ANALYSIS OF ROBOTIC VEHICLE STEERING AND CONTROLLER DELAY

KARL N. MURPHY
National Institute of Standards and Technology
Gaithersburg, MD 20899 / USA

ABSTRACT

Robotic vehicles are driving at highway speeds using computer vision and other methods to detect the vehicle's position on the roadway. Several factors affect system performance and stability. In this paper a simple vehicle and path following model is developed. The effect of vehicle understeer, control delay, and other control parameters is investigated and their effect on stability is presented. A method that uses navigation sensors to compensate for computation delay is presented.

INTRODUCTION

Researchers in many countries are developing control systems that can drive robotic vehicles down highways and off road for both civilian and military applications [1]. In most cases the controller somehow derives a desired path and then steers the vehicle in an attempt to follow it. The paths can come from sensing magnets buried in the roadway, from maps derived from navigation data, or from computer vision systems that detect the road directly in front of the vehicle.

At the National Institute of Standards and Technology (NIST) such research is being performed by the Robot Systems Division in support of the Department of Transportation's Intelligent Vehicle Highway System and the Department of Defense's Robotic Testbed programs. One goal is to increase highway safety as the anti-lock brake system (ABS) does today, increase traffic flow, and reduce driver workload if not replace him altogether.

The NIST vehicle, a military HMMWV, has a video camera mounted on the windshield and motors attached to the steering wheel, brake, and throttle. The controller is based on the NIST–developed RCS control architecture. The current system uses the following steps to track the painted stripes on the road and steer the vehicle along the center of the lane. First, edges are extracted from the video image. Edges occur where the brightness of the image changes, such as where the image changes from the gray road to the white stripe. Then, two quadratic curves that represent each lane boundary as it appears in the video image are updated. The system computes the coefficients of the curves using an exponentially weighted, recursive least squares fit that filters out edges caused by shadows and other image noise. Finally, the steering wheel angle that steers the vehicle along the center of the perceived lane is calculated using the pure pursuit method. This method steers an amount proportional to the lateral position of the lane at a given distance in front of the vehicle. The controller cycles at 15 Hz. and can drive the vehicle at 90 km/h (55 MPH) in bright sunlight, on cloudy days, and at night. For more information about the project see [2,3] and for more information about the vision system see [4,5].

As expected, increasing the speed of the vehicle reduces the stability of the road following while reducing the controller delay improves stability. This paper examines

Presented at the Fifth International Symposium on Robotics and Manufacturing (ISRAM '94), Wailea, Maui, HI, August 15-17, 1994.

the effect of vehicle speed, computation delay, and other control parameters such as look ahead distance on system stability.

In this paper we develop a simple model of the vehicle to analyze stability. The next section develops the transfer function for the vehicle and control system. The following section analyzes driving stability. A method that uses navigation sensors to compensate for delay is presented.

PATH FOLLOWING TRANSFER FUNCTION

Consider a vehicle traveling at a constant speed, v, tracking a straight path (the X axis) as shown in Figure 1. The instantaneous curvature, γ, of the vehicle's path is defined as

$$\gamma = \frac{d\theta}{dp} = \frac{1}{r} \tag{1}$$

where θ is the heading and p is the distance traveled on the path. The instantaneous radius of curvature, r, is the inverse. A path that curves to the left has positive curvature, to the right has negative curvature, and a straight path has zero curvature.

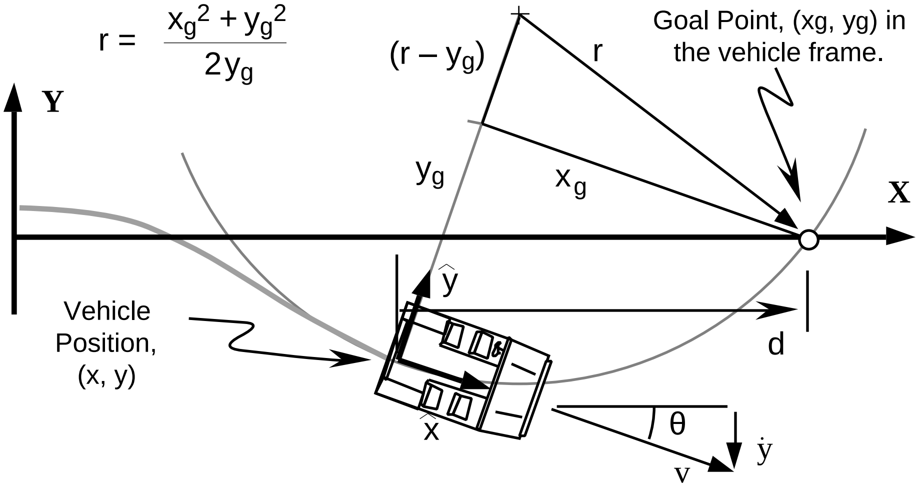


Figure 1. A vehicle tracking a path using pure pursuit steering. The goal point is a point on the path a specified distance, d, ahead of the vehicle. The turn radius, r, and hence the steering angle, is selected to drive the vehicle over the goal point. The goal point, turn radius, and steering angle are repeatedly updated as the vehicle moves. For the special case in this paper, the path coincides with the X axis of the world coordinates.

Assuming small following error in both y and θ, the velocity tangential to the path is proportional to heading

$$\theta = \frac{\dot{y}}{v} \tag{2}$$

and the acceleration tangential to the path is the centripetal acceleration which is proportional to curvature

$$\ddot{y} = \gamma v^2 \tag{3}$$

Thus the vehicle's lateral position, y, is a double integral of the path's curvature.

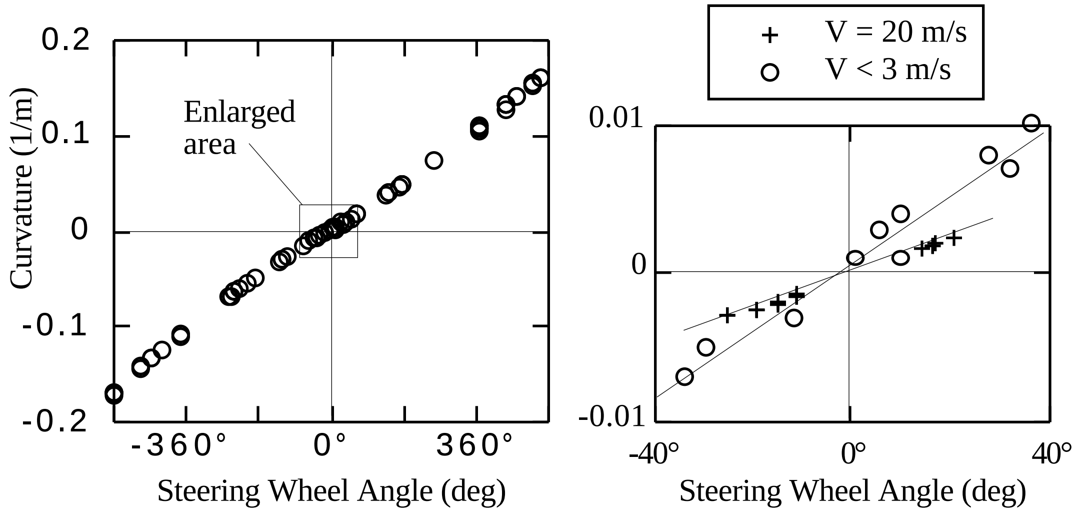
The path's curvature is determined by the steering angle, δ, i.e. the angle that the front wheels make with the body, and varies with surface conditions, vehicle speed, tire stiffness, etc. [6]. This is relation is typically summarized as

$$\delta = L\gamma + K\,\ddot{y} \qquad (4)$$

where L is the distance between front and rear axles, K is the understeer gradient, and $\ddot{y}$ is the lateral acceleration of the vehicle, i.e., the centripetal acceleration.

At low speeds the centripetal acceleration is negligible and curvature is proportional to steering angle. This relationship is shown for the NIST HMMWV in Figure 2 (i). Motivated by this relationship, steering commands for the NIST HMMWV are given as δ/L, in units of curvature, 1/m, rather than the position of the steering wheel in degrees.

At higher speeds, vehicles exhibit understeer when K is positive. As the speed increases, the steering wheel must be turned more to obtain the same turn radius. For most front steer vehicles with two axles, K is independent of steady state speed and turn rate. For the NIST HMMWV, the understeer gradient is 0.8 deg/g as measured at the front wheels. The effect of understeer is shown in Figure 2 (ii).



(i) (ii)

Figure 2. Steering response of the NIST HMMWV. For slow speeds the curvature of the vehicle's path is proportional to the steering angle (i). At higher speeds, wheel slippage and other effects cause the vehicle to curve less than the low speed response (ii).

Even with understeer, the path curvature is still proportional to steering angle, albeit at a reduced gain. Thus, using (3), we rewrite (4) as

$$\gamma = K_{under} \frac{1}{L}\,\delta \qquad (5)$$

where

$$K_{under} = \frac{L}{L + Kv^2}$$

Thus the transfer function of understeer is simply a gain that scales the commanded curvature, δ/L. Note that the gain K_{under}, unlike K, is dependent on velocity. Combining (3) and (5) yield the vehicle's transfer function shown in Figure 3.

Controller Transfer Function

The pure pursuit method, as proposed by Amidi [7], is a very simple and stable steering algorithm. This method steers toward a goal point on the path a set distance in front of the vehicle. The mobility controller repeatedly selects the turn radius so that the vehicle will drive to a goal point on the path a set distance ahead of the vehicle. See Figure 1.

Calculating the turn radius is straightforward. First compute the position of the goal point in the *vehicle frame*, (x_g, y_g). Note the center of the turn circle is (0, r). Solving for r and taking the reciprocal yields the steering curvature for pure pursuit

$$\gamma_{pp} = K_{pp} \frac{2y_g}{x_g^2 + y_g^2} = K_{pp} \frac{2y_g}{d^2} \tag{6}$$

where d is the look ahead distance, K_{pp} is controller gain that is used to counter or augment understeer, K_{under}, and small following errors are assumed.

When the vehicle is tracking the straight path shown in Figure 1, (6) becomes

$$\gamma_{pp} = K_{pp} \left(\cancelto{0}{y_{path}} - \frac{2}{d^2}\, y - \frac{2}{d}\, \theta \right) \tag{7}$$

This has the form of the state space controller [3] where the steering is set proportional to the lateral error, y, and the heading error, θ. The transfer function (in the Laplace Domain) for pure pursuit is obtained by combining (2) and (7)

$$\gamma_{pp} = \frac{-2K_{pp}}{d^2} \left(1 + \frac{d}{v}\, s \right) y \tag{8}$$

The road being followed using pure pursuit steering is detected by the vehicle sensors. Vision based systems often filter image noise using a Kalman filter, an exponentially weighted filter, etc. [1]. In this paper, the detection system will be modeled as a first order filter with time constant τ.

In addition, most vision based systems require substantial computations that take a significant amount of time. This computatinal delay and other system delay is modeled as a pure delay of time T which has the Laplace transform exp(–T s). This is shown in the block diagram in Figure 3.

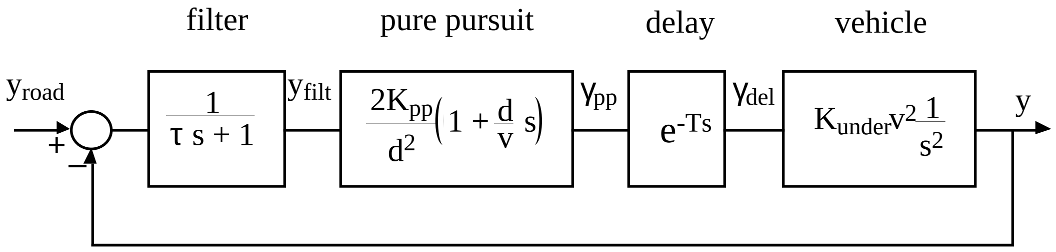


Figure 3. Block diagram for path following using pure pursuit.

STABILITY ANALYSIS

This section examines stability of the system using root locus diagrams and Bode plots. A method to compensate for controller delay and increase stability using navigation sensors is also presented.

Root locus diagrams for path following without delay (T = 0) are shown in

Figure 4. The double integrator in the vehicle transfer function results in two open loop poles at the origin. The pure pursuit portion results in an open loop zero at –v/d. With no filter ($\tau = 0$), the look ahead distance, d, and the steering gain Kpp can be adjusted to provide the desired closed loop poles. The root locus diagram is shown in Figure 4 (i). Note that the system is always stable.

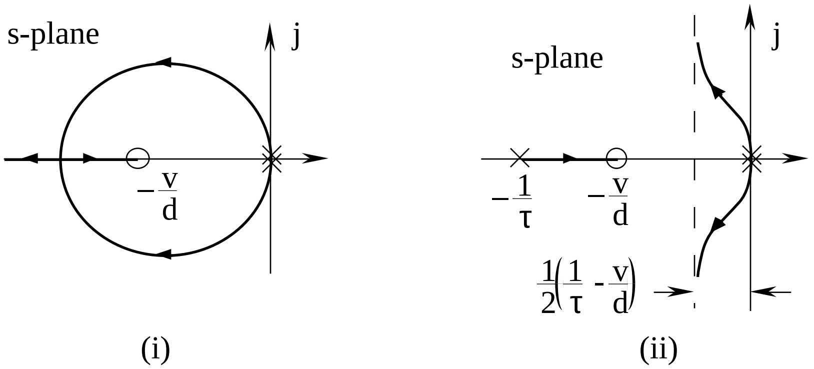


Figure 4. Root Locus Diagrams for zero delay and without the filter (i) and with the filter (ii).

Using the first order filter ($\tau \neq 0$) adds an additional pole at $-1/\tau$, shown in Figure 4 (ii). The system is stable if

$$\frac{1}{\tau} > \frac{v}{d} \qquad (9)$$

If not, the two poles at the origin head immediately into the right half plane.

If the controller has significant time delay ($T \neq 0$) root locus diagrams can't be used. Instead we analyze stability in the frequency domain using the phase shift portion of a Bode plot. See Figure 5.

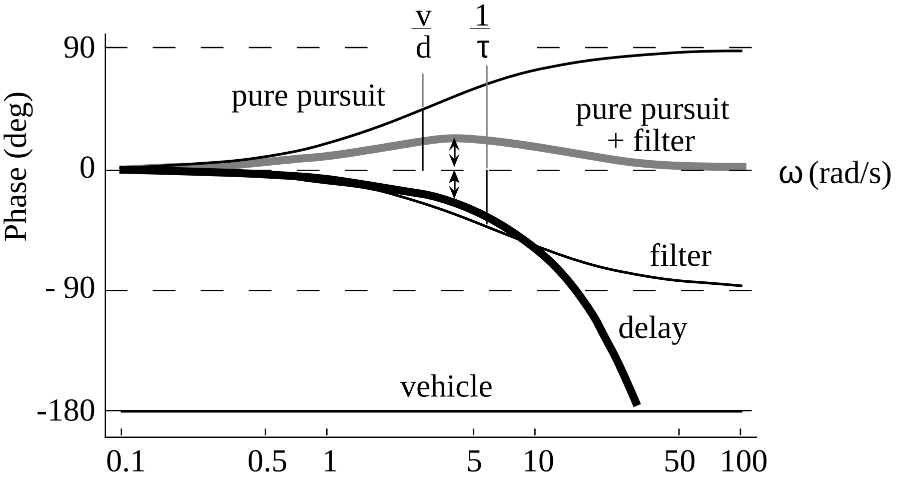


Figure 5. Phase shift portion of the Bode Plot. For stability, the sum of the phases of all the components must be greater than -180° at the crossover frequency. v = 25 m/s, d = 9 m, $\tau = 0.15$ s, T = 0.1 s.

Recall that for Bode plots, s is replaced with $j\omega$ and the magnitude and phase shift of the open loop transfer function are plotted against ω. Adding the phase shift of individual components of the system result in the phase shift of the combined system. A system is stable if the phase shift is greater than –180° at the gain crossover frequency. For the vehicle, the steering gain, Kpp, can be selected to obtain any gain crossover frequency. Thus if the phase is greater than –180° at some frequency, the system can be made stable.

The double integrator of the vehicle transfer function has a constant phase shift of -180°. Thus we start off marginally stable and the controller must provide positive phase for a stable system. Both the filter and delay have negative phase but fortunately, the pure pursuit portion has positive phase.

The break frequency is v/d for the pure pursuit portion and $1/\tau$ for the filter. Their combined phase peaks midway between the two break frequencies with value

$$\varphi_{max} = \sin^{-1} \frac{\alpha - 1}{\alpha + 1} \qquad (10)$$

where α is the ratio of the two break frequencies, $\alpha = d / v \tau$. If this peak phase is greater in magnitude than the phase of the pure delay at the same midpoint frequency, then the overall system will have a phase greater than –180°. In other words, the system will be stable (given the correct Kpp) if

$$\sin^{-1}\left(\frac{d - v\tau}{d + v\tau}\right) > T\sqrt{\frac{v}{d\tau}} \qquad (11)$$

If (11) holds, the system can be made stable. Note stability increases with increasing d and decreasing v, τ, and T. If T = 0, (11) reduces to (9).

Actually, the peak phase of the overall system occures at a frequency slightly less than the midpoint frequency. Thus, if (11) just misses the system might still be able to be made stable. However the destabilizing effects of using a discrete time controller often negates this benefit.

Delay Compensation

The computation delay of the vision system can be effectively reduced by using inertial navigation sensors. Consider one control cycle and the corresponding goal point, just one of a series of goal points on the path selected each cycle. The vision system takes a video frame, selects the goal point, and computes the position of the goal point relative to the vehicle. During this potentially time consuming computation, the vehicle moves forward at velocity v, but the goal point does not. It is a specific point fixed on the path and doesn't move. When the computation is finished the vehicle is closer to the goal point, and most likely at a different heading, than it was at the initial vehicle position when the video frame was taken.

The controller should then compute a steering command for the goal point relative to the current vehicle position, and *not* relative to the initial vehicle position. Inertial navigation sensors can be used to measure the change in position and heading of the vehicle, effectively eliminating the pure delay associated with the vision processing.

REFERENCES

1. H. Schneiderman, J.S. Albus, "Survey of Visual-Based Methods for Autonomous Driving," *IVHS America 1993*, Washington, D.C. (April 1993).
2. M. Juberts, *et al.* , "Development And Test Results for a Vision-Based Approach to AVCS." 26th ISATA meeting, Aachen, Germany. (September 1993).
3. K.N. Murphy, "Navigation and Retro-traverse on a Remotely Operated Vehicle," IEEE SICICI, Singapore. (February 1992).
4. H. Schneiderman, M. Nashman, "Visual Processing for Autonomous Driving." IEEE Workshop: Applications of Computer Vision, Palm Springs, CA. (December 1992).
5. H. Schneiderman, M. Nashman, R. Lumia, "Model-based Vision for Car Following," SPIE 2059 Sensor Fusion VI, Boston, MA. (September 1993).
6. T.D. Gillespie, *Fundamentals of Vehicle Dynamics* (SAE, Inc., PA, 1992)
7. O. Amidi, "Integrated Mobile Robot Control," M.S. Thesis, CMU. (May 1990).