

Self-Balancing Control of a Four Wheeled Vehicle

David Arndt, James E Bobrow*
Department of Mechanical and Aerospace Engineering,
University of California, Irvine
4200 Engineering Gateway
Irvine CA 92697
{jebobrow,darndt}@uci.edu

Steven Peters, Karl Iagnemma, and Steven Dubowsky
Department of Mechanical Engineering,
Massachusetts Institute of Technology
77 Massachusetts Ave.
Cambridge, MA 02139
{scpeters, kdi, dubowsky}@mit.edu

Keywords: vehicle balance control, stability, contact model

* Corresponding author

Abstract – For cars or trucks on the verge of roll-over, the boundary between stability and instability occurs when the vehicle balances on two wheels on the same side. In this work, the open-loop steering servo of an off-road type vehicle is used in a closed-loop balance control system that actuates the front steering angle so that the vehicle self-balances and drives on two wheels. A nonlinear vehicle model was linearized about the equilibrium balance point, and classical control techniques were used to obtain a stabilizing control system. A relatively simple contact force model was created in order to model the vehicle's tire interaction with the ground. It is demonstrated analytically and experimentally that the control law developed for this system is able to retain the equilibrium balance angle of the vehicle even while it encounters significant disturbances. An 8 bit Atmel AVR Butterfly microcontroller was used to implement the control law on-board the vehicle.

1. Introduction

Human stunt car drivers are able to keep cars or trucks balanced while they are tilted in the air and driving on two wheels on the same side. The goal of this research project was to see if this capability could also be achieved by robotic vehicles with limited computing capability. In order to

accomplish this, a remote controlled (RC) vehicle (a “Wheely King truck [17]) was modified so that it self balances while driving on two wheels on the same side. Creating a self balancing driving vehicle is a difficult problem that involves the analysis of vehicle dynamics and embedded control system implementation. Control analysis for simple balancing inverted pendulum Segway type robots have been developed (see e.g. Pathak et al [11] and their references), and the difficulty stabilizing the nonlinear and nonholonomic equations of motion has been demonstrated. In addition, nonlinear models for balancing bicycles and motorcycles have been developed with varying levels of complexity [1, 3, 14, 16, and 19]. Of note is Astrom et al [1] where one of several bicycle models is shown to have a right half plane zero whose location varies linearly with velocity. A similar zero appears in the linearized analysis developed in this paper. Besides being an original and challenging problem, this project could have application to vehicle safety and rollover stability. If the vehicle can be balanced on the rollover threshold then insight can be gained in order to prevent this threshold from being reached.

Much research has been done in the field of vehicle rollover stability and various vehicle models have been developed with varying complexity and physical assumptions. Shim[15] develops a complex 14 degree-of-freedom (DOF) model and compares it to a simpler 8 DOF model, discussing the limitations of different models. Miede[9] develops a 8 DOF model and creates a full scale experimental setup with a freight vehicle. Using an even simpler 3 DOF model, Yoon[20] develops a rollover index to indicate an impending rollover. Peters[12] and Whitehead[18] use small scale remote controlled vehicles to analyze rollover stability and test electronic stability control systems.

An important element of this work is the tire force contact model. It is used for the closed loop stability analysis, and must be a differentiable function of the system state. Just as with the vehicle dynamics model, there exists a wide range of tire force contact models. Tire models can roughly be grouped into two main categories, physical and empirical, with hybrid models being a combination of the two. Physical models are completely analytical and attempt to determine the forces acting upon the tire in a given situation. Empirical models use experimental data and various parameters to create empirical relationships. Empirical models include the widely used "magic formula" model developed by Pacejka[10] and the UniTire model created by Guo[4]. Both of these models require extensive experiments on a test bench and can involve up to 100 different parameters, as described by Porcel[13]. Many tire models are based upon Pacejka's "magic formula" model such as the empirical motorcycle tire model developed by Lot[8] and the hybrid PAC2002 model used in MSC Adams simulation software developed by Kuiper[7]. Analytical models include the rigid ring and 6 DOF springs pressure distribution analysis of Kim[5] as well as the Dynamic Deflection Model developed by Koo[6]. The contact model used in our research is not as complex as those cited above, but it is amenable for control analysis developed in this work and it captures the dominant tire to road force characteristics.

The main contribution of this research is that it is shown both analytically and experimentally that a nonminimum phase control law consisting of roll angle and rate feedback is sufficient to stabilize the roll angle of the vehicle. In order to present this result, we first describe the vehicle model used for the analysis. It has 11 DOF and was simulated in the SimMechanics simulation environment of Matlab. We then develop the tire contact model, which is a spring damper system, modified to be piecewise continuously differentiable. Given the nonlinear vehicle and contact models, we linearize the equations about the balance equilibrium point in order 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 actual behavior of the full nonlinear dynamic system.

2. Vehicle Model

Figure 1 shows a photo of the remote controlled vehicle used for this research. The vehicle was modeled as a rigid body with three translational and three rotational degrees of freedom located at the vehicle center of mass as shown in the figure. In addition to these coordinates, a rotational DOF was used for each wheel and a rotational DOF was used for the front steering angle which was assumed to be the same for both front wheels. This resulted in a model with 11 DOF and required a 22 dimensional state space for the simulation and the stability analysis. Defining the above mentioned 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 usual 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 contribution of generalized force from the two wheels in contact with the ground. The tire to ground contact forces are F_{c1} , F_{c2} an important component of the dynamic model, and the details of this model are provided in the next section. The Jacobian relationships $J_1^T(q)$, $J_2^T(q)$ define how the contact forces influence the motion of each generalized coordinate. The Mathworks SimMechanics toolbox was used to model and simulate the dynamics of the vehicle and to compute the above mentioned Jacobians. The mass properties used to represent the vehicle body and wheels are provided in Appendix A

Figure 2 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 was found from a static analysis to be 52.3 degrees. The RC servo that steers the front wheels (see Figure 3) was modeled as a proportional plus derivative control system. The transfer function of this system has the form

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

where β is the steering angle output from the RC servo, and β_d is the desired steering angle. The desired steering angle was computed by the control law developed in the next section. The coefficients of the transfer function were selected so that the step response of this transfer function approximated that of the actual RC servo, with $\omega = 20$ and $\zeta = .5$. Note that the actual RC servo had characteristics that are not captured with the transfer function including saturation and Coulomb friction. The stock steering servo system had excessive backlash in the linkage and did not provide enough torque to closely hold the desired angle. It was replaced with a Hitec HSR-5995TG titanium gear high torque servo shown in Fig. 3 along with a modified linkage in order to reduce backlash.

3. Contact Force Model

During the simulation of the vehicle model, the tire contact forces must be computed as a function of the system state. Given the orientation and the 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 Fig. 4 as follows. The tire's center of gravity is located at point \mathbf{P} , the axis of rotation is \vec{y}_w , and the point of contact is

located at point P_c . The local coordinate frame of the wheel consists of $R_w = \{\bar{x}_w, \bar{y}_w, \bar{z}_w\}$ as shown, where \bar{x}_w is out of the page. The vector \bar{n} is a unit vector normal to the ground. The radius of curvature of the tire's corner is r_c . The vector \bar{y}_w is obtained directly from the second column of the wheel rotation matrix R_w . The unit vector \bar{z}_w is then calculated from \bar{n} and \bar{y}_w as $\hat{\bar{z}}_w = \bar{n} - (\bar{n}^T \bar{y}_w) \bar{y}_w$, and $\bar{z}_w = \frac{\hat{\bar{z}}_w}{\|\hat{\bar{z}}_w\|}$.

Given the local coordinate frame as computed above, the tire's contact point is calculated as

$$\bar{P}_c = \bar{P} + tilt \cdot \left(\frac{wid_w}{2} - r_c \right) \bar{y}_w - (r_w + r_t - r_c) \bar{z}_w - r_c \bar{n}, \quad (2)$$

where $tilt = -\text{sign}(\bar{n}^T \bar{y}_w)$. If $tilt = +1$, the \bar{y}_w axis is pointing into the ground, which means that the tire is contacting on the outer radius (the positive \bar{y}_w axis points to the outside of the vehicle) as is shown in Fig. 4, while a -1 value corresponds to contact on the inner radius. The velocity of the tire's contact point is computed by differentiating (2) to be

$$\dot{\bar{V}}_c = \dot{\bar{P}} + tilt \cdot \left(\frac{wid_w}{2} - r_c \right) \dot{\bar{y}}_w - (r_w + r_t - r_c) \dot{\bar{z}}_w, \quad (3)$$

where $\dot{\bar{P}}$ is the wheel CG velocity, $\dot{\bar{y}}_w = s(\omega) \bar{y}_w$, $\dot{\bar{z}}_w = s(\omega) \bar{z}_w$, 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 of the contact point from the ground is calculated as $d = n^T P_c$ and its derivative is $\dot{d} = n^T V_c$.

When $d < 0$, the normal force and friction force acting upon the tire must be computed. Figure 5 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 due to the spring and damper is

$$N_s = -kd - c\dot{d}. \quad (4)$$

However, there are three modifications that must be done to this force in order for it to be physically realistic. The first is that the normal force N_s must be greater than or equal to 0 because the ground can push on the tire but cannot pull. Thus, during the simulation, if (4) gives $N_s > 0$, then the normal force N acting upon the tire is set equal to 0. If $N_s < 0$ there are two different cases that can occur depending upon the value of $\vec{d} = \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix}$ as illustrated in Fig. 6. The first case, shown as \vec{d}_1 in the figure, occurs

when $d < 0$ and $\dot{d} > 0$. In this case, the tire is decompressing and moving away from the ground and the normal force N is set equal to N_s . Note that $N=0$ along the bold line in the upper left quadrant of Figure 6, so that the damping action of the tire makes it possible to have a compressed tire with no normal force. In other words, along the bold line (restitution), the compression force from the spring is offset by the force from the damper.

The second case, shown as \vec{d}_2 in the figure, occurs when $d < 0$ and $\dot{d} < 0$, which means that the tire is compressing and the wheel is moving towards the ground. In this case, the normal force calculation is modified so that the normal force is 0 along the negative \dot{d} axis. This is done in order to make the normal force vary continuously from $N=0$ at $d=0$ to its full value at \vec{d}_2 . Continuity of the

normal force is required both from a physical perspective and from a numerical one. That is, the numerical integration routines used in the simulation can not accurately integrate the model equations without this modification. It is achieved by subtracting the vector N_1 from N_s as shown in Fig. 6. From the geometry of the figure it is shown in Appendix 2 that the modified normal force is

$$N = N_s - N_1 = -\frac{d \cdot k}{\cos^2 \gamma}, \quad (5)$$

where $\tan \gamma = c/k$. Note that this modified force has an effective stiffness that is larger than the actual stiffness during the compression phase, and the increase depends on the damping rate c .

Next, the horizontal component of the contact point velocity, \vec{V}_h , is calculated to determine the direction in which the friction force acts. From (2), \vec{V}_c is projected to the horizontal plane as

$$\vec{V}_h = [I - \vec{n}\vec{n}^T] \vec{V}_c. \quad (6)$$

The friction force acting on the tire is then opposes the horizontal component of the velocity as

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

Finally, the generalized force resulting from the normal force and friction are calculated at the CG of the wheel as

$$\vec{T}_{Total} = \vec{T}_{Normal} + \vec{T}_{Friction} = \vec{r} \times \vec{N} + \vec{r} \times \vec{F}_f, \quad (8)$$

where $\vec{r} = \vec{P}_c - \vec{P}$.

4. Model Analysis

The nonlinear vehicle system dynamics were linearized about the tilted equilibrium point to obtain a linear state-space model that contained 22 state variables. As mentioned previously, these states included the 6 degrees of freedom for the car body and their velocities, the 4 wheel rotations and velocities, and the steering angle and its velocity. A number of different feedback control laws were considered for the steering angle β_d in (1) that could potentially stabilize the system. The model contains uncontrollable modes that complicate modern control approaches such as linear quadratic Gaussian designs. In addition, only partial state information can be measured with sensors. The uncontrollable modes correspond to the wheels rolling and other stable motions, so the system is stabilizable but not controllable by the standard definition of controllability (see e.g. Chen [2]). After testing several control strategies, a compensator discussed in the following was found that was easy to implement and was reasonably robust.

In Figure 7, a block diagram of the linearized system and controller is shown. The diagram shows the vehicle dynamics $G(s)$ which represents the transfer function from the desired steering angle command β_d to the output state which includes the roll angle and roll rate. The control compensator used to stabilize the roll angle is represented by $H(s)$ in the figure, which is a linear combination of the roll angle and roll rate as shown. Figure 8 shows a root locus analysis for the linearized system $GH(s)$ from Figure 7. The open loop poles and zeros of the system transfer function are the x's and o's in the plot, respectively. There is a cluster of 8 marginally stable poles at the origin of the figure, and all of the remaining poles are stable except for one, which is due to the unstable nature of the equilibrium point. This pole must be stabilized and is shown in the zoomed-in plot on the right-hand side Figure 8.

A noteworthy observation regarding the left hand plot of Fig. 8 is the zero located in the right-half s-plane. This nonminimum phase zero requires that the controller use positive feedback gains in order to move the steering angle in the same direction as the roll angle error in order to stabilize the system. For instance, if the vehicle has rolled beyond the 52.3 degree equilibrium angle, the front steering angle must be increased in order to create an acceleration of the vehicle body that returns it to the balance point.

In the zoomed-in plot shown in the right-hand side of Fig 8, we see that it is possible to stabilize the unstable pole using a linear combination of roll and roll rate feedback. Physically, the requirement that the roll rate be used (also with positive gain) indicates that one must account for the vehicle roll angular velocity along with the roll error when adjusting the front wheel steering angle. As shown in $H(s)$ in Fig. 7, a roll angle gain of 4.0 (steering deg)/(roll deg) and a roll angular rate gain of 0.3 (steering deg)/(roll deg/s) were used to create the transfer functions for the root locus plots shown. While other choices for the gains resulted in a stable closed loop system, this combination yielded very stable performance for both the full nonlinear simulation and the experimental system. The root locus plots shown represent the location of the closed loop poles as the two gains are scaled together with a loop gain that varies from 0 to infinity. The point shown as “gain=1.0” in right-hand side of Fig. 8 is the location of one of the complex conjugate closed loop poles that branched off the unstable pole location at $s=6.2$. Since the loop gain is 1.0 at this point, the system is stabilized using the output feedback gains 4.0 for roll and .3 for roll rate as mentioned above. Examining the eigenvectors corresponding to marginally stable poles reveals that they correspond to rolling of the wheels about their axis and motion of the vehicle in the x direction. These modes are unimportant in terms of balancing the vehicle as it drives.

5. Simulation

The gains used to stabilize the system in the preceding section were then used in a simulation of the nonlinear system. One example of the simulation results is shown in Figure 9, where the deviation of the roll angle from its balance equilibrium angle and the steering angle (the control input) are plotted versus time. The initial roll angle (the solid line) was set to 40 degrees, so that there was a -12.3 degree error from the 52.3 degree equilibrium angle. This caused the steering angle (the dashed line) to saturate at -22.5 degrees, which was a limit of the steering hardware. The plot shows that the vehicle roll angle error was quickly reduced to zero. During the simulation, zero mean white noise with a variance of 10^{-3} (degree/second)² was injected into the gyro signal. In addition to the initial roll angle error, the vehicle was started with a forward velocity of 40 inches/second. 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 1.0 lb-second was applied to the vehicle. The vehicle recovers and returns to the equilibrium angle after roughly one-half a second. The heading angle shown in Figure 10 was not controlled in this simulation and it changed as necessary to keep the vehicle balanced as it drove.

6. Experimental Hardware and Test Results

The main components used to test the control law experimentally are shown in Fig 11. The flow of information is represented by the black arrows. The RC transmitter sends pulse code modulation (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 also receives analog signals from a potentiometer, gyroscope, and

accelerometer and converts them to digital signals. It 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 output to the steering servo to balance the vehicle. This steering signal is then filtered by a first order low-pass discrete filter whose bandwidth was approximately 4 Hz. The filter was 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.

During the experiments, the steering command sent from the RC transmitter was interpreted by the microcontroller software in two ways as follows. If the vehicle was not tilted more than 20 degrees from horizontal, the steering command was sent directly from the AVR to the steering servo. Otherwise, the steering command added a small bias of up to ± 0.5 degrees to the desired equilibrium roll angle. This small perturbation caused the heading of the vehicle to change while it was balanced on two wheels and thus enabled obstacle avoidance to be controlled manually with the joystick of the RC transmitter. Throughout the testing of the vehicle, sensor data was logged. Due to memory limitations of the microcontroller, only two different variables could be recorded for each test run.

In Figure 12, the roll angle and steering angle are shown for one test run. As the roll angle begins to increase from the equilibrium balance value, the vehicle adjusts 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 was ultimately unsuccessful but it illustrates how the vehicle steers in response to changes in roll angle. The main reason for the failure was that the forward speed of the vehicle was controlled manually with the RC joystick. The vehicle was started from rest and an attempt was made to quickly accelerate it up to the desired speed of 40 inch/sec, which was used for the analysis in simulation section. In the test run shown, the vehicle speed was too fast and the front steering angle could not correct for the roll error quickly enough. Conversely, if the speed was too slow the steering control could not create the acceleration of the CG needed to maintain balance.

Figure 13 shows the roll angle deviation from the equilibrium angle and the steering angle for a successful test run. The roll angle remains in the region around approximately 47 degrees \pm approximately 1.5 degrees while the steering angle oscillates approximately \pm 6 degrees. A number of test runs were performed with manual disturbances applied to the vehicle which demonstrated relatively stable performance with response characteristics similar to that shown in Fig 13. A video of one of the actual experiments can be viewed at the website <http://www.youtube.com/watch?v=GspKkFs595Q>. The video also demonstrates that in addition to balance control, obstacle avoidance can be achieved as mentioned above by making very small adjustments in the roll angle set-point. These adjustments were sent to the vehicle manually from RC transmitter as the experiment was in progress.

7. Conclusions

A control law was developed so that a remote controlled vehicle could drive while self-balancing on two wheels. The vehicle was modeled as a multi-body system with 11 degrees of freedom and simulated using the SimMechanics package of Matlab. Additionally, a simple contact force model was devised to model tire to ground interaction. The tire contact model proved realistic enough to simulate the vehicle's motion and to allow for a linearized analysis of the vehicle stability properties. The linearized analysis contained an unstable zero which is consistent with previous research on the modeling of bicycles. The analysis demonstrates that roll angle and rate feedback are sufficient to stabilize the roll angle of the vehicle. Other, marginally stable states were left uncontrolled and did not

interfere with the overall behavior of the system. The experimental hardware included a low cost Atmel AVR Butterfly microcontroller which was used to decode pulse code modulation signals sent from a remote control transmitter, process sensor signals, and implement the control law by outputting modified PCM signals to the vehicle's steering servo. The vehicle was successfully driven the length of a 25 yard hallway while self balancing the entire way.

8. References

- [1] Astrom, K.J., Klein, R.E., and Lennartsson, A. Bicycle dynamics and control: adapted bicycles for education and research, Control Systems Magazine, IEEE Volume: 25, Issue: 4, Aug. 2005, Pages: 26- 47.
- [2] Chen, C.T. "Linear system theory and design" 1984 - Saunders College Publishing, Philadelphia 1984, PA, USA
- [3] Getz, N.H., and Marsden, J.E., Control for an Autonomous Bicycle, 1995 IEEE International Conference on Robotics and Automation, 21-27 May 1995, Nagoya, Pages:1397 – 1402, vol.2,
- [4] Guo, K., Lu, D., Chen, S.-K, Lin, W. C. and Lu, X.P. (2005). The UniTire model: A nonlinear and non-steady-state tyre model for vehicle dynamics simulation. Vehicle system dynamics, 43, 341-358.
- [5] Kim, S., Nikraves, P. E. and Gim, G. (2008). A two-dimensional tire model on uneven roads for vehicle dynamic simulation. Vehicle system dynamics, 46(10), 913-930.
- [6] Koo, S-L., Tan, H-S., and Tomizuka, M (2006). An improved tire model for vehicle lateral dynamics and control. Proceedings of the American Control Conference, 2006, 5879-5884.
- [7] Kuiper, E. and Van Oosten, J.J.M. (2007). The PAC2002 advanced handling tire model. Vehicle system dynamics, 45(1), 153-167.
- [8] Lot, R. (2004). A motorcycle tire model for dynamic simulations: Theoretical and experimental aspects. Meccanica, 39(3), 207-220.
- [9] Miede, A J. and Cebon, D. (2005). Optimal roll control of an articulated vehicle: Theory and model validation. Vehicle system dynamics, 43(12), 867-884.
- [10] Pacejka, H B. (1993). Magic formula tyre model. Vehicle system dynamics, 21, 1-18.
- [11] Pathak, K, Franch, J. and Agrawal, S.K. "Velocity and Position Control of a Wheeled Inverted Pendulum by Partial Feedback Linearization," Robotics, IEEE Transactions on Robotics and Automation, Volume 21, Issue 3, June 2005 Page(s):505 - 513
- [12] Peters, S. C. and Iagnemma, K. (2006). An analysis of rollover stability measurement for high-speed mobile robots. Proceedings - IEEE International Conference on Robotics and Automation, 2006, 3711-3716.
- [13] Porcel, A., Laurence, P., Basset, M. and Gissinger, G.L. (2001). Tyre model for vehicle simulation: Overview and real time solution for critical situations. IEEE Conference on Control Applications - Proceedings, 817-822.

- [14] Ruijs, P.A.J. and Pacejka, H.B., “Research in the lateral dynamics of motorcycles,” in *Proc. 9th IAVSD Symp. Dynamics Vehicles Roads Tracks*, 1996, pp. 467–478.
- [15] Shim, T. and Ghike, C. (2007). Understanding the limitations of different vehicle models for roll dynamics studies. *Vehicle system dynamics*, 45(3), 191-216.
- [16] Tanaka, Y. and Murakami, T. (2004). Self sustaining bicycle robot with steering controller. *International Workshop on Advanced Motion Control, AMC*, , 193-197.
- [17] Wheel King Truck Manufacturer: <http://www.hpiracing.com/kitinfo/10830/>
- [18] Whitehead, R. et al. (2005). Scaled vehicle electronic stability control. *ESV International Collegiate Student Safety Technology Design Competition. North American Regional Review*.
- [19] Yi, J., Song, D., Levandowski, A., and Jayasuriya, S. (2006). Trajectory tracking and balance stabilization control of autonomous motorcycles. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006, 2583-2589.
- [20] Yoon, J., Kim, D., and Yi, K. (2007). Design of a rollover index-based vehicle stability control scheme. *Vehicle system dynamics*, 45(5), 459-475.

Appendix 1: Mass properties used in the simulation

| Variable | Description | English Units | | SI Units | |
|----------|---|---------------|---------------------|----------|-------------------|
| | | Value | | Value | |
| grav | gravity | 386.4000 | in/s ² | 9.8100 | m/s ² |
| mc | car body mass | 3.4410 | lbm | 1.5608 | kg |
| len | car body length | 9.5000 | in | 0.2413 | m |
| wid | car body width | 8.1875 | in | 0.2080 | m |
| height | car body height | 5.4000 | in | 0.1372 | m |
| lxx | car body moment of inertia about x axis | 27.5840 | lbm*in ² | 0.0081 | kg*m ² |
| lyy | car body moment of inertia about y axis | 34.2408 | lbm*in ² | 0.0100 | kg*m ² |
| lzz | car body moment of inertia about z axis | 45.1015 | lbm*in ² | 0.0132 | kg*m ² |
| mw | wheel mass | 0.4300 | lbm | 0.1950 | kg |
| rw | wheel radius | 1.3125 | in | 0.0333 | m |
| rt | tire thickness | 1.2500 | in | 0.0318 | m |
| wid_w | wheel width | 2.7500 | in | 0.0699 | m |
| la | wheel axial moment of inertia | 1.7821 | lbm*in ² | 0.0005 | kg*m ² |
| lt | wheel transverse moment of inertia | 1.1621 | lbm*in ² | 0.0003 | kg*m ² |
| mtot | total car mass | 5.1610 | lbm | 2.3410 | kg |
| k | tire and suspension spring constant | 2.0000 | lb/in | 350.2537 | N/m |
| c | tire and suspension damping | 0.4000 | lb/(in/s) | 70.0507 | N/(m/s) |
| mu | tire contact friction coefficient | 0.7500 | | 0.7500 | |

Appendix 2: Derivation of Restitution Stiffness

Referring to Figure 6, let \bar{u} be a unit vector in the (d, \dot{d}) plane that is orthogonal to the line defined by $N_s = 0 = [k \quad c] \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix}$. Then for any (d, \dot{d}) along this line, $\bar{u}^T \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix} = 0$ which implies that \bar{u} is parallel to $\begin{Bmatrix} k \\ c \end{Bmatrix}$ or $\bar{u} = \alpha \begin{Bmatrix} k \\ c \end{Bmatrix}$ for some scalar α . Also, from the figure $\bar{u} = \begin{Bmatrix} \cos \gamma \\ \sin \gamma \end{Bmatrix}$ and $\bar{w} = \begin{Bmatrix} -\sin \gamma \\ \cos \gamma \end{Bmatrix}$. Then for the arbitrary point $\bar{d}_2 = \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix}$ as shown in the figure, the magnitude of the modified normal force is

$$N = N_s - N_1 = [-k \quad -c] \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix} - [-k \quad -c] \begin{Bmatrix} 0 \\ \dot{d}_0 \end{Bmatrix} = -kd - cd + cd_0, \quad (\text{A1})$$

where $\dot{d}_0 \cos \gamma = a = \bar{w}^T \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix} = -d \sin \gamma + \dot{d} \cos \gamma$. Substituting for \dot{d}_0 into (A1) we obtain

$$N = -kd - cd + \frac{c}{\cos \gamma} (-d \sin \gamma + \dot{d} \cos \gamma) = -kd - cd \tan \gamma. \quad (\text{A2})$$

Finally using the facts that $\cos \gamma = \alpha k$, $\sin \gamma = \alpha c$, $\tan \gamma = \frac{c}{k}$, (A2) can be simplified to

$$N = -\frac{d \cdot k}{\cos^2 \gamma}. \quad (\text{A3})$$

List of figures

Figure 1: RC vehicle used for research with coordinate frame at vehicle CG, and Euler angles used for analysis.

Figure 2: Free body diagram of vehicle shows contact forces and gravity vector.

Figure 3: Modified high torque steering servo and linkage.

Figure 4: Contact point is located by constructing a local coordinate system using the wheel axis and the surface normal.

Figure 5: Normal force is a modified unilateral spring-damper model

Figure 6: The normal force is computed differently depending on the sign of the contact velocity.

Figure 7: Linearized system dynamics used in root locus analysis of $G(s)H(s)$.

Figure 8: Root locus plot of the transfer function from input steering angle to a linear combination of roll and roll rate. The right-hand plot is a zoom-in version of the left

Figure 9: Nonlinear simulation of roll control algorithm

Figure 10: Heading angle for previous simulation. Note change in heading when impulse is applied.

Figure 11: The main hardware components used for the experiments. The arrows indicate the direction of information flow.

Figure 12: Unsuccessful test run. Steering angle adjusts quickly to roll errors, but vehicle forward velocity was too fast to stabilize.

Figure 13: Successful test run with vehicle speed approximately 40 inch/sec. Solid line is roll angle error, dashed line is steering angle.

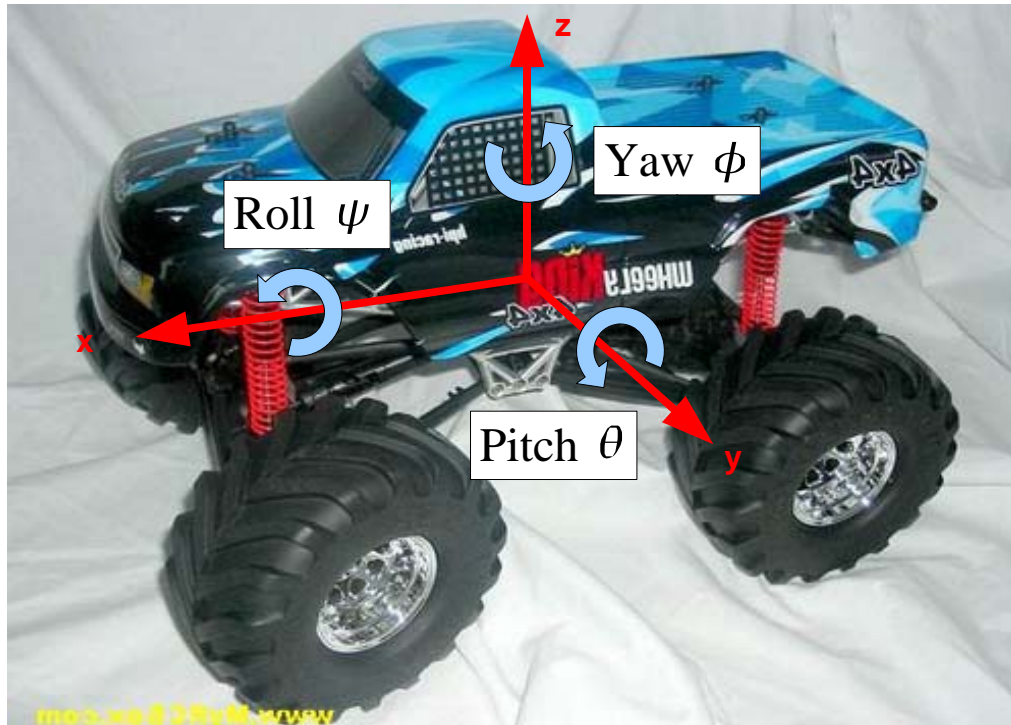


Figure 1: RC vehicle used for research with coordinate frame at vehicle CG, and Euler angles used for analysis.

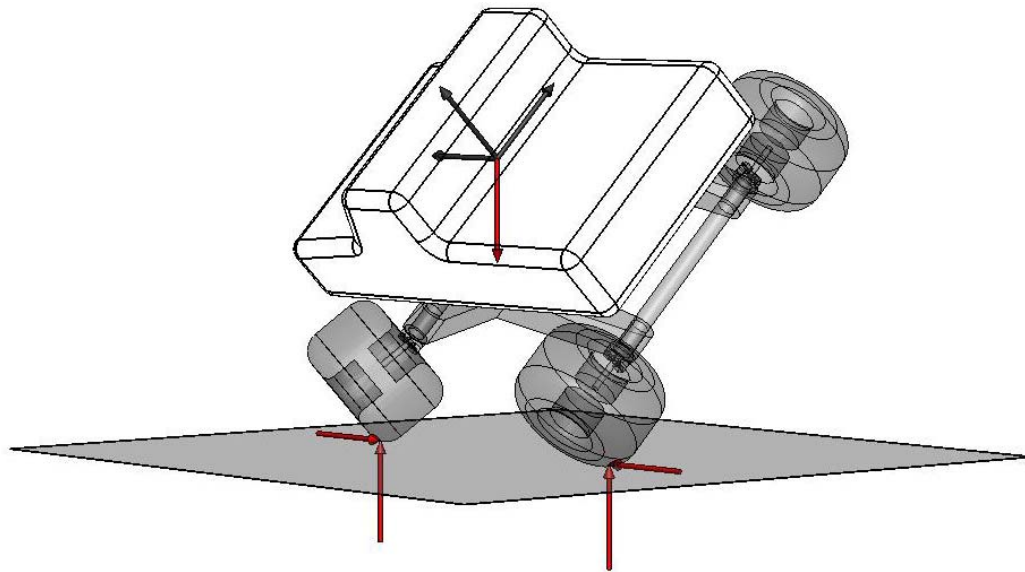


Figure 2: Free body diagram of vehicle shows contact forces and gravity vector.



Figure 3: Modified high torque steering servo and linkage.

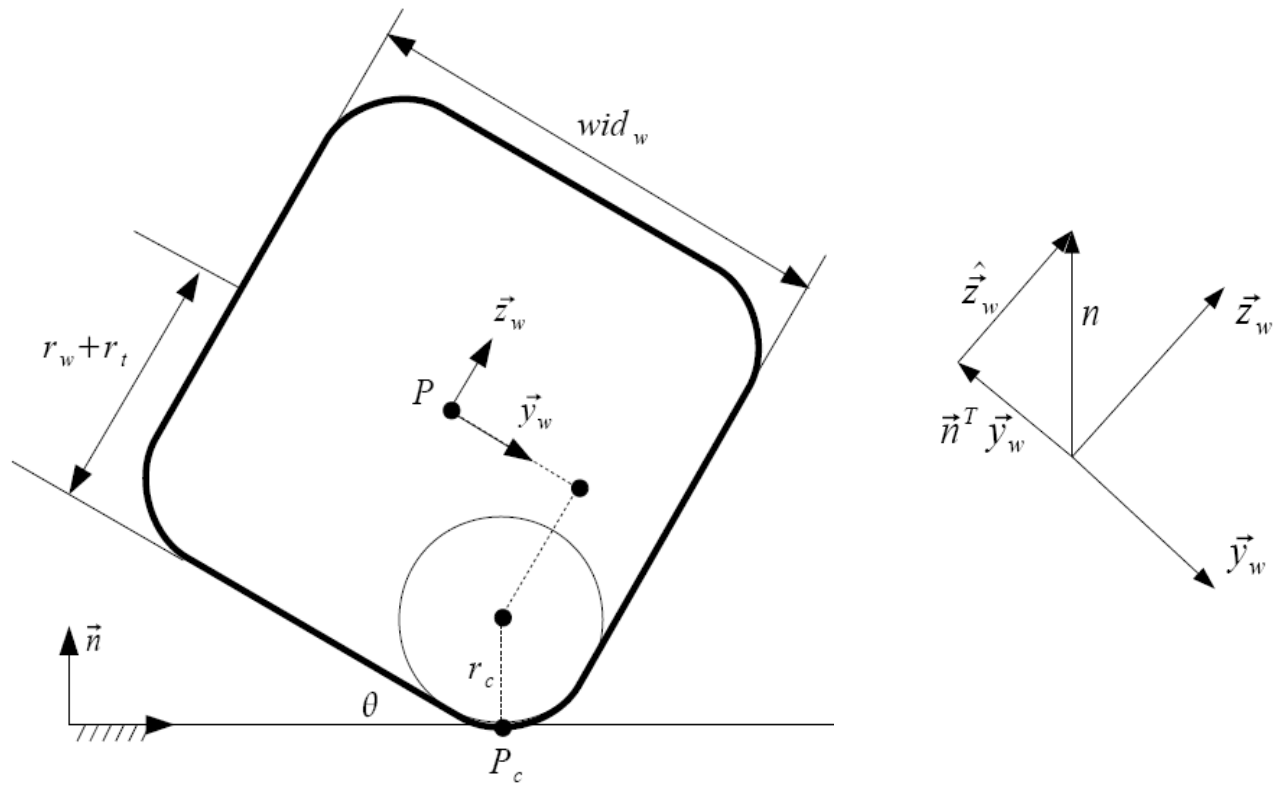


Figure 4: Contact point is located by constructing a local coordinate system using the wheel axis \vec{y}_w and the surface normal \vec{n} .

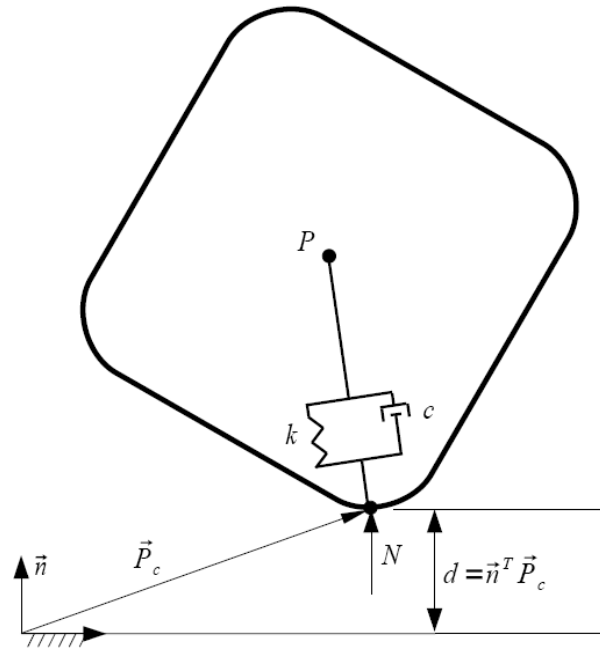


Figure 5: Normal force is a modified unilateral spring-damper model

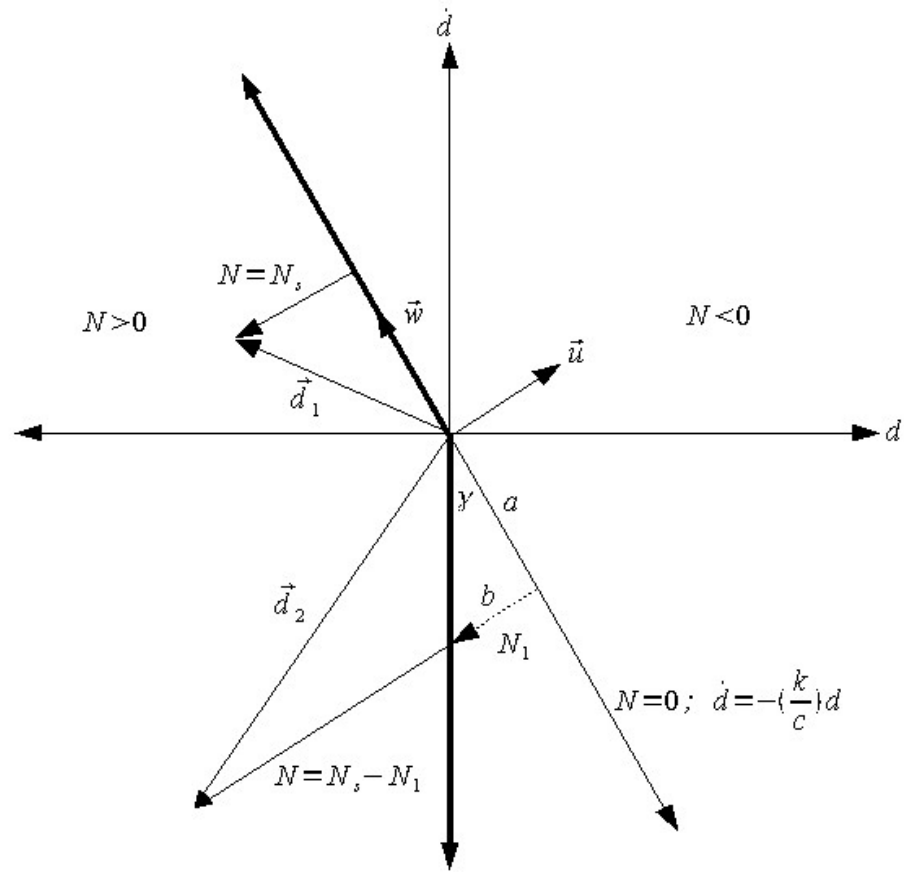


Figure 6: The normal force is computed differently depending on the sign of the contact velocity.

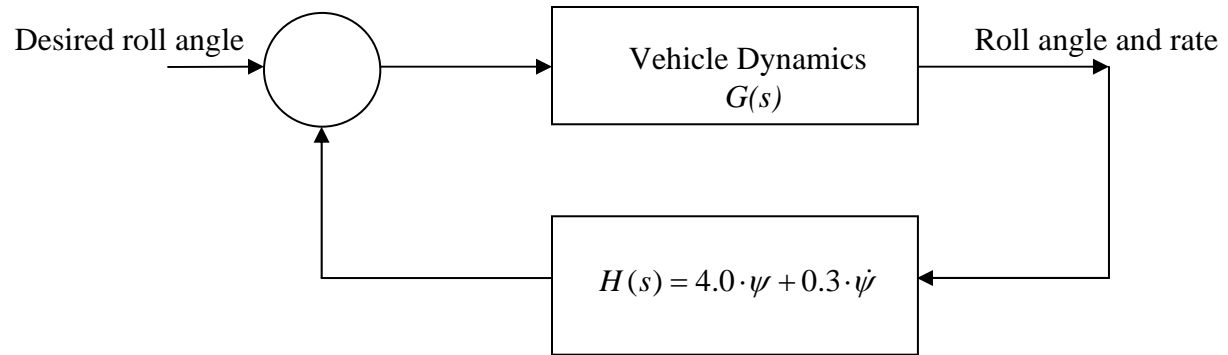


Figure 7: Linearized system dynamics used in root locus analysis of $G(s)H(s)$.

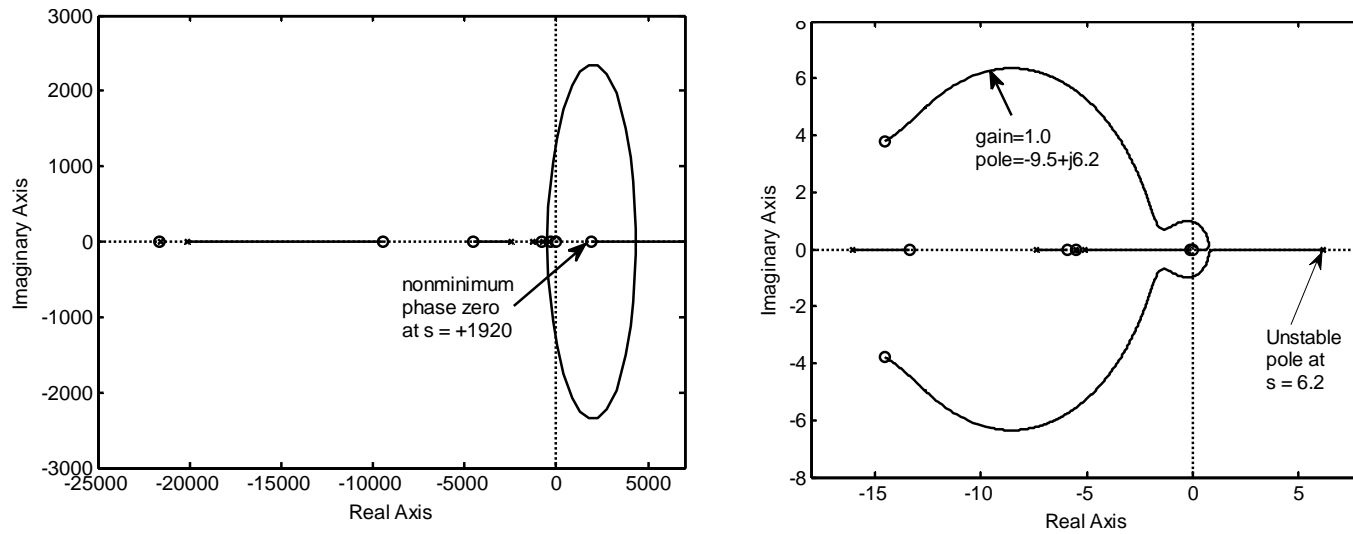


Figure 8: Root locus plot of the transfer function from input steering angle to a linear combination of roll and roll rate. The right-hand plot is a zoom-in version of the left.

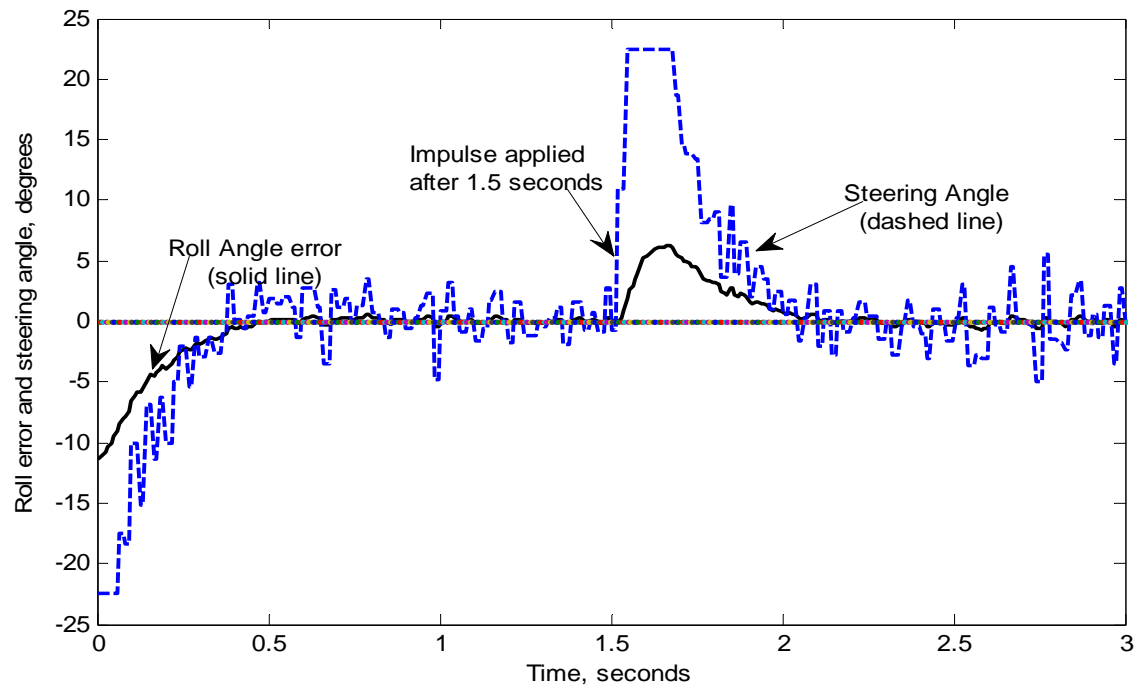


Figure 9: Nonlinear simulation of roll control algorithm.

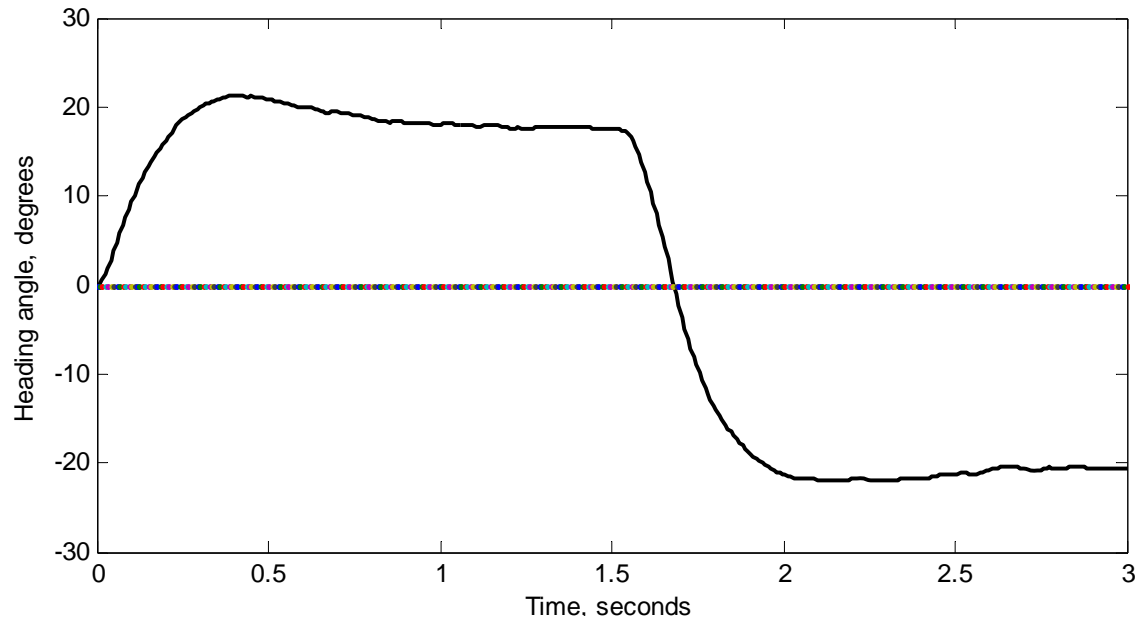


Figure 10: Heading angle for previous simulaiton. Note change in heading when impusle is applied.

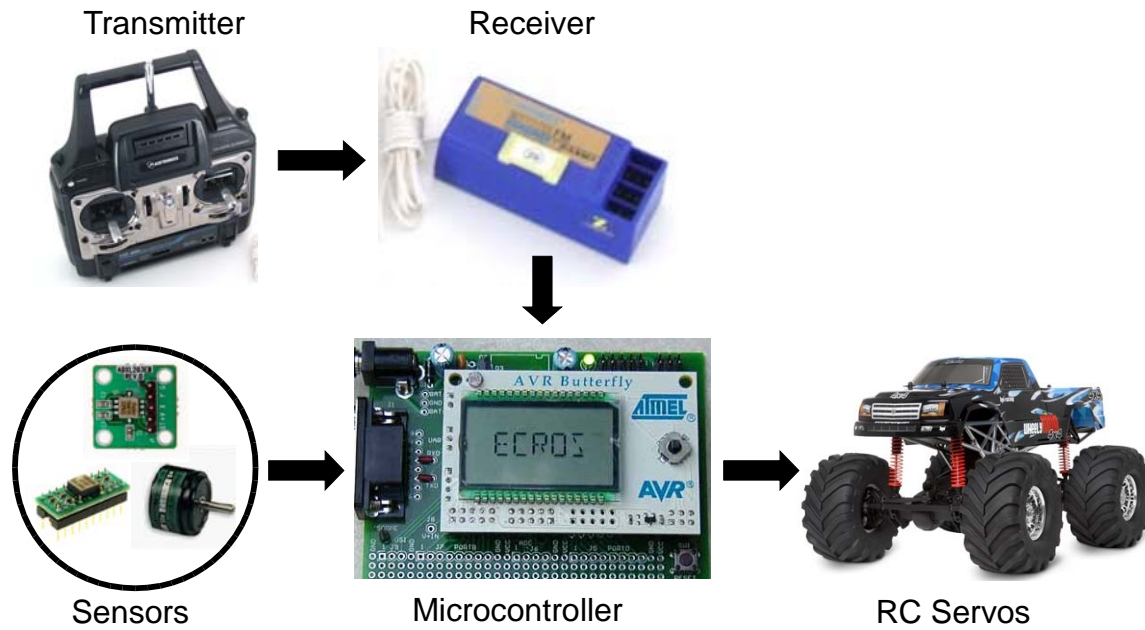


Figure 11: The main hardware components used for the experiments. The arrows indicate the direction of information flow.

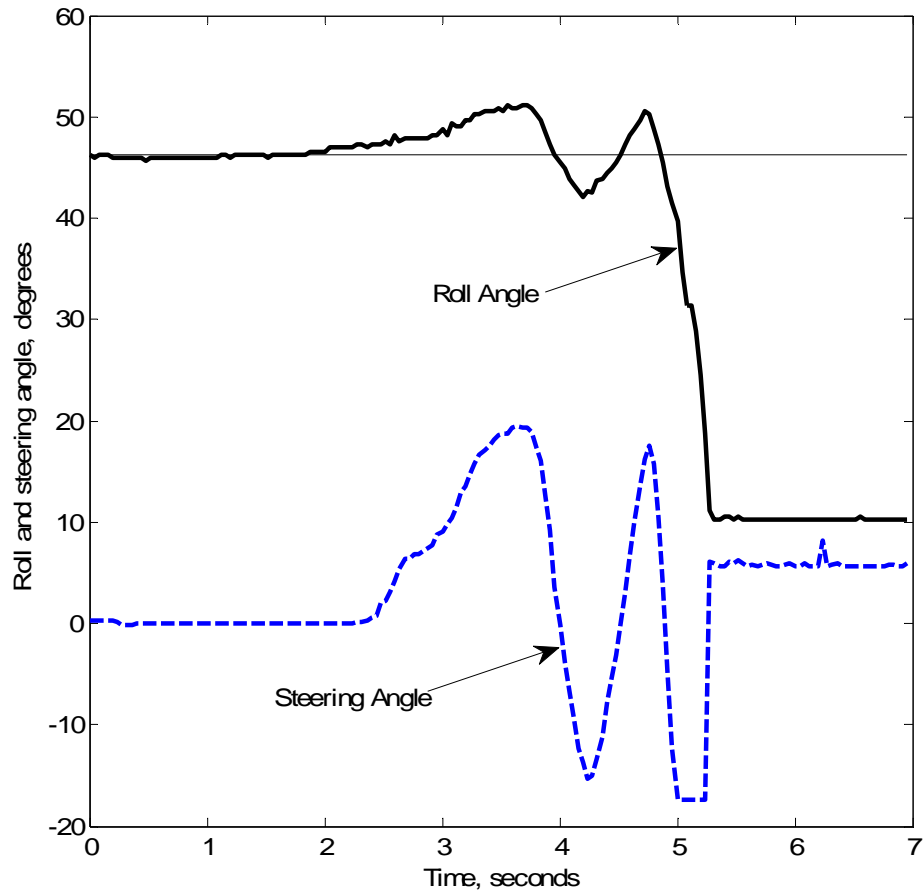


Figure 12: Unsuccessful test run. Steering angle adjusts quickly to roll errors, but vehicle forward velocity was too fast to stabilize.

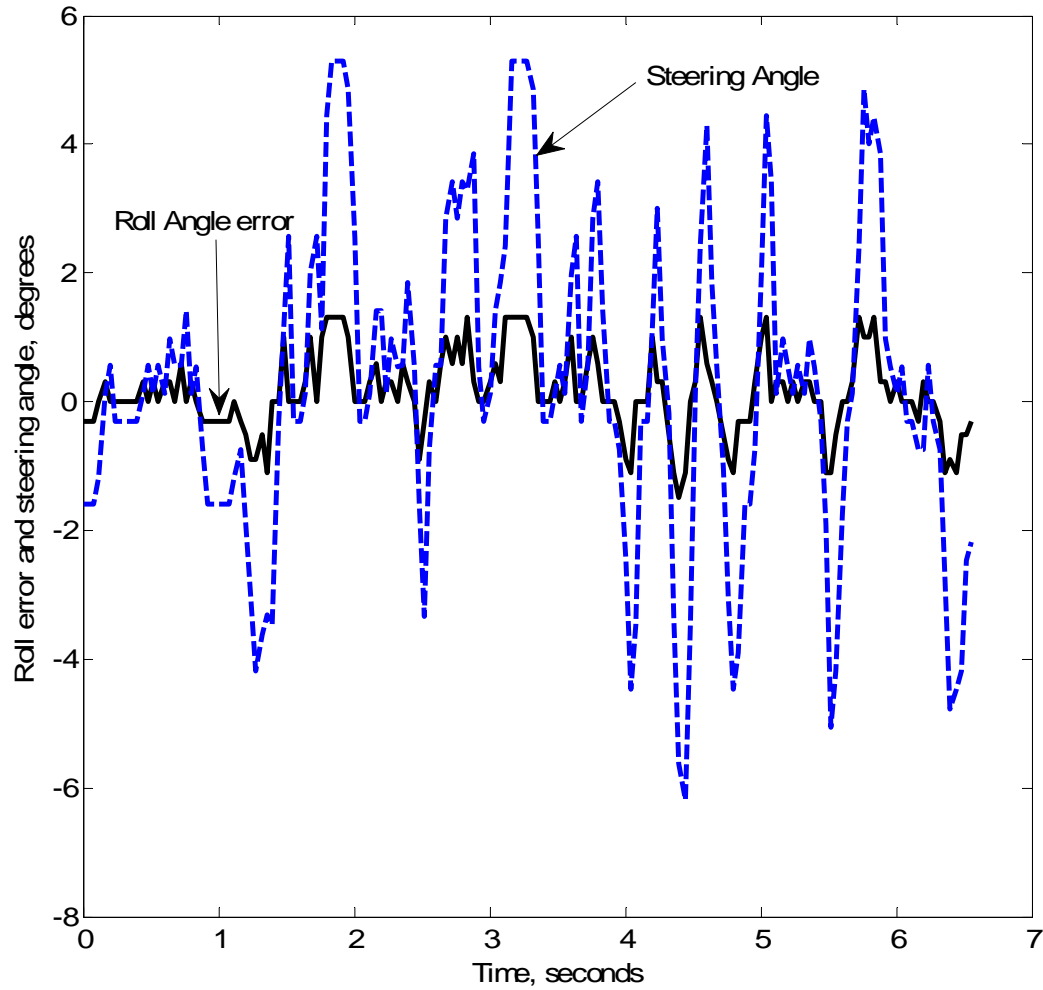


Figure 13: Successful test run with vehicle speed approximately 40 inch/sec. Solid line is roll angle error, dashed line is steering angle.