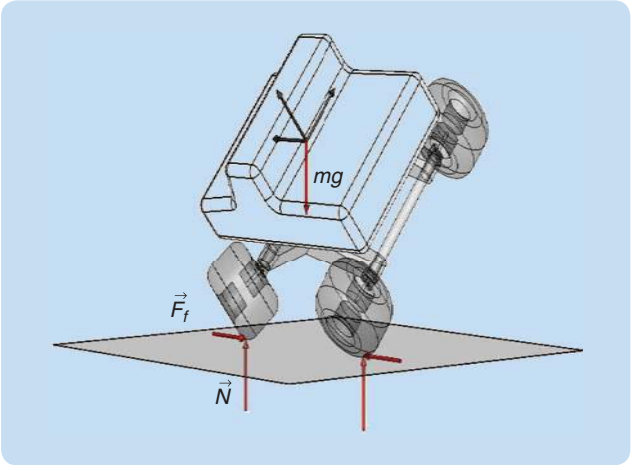

FIGURE 5 Free body diagram of the vehicle near its balance equilibrium point. The friction and normal forces that arise from tire contact are shown along with the force of gravity acting through the vehicle center of mass. The steering angle of the front wheel shown is turned inward, which causes the roll angle to increase as the vehicle moves forward.

TABLE 1 Mass properties used in the simulation.

Variable	Description	Value
grav	Gravity	9.81 m/s^2
mc	Car body mass	1.561 kg
len	Car body length	0.241 m
wid	Car body width	0.208 m
height	Car body height	0.137 m
I_{xx}	Car body moment of inertia about x axis	0.0081 kg-m^2
I_{yy}	Car body moment of inertia about y axis	0.0100 kg-m^2
I_{zz}	Car body moment of inertia about z axis	0.0132 kg-m^2
m_w	Wheel mass	0.195 kg
r	Tire radius	65.0 mm
r_c	Tire corner radius	3.17 mm
w	Wheel width	0.070 m
I_a	Wheel axial moment of inertia	0.0005 kg-m^2
I_t	Wheel transverse moment of inertia	0.0003 kg-m^2
mtot	Total car mass	2.341 kg
k	Tire and suspension spring constant	350.0 N/m
C	Tire and suspension damping	70.0 N/(m/s)
μ	Tire contact friction coefficient	0.75

model. Rather than explicitly modeling the suspension, the tires and suspension are viewed as a series combination of spring-damper systems in the modified spring-damper tire contact model.

Figure 5 shows a free body diagram of the vehicle as it rolls near its tilted balance equilibrium configuration. The forces acting on the vehicle include gravity and the lateral and longitudinal tire contact forces for the two tires in contact with the ground. The equilibrium tilt angle is found from a static analysis to be 52.3°. The RC servo that steers the front wheels shown in Figure 2 is modeled as a proportional-plus-derivative control system. The transfer function of this system has the form

$$G_{\text{steer}}(s) = \frac{\beta(s)}{\beta_d(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}, \tag{1}$$

where β is the steering angle output from the RC servo, and β_d is the desired steering angle input to the servo. The desired steering angle is defined by the control law in the next section. The coefficients of the transfer function (1) are selected so that the step response of this transfer function approximates that of the RC servo, with ω = 20 rad/s and ζ = 0.5. The RC servo has characteristics that are not captured by the transfer function, for example, saturation and Coulomb friction.

CONTACT FORCE MODEL

Simulation of the vehicle model requires tire contact forces to be computed as a function of the system state. Given the orientation and position of the vehicle cg, the position of each wheel's center of gravity is used to calculate the position of the contact point shown in Figure 6 as follows. The tire's center of gravity is located at the point $\vec{P}$, and the point of contact is located at the point $\vec{P}_c$. A local coordinate frame for the wheel is defined with the origin at $\vec{P}$, and the associated rotation matrix is given by $R_w = [\hat{x}_w, \hat{y}_w, \hat{z}_w]$, where $\hat{y}_w$ is a unit vector along the axis of wheel rotation, $\hat{z}_w$ is a unit vector in the direction of the steering axis of rotation, and $\hat{x}_w$ is a unit vector orthogonal to $\hat{y}_w$ and $\hat{z}_w$. Note that the directions of the unit vectors $\hat{x}_w, \hat{y}_w$ depend on the steering angle β. The vector $\hat{n}$ is a unit vector normal to the ground. The vector $\hat{y}_w$ is available from SimMechanics, while the unit vector $\hat{z}_w$ is calculated from $\hat{n}$ and $\hat{y}_w$ as $\vec{z}_w = \hat{n} - (\hat{n}^T\hat{y}_w)\hat{y}_w$, where $\hat{z}_w = \vec{z}_w / \|\vec{z}_w\|$. Finally, $\hat{x}_w = \hat{y}_w \times \hat{z}_w$ completes the vector triad.

Given the local coordinate frame defined above, the tire's contact point is calculated using Figure 6 as

$$\vec{P}_c = \vec{P} + \left(\frac{1}{2}w - r_c\right)\hat{y}_w - (r - r_c)\hat{z}_w - r_c\,\hat{n}, \qquad (2)$$

where w is the wheel width, r is the tire radius, and r_c is the tire corner radius as shown in Figure 6. Differentiating (2), the velocity of the tire's contact point is

$$\vec{V}_c = \dot{\vec{P}} + \left(\frac{1}{2}w - r_c\right)\dot{\hat{y}}_w - (r - r_c)\dot{\hat{z}}_w, \qquad (3)$$

where $\dot{\vec{P}}$ is the wheel velocity, $\dot{\hat{y}}_w = S(\omega)\,\hat{y}_w$, $\dot{\hat{z}}_w = S(\omega)\hat{z}_w$, and $S(\omega) = \dot{R}_w R_w^T$ is the skew-symmetric angular velocity matrix of the wheel in the inertial frame. The distance from the contact point to the ground is calculated as $d = \hat{n}^T\vec{P}_c$, and its time derivative is $\dot{d} = \hat{n}^T\,\vec{V}_c$.

When $d < 0$, the normal force and friction force acting on the tire must be computed. Figure 7 shows the direction of the positive normal force N, which is zero in the configuration shown since the tire is off the ground with $d > 0$. Assuming $d < 0$, the tire is contacting the ground and deforming, resulting in a normal force. The magnitude of the resultant normal force is assumed to be represented by a modified spring and damper system. All tire models used in vehicle simulations must have some stiffness and damping characteristics. Damping is needed to represent energy dissipation in the suspension and tires, while stiffness is needed for the tires to rebound. Because the expected normal force is continuous and differentiable, for $d < 0$ we define the normal force as

$$N = \begin{cases} -kd - c\dot{d}, & \dot{d} > 0 \text{ (restitution)}, \\ -kd, & \dot{d} \le 0 \text{ (compression)}. \end{cases} \qquad (4)$$

Note that damping $c\dot{d}$ is not included during compression. This omission allows the tire to fall and hit the ground with nonzero velocity $\dot{d}$ without the normal force N changing discontinuously from zero when the tire is in the air to the nonzero value $-c\dot{d}$ when ground contact occurs.

The normal force N in (4) is continuous in $(d, \dot{d})$ but not differentiable at the transition between $d \le 0$ and $d > 0$. To create a differentiable model, we further modify the normal force with a smoothing function that satisfies the boundary conditions

$$s(x) = \begin{cases} 0, & x \le 0, \\ 1, & x \ge 1, \end{cases} \qquad (5)$$

with $s(0) = s'(0) = s'(1) = 0$ and $s(1) = 1$. For $0 \le x \le 1$, the cubic polynomial $s(x) = 3x^2 - 2x^3$ is used for smoothing. The smoothed contact force is then defined as

$$N_s = s(N/\Delta)N, \qquad (6)$$

where the constant $\Delta > 0$ defines the transition region for the smoothed contact force N_s.

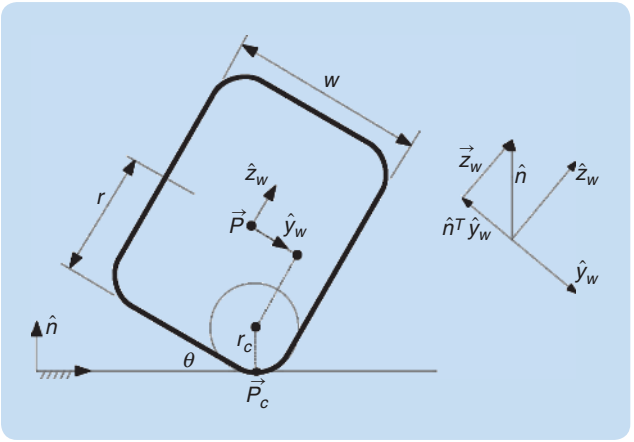

FIGURE 6 Geometry used to determine the location of the contact point given the wheel center of gravity and the axis of rotation. The contact point is located by constructing a local coordinate system using the wheel rotation axis $\hat{y}_W$ and the surface normal $\hat{n}$.

Use of the smoothed contact force (6) enables the numerical simulations to execute quickly and, after adjustment of k, c, and Δ, the simulations are visually indistinguishable from experimental motions of the vehicle. Next, the horizontal component of the contact point velocity $\vec{V}_h$ is calculated to determine the direction of the friction force. From (3), $\vec{V}_c$ is projected to the horizontal plane as $\vec{V}_h = [I - \hat{n}\hat{n}^T]\vec{V}_c$. The friction force acting on the tire opposes the horizontal component of the velocity as

$$\vec{F}_f = -\mu N_s \frac{\vec{V}_h}{\|\vec{V}_h\|}. \qquad (7)$$

Finally, the torques resulting from the normal force and friction are calculated at the wheel as

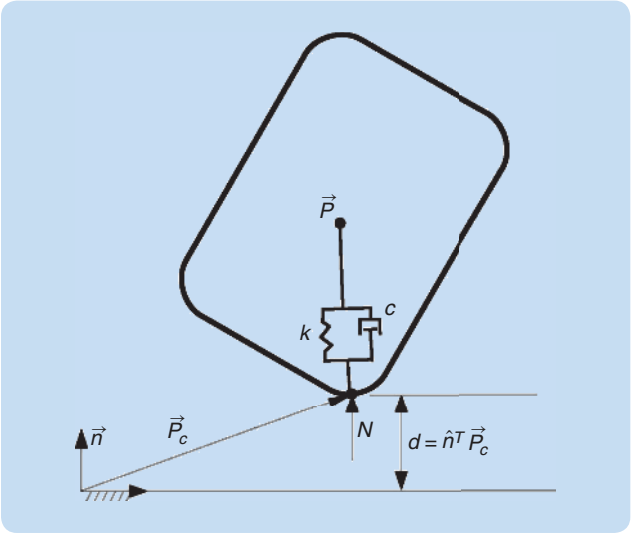

FIGURE 7 Normal force is determined from the contact point and ground-penetration distance. The normal force is obtained from a unilateral spring-damper model.

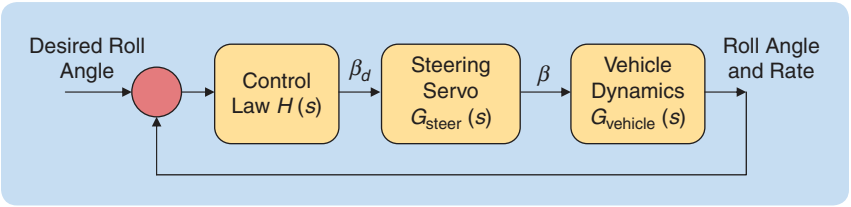

FIGURE 8 Linearized system dynamics used for root locus analysis. A 22-dimensional state-space model of the nonlinear vehicle dynamics is linearized about the equilibrium configuration to obtain $G_{vehicle}(s)$. The control law and steering servo dynamics are added to the vehicle dynamics to obtain a root locus plot for the open-loop transfer function $H(s)G_{steer}(s)G_{vehicle}(s)$.

$$\vec{T}_{\text{Total}} = \vec{T}_{\text{Normal}} + \vec{T}_{\text{Friction}} = \vec{r} \times \vec{N}_s + \vec{r} \times \vec{F}_f,$$

where $\vec{r} = \vec{P}_c - \vec{P}$. Hence, the total force and torque vectors applied to the vehicle at the contact point of each wheel are $\vec{N}_s + \vec{F}_f$ and $\vec{T}_{\text{Total}}$, respectively.

MODEL ANALYSIS

The nonlinear vehicle system dynamics are linearized about the tilted equilibrium to obtain a linear state-space model that contains 22 state variables. These states include the 6 DOFs for the car body and their velocities, the four wheel rotations and velocities, and the steering angle and its velocity. The equilibrium point is found from the geometry and system mass properties to be 52.3°. The system is linearized at this tilt angle with a nominal vehicle speed of 1.0 m/s. The linearization tool available with Simulink is used numerically to compute sensitivity information. This tool uses finite differences to compute the required Jacobian matrices, which is facilitated by the smoothing function (5) used in (6).

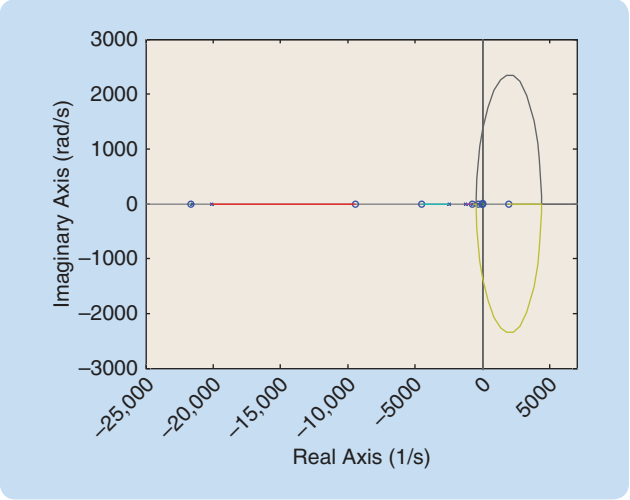

FIGURE 9 Root locus plot of $G_{steer}(s)G_{vehicle}(s)\,H(s)$. The transfer function is from the input steering angle to a linear combination of roll and roll rate. The nonminimum-phase zero located at s = 1920 is due to the effect that increasing the steering angle has on decreasing the roll angle.

Several feedback control laws are considered to provide a steering angle β_d in (1) that stabilizes the linearized system. The linearized model contains uncontrollable modes that correspond to the rolling wheels and other stable motions, and thus the system is stabilizable but not controllable. In addition, only partial state information can be measured with sensors. Linear quadratic Gaussian (LQG) control laws designed for the linearized system may not be robust to model uncertainty and may require more computation than can be achieved with the microcontroller used for the experiments. A classical control approach is thus used.

Figure 8 shows a block diagram of the linearized system. This diagram shows the vehicle dynamics and steering servo dynamics, whose product $G_{steer}(s)G_{vehicle}(s)$ is the transfer function from the steering-angle command β_d to the output roll angle ψ and roll rate $\dot{\psi}$ of the system. The control compensator used to stabilize the roll angle is represented by $H(s)$ in Figure 8. After testing several control strategies, the chosen compensator is the proportional-plus-derivative control law $\beta_d = K_p\psi + K_V\dot{\psi}$, where ψ is the deviation of the roll angle from its equilibrium value. The chosen compensator $H(s)$ is thus a static output feedback compensator with inputs ψ and $\dot{\psi}$ and with output β_d. The reason for this choice is that, near the balance equilibrium, the vehicle behaves similarly to an inverted pendulum on a moving cart system, where the cart motion is created by the lateral acceleration resulting from the steering angle. For the cart-pole system, proportional-plus-derivative gains can stabilize the system [16].

Figure 9 shows a root locus analysis for the linearized model $H(s)G_{steer}(s)G_{vehicle}(s)$ from Figure 8. The root locus plots shown in figures 9 and 10 assume a roll-angle gain of $K_p = 4$ (steering deg)/(roll deg) and a roll angular rate gain of $K_V = 0.3$ (steering deg)/(roll deg/s). The open-loop poles and zeros of the system transfer function are the x's and o's in the plot, respectively. The plots show the location of the closed-loop poles as the two gains are scaled simultaneously by means of a loop gain that varies from zero to infinity. A cluster of eight marginally stable poles are located at the origin in Figure 9, while all of the remaining poles are asymptotically stable except for one, which is due to the unstable nature of the equilibrium point. This pole, which must be stabilized, is shown in a zoomed-in plot in Figure 10.

The zoomed-in plot in Figure 10 shows that it is possible to stabilize the unstable pole with the selected compensator when the root locus gain is sufficiently large. The point shown as "Gain: 1" in Figure 10 is the location of one of the complex conjugate closed loop poles that branch off the unstable pole location at $s = 6.2$. Since the loop gain

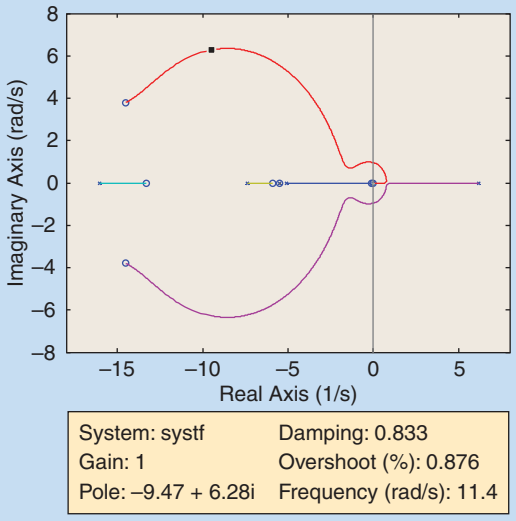

FIGURE 10 Zoomed-in portion of Figure 9. The unstable pole at s = 6.2 is stabilized with a loop gain of 1.0, as shown by the black square in the left-half s-plane. When the loop gain is 1.0, the roll-angle gain is $K_p = 4$ (steering deg)/(roll deg) and the roll angular rate gain is $K_V = 0.3$ (steering deg)/(roll deg/s).

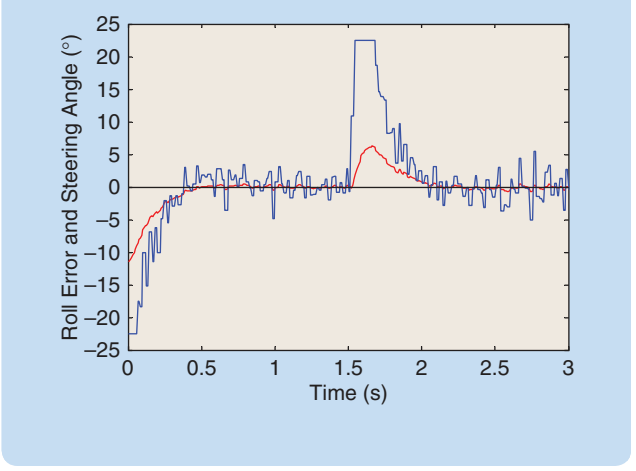

FIGURE 11 Simulation of the nonlinear system with the roll-control algorithm. The blue signal is the steering angle, while the red signal is the vehicle roll-angle error from equilibrium. Noise is injected into the roll feedback signal to replicate experimental conditions. After 1.5 s, an impulsive lateral force is applied to the vehicle to test the ability of the controller to reject disturbances. The vehicle recovers and continues to travel forward while maintaining the equilibrium roll angle.

is 1.0 at this point, the system is stabilized using the output feedback gains mentioned above. Examining the eigenvectors corresponding to the marginally stable poles reveals that they correspond to rolling of the wheels about their axis and motion of the vehicle in the forward direction. These modes are unimportant in terms of balancing the vehicle as it drives.

A noteworthy observation regarding the plot of Figure 9 is the nonminimum-phase zero. It is likely that this zero is responsible for the need for positive proportional-plus-derivative gains, which cause the steering angle to move in the same direction as the roll-angle error to stabilize the system. For instance, if the vehicle rolls beyond the 52.3° equilibrium angle, the front steering angle must be increased to create an acceleration of the vehicle body that returns it to the balance point. While other choices for the proportional-plus-derivative gains also result in a stable linearized closed-loop system, the combination used yields stable performance for both the full nonlinear simulation and the experimental system.

SIMULATION

The gains used to stabilize the linearized system in the preceding section are then used in a simulation of the nonlinear system. The deviation of the roll angle from its balance equilibrium angle and the steering angle are plotted versus time in Figure 11. The initial roll angle is set to 40°, so that there is a –12.3° error from the 52.3° equilibrium angle. The resulting steering angle commanded by the control compensator saturates at the 22.5° limit of the steering hardware. The plot shows that the vehicle roll-angle error is reduced to zero. During the simulation, zero-mean white noise with a variance of 10^{-3} $(°/s)^2$, which approximates the noise measured from the sensor, is added to the gyro signal. In addition to the initial roll-angle error, the vehicle is initialized with a forward velocity of 1.0 m/s. The vehicle reaches the equilibrium angle at approximately $t = 0.5$ s and drives at this angle until $t = 1.5$ s when an impulse disturbance of 4.4 N-s is applied to the vehicle. The vehicle recovers and returns to the equilibrium angle after roughly 0.5 s. The heading angle, which is shown in Figure 12 and which is not controlled in this simulation, changes as necessary to keep the vehicle balanced as it drives.

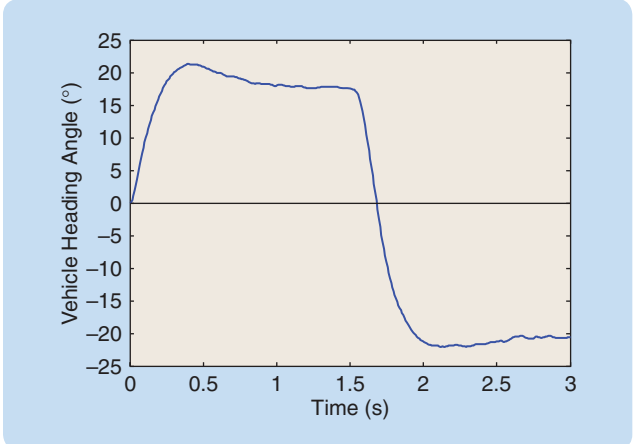

FIGURE 12 Heading angle from the simulation of Figure 11. The blue curve shows the change in vehicle heading caused by the steering commands. Note the immediate change in direction after 1.5 s when the impulse is applied.

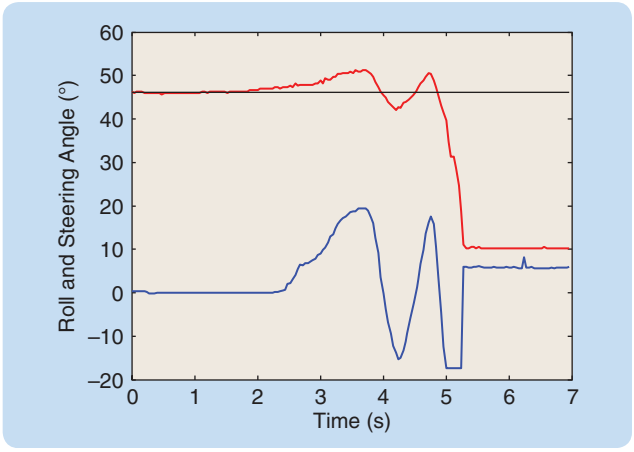

FIGURE 13 Unsuccessful experimental test run. The red curve is the roll angle, while the blue curve is the steering angle. The vehicle is released from a tilted position, but the vehicle's forward velocity is too fast for the steering angle to compensate for roll as the vehicle begins to fall.

TEST RESULTS

The RC transmitter shown in Figure 3 sends PCM signals for the throttle and steering to the receiver. The receiver sends the signals to the AVR Butterfly microcontroller, which uses interrupts to detect the pulses and calculate the pulse times for each signal. The AVR Butterfly receives analog signals from a potentiometer, gyroscope, and accelerometer and converts them to digital signals. The AVR Butterfly then uses the sensor signals to calculate the roll angle and angular rate. These values are used to implement the control law and calculate a PCM signal to be sent to the steering servo to balance the vehicle. This steering signal is then filtered by a first-order lowpass discrete-time filter whose bandwidth is approximately 4 Hz. The filter is needed to remove some of the effects of noise in the sensor measurements and also to deal with the limited bandwidth and torque and speed constraints of the RC steering servo. The filtered steering signal is output by the AVR Butterfly to the steering servo along with the throttle signal originally sent by the RC transmitter.

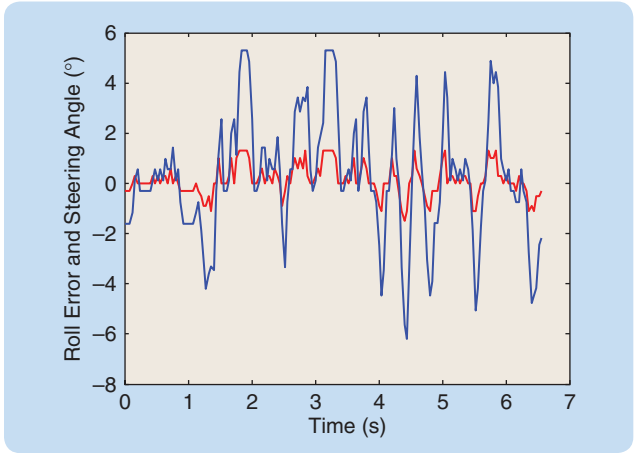

FIGURE 14 Experimental run with vehicle speed approximately 1.0 m/s. The red curve is the roll-angle error, and the blue curve is the steering angle. The steering angle holds the vehicle roll angle to within +/− 2° of its equilibrium value.

During the experiments, the steering command sent from the RC transmitter is interpreted by the microcontroller software in two ways as follows. If the vehicle is not tilted more than 20° from the horizontal, then the steering command is sent directly from the AVR to the steering servo. Otherwise, the steering command adds a small bias of up to $+/-0.5°$ to the desired equilibrium roll angle. This perturbation causes the heading of the vehicle to change while it balances on two wheels, which enables obstacle avoidance to be controlled manually with the joystick of the RC transmitter. Throughout the testing of the vehicle, sensor data is logged. Due to memory limitations of the microcontroller, two variables are recorded for each test run.

Figure 13 shows the roll angle and steering angle for one test run. As the roll angle begins to increase from the equilibrium balance value, the vehicle compensates by increasing the steering angle. The roll angle then dips lower than the balance angle, and the vehicle steers in the opposite direction to try to stay balanced. This run illustrates how the vehicle steers in response to changes in roll angle, but ultimately the control law does not stabilize the vehicle. The main reason for the failure is that the forward speed of the vehicle is controlled manually with the RC joystick. The vehicle starts from rest and accelerates to the desired speed of 1.0 m/s, which is used for the analysis in the simulation described above. In the test run shown, the vehicle speed is too fast, and the front steering angle cannot correct for the roll error quickly enough. Conversely, if the speed is too slow, the steering control cannot create an acceleration of the cg that is sufficient to maintain balance.

For another test run, Figure 14 shows the roll-angle deviation from the equilibrium angle as well as the steering angle. The roll angle remains within $\pm 1.5°$ of 47°, while the steering angle oscillates with an amplitude of approximately 6°. Several test runs are conducted with manual disturbances applied to the vehicle. The results demonstrate stable performance with a response similar to that shown in Figure 14. A video of one of these experiments can be viewed on YouTube [17]. This video demonstrates that, in addition to balance control, obstacle avoidance can be achieved by making adjustments in the roll-angle setpoint. These adjustments are sent to the vehicle manually from the RC transmitter while the experiment is in progress.

CONCLUSIONS

A control law that enables a remote-controlled vehicle to drive while self-balancing on two wheels is developed. The vehicle is modeled as a multibody system with 11 DOFs and simulated using the SimMechanics package of Matlab. Additionally, an elementary tire-contact force model is devised to model tire-to-ground interaction. This tire-force

model is sufficiently realistic to simulate the vehicle's motion and allow for a linearized analysis of the vehicle stability properties. Consistent with previous research on the modeling of bicycles, the linearized analysis in this article contains an unstable zero. The analysis demonstrates that roll angle and rate feedback are sufficient to stabilize the roll angle of the vehicle. Several marginally stable modes are left uncontrolled but do not interfere with the overall behavior of the system. The vehicle achieved self-balance while driving the length of a 25-yd hallway.

AUTHOR INFORMATION

David Arndt received the B.S. and M.S. in mechanical and aerospace engineering from the University of California, Irvine. He is currently a cryogenic systems research and development engineer at L3 Communications, Inc, in Pasadena, California.

James E. Bobrow is a professor of mechanical and aerospace engineering at the University of California, Irvine. He received the Ph.D. in 1983 from the University of California, Los Angeles. As a senior programmer analyst at McDonnell Douglas Automation Company, he developed CAM software for the Unigraphics system. Since joining University of California, Irvine, in 1984, his research interests have included optimal control and motion planning for robots, design of pneumatic actuators and sensors for automation systems, robots for rehabilitation, and machine learning systems. He has been a visiting professor at Stanford University and MIT, and he has created automation devices for industry. He has served on organizing committees for several conferences in control systems and robotics, and he has been an associate editor of the *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*

Steven Peters received the B.S. from the University of California, Davis, in 2004 and the M.S. from MIT in 2006, both in mechanical engineering. His research interests include improving energy and transportation technology. He is a member of the MIT Energy Club and cofounder of the MIT Energy Map. His current Ph.D. research at MIT is related to planning and control techniques for mobile robots operating at high speed on sloped and rough terrain.

Karl Iagnemma received the B.S. in mechanical engineering from the University of Michigan, Ann Arbor, in 1994, and the M.S. and Ph.D. in mechanical engineering from MIT in 1997 and 2001, respectively. He is currently a principal research scientist in the Department of Mechanical Engineering, MIT, where he was also a National Science Foundation graduate fellow. He has been a visiting researcher at the Jet Propulsion Laboratory and the National Technical University of Athens (Greece). His current research interests include the design, sensing, motion planning, and control of mobile robots in outdoor terrain, including modeling and analysis of robot–terrain interaction. He is the author of the monograph *Mobile Robots in Rough Terrain: Estimation, Planning and Control With Application to Planetary Rovers* (Springer, 2004). He is currently an associate editor of *IEEE Transactions on Robotics* and *Journal of Field Robotics.*

Steven Dubowsky received the B.S. from Rensselaer Polytechnic Institute, and the M.S. and Sc.D. from Columbia University. He is currently a professor of mechanical engineering at MIT. His research has included the development of modeling techniques for manipulator flexibility and the development of optimal and self-learning adaptive control procedures for rigid and flexible robotic manipulators. He has authored or coauthored over 200 papers in dynamics, control, and design of high-performance mechatronic and robotic systems. He has served as an advisor and consultant to the National Science Foundation, the National Academy of Science/Engineering, the Department of Energy, the U.S. Army, and industry. He is a Fellow of the American Society of Mechanical Engineers, and a member of Sigma Xi and Tau Beta Pi.

REFERENCES

[1] U.S. Department of Transportation, National Highway Traffic Safety Administration. (2005). Traffic safety facts. Document DOT HS 810 631 [Online]. Available: http://www-nrd.nhtsa.dot.gov/pubs/tsf2005.pdf

[2] D. Odenthal, T. Bunte, and J. Ackermann, "Nonlinear steering and braking control for vehicle rollover avoidance," in *Proc. European Control Conf.*, Karlsruhe, Germany, 1999, pp. 1–6.

[3] B. Chen and H. Peng, "Differential-braking-based rollover prevention for sport utility vehicles with human-in-the-loop evaluations," *Veh. Syst. Dyn.*, vol. 36, no. 4-5, pp. 359–389, 2001.

[4] C. R. Carlson and C. J. Gerdes, "Optimal rollover prevention with steer by wire and differential braking," in *Proc. ASME Dynamic Systems and Control Division,* Nov. 2003, pp. 345–354.

[5] M. W. Spong, "Energy-based control of a class of underactuated mechanical systems," in *Proc. 1996 IFAC World Congr.*, San Francisco, CA, July 1996, pp. 431–435.

[6] Guiness World Records (2000, Nov. 26). Most cars driven on two wheels simultaneously [Online]. Available: http://www.youtube.com/watch?v=pobOY1g0QBI

[7] K. J. Astrom, R. E. Klein, and A. Lennartsson, "Bicycle dynamics and control: Adapted bicycles for education and research," *IEEE Contr. Syst. Mag.*, vol. 25, no. 4, pp. 26–47, 2005.

[8] N. H. Getz and J. E. Marsden, "Control for an autonomous bicycle," in *Proc. IEEE Int. Conf. Robotics and Automation,* Nagoya, Japan, 1995, vol. 2, pp. 1397–1402.

[9] J. Yi, D. Song, A. Levandowski, and S. Jayasuriya, "Trajectory tracking and balance stabilization control of autonomous motorcycles," in *Proc. IEEE Int. Conf. Robotics and Automation,* Orlando, FL, 2006, pp. 2583–2589.

[10] T. Shim and C. Ghike, "Understanding the limitations of different vehicle models for roll dynamics studies," *Veh. Syst. Dyn.*, vol. 45, no. 3, pp. 191–216, 2007.

[11] A. J. Miege and D. Cebon, "Optimal roll control of an articulated vehicle: Theory and model validation," *Veh. Syst. Dyn.*, vol. 43, no. 12, pp. 867–884, 2005.

[12] J. Yoon, D. Kim, and K. Yi, "Design of a rollover index-based vehicle stability control scheme," *Veh. Syst. Dyn.*, vol. 45, no. 5, pp. 459–475, 2007.

[13] S. C. Peters and K. Iagnemma, "An analysis of rollover stability measurement for high-speed mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation,* Orlando, FL, 2006, pp. 3711–3716.

[14] H. B. Pacejka, "Magic formula tyre model," *Veh. Syst. Dyn.*, vol. 21, pp. 1–18, 1993.

[15] HPI Racing. (2007). Wheel king truck [Online]. Available: http://www.hpiracing.com/kitinfo/10830/

[16] K. Ogata, *Modern Control Engineering.* Englewood Cliffs, NJ: Prentice-Hall, 1970, pp. 277–279.

[17] D. Arndt. (2009). RC car automatically balancing and driving on two wheels #3—Body added [Online]. Available: http://www.youtube.com/watch?v=GsPkKFs595Q