IEEE/RSJ International Workshop on Intelligent Robots and Systems '89, Sep. 4-6, 1989, Tsukuba, Japan

Position Based Path Tracking for Wheeled Mobile Robots

Sanjiv Jeff Singh and Dong Hun Shin
Field Robotics Center
Carnegie-Mellon University
Pittsburgh, Pa 15213

Abstract

This paper discusses two issues, path representation and path tracking, in navigation for wheeled mobile robots. Vehicle and path models are proposed such that steering control can be decoupled from velocity control of the vehicle. "Good" paths are defined, and it is shown that tracking performance is superior when the path is intrinsically easier to track. Two methods of tracking paths are proposed: a derivative of standard inverse kinematics methods and a polynomial curve fit in error space. Simulation results are compared for different paths and tracking schemes.

Introduction

Robot vehicle navigation typically consists of two sub problems, *path generation* and *path tracking*, which are solved separately. Path generation uses intermediate goals from a high level planner to generate a detailed path for the robot to follow. There is a distinct trade-off between simplicity of representation of such plans and the ease with which they can be executed. For example a simple scheme is to decompose a path into straight lines and circular curves. However, such paths cannot be tracked precisely simply because of discontinuities in curvature at transition points of segments that require instantaneous accelerations [1,2]. Similar methods have been used in several robot systems [3,4], where the robot plans its motion through analysis of a video image of the road ahead; an arc path is planned and the robot begins to traverse it. The robot compensates for tracking errors by planning a new arc and issuing the corresponding command before the current arc is fully traversed. Such methods have concentrated on vision processing and have succeeded at low speeds without concern for physical control of the vehicle. Other schemes have been suggested to ensure ease of tracking [5,6,7]. We will compare the performance of the proposed tracking schemes using two different kinds of paths.

Following path generation, path tracking takes, as input, the detailed path generated and controls the vehicle to follow the path as precisely as possible. It is not enough to simply follow a premade list of steering commands because failure to achieve the required steering motions exactly, results in steady state offset errors, which accumulate in the long run [8]. This paper presents two methods that use global position feedback to compensated for less than ideal actuators.
Both methods deviate from traditional robot control schemes used for similar problems in which a time history of position (*a trajectory*) is implicit in the plan specified to the robot. The methods proposed are appropriately labeled "*path*" tracking in that the steering motion is time decoupled, that is, steering motions are directly related to the geometric nature of the specified path, making speed of the vehicle an independent parameter. (Note: this analysis applies to steered wheel robots and not to skid-steer and omni-directional robots.) Dynamic characteristics of inertia and slippage are not modeled explicitly in the control scheme; a simpler kinematic approach is used instead.

Vehicle Model

For our analysis, we use a bicycle model as an archetype. The guide point of the vehicle, which is guided along the reference path, is chosen as the midpoint of the rear axle, as in Figure 1, resulting in the following advantages:

- The steering angle at any point on the path is determined geometrically, independent of the speed, in the following manner:

$$\tan\phi = \frac{l}{r}; \qquad \phi = \tan^{-1} c\,l \tag{1}$$

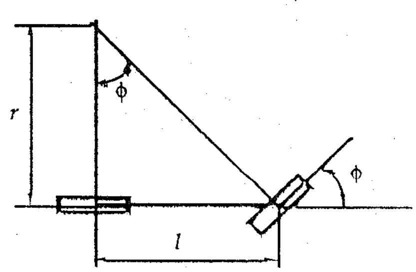


Figure 1: Geometry of a Bicycle

where l is the wheelbase of the vehicle and c is the curvature of the path. Also, the angular velocity of the driving rear wheel is determined only by the vehicle speed (v) and the radius of the wheel (R_w):

$$\omega = \frac{v}{R_w} \quad (2)$$

If the guide point is placed elsewhere, expressions for ϕ and ω are more complex than (1, 2). The reference steering angle and angular velocity of the driving wheel must be obtained by numerical integration [6].

- The vehicle to follow the minimum turning radius for the maximum steering angle [6]. In other words, the peak steering angle is smaller than than for any other choice of guide point.
- The heading of the vehicle is aligned with the tangent direction of the path. This gives a more reasonable vantage point for a vision camera or a range scanner mounted at the front of the vehicle, as in Figure 2, and a smaller area is swept by the vehicle.

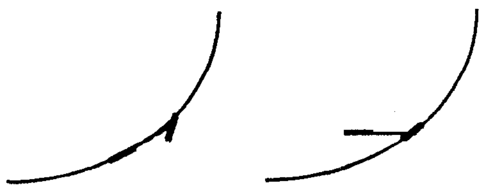


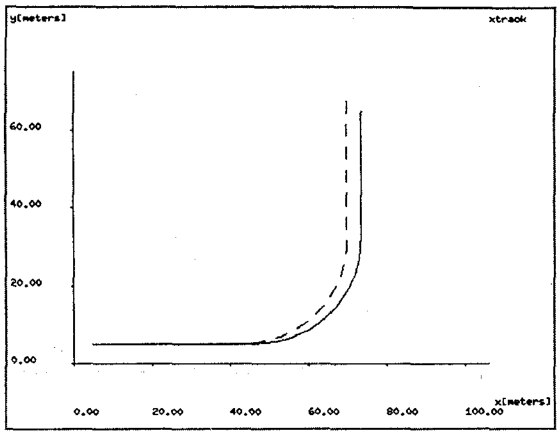
Figure 2: The heading alignment along a path

For the guide point at the center of the rear axle, certain paths will require infinite acceleration of the steering wheel. This is not the case if the guide point is moved from the rear axle, for example, to the front wheel. However, this choice loses the advantages discussed above.

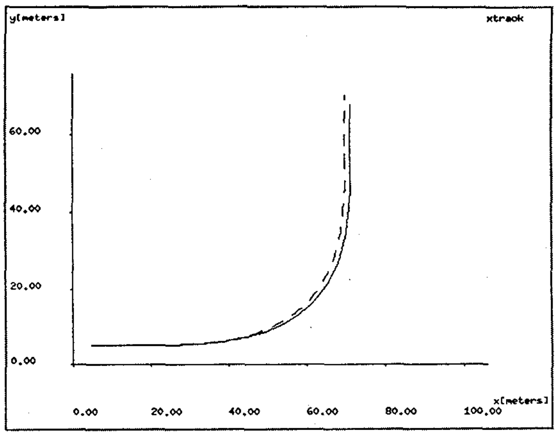
Path Representation

The response of an autonomous vehicle in tracking a path depends partly on the characteristics of the path. In particular, continuity of the curvature and the rate of change of curvature (*sharpness*) of the path are of particular importance, since these parameters govern the idealized steering motions to keep a vehicle on the desired path. In the case where a path is specified as a sequence of arcs and lines, there are discontinuities of curvature at the point where two arcs of differing radii meet. Discontinuities in curvature are troublesome, since they require an infinite acceleration of the steering wheel. A robot traveling through such transition points with non-zero velocity will experience an offset error along the desired path.

In general, if we define a *posture* as the quadruple of parameters- position, heading, and curvature (x,y,θ,c), then we will require a path to be posture-continuous. In addition, the extent to which steering motions are likely to keep a vehicle on a desired path correlates with the linearity of sharpness of the path, since linear curvature along a path means linear steering velocity while moving along the path. For example, certain spline curves [6] are candidates because they guarantee posture continuity. However, these spline curves do not guarantee linear gradients of curvature along curves. Clothoid curves [5] are known to have the "good" property that their curvature varies linearly with distance along the curve. For our purposes, we will use paths composed of (a) arcs and straight lines and (b) clothoid segments.



(a) arc path



(b) clothoid path

Figure 3: Open-loop Responses

[7] compares the response of a vehicle to different paths. Simulation results show that in the open loop case, a path that has discontinuities in curvature results in larger steady state tracking errors. This is particularly the case when the actuators are slow. The robot is made to follow a set of steering commands (obtained from the path specification) without trying to compensate for tracking errors. Open loop response of a vehicle following a path composed of (a) arcs and lines and (b) clothoid segments can be seen in Figure 3. Reference paths are denoted by dashed lines and actual paths are denoted by solid lines. The dynamic response of the robot is modeled as a first order lag with hard limits on acceleration and hysteresis.

Path Tracking

For an autonomous vehicle to track specified paths it is necessary to generate reference inputs for the vehicle servo-controllers. Thus, path tracking can be formulated as a problem of obtaining a reference steering angle and a reference speed for the next time interval in order to get back to the reference path ahead from the current deviated position. Nelson [8] used a simple proportional control law to guide an indoor tricycle robot that operates at speeds on the order of a few inches per second. Given a prespecified steering angle, driven wheel velocity values (ϕ_{old}, ω_{old}), and error components, the command steering and driving inputs are computed as:

$$\phi = \phi_{old} + K_1 e_n + K_2 e_h$$

$$\omega = \omega_{old} + K_3 e_t + K_4 e_v \qquad (3)$$

where, e_n is a lateral position error, e_t is a tangential error, e_h is a heading error, e_v is a speed error, and K_1, K_2, K_3, K_4 are control gains. In the same way, Kanayama [9] applied a PID control law to the path tracking of his differential wheeled mobile robot.

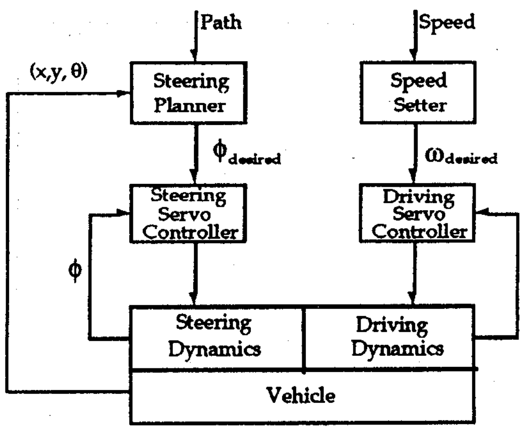


Figure 4: Path Tracking Structure of Autonomous Vehicle

The proposed path tracking scheme is distinct in the following ways:

- *Global position feedback*: the path to be tracked is specified in cartesian coordinates. If the control scheme consists of only servo-control to reference steering commands, vehicle position and heading errors accumulate because position and heading result from integrating the whole history of steering and driving,[1] it is necessary to feed back vehicle position and heading in cartesian space. Consequently, reference inputs to the servo-controllers are generated in real time, based on position feedback, as in Figure 4.
- *Separate steering and driving control*: since we locate the guide point on the center of the rear axle, steering and driving reference inputs are computed from the given path and vehicle speed, respectively, as in Eq 1, 2. This enables easy integration of path tracking with other modules like collision avoidance.

In the setup shown in Figure 4, the inner loop is executed on the order of 10 ms, while the outer loop is closed at the rate of 0.25-0.5 second. The following procedure is used to close the loop on position(Figure 5). After sensing the current posture ($P_{a,k}$), the posture at the end of the current time interval ($\underline{P}_{a,k+1}$) is expected. Then, the desired posture at the end of the next time interval ($P_{d,k+2}$) is computed and the reference steering angle between $\underline{P}_{a,k+1}$ and $P_{d,k+2}$ are determined.

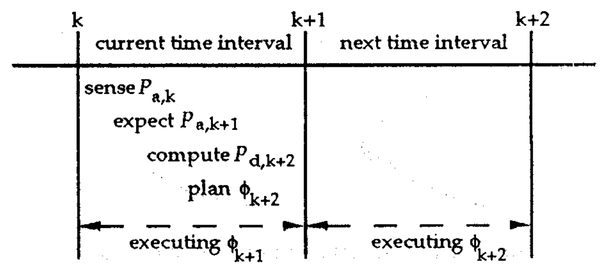


Figure 5: Timing of Steering Planning

Ideally, $\underline{P}_{a,k+1}$ should be computed from $P_{a,k}$, ϕ_{k+1}, the speed of the robot, planning time interval, and robot dynamics. However, since this is too time-consuming, $\underline{P}_{a,k+1}$ is computed approximately as the following:

$$e_k = P_{a,k} - \underline{P}_{a,k}$$

$$\underline{P}_{a,k+1} = P_{d,k+1} + e_k \qquad (4)$$

Two methods are proposed for planning steering motions (ϕ) such that the vehicle converges back to the path: an inverse kinematics method and a polynomial curve fit in error space.

Inverse Kinematics Method

The Inverse Kinematics method is a derivative of standard inverse kinematic methods for robot manipulators. The inputs to the steering planner in Figure 4 are specified in global coordinates (x, y, θ, c), while the output is a local coordinate ϕ. It is, therefore, natural to use

[1] This is a characteristic of a non-holonomic system, where a geometric constraint cannot be expressed analytically as an equation relating position and time. The most important type of non-holonomic constraint is one which can only be expressed as an equation relating infinitesimal variations to coordinates. In systems that have such constraints, the number of degrees of freedom is less than the number of independent generalized variables in a complete set.

inverse kinematics for the steering planning, since this method is often used to relate global and local coordinates. However, kinematics for vehicles are known only in the form of differential equations.[1] For example, if we use a bicycle model for the vehicle, we have:

$$\phi = \tan^{-1} l \frac{\dot{\theta}}{\dot{\eta}} \qquad (5)$$

where η is the coordinate in the heading direction. Since (5) is in the form of desired velocities, the following equation is used as an approximation:

$$\phi = \tan^{-1} l \frac{\Delta\theta/\Delta t}{\Delta\eta/\Delta t} = \tan^{-1} l \frac{\theta_{d,k+2} - \theta_{a,k+1}}{\eta_{d,k+2} - \eta_{a,k+1}} \qquad (6)$$

where $\theta_{d,k+2}$ and $\eta_{d,k+2}$ are the desired heading and the longitudinal coordinate at the end of the time interval, respectively, and $\theta_{a,k+1}$ and $\eta_{a,k+1}$ are currently sensed values.

Quintic Polynomial Approach

The Quintic Polynomial method replans a simple, continuous path that converges to a desired path in some look-ahead distance and computes a steering angle corresponding to the part of the replanned path to be followed for the next time interval.

If the desired path is considered as a continuous function of position and the vehicle is currently at P_a, an error vector can be calculated (Figure 6) that represents error in the distance transverse to the path (ε_o) relative to P_o, in heading (β_o), and in curvature (γ_o). If the vehicle is to be brought back onto the specified path within distance L (measured along the reference path), six boundary conditions can be stated corresponding to the initial errors and to zero errors at P_L.

$$\begin{aligned} \varepsilon(P_o) &= \varepsilon_o \;; & \varepsilon(P_L) &= 0 \\ \beta(P_o) &= \beta_o \;; & \beta(P_L) &= 0 \\ \gamma(P_o) &= \gamma_o \;; & \gamma(P_L) &= 0 \end{aligned} \qquad (7)$$

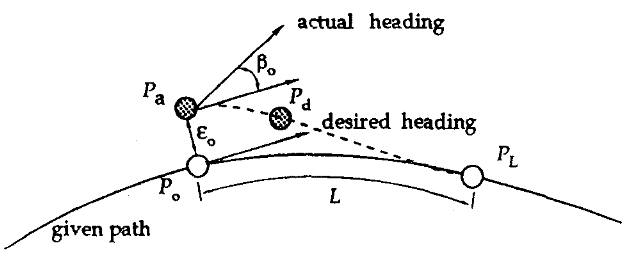


Figure 6 : Computing an Error Vector

A quintic polynomial can be constructed to describe the replanned path (in error space) as follows:

$$\varepsilon(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4 + a_5 s^5 \qquad (8)$$
$$\text{where } s \in [0, L]$$

The expression for $\varepsilon(s)$ gives the error along the replanned path from P_o to P_L. Variation in the steering angle from the replanned path (or in error space) is computed from the second derivative of error function $\varepsilon(s)$. Then, curvature along the new path can be computed as:

$$C_{new}(s) = C_{old}(s) + \frac{d^2\varepsilon(s)}{ds^2} \qquad (9)$$

The reference steering angle along the new path can be converted from curvature using (1). Since this procedure is executed at every planning interval, the entire new path back to the reference path is not required. Only the steering angle for the next time interval is computed from the curvature at the point on the new path that can be achieved in the next time interval.

The look-ahead distance, L , is a parameter that can be used to adjust how rapidly the vehicle steers to converge to the desired path. Additionally, better performance is obtained if L is chosen proportional to the vehicle speed because for small values of L , the vehicle oscillates around the path, while for large values of L the variation introduced by the quintic polynomial is small enough that the tracking performance is poor.

Results and Conclusion

Computer simulation was carried out to test the proposed methods. Figure 7 shows the performance of a robot following a specified path without compensating for tracking errors in position and heading.

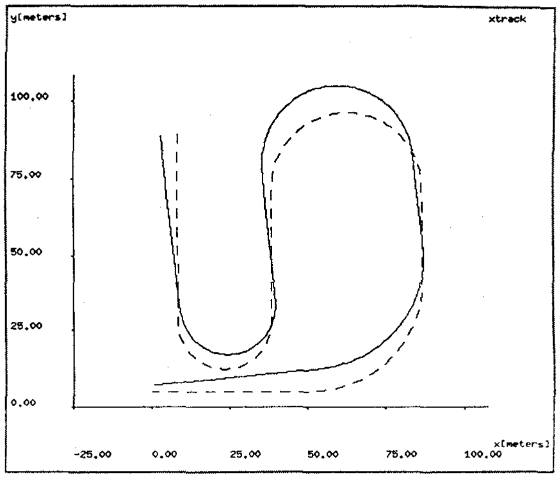
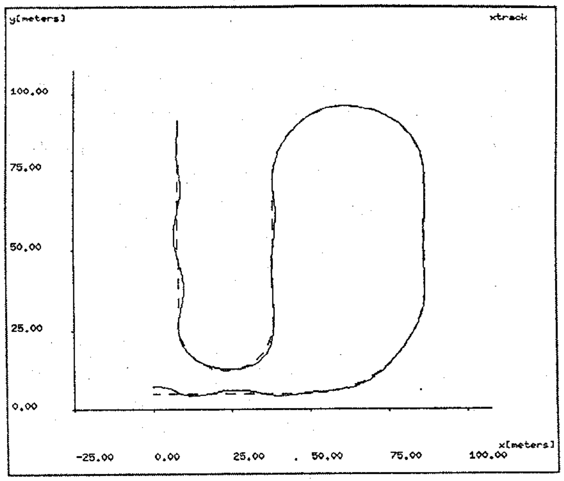


Figure 7: Open-loop Response with Initial Errors in Position and Heading

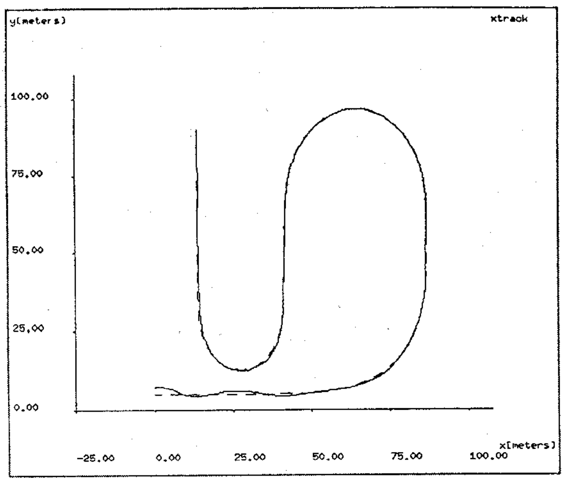
Figure 8 shows the performance of the Quintic method using (a) an arc path and (b) a clothoid path. Larger

[1] Non-holonomic constraints

deviations from the specified path are noted in the former case. This difference becomes even more noticeable when the speed of the vehicle is doubled from 5 m/s to 10 m/s as shown in Figure 9.



(a) arc path



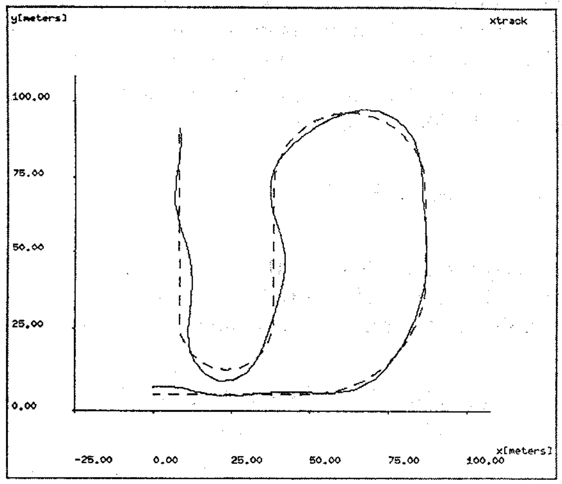
(b) clothoid path

Figure 8: Closed-loop Response Using the Quintic Method at 5 m/s

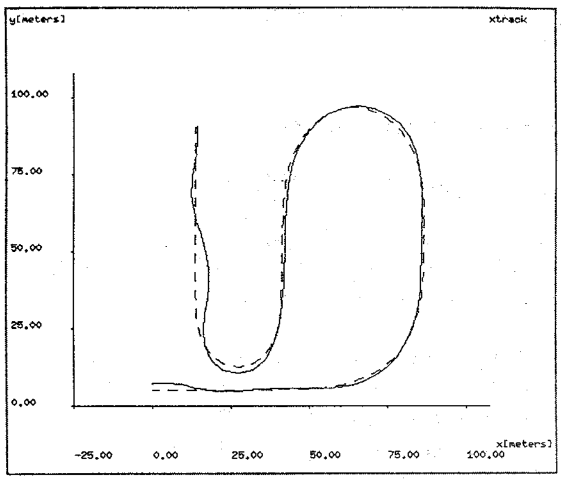
In Figure 7, 8, and 9, the following parameters were used:

Steering planning frequency	2	Hz
Hysteresis in steered wheel	2	deg
Time constant of steered wheel	0.5	s
Max. steering acceleration	100	deg/s^2
Max. steering velocity	25	deg/s^2

The parameters used above were deliberately exaggerated to show path deviations. In reality, the actuators are faster and the paths have milder curvatures.



(a) arc path

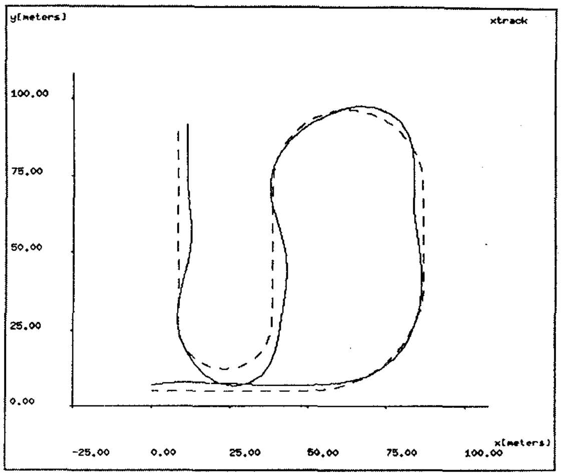


(b) clothoid path

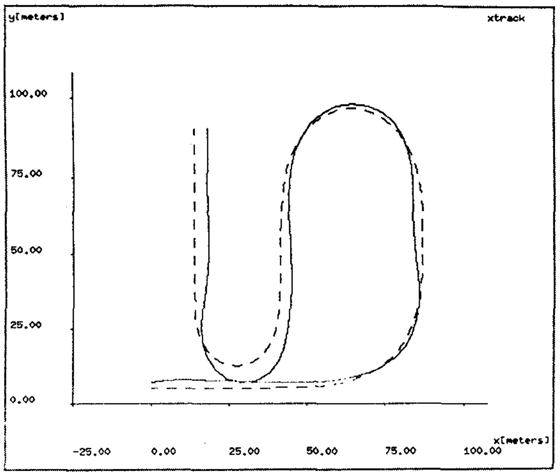
Figure 9: Closed-loop Response Using the Quintic Method at 10 m/s

Both Quintic method and Inverse Kinematics method showed comparable results in following arc and clothoid paths at a speed of 5 m/s. However, at 10 m/s, the difference in tracking performance between the two paths with either method was clear. Results of Inverse Kinematics method are shown in Figure 10.

Simulation results indicate that path tracking schemes perform better when the path specified is intrinsically easier to track. This is especially the case when the steering actuators are slow compared to the speed of the vehicle. Other vehicle characteristics, like steering response, steering backlash, vehicle speed, sampling, and planning time intervals significantly affect vehicle performance. As expected, at higher vehicle speeds, faster and more accurate actuators are necessary, if sensing and planning time intervals are kept constant.



(a) arc path



(b) clothoid path

Figure 10: Closed-loop Response Using the Inverse Kinematics Method at 10 m/s

One reason that the inverse kinematics method shows poorer performance is that it does not correct for errors in the transverse direction of the vehicle in the case where the heading and curvature errors are zero. In this case, the robot travels parallel to the desired path with steady-state lateral errors, as in Figure 10.

The advantage of the quintic polynomial method is that it is simple, and reference steering angles can be computed very easily. However, since there is no consideration of vehicle characteristics (mass, inertia, time delays, vehicle ground interaction, etc) in the control scheme, stability and convergence cannot be guaranteed. The parameter *L* (look-ahead distance) can be adjusted to modify response of the vehicle, but the value of *L* must be chosen based on trial and error. This scheme has been implemented on the NavLab[10] a robot testbed at CMU, and good results have been obtained at speeds up to 25 Km per hour.

In conclusion, this paper has pointed out some considerations relevant to robot path tracking. Vehicle and path models were proposed such that steering prescriptions can be obtained directly from the path alone, making it possible to decouple steering control from velocity control. Two methods of tracking specified paths were presented and computer simulation results were compared. The quintic method was shown to perform better in a variety of conditions. Since the proposed method idealizes the vehicle as massless entity with perfect response, future work will attempt to incorporate vehicle physics into the control scheme.

References

1. T. Hongo, H. Arakawa, G. Sugimoto, K. Tange and Y. Yamamoto, "An Automatic Guidance System of a Self-Controlled Vehicle--The Command System and Control Algorithm," *Proceedings IECON. 1985*, MIT Press, 1985
2. Y. Goto and A. Stentz, "The CMU System for Mobile Robot Navigation", *IEEE International Conference on Robotics and Automation*, 1987, pp99-106.
3. T. Kanade, C. Thorpe and W. Whittaker, "Autonomous Land Vehicle Project at CMU", *ACM Computer Conference*, February 1986.
4. E. D. Dickmanns, "Vehicle Guidance by Computer Vision," Proc. Intl. Workshop on High Precision Navigation, May 1988, pp3.2-1 to 3.2-11
5. Y. Kanayama and N. Miyake, "Trajectory Generation for Mobile Robots", *Robotic Research: The Third International Symposium on Robotics Research*, Gouvieux, France, 1985, pp. 333-340.
6. N. L. Winston, "Continuous Steering Function Control of Robot Cart", *IEEE Transactions on Industrial Eletronics*, Submitted for publication 1988, AT&T Bell Laboratories.
7. D. H. Shin and S. J. Singh, "Path Generation for Robot Vehicles Using Composite Clothoid Segments", Tech. Report, Robotics Institute, Carnegie-Mellon University, in press 1989.
8. N. L. Winston and I. J. Cox, "Local Path Control for an Autonomous Vehicle", *IEEE Intl. Conference on Robotics and Automation*, Philadelphia, 1988, pp1504-1510.
9. Y. Kanayama, A. Nilipour and C. A. Lelm, "A Locomotion Control Method for Autonomous Vehicles," *IEEE Intl. Conference on Robotics and Automation*, Philadelphia, 1988, pp1315-1317.
10. K. Dowling, R. Guzikowski, J. Ladd, H. Pangels, S. Singh, & W. Whittaker, "NAVLAB: An Autonomous Navigation Testbed", Technical Report, Carnegie Mellon University, CMU-RI-TR-87-24