Local Path Control for an Autonomous Vehicle

Winston L. Nelson
Ingemar J. Cox

AT&T Bell Laboratories
Murray Hill, NJ 07974, USA

ABSTRACT

This paper describes a control system for an autonomous robot cart designed to operate in well-structured environments such as offices and factories. The onboard navigation system comprises a reference-state generator, an error-feedback controller, plus cart location sensing using odometry. There is a convenient separation between the path guidance and control logic. Under normal operating conditions, the controller ensures that the errors between the measured and reference states are small. These errors only exceed set limits if the cart is malfunctioning. Major hardware failures can be detected in this way and failsafe procedures invoked. Results on the control system performance derived from a computer simulation of the cart and its operating environment, and from an experimental cart, indicate that it can provide reliable, accurate, and safe operation of autonomous robot carts.

1. INTRODUCTION

Robot carts are coming into increased use in automated factories and other such reasonably well-structured environments.[1] Most of the automated carts in present use, however, are not really autonomous (i.e., self-guided) but rely on "tracks" imbedded in, or painted on, the factory or office floor to guide them from one station to another.[2]

This paper describes some preliminary results on a local path control system for an experimental cart, "Blanche",[3] shown in Figure 1. It is a tricycle configuration with a single front wheel which serves both for steering and driving the cart and two passive load-bearing rear wheels. The superstructure in the front supports an optical rangefinder unit.[4]

The cart is capable of navigating in structured environments such as offices and factories without external guide tracks. The onboard control system uses odometry to provide the position and heading information needed to guide the cart along specified paths between stations. The overall path control system for the cart is diagrammed in Figure 2. The external, or off-board, part of the system consists of a global path planner which stores the structured environment data and produces the path plans from one station to the next.[5] It also includes a communication link between this supervisory controller and the cart.

The path plan directing the cart from one station to the next comprises a list of path segments of three basic types: lines, arcs, and splines. Although line and arc segments provide optimal (minimum distance) paths between any initial and final positions and headings,[6] spline segments have been included because they reduce the number of segments needed to specify paths for cases such as simple lane-change maneuvers.

The local path planner takes the path segment data and passes each segment in turn to the reference state generator, which serves the function of converting the sparse *spatial* data of the path plan into detailed *temporal* path data produced as the cart moves along the path. Having the RPG on board the cart facilitates autonomous operation of the cart with only an occasional low-data rate link to the supervisory controller. As indicated in Figure 2, this reference state is input to the cart controller, which compares it with the measured state of the cart

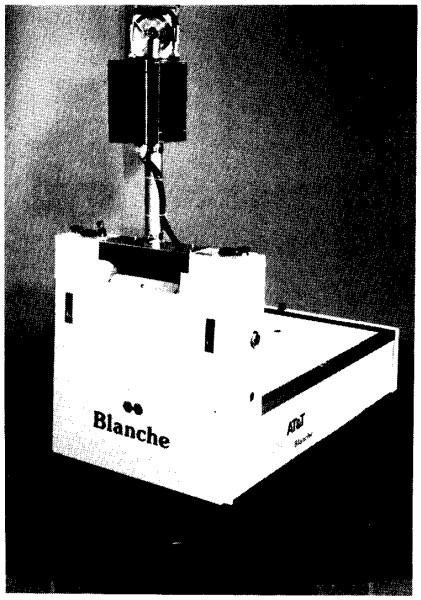


Figure 1. Experimental robot cart.

(derived from the odometry sensors) and outputs control torque signals to the cart steering and drive motors.

Algorithms for the autonomous cart navigation system are being developed and tested both on a computer-aided design workstation and on the experimental cart. The simulation permits the testing of path-planning, path-generation, and path-control algorithms in a safe and readily re-structurable operating environment before porting them to the computer on board the cart. The computer system and sensors on the cart, and the path planning system system in the supervisory controller, are discussed in the companion papers.[3]-[5]

2. REFERENCE PATH GENERATOR

2.1 Rationale

The reference path generator (RPG) functions as an onboard navigator that continually provides an appropriate reference state for the cart controller to track. The reference state is determined by three factors: (1) the path plan, (2) the acceleration/deceleration capabilities of the cart, and (3) the current operating status of the cart.

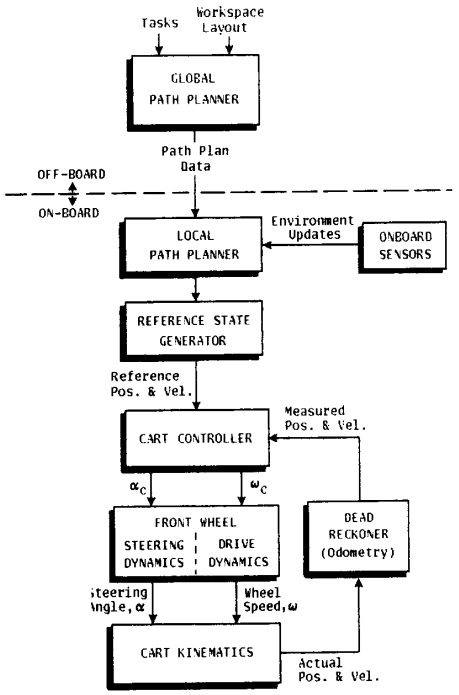


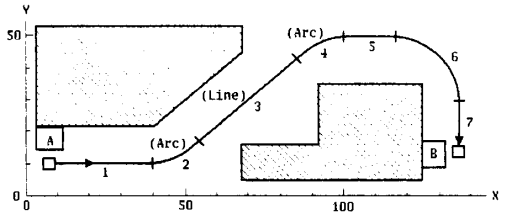
Figure 2. Block diagram of path control system.

The path plan received from the global path planner consists of a list, each line of which specifies a segment of the path. The segment specification includes the segment type and the desired state at the end of the segment. Figure 3(a) shows a possible path plan between two cart stations (A and B). This sample path plan contains seven segments, four line segments separated by three arc segments. The corresponding path data used by the RPG to evolve the reference path are the seven lines of numbers listed in Figure 3(b).

Although the sample plan in Figure 3(a) consists only of lines and arcs, the RPG, as presently implemented, can also handle spline segments. Splines are especially convenient for "lane-change" maneuvers, because one spline segment can replace an arc-line-arc sequence. For example, segments 2, 3, and 4 of the path shown in Figure 3(a) could be represented by one spline segment.

The RPG computes at each control update cycle the reference state on the path, the reference steering angle and drive speed, and the distance to the end of the current path segment. When the distance to the end-point of the segment becomes less than the distance between the preceding reference points, the RPG requests the next segment in the path plan and begins generating reference states along the new segment. An illustration of the reference points generated for the path shown in Figure 3(a) is given in Figure 3(c).*

The main advantage of the reference path guidance scheme over point-to-point guidance schemes[7], [8] is that the cart's onboard system is always providing a current point on the path where the cart should be, with smooth transitions across path-segment boundaries. A point-to-point control scheme, because of measurement errors, may not detect the point at which the control algorithm should switch to a new target

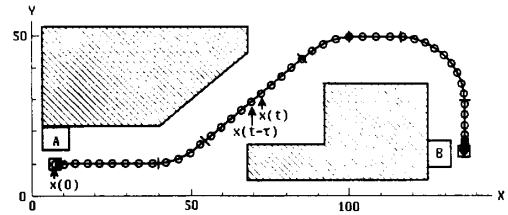


(a) PATH SEGMENTS

Segment Number	Segment Type*	Desired End State x_d	y_d	θ_d	v_d
1	1	40.00	10.00	0.00	0.50
2	2	56.97	17.03	45.00	0.50
3	1	83.03	42.97	45.00	0.50
4	2	100.00	50.00	0.00	0.50
5	1	116.20	50.00	0.00	0.50
6	2	136.20	30.00	-90.00	0.40
7	1	136.20	14.22	-90.00	0.00

* 1 = line, 2 = arc

(b) PATH PLAN DATA



(c) REFERENCE POINTS

Figure 3. Example of a path plan, data, and reference path.

point, and hence may fail to make the proper transitions from one path segment to the next.

The RPG scheme also provides a means for onboard monitoring of the operating status of the cart. Under normal operating conditions, the difference between the measured state of the cart and the current reference state should be small. If this state error ever exceeds some established limits, then the cart must be functioning abnormally. The cart controller can test for abnormal errors and signal the local path planner to take remedial action. By this means, system malfunctions, such as failure of motors or odometry sensors, can be detected in time for the cart to be safely stopped.

A third advantage of the RPG scheme is that it provides a convenient separation between the path-guidance logic and the controller logic, which must be tuned to the characteristics of the closed-loop dynamics of the steering and drive systems of the cart. Once properly adjusted, the controller design remains fixed throughout all the different operating situations that may arise along the path; all the controller needs to know is what the current reference and measured states are. The RPG takes care of dealing with the geometric computations necessary to evolve the path plan as points in time, and also with anomalies which may require local path modifications, either spatially, temporally, or both.

2.2 Reference Speed

In order for the RPG to be a "considerate" guide to the cart controller, it should use speed profiles compatible with the recommended acceleration and speed limits of the drive system. For start-up from a stop, and for speed changes across path segments, a ramp change in reference speed is programmed using the recommended acceleration. For the terminal segment of the path, or for emergency

* The actual reference points form a very dense set of points along the path; a sparse subset of points is shown in Figure 3(c) for clarity.

stops along the path, a ramp-down of speed, based on a gain G_r, multiplying the remaining distance to the stop point, is programmed to provide smooth stops without overshoot. The value of G_r is chosen to produce an appropriate compromise between the competing requirements for docking. The value should be large enough so that the cart remains maneuverable up to a point near the stop point, but not so large that the decrease in the reference speed exceeds the deceleration capability of the cart. Maneuverability is needed near the docking point to correct for any heading error or position error normal to the path, while reasonable deceleration is needed to correct for position error tangential to the path.

2.3 Reference Position and Heading

The reference position and heading on the desired path at the current update time, t, depends on the position and heading at $t-\tau$, the current path-segment type, and the reference speed, $v_r(t)$, discussed above. The geometric computations needed to generate the new reference position and heading depend on the segment type. Before describing these, it is useful to introduce some notation: The *desired* position-heading vector at the end point of the current segment is denoted by $\mathbf{z}_d \equiv [x_d, y_d, \theta_d]^T$, where the superscript T denotes transpose. The *beginning* position-heading vector for the current segment (which is the $\mathbf{z}_d$ of the previous segment) is denoted by $\mathbf{z}_b \equiv [x_b, y_b, \theta_b]^T$. The *reference* position-heading vector at the update time t is denoted by $\mathbf{z}_r(t) \equiv [x_r(t), y_r(t), \theta_r(t)]^T$. We define $\mathbf{z}_f \equiv [x_f, y_f, \theta_f]^T$ to be the *former* reference position-heading vector, transformed to a coordinate system whose origin is at (x_b, y_b) and whose X-axis is aligned with θ_b, i. e.,

$$\mathbf{z}_f = \mathbf{B}\,(\mathbf{z}_r(t-\tau) - \mathbf{z}_b) \tag{2.1}$$

where the base transformation matrix, $\mathbf{B}$, is

$$\mathbf{B} = \begin{bmatrix} \cos\theta_b & \sin\theta_b & 0 \\ -\sin\theta_b & \cos\theta_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

This transformation of reference positions and headings with respect to the beginning of the current path segment simplifies the update computations within the different segment types. We also define $\mathbf{z}_p \equiv [x_p, y_p, \theta_p]^T$ to be the *projected* position-heading vector in this transformed coordinate space. Finally, we denote by α_r the reference steering angle and by ω_r the reference drive rotational speed needed to move the cart along the current path segment when the error between the reference and the actual state of the cart is zero.

2.3.1 Line segments. If the segment type is a line, the incremental path motions are a sequence of $v_r\tau$ steps along the transformed X-axis. Hence, the projected position-heading vector along a line segment is

$$\mathbf{z}_p = [x_f + v_r(t)\tau \;,\; 0 \;,\; 0]^T \tag{2.3}$$

The reference steering angle and drive speed for straight line segments are obviously

$$\alpha_r = 0 \quad , \quad \omega_r = v_r / R \tag{2.4}$$

where R is the radius of the front wheel of the cart.

2.3.2 Arc segments. When the path seqment is a circular arc, a radius of curvature parameter is computed as

$$r_c = y_e / (1 - \cos\theta_e) \tag{2.5}$$

where y_e and θ_e are the second and third elements of the transformed desired end-point vector,

$$\mathbf{z}_e = \mathbf{B}\,(\mathbf{z}_d - \mathbf{z}_b) \tag{2.6}$$

The parameter r_c has a magnitude equal to the radius of the arc and a sign which is positive for a counter-clockwise turn and negative for a clockwise turn.

The projected position-heading vector along an arc is

$$\mathbf{z}_p = [r_c \sin\theta_p \;,\; r_c(1-\cos\theta_p) \;,\; \theta_p]^T \tag{2.7}$$

where $\theta_p = \theta_f + v_r(t)\tau / r_c$. The reference steering angle and drive speed for arc segments are constants, given by

$$\alpha_r = \tan^{-1}(b/r_c) \quad , \quad \omega_r = v_r / (R\cos\alpha_r) \tag{2.8}$$

where b is the wheelbase of the cart (see Figure 4).

2.3.3 Spline segments. A cubic spline is useful for connecting two line segments with a smooth curve that matches both the position and slope conditions of the line segments at the junction points.* Its general form is a cubic polynomial having four coefficients to match the four end-point conditions:

$$y = Kx^3 + Lx^2 + Mx + N$$

In the base-transformed space in which the RPG computes the reference path, however, only the first two coefficients are non-zero. These coefficients are determined from the elements of the vector $\mathbf{z}_e$ in (2.6) as follows:

$$\begin{aligned} K &= (\tan\theta_e - 2y_e/x_e)/x_e^2 \quad , \\ L &= (3y_e/x_e - \tan\theta_e)/x_e \end{aligned} \tag{2.9}$$

The elements of the projected position-heading vector, $\mathbf{z}_p \equiv [x_p, y_p, \theta_p]^T$, along the spline segment are given by

$$\begin{aligned} x_p &= x_f + v_r(t)\tau\cos\theta_f \quad , \\ y_p &= Kx_p^3 + Lx_p^2 \quad , \\ \theta_p &= \tan^{-1}[(y_p - y_f)/(x_p - x_f)] \end{aligned} \tag{2.10}$$

The reference steering angle and drive speed for spline segments are

$$\alpha_r = \tan^{-1}(b\,c_p) \quad , \quad \omega_r = v_r / (R\cos\alpha_r) \tag{2.11}$$

where c_p is the spline curvature at x_p, given by

$$c_p = (6Kx_p + 2L)/(1 + (3Kx_p^2 + 2Lx_p)^2)^{3/2} \tag{2.12}$$

The position-heading vector for the reference state at time t along the line, arc, and spline segments is obtained by the transformation of the position-heading vector, $\mathbf{z}_p$, back to the original XY-space:

$$\mathbf{z}_r(t) = \mathbf{B}^{-1}(\mathbf{z}_p) + \mathbf{z}_b \tag{2.13}$$

where $\mathbf{B}^{-1}$ is obtained simply by interchanging the off-diagonal terms in (2.2). The position-heading vector given by (2.13), and the programmed speed, $v_r(t)$, described in Section 2.2, form the new reference state, which with the new reference steering angle, α_r, and reference drive speed, ω_r, comprise the data output by the RPG to the cart controller in each control update cycle.

3. CART CONTROLLER

3.1 Cart Coordinates and Kinematics

The coordinates by which the cart is controlled are the front wheel steering angle, α, and the front wheel drive speed, ω. If the location of the cart is measured with respect to the the center of rotation (CR) of the cart, located at the mid-point between the rear odometry wheels, (see Figure 4), the speed, v, heading, θ, and position (x,y) of the cart are related to the steering angle and drive speed coordinates (α,ω) by the kinematic equations

$$\begin{aligned} v &= R\omega\cos\alpha \quad , \quad \dot{\theta} = (R/b)\omega\sin\alpha \\ \dot{x} &= v\cos\theta \quad , \quad \dot{y} = v\sin\theta \end{aligned} \tag{3.1}$$

where R is the radius of the drive wheel, and b is the wheelbase, as shown in Figure 4.*

The task for the cart controller is to steer and drive the front wheel of the cart so that the CR point accurately tracks the reference points "laid down" by the reference path generator. The cart position and velocity measurements necessary for performing this task are obtained from the odometry system indicated in the cart control block diagram shown in Figure 5. The odometry for this system is provided by optical

* If a continuity of curvature at the junction points is also desired a quintic spline may be used.

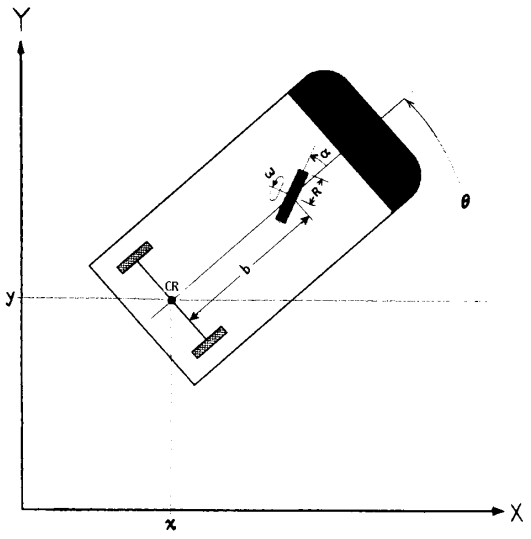


Figure 4. Workspace and cart coordinates.

encoders mounted on two non-load-bearing, knife-edge wheels located near the rear wheels. Since any odometry system has cumulative errors, provision for nullifying these errors at regular calibratrion points in the workspace is needed. The optical ranger can be used to obtain these position fixes.[3]

The cart controller, shown as a shaded block in Figure 5, consists of a path controller and two motor-control units. The motor-control units provide inner control loops for the steering and drive motors, while the path controller provides the outer control loop for path tracking.

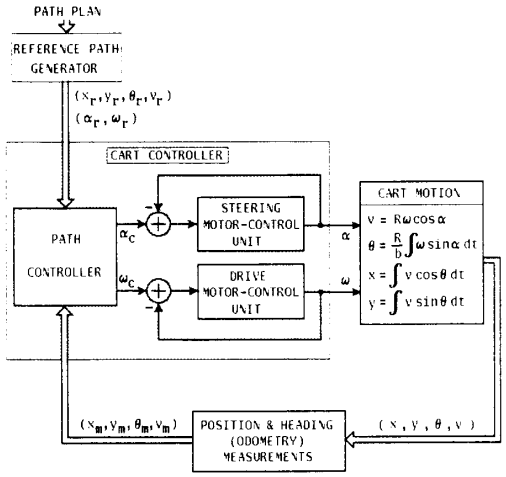


Figure 5. Block diagram of the cart control structure.

* The choice of x,y position determines what point on the cart tracks the reference path, as well as the form of the kinematic equations. The choice to have the CR point track the path, rather than, say, the front wheel, was made for reasons related to path planning and cart maneuverability.

3.2 Motor Error Control

The front wheel steering angle and the front wheel drive speed depend on the motor-load dynamics in response to the control signals to the motors. The purpose of the motor-control units is to compensate for the dynamics in such a way that the motor outputs (α,ω) respond rapidly and smoothly to accurately follow the *commanded* values (α_c,ω_c) output from the path controller. This is done by conventional feedback compensation, based on error signals between the commanded values and the motor outputs values, measured by angle encoders on the motor shafts. To achieve reasonably high-gain error control without causing undesirable, or even unstable response, controller chips containing digital pole-zero compensating filters are used.

The dynamics of the motor-load combination for the two units are represented to a close approximation by the linear model:

$$\begin{aligned} \ddot{\alpha} &= a_1 - H_1\dot{\alpha} \\ \dot{\omega} &= a_2 - H_2\omega \end{aligned} \tag{3.2}$$

where a_1 and a_2 are the control accelerations (torque per unit moment of inertia) developed by the steering and drive motors, respectively, and H_1 and H_2 are the friction coefficients of the two motor-load systems. The digital error signals driving the control units are

$$e_\alpha(k) = C_r(\alpha_c(k)-\alpha(k)) \quad , \quad e_\omega(k) = C_r(\omega_c(k)-\omega(k)) \tag{3.3}$$

where k denotes the k-th sampling period, and C_r is the angle encoder gain (counts/radian). The difference between the two control units is that one is controlling the angle while the other is controlling the angular velocity. Because of the extra integration in the angle control dynamics, the steering angle is the more difficult one to compensate properly.

3.3 Path error control

The path controller generates the steering and drive command signals (α_c,ω_c), using the reference values (α_r,ω_r) from the RPG, and path-errors derived from the measured state of the cart relative to the reference state, as shown in Figure 5. The path error is resolved into four components. As diagrammed in Figure 6, the distance error is resolved into *tangential* error, e_t, and *normal* error, e_n:

$$\begin{aligned} e_t &= (x_r-x_m)\cos\theta_r + (y_r-y_m)\sin\theta_r \\ e_n &= -(x_r-x_m)\sin\theta_r + (y_r-y_m)\cos\theta_r \end{aligned} \tag{3.4}$$

The velocity error is resolved into *heading* error, e_h, and *speed* error, e_v:

$$e_h = \theta_r - \theta_m \quad , \quad e_v = v_r - v_m \tag{3.5}$$

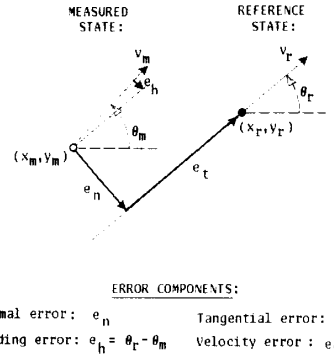


Figure 6. Error components used for path control.

Given the reference values α_r, ω_r and these error components, the command steering and drive signals generated by the path controller in each update cycle are:

$$\alpha_c = \alpha_r + C_1 e_n + C_2 e_h \quad , \qquad (3.6)$$
$$\omega_c = \omega_r + C_3 e_t + C_4 e_v$$

Note that when the cart is following the reference path without error, the command steering and speed are equal to their reference values.

The weighting constants, $C_1,...,C_4$ in (3.6), are chosen to optimize the overall path-tracking performance of the cart. As with the motor control units, it was the steering control part of the outer loop design which was the most sensitive to the choice of the control weights in (3.6). Because the cart speed affects the optimum steering gains through the kinematic relations (3.1), work is currently under way to find an appropriate functional dependence on speed for these weights, such that path control is maintained at near optimum during the critical docking segments, when the speed is being reduced to zero.

4. RESULTS

4.1 Implementation

The reference path generator and cart controller procedures outlined in Sections 2 and 3 have been implemented both in a simulation and on a real vehicle. The simulation program is written in the C language and runs on a VAX11/750 and Silicon Graphics IRIS workstation. The current version of the simulation program requires as input a file containing the path plan in a format like that shown in Figure 3(b), and a file containing system and run-control parameters. It produces as output a run parameter file, two files of system variables as a function of time, and a file containing reference path and cart motion data for the cart animation program. The animation program shows the cart moving in response to the steering and drive commands, along with the reference path and the layout of the workspace obstacles and corridors.

The C++ language[9] is used to program the experimental cart. Consequently, only minor changes in the simulation code are required for it to execute directly on the vehicle. This language compatability has facilitated rapid transfer of program developments between the simulation environment and the actual robot cart. The simulation also permits the testing of the cart's performance in workspace conditions that cannot be duplicated in the laboratory environment.

Program development for the vehicle is performed under UNIX.* The compiled code is then downloaded via ethernet to the onboard real-time system consisting of a Multibus-based MC68020 microprocessor running a real-time UNIX-derived executive called NRTX[10].

During program execution, the cart is completely autonomous. Any interaction is performed using a portable battery-powered Data General One PC acting as a terminal emulator. Diagnostic data from the cart may be temporarily saved in the PC and later downloaded to the UNIX host for analysis.

4.2 Motor Control Units

The mathematical model of the cart dynamics used in the simulation is a simple approximation of the actual cart dynamics. Nevertheless, it was found to be an adequate representation for obtaining good agreement of the simulated response with the observed response of the cart steering angle and drive speed to changes in their command values, once the model parameters were properly adjusted.

The model for the steering motor-control unit involves three unknown parameters: the friction coefficient H_1, the motor torque/inertia gain factor G_I, and the motor acceleration limit A_1. The other parameter values needed for both the motor control units on the cart and in the simulation program are the digital filter gain, C, and zero and pole values, A and B. To obtain appropriate parameter values, an iterative design procedure was used, in which initial estimates of the unknown parameters were made. The small-error (linear) closed-loop system model was then analyzed to determine loop gain and pole-zero values that provided good transient response. These values were used in the cart controller chip[11] to get experimental data on step response to different command steering angles. The friction and motor torque parameters in the simulation model were then adjusted to get similar step responses. After a second iteration of this procedure, a good agreement between the simulation model and the actual cart response was obtained. Figure 7 shows the steering angle response times, for positive and negative steering commands in the cart motor control unit, compared with the response times obtained from simulation runs. The response times were the times at which the wheel angle, α, reached 90% of the command angle, α_c. Also listed in Figure 7 are the design parameters obtained from this procedure.

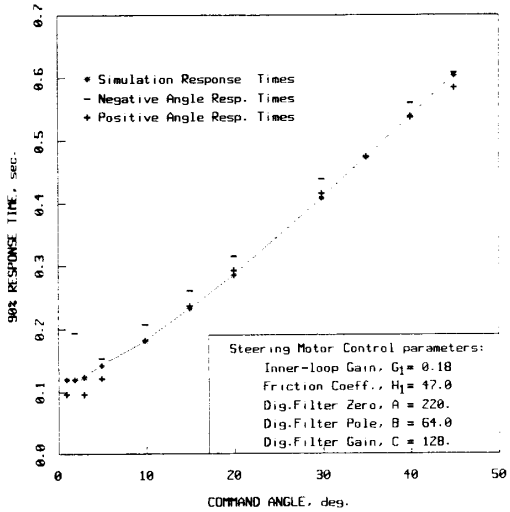


Figure 7. Steering response times for a range of commanded steering angles.

The response time data in Figure 7 indicate a good agreement between the cart and the simulation over the full ±45 degree range of possible steering angles, in spite of the simple model (3rd-order linear system dynamics with hard acceleration limits) used in the simulation. At large steering angles, the motor control is at the acceleration limit during most of the response. At small steering angles, the acceleration limit is a less significant factor in the response, but nonlinear friction effects may become more significant, which probably accounts for why the positive and negative angle response times for the cart diverge somewhat. Note that the simulation data for angles below about 4 degrees lies between the positive and negative experimental data, at a relatively constant response time of about 0.12 s.

The drive motor control unit has essentially the same elements as the steering motor unit, the main difference being that velocity-error rather than angle-error is used to control the drive motor. This reduces the control loop to a 2nd-order system, which is simpler to stabilize. As a result, the digital filter gain for the drive unit could be increased to four times that used in the steering unit (with approximately the same pole-zero values) without causing oscillations in the speed response. The present cart drive control yields a speed response rate of about 10 in/s/s. This is adequate for the cart to track changes in the path reference speed, which are currently programmed at 6 in/s/s.

4.3 Path Controller

4.3.1 Simulation Results. The reference path procedures described in Section 2 are essentially independent of the cart control, once the

* UNIX is a trademark of AT&T Bell Laboratories

physical constraints, such as the angle, velocity, and acceleration limits discussed above have been properly taken into account. The path plans represent feasible, collision-free paths in terms of such constraints as the minimum turning-radius of the cart (about two feet) and the physical area swept out by the cart as it moves along the prescribed path.

Given a feasible path to track, then, it is the task of the path controller, described in Section 3.3, to keep the tracking errors small enough to ensure smooth, collision-free motion along the paths. This task is complicated by the nonlinear kinematics (3.1) linking the cart position and velocity to the steering angle and drive speed and by the nonlinear transformations in the computation of the error components (3.4). At this stage of the study, a reasonably good design for the path controller has been achieved through a combination of small-error analysis and stabilization methods on the overall closed-loop system, followed by verification and fine-tuning of the control algorithm, using the full nonlinear system simulation program.

When the control weights in the control law (3.6) are properly adjusted, a satisfactory tracking of the reference path is achieved, as indicated by the simulation run shown in Figure 8. Because the scale of the overall trajectory is large compared to the path tracking errors, it is difficult to distinguish the actual cart path from the reference path in Figure 8. The cart speed in this run was 4 in/s. The errors in the position measurement system were set to zero in this simulation run, so any path errors were due to control errors, rather than measurement errors. The position control errors for the run shown in Figure 8 were less than 1/4 inch normal to the path, and less than 3/4 inch tangential to the path, while the heading error did not exceed 3 degrees. At the three stop points (A, B, and C), the normal and tangential errors were less than .06 and .53 inch, respectively, while the heading errors were less than half a degree.

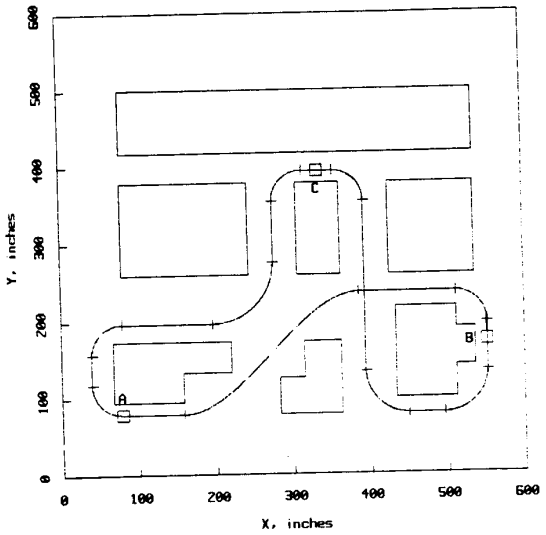


Figure 8. Example of a simulation run.

4.3.2 Experimental Results. The test runs for the experimental cart have so far been limited to a 14 by 12 foot "workspace" in the robot laboratory. Hence the test paths have been mostly figure-eight patterns, with tight turns in which the steering angle is at or near the hard limit of 45 degrees. Figure 9 illustrates the performance of the cart over such a test path. The reference trajectory is the solid line, while the cart's measured trajectories are shown for three operating speeds: 2 in/sec (dotted), 4 in/sec (dashed) and 6 in/sec (large dashed). The control algorithm clearly performs better at the slower speeds as the gains have been optimized for the slower speeds. The performance degradation at

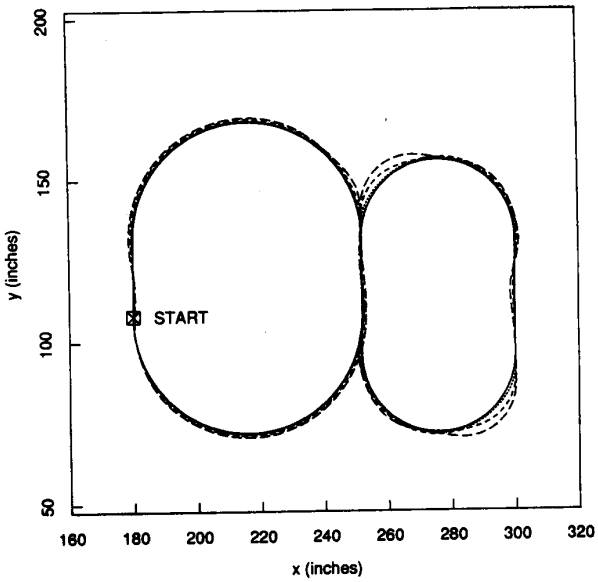


Figure 9. Measured path trajectories for the experimental cart.

higher speeds is usually acceptable since high speed transit maneuvers typically do not require as much precision as low speed docking maneuvers. Some overshoot at the beginning of the arc segments is unavoidable, since the control cannot make the steering wheel follow the angle discontinuity at the transition point between line and arc segments. Furthermore, as the vehicle enters the tighter loop on the right, the overshoot at high speeds is exaggerated due to the steering wheel hitting the hard limit stop.

The discontinuity in steering angle could be avoided if we were to control the point directly under the drive wheel rather than the midpoint between the rear wheels. However, controlling the orientation of the vehicle becomes more complex, as does path planning. A further alternative would be to replace the arc segments with segments of curves which are continuous in curvature[12].

5. CONCLUSION

Guidance and control procedures for enabling a robot cart to accurately and autonomously navigate around a structured environment, given a set of path data from a global path planner, have been described. These procedures have been implemented in an experimental robot cart and in a computer simulation that closely models the physical characteristics of the experimental cart. The use of an onboard reference path generator (RPG) has a number of advantages over generation of the path by the global path planner: (1) it reduces greatly the amount of data needed from the path planner, so even a very low data-rate, intermittant link to the cart is sufficient; (2) it provides a convenient separation between the path guidance and the cart control logic; (3) it supports fail-safe operation through continuous monitoring of the cart's control status; and, (4) it allows the cart relative independence in evolving the detailed path in time, once it is commanded to start. For example, unexpected obstacles detected by the onboard sensors can immediately trigger the local path planner to insert a stop segment at the current path location, and then generate a startup segment to continue the cart along the original path. This logic has been tested in the simulator and implemented on the cart, so that if a person is in the cart's path, for example, it stops until the person moves out of the way, then starts up automatically and continues along the path. Fixed obstacles within the structured environment are known to the path planner. If an unknown obstacle were encountered by the cart, the vehicle would stop, relay the obstacle data back to the global path

planner and await an updated path plan.

As mentioned in Sec. 3.3, work is under way to find an appropriate functional dependence of the control weights in (3.11) on speed, in order to optimize the positioning accuracy of the cart during docking maneuvers, where the cart's position and heading as the speed reaches zero are more critical than while the cart is moving. The current performance with fixed weights gives docking accuracies of less than an inch in position and a few degrees in heading, provided the final path segment is a line several cart lengths long. The aim of the parametric control design is to reduce docking errors to less than about 1/10 inch and 1/10 degree, for arcs as well as line approach-segments.

The experience gained so far in this study indicates that the basic approach described above for guidance and control of a robot cart is sound. The separation of navigation and control functions in the RPG and the cart controller units provides a robust method for maintaining accurate tracking of any feasible path, once the controller unit has been properly adjusted to stabilize the closed-loop dynamics of the path control loop. Because the computer language of the simulation is the same as that used by the cart's onboard computer, program improvements developed in the simulation environment can be readily transferred to the onboard computer for experimental verification. The simulation environment includes a graphics workstation on which a wide range of factory and office layouts can be set up. A path-planning program[5] generates minimum-turn, collision-free paths, allowing the cart to navigate between any selected points in the layout. These can then be input to the cart simulation program for verification of cart performance over the prescribed paths.

The work reported in this and the companion papers[3]-[5] is continuing with the goal of achieving a versatile, autonomous robot cart that will be useful for transport of materials in reasonably well-structured environments, such as factories, warehouses, and offices.

5.1 Acknowledgements

The authors wish to thank W. J. Kropfl, G. L. Miller, E. R. Wagner, and G. T. Wilfong, for their help and useful discussions during this work.

REFERENCES

1. *Proc. of the Second International Conf. on Automated Guided Vehicle Systems*, Stuttgart, W. Germany, 7-9 June 1983, H. J. Warnicke, Ed., North Holland Publ. Co., 1983.
2. T. Tsumura, "Survey of Automated Guided Vehicle in Japanese Factory," *Proc. 1986 IEEE International Conf. on Robotics and Automation*, San Francisco, CA, Vol. II, p. 1329.
3. I. J. Cox, "Blanche: An autonomous robot vehicle for structured environments," *Proc. 1988 IEEE International Conf. on Robotics and Automation*, Philadelphia, PA.
4. G. L. Miller and E. R. Wagner, "An Optical Rangefinder for Autonomous Robot Cart Navigation," *Proc. IECON/SPIE '87*, Cambridge, MA, Nov. 1-6, 1987.
5. G. T. Wilfong, "Motion Planning for an Autonomous Vehicle," *Proc. 1988 IEEE International Conf. on Robotics and Automation*, Philadelphia, PA.
6. L. E. Dubins, "On Curves of Minimal Length With a Constraint on Average Curvature, and With Prescribed Initial and Terminal Positions and Tangents", *American Journal of Mathematics* **79**, 1957 pp. 497-516.
7. T. Hongo *et al.*, "An automatic guidance system of a self-controlled vehicle," *IEEE Trans. on Ind. Electronics*, Vol. IE-34, No. 1, 1987, pp. 5-10.
8. S. Uta *et al.*, "An implementation of michi-a locomotion command system for intelligent mobile robots," *Proc. 15th ICAR*, 1985, pp. 127-134.
9. B. Stroustrup, *The C++ Language*, Addison Wesley, New York(1986).
10. D. A. Kapilow, "Real-Time Programming in a UNIX Environment," *1985 Symposium on Factory Automation and Robotics*, Courant Institute of Mathematical Sciences, New York University, p. 28.
11. "General Purpose Motion Control IC HCTL-1000" Hewlett Packard Technical Data Sheet, 1985.
12. W. L. Nelson, "Continuous Steering-Function Control of Robot Carts", Bell Labs Internal Document, Oct. 1987, submitted for publication, 1988.