Robotica (1992) volume 10 pp 539–554

Explicit path tracking by autonomous vehicles

Dong Hun Shin, Sanjiv Singh, and Ju Jang Lee

Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa 15213 (USA)

(Received in Final Form; December 2, 1991)

SUMMARY

We have suggested a novel approach to autonomously navigate a full sized autonomous vehicle that separately treats vehicle control and obstacle detection. In this paper we discuss the vehicle control that has enabled our autonomous vehicle to travel at speeds upto 20 mph. We point out the limitations of existing schemes that restrict their consideration to kinematic models and show that it is possible to obtain an increase in performance through the use of approximate dynamical models that capture first-order effects. Our approach combines such a modeling philosophy with accurate feedback in world coordinates from sensors that have only recently become available. Experimental results of our implementation on NavLab, a modified van at CMU, are presented.

KEYWORDS: Path tracking; Autonomous vehicles; Obstacles; Dynamic models

1. INTRODUCTION

Early work in the area of autonomous mobile robot navigation[1,2] has emphasized symbolic approaches for interpreting sensed data of the world. In such approaches, cycle times of several seconds for perception and simple low speed vehicle control have been sufficient. Further work has concentrated on improving mobile robot navigation by closing the sense-plan-act cycle at shorter intervals. However, this paradigm of mobile robot navigation has been limited in the outdoors by the competence and speed of sensing, and, the diversity of real world conditions.

Recently, inertial sensors and satellite positioning systems have emerged as accurate devices with high data rates, providing position sensing data outdoors and are now widely used for aircraft and ship navigation. They make it possible for autonomous vehicles to obtain rapid feedback in world coordinate during outdoor operation. This sort of position based navigation is particularly useful in scenarios where it is not possible to distinguish explicit features such as road edges that can be tracked. We have proposed a paradigm for autonomous robot navigation that separately considers the issues of vehicle control and obstacle detection. In our scenario, an explicit path composed of a sequence of position tags is given to the vehicle. The vehicle travels along this path till it finds an obstacle. In the case, an obstacle is found, the vehicle stops, and replans its path around the obstacle to return to the specified path. In this paper we will discuss the issues of vehicle modeling and control that enable the vehicle to travel along the specified path at high speeds. Obstacle detection, the process that is responsible for bringing the vehicle to a halt upon detection of an obstacle is separately discussed in reference 3, while obstacle avoidance, the process of planning a path around the obstacle is discussed in reference 4.

Among existing path tracking approaches, some methods[5,6] continuously generate paths that converge to the reference path from the deviated vehicle position; the generated paths are converted into steering angles and wheel velocities using simple vehicle kinematics. Other methods[7,8] obtain steering angles by multiplying gains to vehicle heading and position errors. These gains are chosen by trial and error until satisfactory results are produced. Both these schemes are simple but suffice only for low performance, since the first scheme assumes the vehicle dynamics are perfect while the second scheme lacks generality. In pursuing a dynamic model based approach, Muir[9] has developed a version of Newton–Euler recursive formulation of manipulator dynamics for the dynamic modeling of mobile robots. Full-sized mobile robots – computer controlled cars, buses, and trucks – contain a number of subsystems, such as hydraulic power steering, geared transmission, combustion engines, and tires, which exhibit non-linear and time-varying behavior of different types. This makes it difficult to justify such a model-based scheme that treats the robot as a chain of rigid bodies as is commonplace in the treatment of manipulators. In addition, the non-holonomic constraints of wheeled vehicles make the equations of motion very complex, and it is very difficult to accurately model the interaction between the ground and the vehicle[10,11]. We suggest that modeling vehicle dynamics at such a level of detail is not practical for the control of full-sized autonomous vehicles.

The proposed scheme further separates the steering and the velocity control of the vehicle through a judicious choice of a guide point, the point on the vehicle that is guided over the reference path. Thus we are able to reduce the navigation problem to one of planning steering motions that will keep the vehicle on the specified path. Vehicle speed is independantly decided by several factors like curvature of the path, and proximity to possible obstacles. We circumvent the explicit modeling of vehicle dynamics and the interaction with the ground by identifying first order effects that can be obtained through relatively simple experimentation. This model of the vehicle dynamics is incorporated in a feedforward module which is combined with a more traditional feedback scheme to select steering motions at

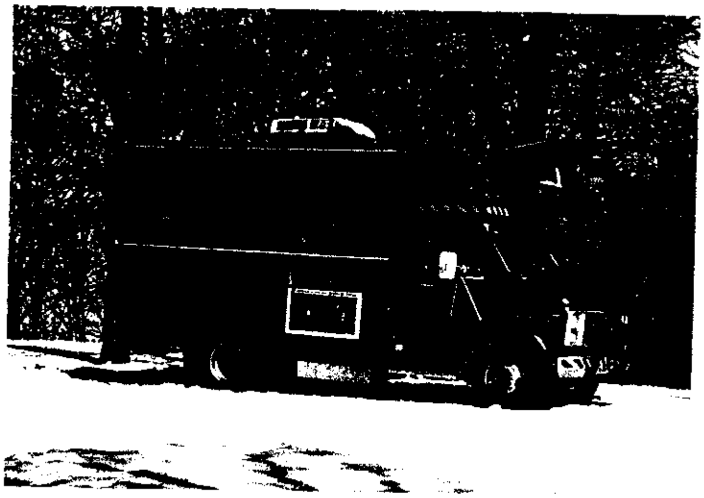

Fig. 1 Navlab: autonomous vehicle developed at Carnegie–Mellon Univ

a subsecond interval We have implemented this methodology on the Navlab[12], a testbed vehicle at Carnegie–Mellon University, and have been able to obtain autonomous navigation at speeds upto 20 mph

The following section formulates the path tracking control problem and Section 3 presents the proposed method. In Section 4, the performances of the proposed method are exhibited utilizing computer simulations and actual vehicle implementation where an inertial navigation system in a testbed vehicle in Figure 1 is used to sense the vehicle position. Results are summarized and directions for future research are discussed in Section 5

2. PROBLEM FORMULATION

The following assumptions for path tracking by a mobile robot distinguish the class of problems to which the path tracking methodology applies.

1) The robot is full-sized conventionally steered vehicle, such as a car or truck. Thus, it has two actuation trains: steering and propulsion.
2) The robot moves on a quasi-planar surface.
3) Global position, orientation, and velocity of the robot are sensed.

2.1 Problem reduction

Control modifies the input–output behavior of a system to achieve a desired response Good control methodology reflects and exploits the characteristics of the system behavior The path tracking problem is approached first by formulating a control problem with the consideration of characteristics of conventionally steered vehicles; non-holonomic constraints and kinematically decoupled propulsion and steering The kinematic bicycle model reveals two main characteristics that impact path tracking control[13]:

- For such a non-holonomic system that is constrained by non-algebraic equations describing the relationship between the steering angle and vehicle position, the vehicle position error cannot be compensated only by steering feedback control. Thus, the control scheme must be based on the global position feedback Typically, sampling rates for global position feedbacks are very low; the resulting closed-loop system succeeds only at a low frequency. This degrades the stability of the system, as well as its capability to reject disturbance.
- The center of the rear axle is selected for the position of the guide point[13] Consequently, steering and propulsion are kinematically decoupled in the following manner:

$$\tan\phi = \frac{l}{r}; \quad \phi = \tan^{-1} cl \tag{1}$$

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):

$$\omega = \frac{v}{R_w} \tag{2}$$

In other words, geometric path following relies only on steering Vehicle speeds rely purely on propulsion.* In order to overcome the former and exploit the latter, the following three strategies are used:

(1) *Path tracking*: geometric "path" tracking is pursued rather than trajectory tracking, which pursues a "time history of position." In the typical manipulator tracking problem, a trajectory (time history of position) is planned off-line from a given geometric path, and then is followed in real time

---

* Of course steering and propulsion are weakly coupled dynamically. But the practical implication is only that slightly different force/torque is required to obtain the desired steering motion depending on propulsion motion.

For mobile robots path tracking is appropriate for the following reasons:

- When a human drives a car along a road, he or she cares most about keeping a car within the road boundaries. Vehicle speed is not changed to follow a time history of vehicle positions. Speed is changed in response to some higher level characteristics such as a road curvature, proximity of obstacles, or conditions of the road, vehicle states, and the human driver.
- Vehicle tracking is more concerned with lateral errors from a reference path than with longitudinal errors along the path. Trajectory tracking must concern itself with the coupling between both types of error and will, therefore, be more difficult to accomplish. For example, a speed error, such as loss of traction from a disturbance, will result in accelerated propulsion and jerky steering motions in order to recover.
- Roadworthy vehicles have more limited acceleration than electrically driven robots and usually require a multi-step transmission for vehicle speed change. However, trajectory tracking requires frequent, responsive changes of vehicle speed to compensate for longitudinal error, whch is not desirable for tracking a roadworthy vehicle.
- Trajectory planning is usually a separate task; path tracking is less complex and can be performed 'on-line'.

(2) *Reduction to steering problem*: By locating the guide point on the center of the rear axle, the path determines only steering motion for tracking, independent of vehicle speed. Since the solution of path tracking for mobile robots does not prescribe speed, the path tracking problem is then reduced to steering along a given path.

(3) *Nested feedback control loops*: When a path to be tracked is specified by a series of points given in global coordinates, the control of a vehicle should be based not only on actuator coordinates but also on feedback of global position. If closed loop control of steering and propulsion is used alone (without global position and heading feedback information like the outer feedback loop in Figure 2) in an attempt to follow this prespecified path, vehicle position and heading errors will accumulate. This is because position and heading result from integrating the entire time history of steering and propulsion. There is no means to determine, and thus regulate, errors in heading and position, even though it is possible to regulate errors in steering.

For example, imagine that a step input is given to the steering system. After the transient response has subsided, the steering mechanism will have reached the desired steering angle. However, in failing to track the step input exactly, offset errors have been introduced in the resultant path, which will have the required curvature but will be parallel to the reference path (Figure 3). Alternately closed-loop control, based only on global coordinates, is not desirable, either. In this case, actuators would be servo-controlled based on global errors, as are global control schemes for manipulators. The resulting system would succeed only at low frequency, because vehicle position estimates are only possible at slow rates, either by dead reckoning or direct position sensing. This would, in general, degrade the stability and disturbance-rejection capabilites of the system. Consequently, it is necessary to feedback global vehicle position and heading, in addition to servo-control feedback.

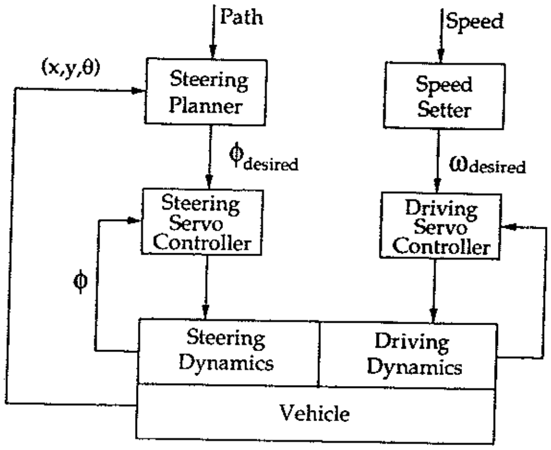

Fig. 2. Nested feedback control loops

The advantages of geometric path tracking, a nested feedback loop, and kinematically decoupled steering and propulsion reduce the path tracking problem to a tractable steering planning control problem and actuator servo control. In other words, for a mobile robot to track paths satisfactorily, it is necessary to generate steering and propulsion commands for the servo-controllers with global position feedback and to execute these commands at the real time servo level. Since servo-control is already a mature technology and steering decides how well the path is tracked, this research puts an emphasis on *real-time steering command generation* which is called *steering planning*.

2.2 Preview control and partitioned scheme

In general tracking control, the problem of generating the control inputs U_i has both feedforward and feedback compensation:

$$U_i = R_i + Ke_i \tag{3}$$

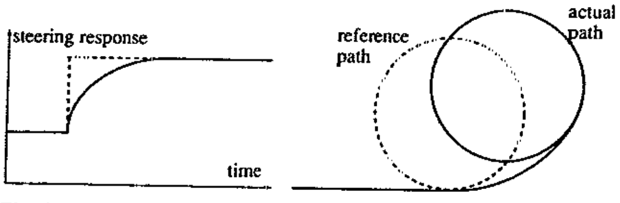

Fig. 3. Steering response and tracking a path

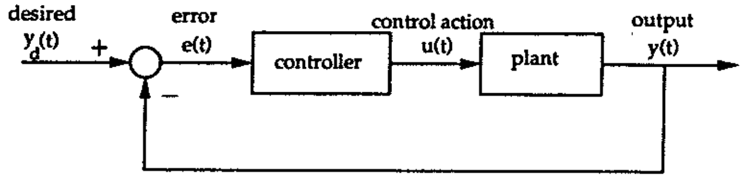

Fig 4 Conventional feedback control

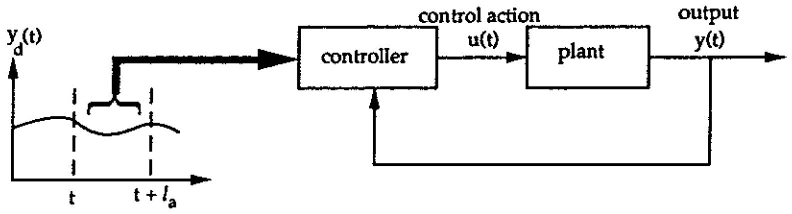

Fig 5. Preview control

R_i denotes the feedforward compensation, e_i is the error between the reference input and actual state of the system, and K is a gain. The Ke_i term denotes an error feedback compensation; for example, it can be any type of Proportional-Integral-Derivative (PID) action. A feedforward compensation is, for example, the rejection of expected disturbances to the system or open-loop control through the computation of the inverse dynamics.

The typical way to solve the tracking problem is to formulate it as a regulator problem and to change its references as the desired input states change. Then, the method is basically a feedback scheme which stabilizes *bounded input and bounded output*. However, a vehicle driving at high speeds requires more than this type of control in order to overcome the limited rate of response of the system dynamics. *Preview control* is introduced as an additional control scheme to improve vehicle tracking ability.

Preview control is a method that uses future information, as well as present and past information, to calculate control commands. The term *preview* implies that the future desired states of the commands are known in advance. Human vehicle driving is known as a typical example of preview control.[14,15] Thus, its concept seems promising as a path tracking method for autonomous vehicles. If a vehicle driver cannot look ahead but looks at the road close to the vehicle, he or she will not be able to drive a car at high speed. By looking at the road sufficiently far ahead, one can decide what kind of action should be taken to keep the vehicle on the road, considering vehicle and human dynamics.

One characteristic of preview control is that *a priori* knowledge of the future inputs can be used to improve performance. Intuitively, it is possible that some of the system errors are fortuitous in light of the future reference path, and thus can be exploited. Figure 4 and 5 contrast a conventional feedback scheme with a scheme that uses a "preview" of future inputs. In Figure 5, at time t, the controller uses knowledge of the desired states in the interval $[t \rightarrow t + l_a]$ where l_a denotes the preview distance, to plan the next control inputs.

Another characteristic of preview control, which is not described in Figure 2–7, is that anticipation of the system behavior can be used to improve performance. Thus, another way of looking at preview control is that feedback control only compensates for errors after the fact; whereas, knowledge of future desired states and an approximate dynamical model of imperfect system behavior can be used to compensate for errors before the fact. This sort of anticipatory control is a well-known part of human performance in tasks like vehicle driving.*

Thus, a viable controller is high performance tracking of autonomous vehicle should have both feedback and feedforward compensation capabilities (Figure 6). The steering planner is partitioned into two parts:

- *feedback compensator,* which provides closed-loop compensation for errors between the vehicle's actual path and its desired path;
- *feedforward compensator,* which uses current reference inputs but also incorporates the future course of the path anticipating the behavior of the system.

Partitioning of the controller helps to understand the task of the controller more clearly and makes it easier to design and implement.

3. PROPOSED METHOD

In this section, a systematic methodology is presented to obtain an anticipatory control and for error feedback compensation, as shown in the partitioned block diagram of Figure 6.

3.1 Feedforward compensator

The output response of the simple open-loop system without any compensation is different from the desired

---

* When a human drives a new car, he or she would have some difficulty in driving before becoming accustomed to its dynamic behavior.

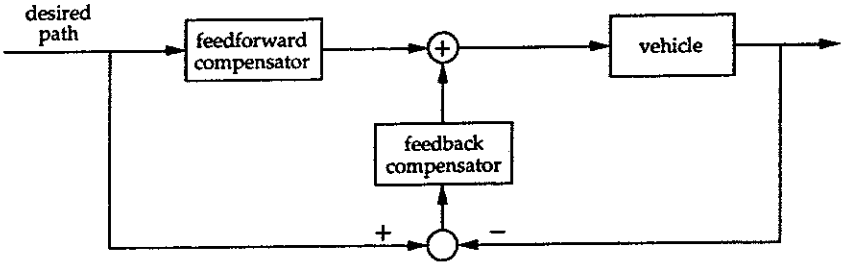

Fig 6 Partitioned scheme using both feedforward and feedback compensation scheme

response because the dynamic behavior of the controlled system is less than ideal – not as fast and not as accurate as the ideal – and there are disturbances to the system.

feedback compensation, the control is determined on a continual basis by compensating errors, as caused by the above reasons. However, feedback compensators act after the fact. If the system dynamic behavior is less than ideal and known disturbances can be compensated before their effects, the system performance can be much improved. This action is referred to as feedforward compensation. As mentioned earlier, path tracking for a roadworthy vehicle should include preview control which is characterized as using *a priori* knowledge of the reference input along with the anticipation of the system dynamic behavior.

The dynamic behavior of a mobile robot can be anticipated by using an approximate dynamic model of the system. Since vehicle dynamics are too complicated to model and to compensate for the dynamics exactly, we have developed a scheme that compensates for most dominant characteristic behavior of the system dynamics – latency. Since this phenomenon can be described by a time constant in a first-order lag system, the entire steering system is modeled as a lumped system of first-order lag.*

The effect of the feedforward compensation can be seen in Figures 7, 8, 9 and 10. Figure 7 shows the response (solid lines) of the steering system to given reference commands (dashed lines). The reference commands correspond to a path composed of a straight line, a circular arc, and a straight line. The reference and resultant paths in this case are shown in Figure 8.

Feedforward compensation is accomplished by sending reference commands in advance. In this manner, the steering starts moving before the arc is reached. A useful amount of advance is the time constant of the first-order model. If the vehicle dynamics are modeled exactly by a first-order lag, then the steering would reach 63% of the desired value by the time the arc path transition is encountered. An intituive argument can be made that such a scheme produces an area under the responsive curve, (solid line in Figure 9) that approximates the area under the reference curve (dashed line in Figure 9) and at the required time. Figure 10 shows the improvement produced by anticipating the response of the vehicle dynamics. Note that the future reference steering command is used. Thus the feedforward compensator satisfies both characteristics of preview control. Real vehicle responses are not as nicely predictable as shown, but the above scheme works well especially since reference steering commands have much smaller step changes (due to discrete time control) than those shown in the example.

Another virtue of the feedforward compensation method is robustness against systemic effects, such as pure time delay due to the vehicle's communications between the multi-tasking systems of multi-processors.

3.2 Feedback compensator

The feedback compensator guides a vehicle by closed-loop compensation for instantaneous deviation from the prescribed path. At every control cycle the proposed scheme replans a simple, continuous path that converges to a desired path in some look-ahead distance. It also computes a steering angle corresponding to the part of the replanned path, which will be followed for the next time interval. Since it does not require explicit dynamic models of the system, this scheme does not guarantee stability of the closed-loop system analytically, but provides a simple convergence of errors to the desired path, especially for a low frequency system, like the steering control of a roadworthy vehicle.

At any given moment, the vehicle will have errors in lateral position, heading, and curvature. The scheme uses the geometry of the errors to obtain a continuous quintic polynomial function that converges to zero errors

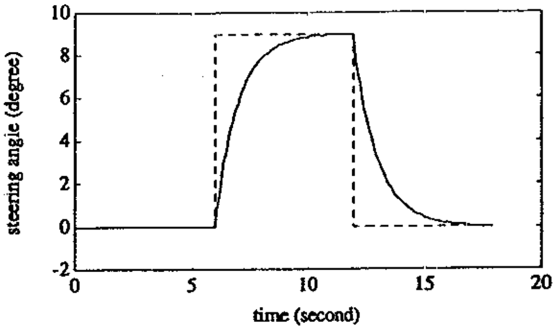

Fig. 7. Performance of the first order model of the steering system, in response to a path consisting of straight lines and an arc

---

* Higher order dynamic models would add supplementary effects like vibratory phenomena along with the first-order model response with complexity.

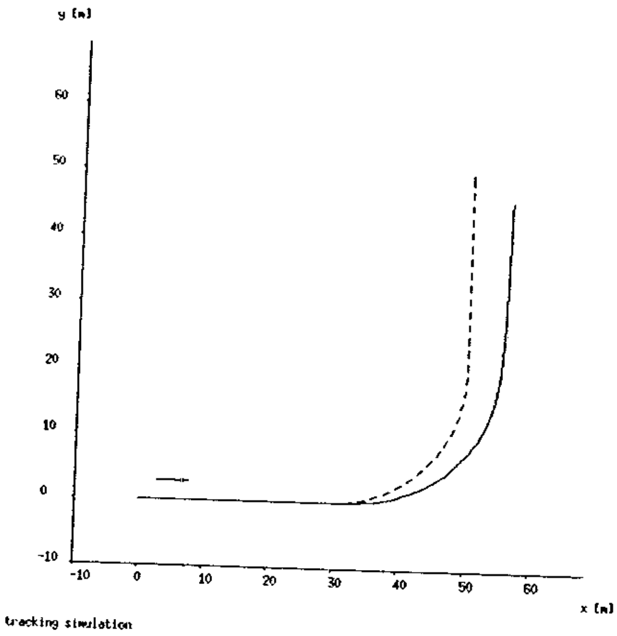

Fig. 8 Open-loop response without feedforward compensation

at some prespecified look-ahead distance. Consider a desired continuous path $(x(s), y(s))$, a function of traveling distance, and a vehicle that is currently at P_a. An error vector can be calculated that represents error in the distance transverse to the path (ε_0) relative to P_0 in heading (β_0), and in curvature (γ_0), as in Figure 11. 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 (using Taylor series approximation) and to zero errors at P_L:

$$\begin{aligned}
&\varepsilon(P_0) = \varepsilon_0; && \varepsilon(P_L) = 0 \\
&\beta(P_0) = \beta_0 \approx \left[\frac{d\varepsilon(s)}{ds}\right]_{s=0}; && \varepsilon(P_L) = 0 \approx \left[\frac{d\varepsilon(s)}{ds}\right]_{s=L} \\
&\gamma(P_0) = \gamma_0 \approx \left[\frac{d^2\varepsilon(s)}{ds^2}\right]_{s=0}; && \gamma(P_L) = 0 \approx \left[\frac{d^2\varepsilon(s)}{ds^2}\right]_{s=L}
\end{aligned} \tag{4}$$

These boundary conditions can be sufficiently satisfied by a polynomial of fifth degree. A quintic polynomial can be constructed to describe the replanned path (in error space) as follows:

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

Solving for a_0, a_1, a_2, a_3, a_4, a_5:

$$\begin{aligned}
a_0 &= \varepsilon_0 \\
a_1 &= \beta_0 \\
a_2 &= 2\gamma_0 \\
a_3 &= \frac{-(13\gamma_0 L^2 + 12\beta_0 L + 20\varepsilon_0)}{2L^3} \\
a_4 &= \frac{9\gamma_0 L^2 + 8\beta_0 L + 15\varepsilon_0}{L^4} \\
a_5 &= \frac{-(7\gamma_0 L^2 + 6\beta_0 L + 12\varepsilon_0)}{2L^5}
\end{aligned} \tag{6}$$

The expression for $\varepsilon(s)$ gives the error along the pat from P_0 to P_L. The curvature in error space can be written as:

$$\frac{d^2\varepsilon(s)}{ds^2} = 2a_2 + 6a_3 s + 12a_4 s^2 + 20a_5 s^3 \tag{7}$$

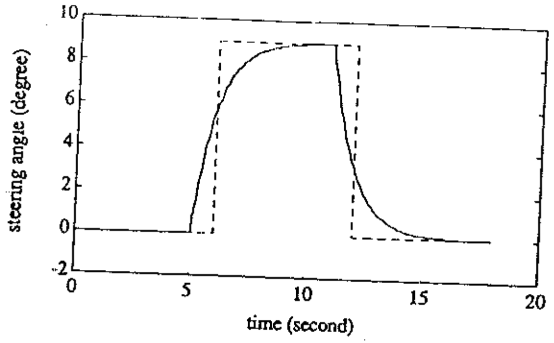

Fig. 9. Compensated steering performance modeled as a first order model

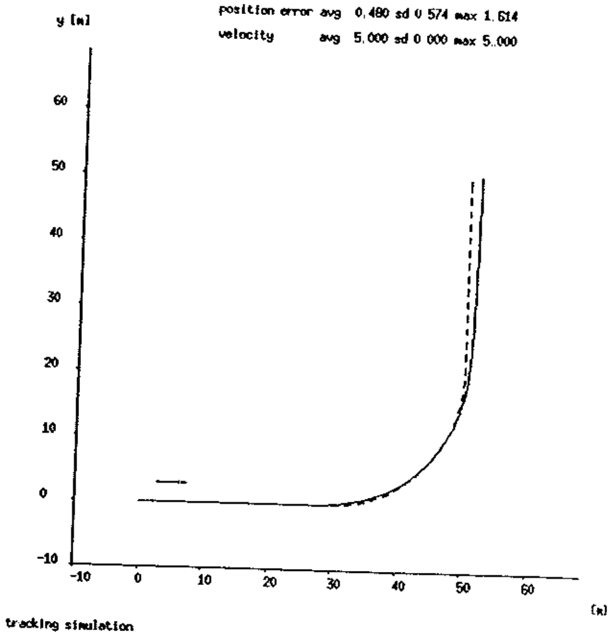

Fig. 10 Open-loop response with feedforward compensation

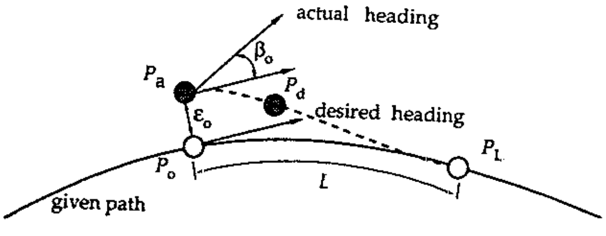

Fig. 11. Computing an error vector

Steering angle variation can be obtained from (7) and (1) as follows:

$$\phi_{\text{feedback}}(s) = \tan^{-1}\left\{\frac{d^2\varepsilon(s)}{ds^2} l\right\} \qquad (8)$$

Computing the steering angle $\phi_{\text{feedback}}(s)$ over the entire interval $[0 \rightarrow L]$ is not necessary. Instead, during every control loop cycle, the coefficients $a_1 \cdots a_5$ are computed based on the most recent estimate of position, heading, and curvature errors, and $\phi_{\text{feedback}}(s)$ is computed only for a small path distance immediately ahead of the vehicle

The sensitivity of the scheme described above can be modulated by the adjustment of the parameter L which represents the distance at which all three parameter (position, heading, and curvature) errors are forced to be zero. Figure 12 shows the difference in variation produced by the quintic polynomial for three different values of L. The solid lines denote the variation in position error, while the dashed lines indicate the variation in the curvature

The parameter L (look-ahead distance) is chosen based on vehicle speed, much as a human driver would choose to look out farther along the path the higher the vehicle speed. A larger value of L corresponds to a milder contribution from the feedback scheme than for smaller values Once again, a simple linear model is used to vary L with vehicle speed as in (9), the gains for which are determined experimentally:

$$L = \text{slope}^* \,(V - V_{ref}) + L_{ref} \qquad (9)$$

where V is the speed of a vehicle.

3.3 *Speed planning*

As mentioned in Section 2.1, one of the behefits of choosing the guide point to be the center of the rear axle is that steering and propulsion can be controlled separately Thus, speed (propulsion) is based on the following constraints:

- maximum speed limit in a particular area
- distance to possible obstacles and the destination
- path curvature, i.e. for constant lateral acceleration, curvature of the section of the path immediately in front of the vehicle can be used to constrain the vehicle speed as the following:

$$a_{\text{lateral}} = \frac{v^2}{r} = cv^2 \qquad (10)$$

Speed can be chosen to be the minimum of the above constraints and this speed is sent as input to the propulsion servo control

4. RESULTS

4.1 *Implementation*

To verify and evaluate the performance of path tracking algorithm, the algorithm was applied to vehicle navigation, both in simulations and experiments. In computer simulations a vehicle with four wheels is simulated. A first-order lag system with pure time-delay is used as a model for dynamics of steering and driving.

In the experiments Navlab[14], as in Figure 1, has been used as a testbed vehicle. It is modified from a truck and its actuation functions are described in Figure 13. An inertial sensor, referred to as "Vehicle Positioning System" (VPS), is used to sense information, such as the position and velocity of the Navlab. It consists of three ring laser gyros, which measure angular acceleration about three orthogonal axes, and three accelerometers, which measure linear acceleration in three inertial and orthogonal directions. In general, this type of inertial sensor can provide position, heading, and other key information at a higher data rate than any other sensor, such as vision, sonar, or stellar sensors The system control architecture, which realize the proposed path tracking algorithms is described functionally in Figure 14.

All tasks in the system were loosely coupled through shared memory, with data being stored/fetched on an as available/needed basis. The supervisor initiated system tasks and monitored overall system status The position estimator provided global position data, and the path manager updated the path store to reflect vehicle movement. The path tracker used the position and path

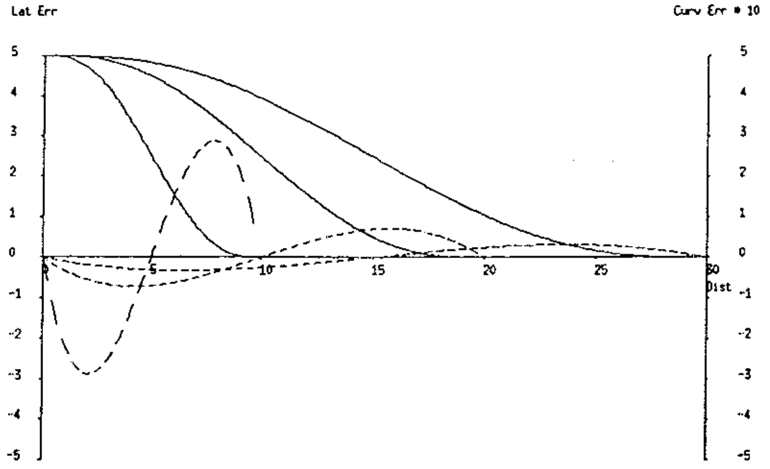

Fig. 12 Modulation for the parameter L

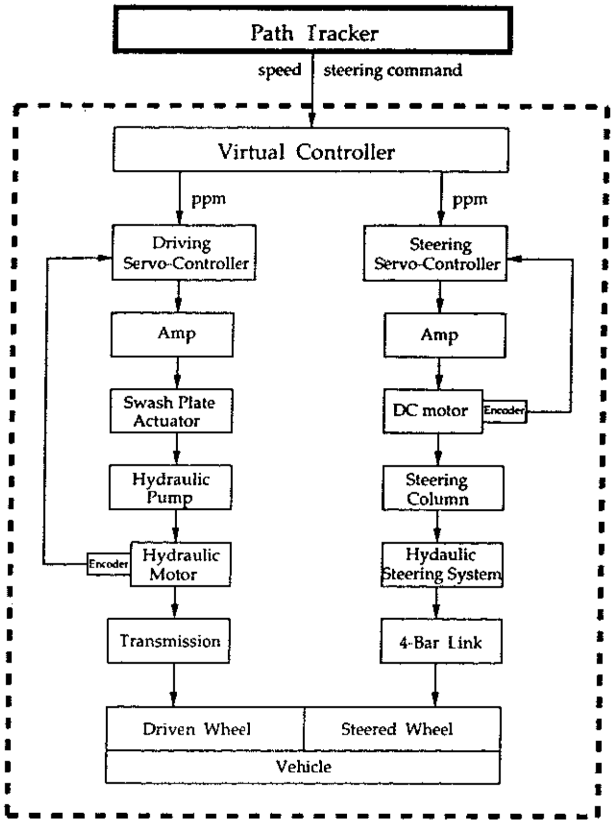

Fig. 13 Navlab actuation functional diagram

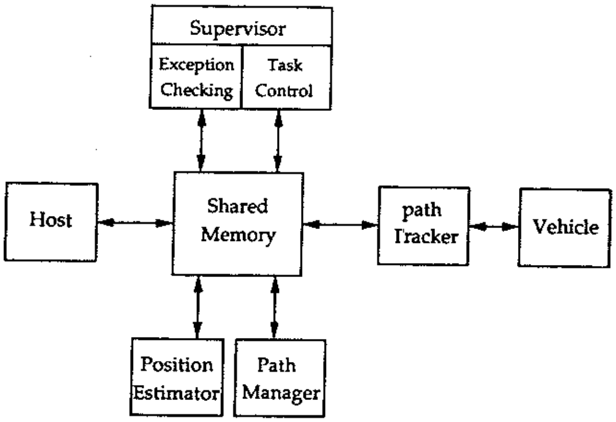

Fig. 14. Navlab control system architecture

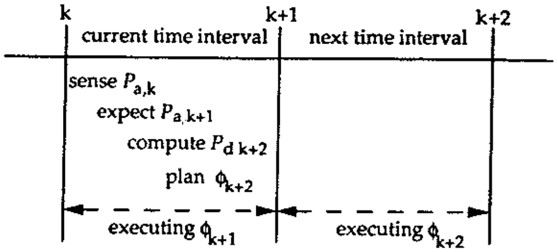

Fig. 15. Timing of steering planning

data to control the vehicle. The path tracking system is closed with nested feedback loops (Fig. 2). The inner loop for servo-controllers in the actual vehicle system is executed on the order of 10 mS, while the outer loop is closed normally at the rate of 0.25 second – steering and speed commands are sent to the servo-controllers at this same rate. The time interval for the outer loop is governed by the VPS time interval (normally 4 Hz, maximally 20 Hz), which is much longer than the computing time required for the Path Tracker (16 mS at SUN 3/75). Since the time interval is quite long, the steering planning scheme for the Path Tracker is required to predict the vehicle position.

Figure 15 illustrates steering planning procedure in time domain; After sensing the current posture ($P_{a,k}$), the posture at the end of the current time interval ($\bar{P}_{a,k+1}$) is projected. Then, the desire posture at the end of the next time interval ($P_{d,k+2}$) is computed and the reference steering angle between $\bar{P}_{a,k+1}$ and $P_{d,k+2}$ is determined. Ideally, $\bar{P}_{a,k+1}$ should be computed fro $P_{a,k}$, ϕ_{k+1}, the speed of the robot, the planning tim interval, and the robot's dynamics. However, because this is too time-consuming, $\bar{P}_{a,k+1}$ is approximated through the following:

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

$$\bar{P}_{a,k+1} = P_{d,k+1} + e_k \tag{11}$$

4.2 Results

In order to show performance improvements by the feedback and feedforward schemes developed in Section 3, the following four simulations are considered:

- The vehicle is made to follow a path in open-loop fashion. In this case only the steering angle is regulated, while vehicle position errors are not compensated (Figure 16-A).
- Vehicle position errors are compensated using only the feedback scheme discussed in Section 3.2 (Figure 16-B).
- The only compensation made is a feedforward one, using the control scheme discussed in Section 3.1 (Figure 16-C).
- Both feedback and feedforward schemes used in Figure 16-B and 16-C are applied to compensate fo the vehicle errors (Figures 16-D).

In the open-loop case, the resultant path is often parallel to the desired path. This is because, after the desired curvature is obtained, no further correction is made unless the input is changed. In the case of feedback only, the vehicle oscillates around the nominal path because the feedback gain makes the vehicle sensitive to position errors. For lower value of feedback gain (longe look-ahead distance), much less overshoot is noticed, but the feedback response is very sluggish. In the feedforward only case, the algorithm was able to compensate for the vehicle in a predictive fashion. However, such errors accumulate without feedback in the general case. In the last case, it can be seen that the combined feedback and feedforward schemes comple

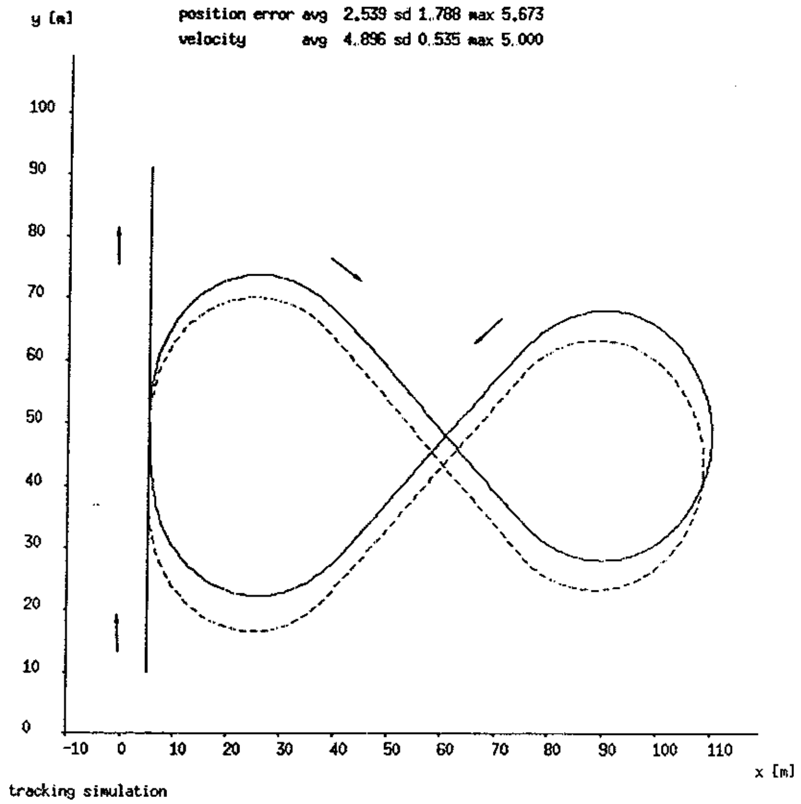

Fig 16-A.

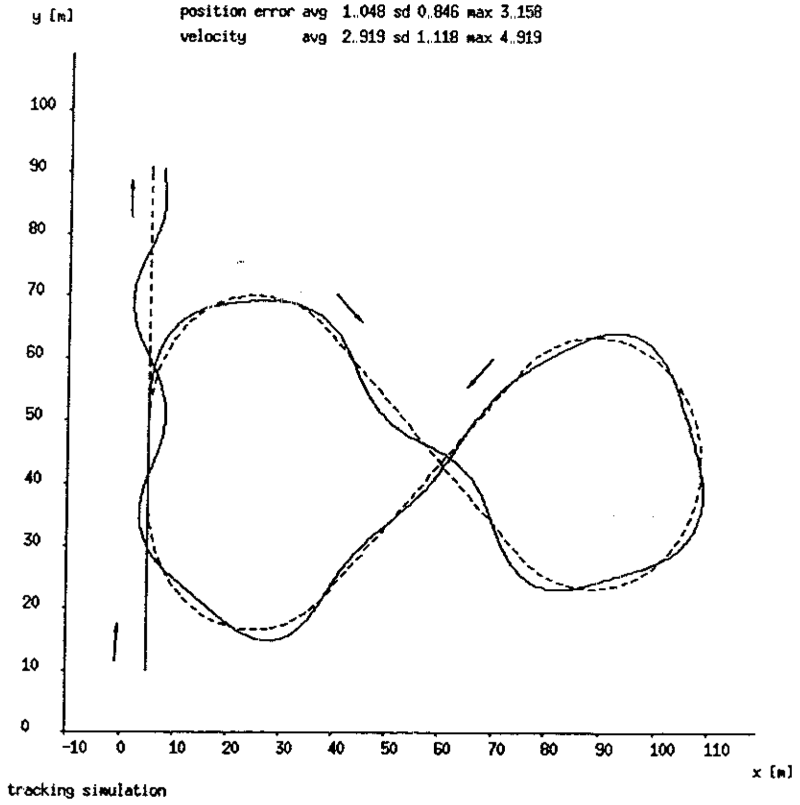

Fig. 16-B.

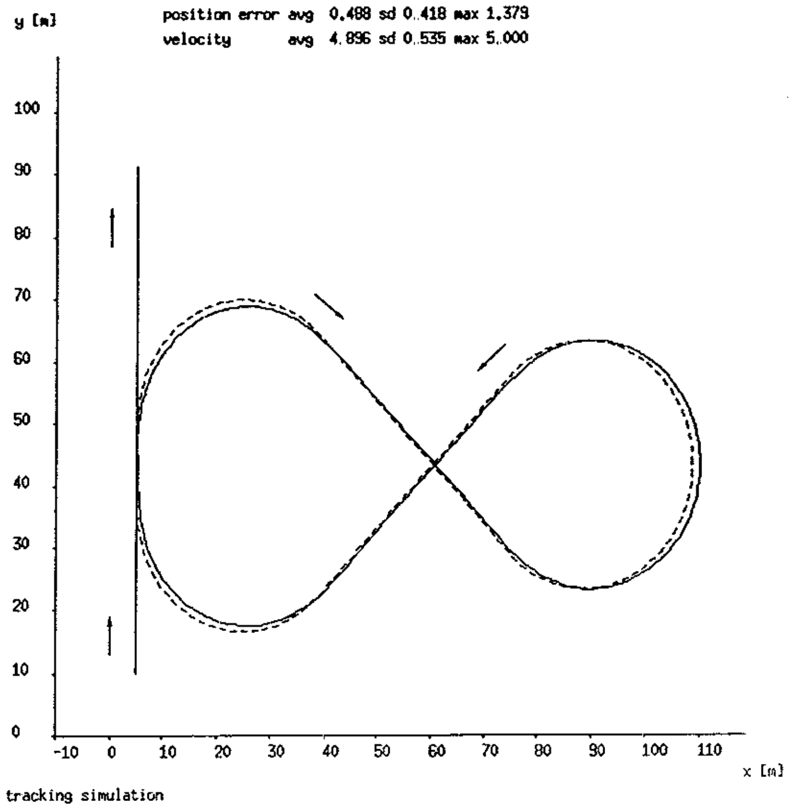

Fig. 16-C.

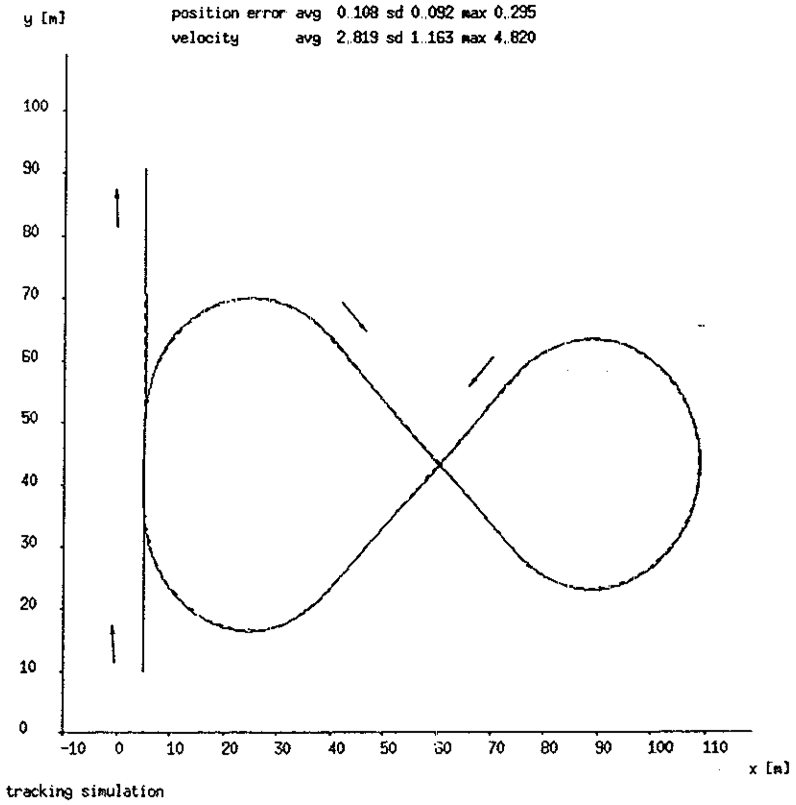

Fig. 16-D.

---

ment
super
meth
Ex
simul
result
variat
desire
jump
from
feedb
path
is sim
the f

Fig. 17

nent each other and the resultant path is qualitatively superior to those obtained by application of the other methods.

Experiments have shown results similar to those in the simulations. Figures 17-A, B, C, and D show graphical results from real time navigation of the Navlab with variations of feedforward compensating time. The desired path consists of a 20*m* straight line, a sudden jump by 5*m* in lateral direction, and an 80*m* straight line from the initial offset to see the step response. Only with feedback compensation alone does the actual vehicle path shows serious oscillation, as in Figure 17-A, which is similar to the simulation results of Figure 16-B. When the feedforward compensation time was 0.4 seconds (Figure 17-C) or 0.6 seconds (Figure 17-D), the experimental results showed the best tracking performance. These figures indicate that the Navlab has inherent latency of 0.5 seconds Note that the look-ahead distance in the feedback compensator was fixed at 15*m* to concentrate on the feedforward compensator evaluation. Evidently, the feedforward compensation has a dominant effect on the tracking performance, as the simulation and experimental results (Figure 16-D and Figure 17-C) show dramatic improvement from those without feedforward compensation.

Figure 18 presents the tracking performance of an arbitrarily recorded path is more than 500*m*. Navigations succeeded at speeds upto 10*m*/*second* which is the limit

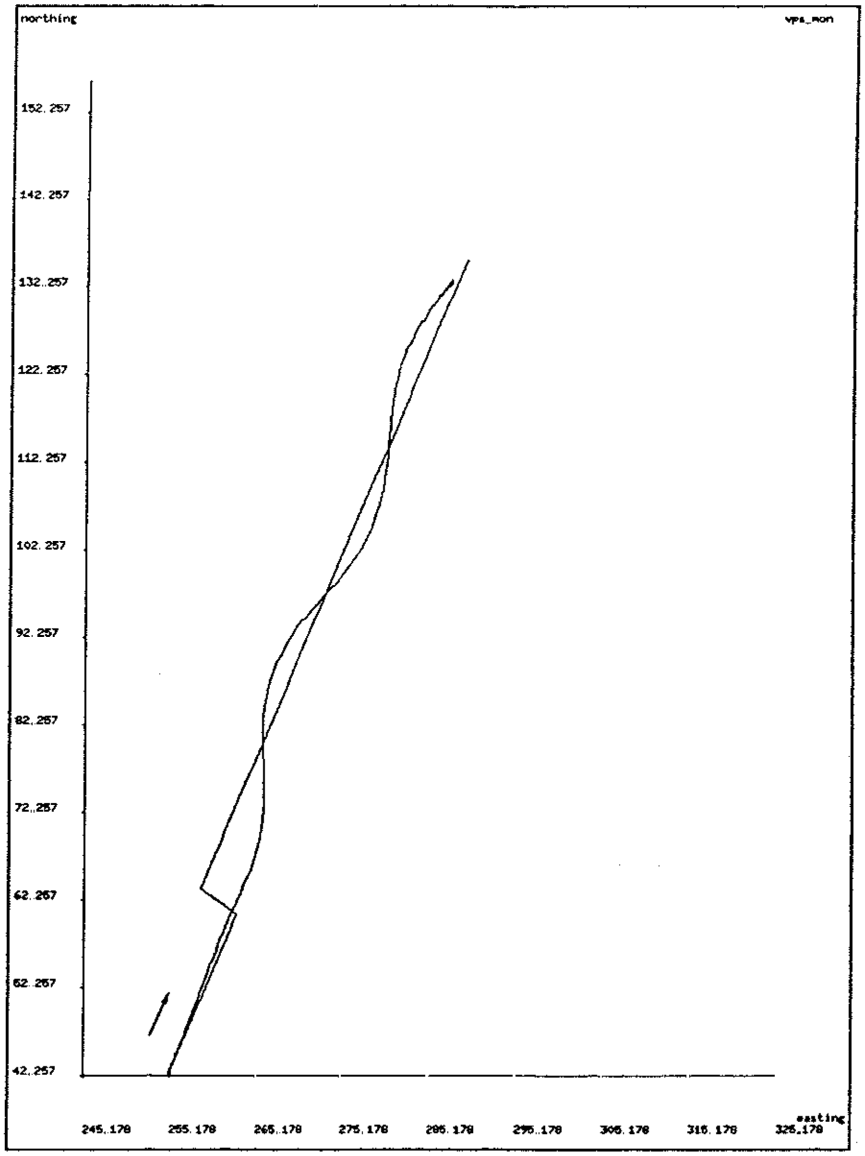

Fig. 17-A.

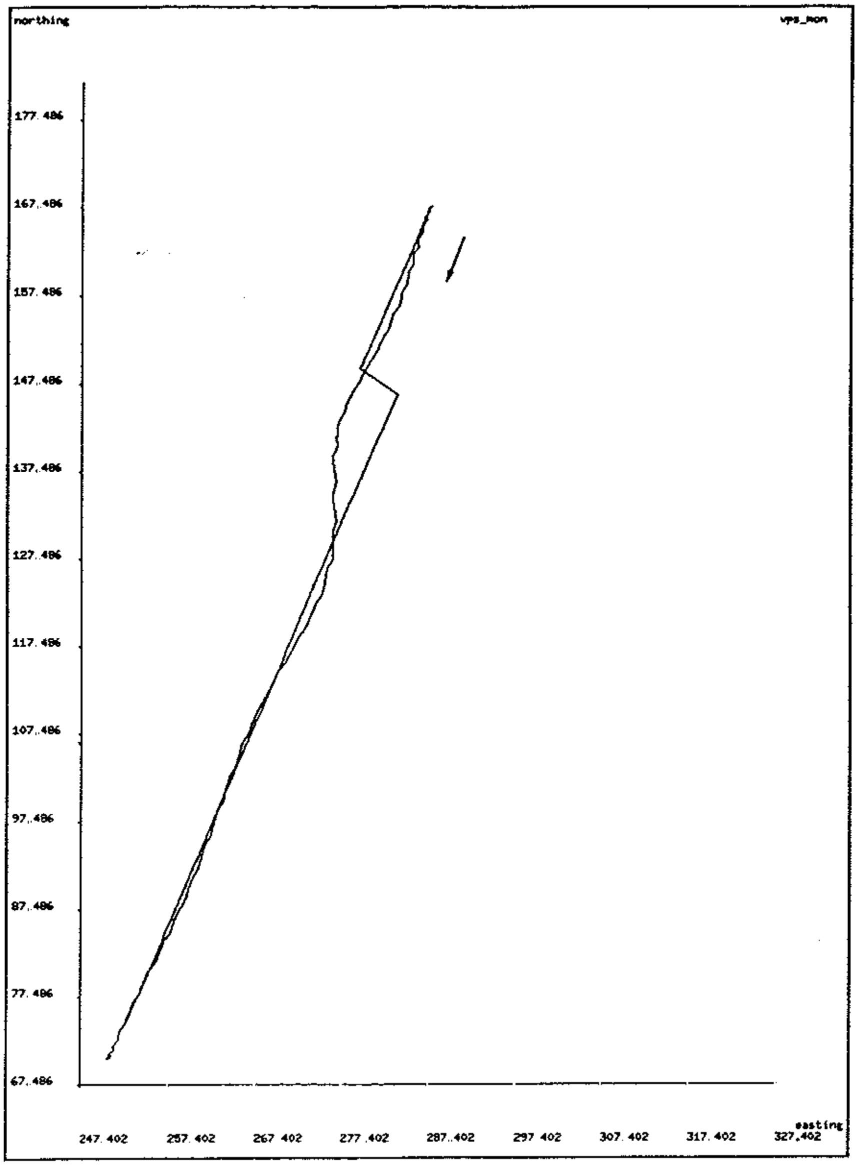

Fig. 17-B

speed of Navlab. Note that there are some deviations from an intended path near the ends of the courses. The inertial position sensor in Navlab often drifted in heading. The desired path in Figure 18 was found to have serious discontinuous jumps in heading and curvature, recorded while driving the Navlab for teaching a desired path. However, the physical vehicle converged to the desired path, even when the path had discontinuities due to VPS sensor problems. Actual tracking path in Figure 18 also shows some notches like the teeth of a saw, but it was due to our sensor and the actual nagivator was smooth. These results show that the performance of the tracking algorithms for autonomous vehicles are quire acceptable.

5 CONCLUSION

This paper develops and demonstrates a mode of high-speed autonomous navigation for full-sized outdoor vehicles. This mode differs from precedent mobile robot path tracking in that it follows explicit outdoor paths and uses accurate, explicit robot vehicle positions at high periodic rates.

Together, the following three strategies distinguish the methodology for path tracking developed in this work: (1) geometric "path" tracking similar to wire-following by factory AGVs versus trajectory tracking which has

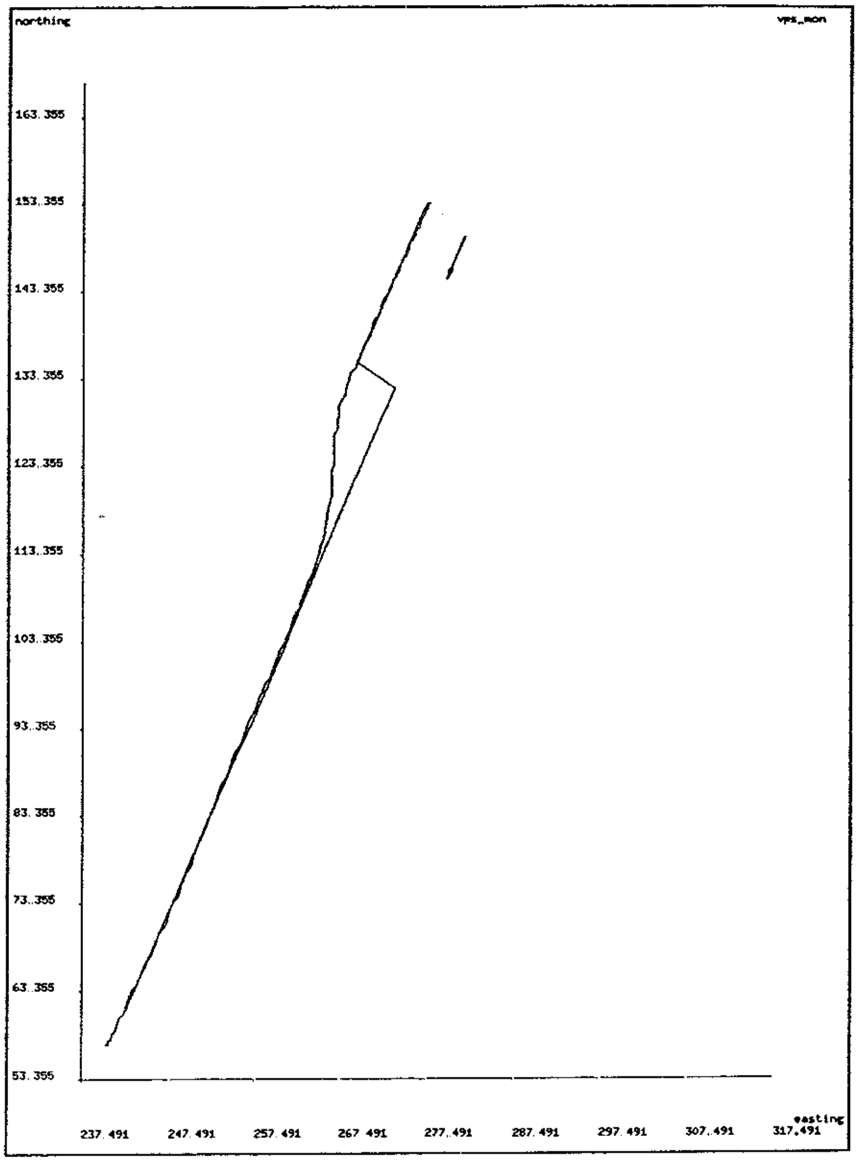

Fig 17-C.

been common for robot navigation heretofore: (2) multiple nested feedback control loops on actuators, vehicle geometry and path tracking error; and (3) the selection of a guide point at the center of the rear axle, which exploits kinematic decoupling of steering and propulsion. The three follow naturally from the kinematic characterics of mobile robots; non-holonomic constraints and kinematically decoupled steering and propulsion. The strategies simplify the robot vehicle tracking problem as a steering control problem. Propulsion speed is a parameter set outside the steering loop by criteria such as obstacle avoidance or human interaction. Furthermore, the nested feedback loop strategy decomposes the problem into a steering actuator servo control problem and an on-line steering planning problem. Consequently, these strategies lead to the following advantages: (1) a simpler navigation problem formulation; (2) modularized control hardware implementation; and (3) smooth and stable navigation.

A partitioned scheme for steering planning is implemented in two parts: a feedforward compensator and a feedback compensator. The former guides a robot along an intended path by producing, with the use of *a priori* knowledge of the future path, an anticipatory control; and the latter compensates for errors by finding a curve which will converge with the desired path while insuring a smooth reduction of all errors; the quintic polynomial scheme. This method does not require a

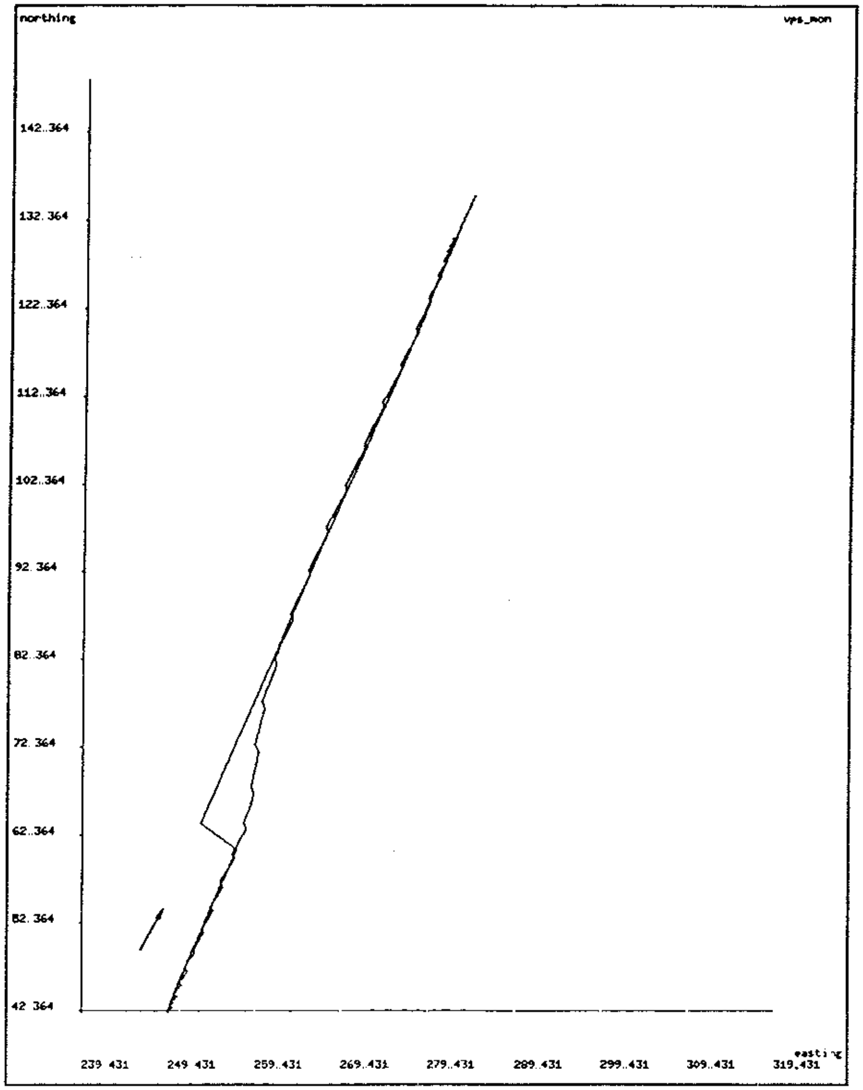

Fig. 17-D.

complete modeling of the full-sized vehicle system which, in any case, defies complete and precise description. Rather, it succeeds with a first order model by compensating for the total system latency, which appears to be the dominant characteristic in the behavior of slow response systems, such as roadworthy vehicles. This circumvents the problem of complex physics in modeling and control and yet provides high performance through smooth error compensation and anticipatory control, which are important characteristics of human driving.

These developments were then implemented on simulation and navigation hardware. Tracking performance was found in experiments and simulation to be more dramatically improved by anticipatory control, as the vehicle speed increased. Outdoor tracking of paths by a real-time autonomous vehicle has been demonstrated at speeds up to 20 miles per hour, which is an American record for the highest speed achieved by an autonomous vehicle, but more significantly it is a first benchmark and world record for navigation by this promising navigation paradigm.

Directions for future research include the following. Firstly, dynamic characters of inertia and slippage could be modeled and used in feedback compensator which currently uses a geometric scheme. This would help in ensuring (though not guaranteeing) the stability of the compensator which is the weakpoint of the closed-loop system. Secondly, the feedforward compensator could be improved to get the better open-loop response. Higher order dynamics or neural net could be applied. Finally, the proposed path tracking method could be extended to other tasks like convoying vehicles.

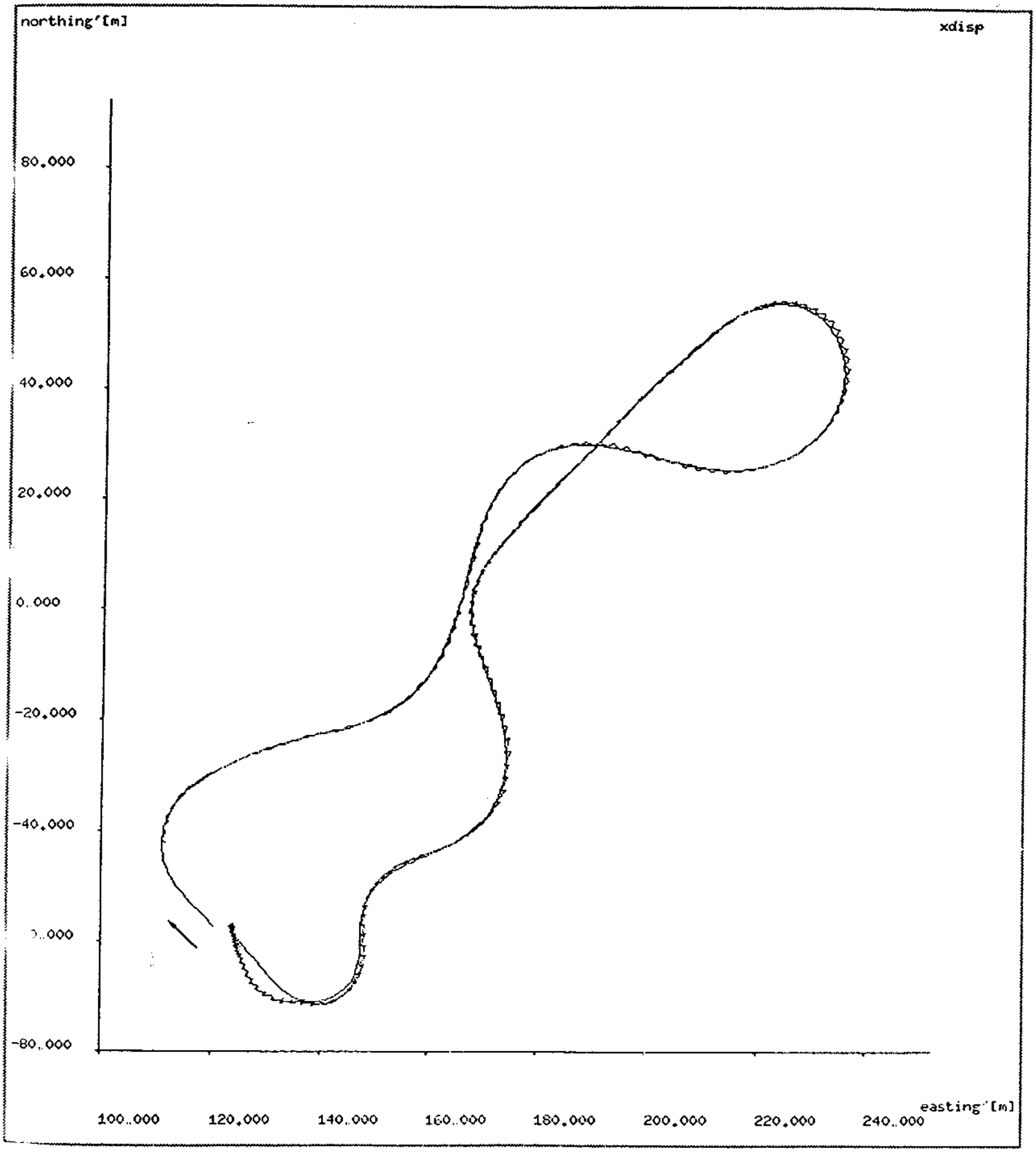

Fig 18.

References

1 H.P. Moravec, "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover" *Ph.D. Dissertation* (Department of Computer Science, Stanford University, Stanford, CA, 1980).

2 Goto and A. Stentz, "The CMU System for Mobile Robot Navigation" *IEEE International Conference on Robotics and Automation* (1987) pp. 99–106.

3 S Singh and P. Keller, "Obstacle Detection for High Speed Autonomous Navigation" *Proceedings of IEEE Conference on Robotics and Automation, Sacramento* (April, 1991) pp 2798–2805.

4 D. Feng, S. Singh and B.H. Krogh, "Implementation of Dynamic Obstacle Avoidance on the CMU NavLab" *Proc IEEE Conference on Systems Engineering* Pittsburgh (August, 1990) pp. 208–211.

5 R. Wallace, A. Stentz, M. Moravec, W. Whittaker, C. Thorpe and T. Kanade, "First Results in Road Following" *Proceedings of IJCAI, Los Angeles* (Ausist 1985) pp 1089–1095.

6 D.H. Shin and S Singh, "Position Based Path Tracking for Wheeled Mobile Robots" *Proceedings IEEE/RSJ International Workshop on Intelligent Robots and Systems '89* (1989) pp 386–391.

7. Y. Kanayama, A. Nilipour and C.A. Lelm, "A Locomotion Control Method for Autonomous Vehicles" *IEEE Intl. Conference on Robotics and Automation*, Philadelphia (1988) pp. 1315–1317.

8 W.L. Nelson and I.J. Cox, "Local Path Control for an Autonomous Vehicle" *IEEE Intl. Conference on Robotics and Automation, Philadelphia* (1988) pp. 1504–1510.

9 P.F. Muir, "Modeling and Control of Wheeled Mobile Robots" *Ph D. Thesis* (Carnegie–Mellon University, Pittsburgh, Pennsylvania, 1988).

10. H. Dugoff, P.S. Fancher and L. Segel, "An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance" *SAE paper* 700377 (1970).

11 J.C. Dixon, "Linear and Non-linear Steady Vehicle Handling" *Proc Instn. Mech. Engrs.* **202,** No D3 173–186 (1988).

12. K. Dowling, R. Guzikowski, J. Ladd, H. Pangels, S. Singh and W. Whittaker, "NAVLAB: An Autonomous Navigation Testbed" *Technical Report Carnegie Mellon University, CMU-RI-TR*-87-24 (1987).

13. W.L. Nelson, "Continuous Steering Function Control of Robot Cart" *IEEE Transactions on Industrial Electronics* **36,** No. 3, 330–337 (August, 1989).

14 T.B. Sheridan, "Three Models of Preview Control" *IEEE Transactions on Human Factors in Electronics* **HFE**-7, No 2, 91–102 (June, 1966).

15 E. Donges, "A Two-level Model of Driver Steering Behavior" *Human Factors* **20**(6), 691–707 (1978).