GENERAL DESIGN OF A DRIVEN WHEEL MOBILE ROBOT WITH A PATH STRORED IN MEMORY

Mircea Nițulescu
University of Craiova
Faculty of Automation, Computers and Electronics
5 Lăpuş, 1100, Craiova, Romania
nitulescu@robotics.ucv.ro

Abstract: This paper describes a control system for a driven wheel autonomous mobile robot designed to operate in well-structured environments. The path is stored in the memory and given by a global path planner generator, as a concatenation of basic and pre-defined elements. So, the onboard navigation system includes an error-feedback controller and a mobile robot location sensing using odometry. There is a convenient separation between the path guidance and logic control. Under normal operating conditions, the controller ensures that the errors between the measured and reference states are small. A study case and associated path plan data stored in the external memory for this case is included.

Key words: Mobile robot, Modeling, Control design, Navigation.

INTRODUCTION

In manufacturing applications, mobile robots have received a little attention. The preponderance of the current research in mobility surrounding Flexible Manufacturing Systems (FMS) involves the use of Automatic Guided Vehicles (AGV's). These vehicles simplify the problem of navigation by restricting their path to predetermined routes, which are typically demarcated by striping the floor in some manner or by using buried cables. A major issue is just how "flexible" such systems are (Biekert 1993).

AGVs are attractive because they provide labor savings, efficiency and they reduce damage to transport materials. However, they have two significant disadvantages.

First, they utilize a limited drive-path. An external and fixed path (typically either buried cables or chemical stripes painted on the plant floor) guide commercially available AGV's. These AGV's treat the path as a closed loop, which is usually traversed in one direction, with mandatory stops at preplanned location. The workplaces and production lines of a FMS must be designed and structured to accommodate the AGV's drive-path, thereby sacrificing some of the flexibility of the system from the start. Modifications of production lines in an existing FMS are made at great expense. To increase the vehicle's intelligence, the alternative is to modify the workplace for providing perceptual information for AGV's. Typical solutions (Nițulescu 2003) include using magnetic markers, the placement of retro-reflective landmarks (ultrasonic sometime), infrared beacons or an accurate dead-reckoning system. In all these methods, world modeling is generally kept to a minimum because travel is usually severely restricted within the FMS.

Second, AGV's also have limited interaction with the workstations (Groover 1987). They only transport material to and from workstations. Usually, other material-handling equipment attached to the workstations assures the load transfer between an AGV and the workstation operation envelope. The additional and important problem of docking (the correct position and orientation with respect to the workstation in order to meet tolerances for interaction) is usual solved by the external path.

There are a number of different types of AGV's, all off which operate according to the preceding description. Historically, the usually types in industrial medium can be classified (Groover 1987; Lee 1993; Nițulescu 2003) as follows:

Driverless trains were the first type of AGV's to be introduced. This type consist of a towing vehicle that pulls one or more trailers to form a train and it is use in applications where heavy payloads must be moved large distances in warehouses or factories with intermediate pickup and drop-off points along the route. An example of this type of AGV is included in Figure 1.

AGV's pallet trucks are used to move palletized loads along predetermined routes (Figure 2). In the typical application the vehicle is backed into the loaded pallet by a human worker, who steers the truck and uses its forks to elevate the load slightly. Then the worker drives the pallet truck to the guide-path, programs its destination, and the vehicle proceeds automatically to the destination for unloading.

AGV's unit load carriers are used to move unit loads from one station to another station. They are often equipped for automatic loading and unloading by means of powered rollers, moving belts mechanized lift platforms or other conventional devices (Figure 3). The assembly line AGV's is designed to carry a partially completed subassembly through a sequence of assembly workstations to build the final product (Figure 4).

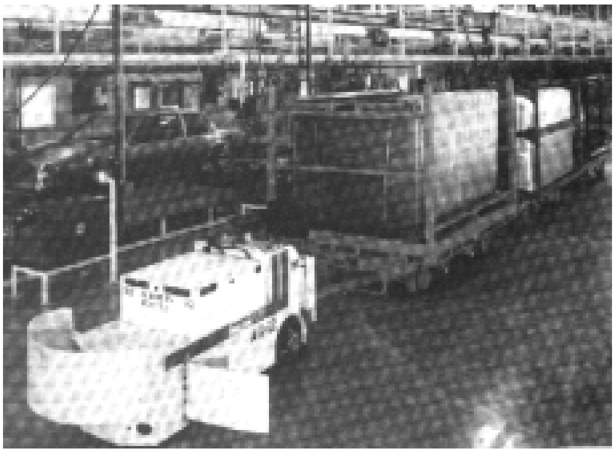

Figure 1. AGV driverless trailer type.

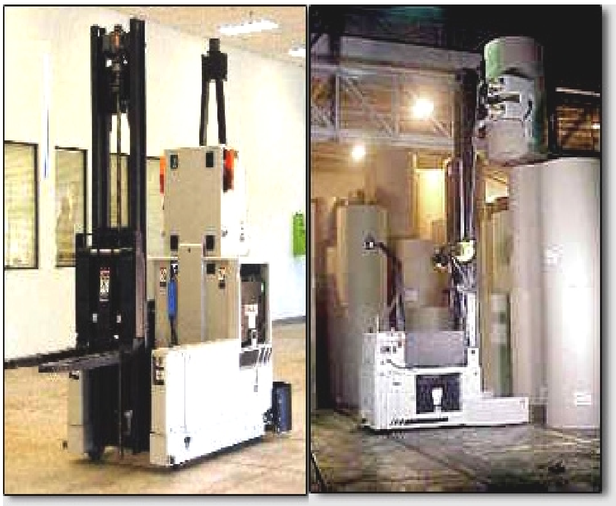

Figure 2. AGV forklift vehicle type.

Figure 3. AGV unit load carrier type.

Figure 4. AGV unit load carrier in application.

To eliminate the disadvantage of the external drive-path, specific of the AGV's (to see for example Figure 4), a solution is to design a mobile robot which have a trajectory stored in their memory. The trajectory is stored as a set of pre-defined elements, namely here segments, each of them being a task for local navigation of the mobile robot. The global trajectory must be easy modifying using an external computer, in concordance with the workspace layout and global path planner decisions. Globally, the robot must be capable of navigating in structured environments such as factories or offices, without external guide-path (Kriegman 1987).

The onboard control system uses odometry to provide the position and heading information needed to guide the mobile robot along specified paths between stations. However, to compensate the cumulative errors introduced by odometry, periodical absolute localization are needed to reset these errors. The must common solution uses laser or ultrasonic absolute localization and predefined landmarks placed into the workspace.

GENERAL DESIGN

In this paper, the general model is a driven wheel mobil robot (Figure 5). This is a tricycle mechanical configuration with a single front wheel (FW), which serves both for steering and driven the mobile robot, and two passive load-bearing rear wheels (RW).

The coordinates by which the mobile robot is controlled are the front wheel steering angle (α) and the front wheel drive speed (ω).

The task for the controller is to steer and drive the front wheel of the mobile robot so that the characteristic point (CP), which is placed in the middle axis between the rear wheels, accurately tracks the reference points given by reference path generator of the mobile robot.

The mobile robot position and velocity measurements necessary for performing the navigation task are obtained from the odometry system, which is provided by optical encoders mounted on two-load bearing rear wheels (Nițulescu 1995).

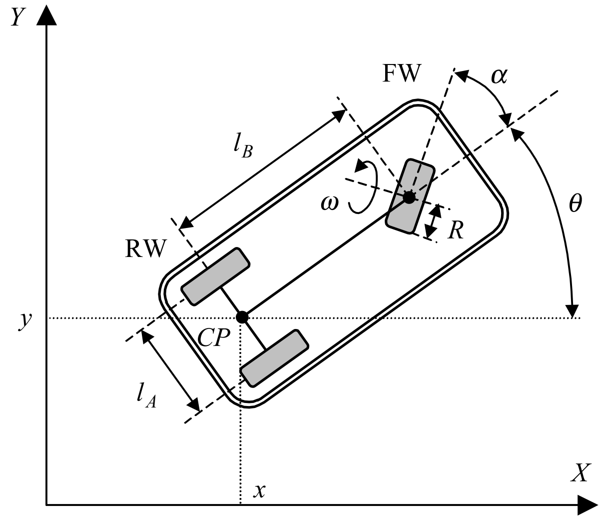

Figure 5. Mobile robot in a tricycle mechanical configuration.

THE PATH CONTROL SYSTEM

Figure 6 present the overall path control system for this type of mobile robot. It functions as an onboard navigator that continually generates an appropriate reference state for the cart controller.

In a minimal structure, the reference state must be determined by three factors: the global path plan, the dynamic capabilities of the cart in acceleration and deceleration, and the current operating status of the cart.

The global path planner gives a data list, each line of which specifies a segment of the path. In a minimal structure, the segment specification includes two elements: the segment type and the desired state for the mobile robot at the end of the current type segment. It is usual that an external computer assures the global path planner and transfers the data list to the local path planner, placed on-board of the mobile robot.

The segment type must be in accord with the mobile robot capabilities. For this mechanical type of mobile robot, to generate complex trajectories it is basically necessary a line type segment, an arc line segment and spline segment. The spline segment is useful especially for lane-change maneuvers, because one spline segment can replace an arc-line-arc sequence (Nițulescu 2003).

At each on-board control update cycle, the Path Control System calculates the reference state on the path, the reference for the steering angle and for the drive speed and the distance to the end of the current path segment. When the distance to the end-point of the current segment becomes less than the distance between the preceding reference points, the Path Control System requests the next from the off-board Global Path Planner and begins generating reference state along the new segment. For start-up from a stop and for speed changes across the path segments, a ramp change in reference speed must be programmed using the recommended acceleration and deceleration.

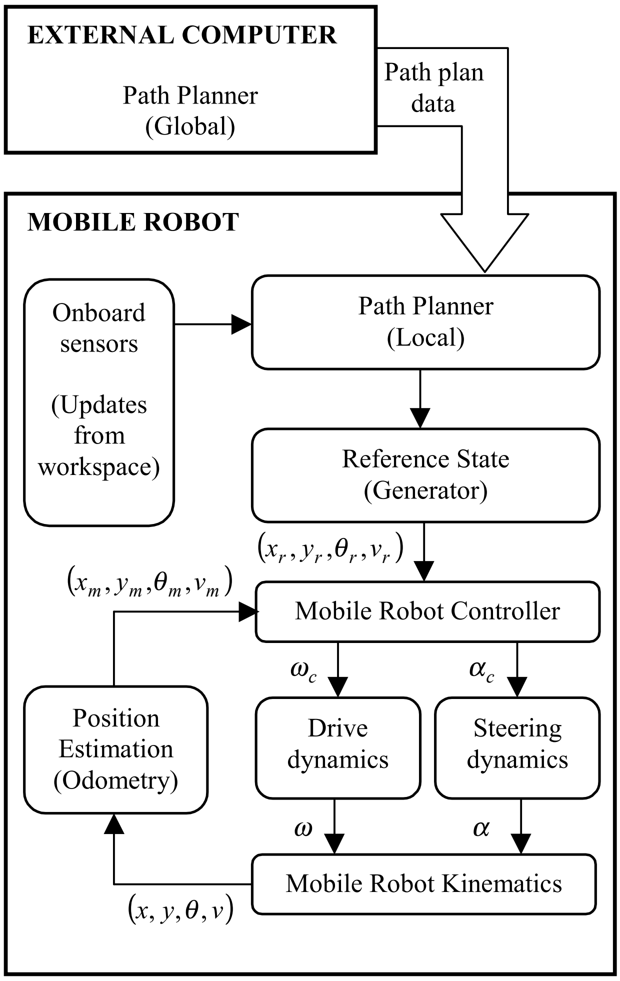

Figure 6. A block diagram for the general path control system of the mobile robot.

The reference position and heading on the desired path at the current update time t depends on the position and heading at $(t-\sigma)$, the current path-segment type and the reference speed $v_r(t)$. It is necessary to introduce some notation:

- $z_d = [x_d, y_d, \theta_d]^T$, the desired position-heading vector at the end point at the current segment;
- $z_b = [x_b, y_b, \theta_b]^T$, the beginning position-heading vector for the current segment (the z_d of the previous segment);
- $z_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$, the reference position-heading vector at the update time (t);
- z_f, the former 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);

$$z_f = Rot(\theta_b) \cdot [z_r(t-\sigma) - z_b]^T =$$

$$= \begin{bmatrix} \cos\theta_b & \sin\theta_b & 0 \\ -\sin\theta_b & \cos\theta_b & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[z_r(t-\sigma) - z_b\right]^T \quad (1)$$

- $z_p = \left[x_p, y_p, \theta_p\right]^T$, the projected position-heading vector in this transformed coordinate space;
- (α_r) and (ω_r), the references for steering angle and for drive rotational speed.

The three type segments characteristics are:

Line Segments Type

In this case, the incremental path motion is a sequence of $(v_r \cdot \sigma)$ steps along the transformed X-axis. The projected position-heading vector along a line segment is:

$$z_p = \left[x_f + v_r(t)\cdot\sigma, 0, 0\right]^T \quad (2)$$

and:

$$\alpha_r = 0,\ \omega_r = v_r / R \quad (3)$$

where (R) is the radius of the FW (see Figure 5).

Circular Arc Segments Type

The radius of curvature parameters is:

$$r_c = y_e / (1 - \cos\theta_e) \quad (4)$$

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

$$z_e = Rot(\theta_b)\cdot(z_d - z_b) \quad (5)$$

The parameter (r_c) has a positive sign for counter-clockwise turn and negative for a clockwise turn.

The projected position-heading vector along an arc is:

$$z_p = \left[r_c \cdot \sin\theta_p, r_c \cdot (1 - \cos\theta_p), \theta_p\right]^T \quad (6)$$

where:

$$\theta_p = \theta_f + \sigma \cdot v_r(t) / r_c \quad (7)$$

Finally (see Figure 5 for the mechanical parameter l_B):

$$\alpha_r = \tan^{-1}(l_B / r_c),\ \omega_r = v_r / (R \cdot \cos\alpha_r) \quad (8)$$

Spline Segments Type

Generally, a spline-arc is a cubic polynomial having four coefficients:

$$y(x) = Ax^3 + Bx^2 + Cx + D \quad (9)$$

In the base-transformed space, finally (Nitulescu 2003):

$$A = (\tan\theta_e - 2y_e/x_e)/x_e^2$$
$$B = (3y_e/x_e - \tan\theta_e)/x_e$$
$$C = D = 0 \quad (10)$$

The elements of the projected position-heading vector $z_p = \left[x_p, y_p, \theta_p\right]^T$ along the spline segment are given by:

$$x_p = x_f + v_r(t)\cdot\sigma\cdot\cos\theta_f$$
$$y_p = A\cdot x_p^3 + B\cdot x_p^2$$
$$\theta_p = \tan^{-1}\left[(y_p - y_f)/(x_p - x_f)\right] \quad (11)$$

Finally (see Figure 5):

$$\alpha_r = \tan^{-1}(l_B \cdot s_p)$$
$$\omega_r = v_r / (R \cdot \cos\alpha_r) \quad (12)$$

where (s_p) is the spline curvature at (x_p):

$$s_p = \frac{6\cdot A\cdot x_p + 2\cdot B}{\left[1 + (3\cdot A\cdot x_p^2 + 2\cdot B\cdot x_p)^2\right]^{3/2}} \quad (13)$$

The position-heading vector for the reference state at time (t) along the line, arc or spline segments is obtained by the transformation of the position-heading vector (z_p) back to the original XY-space:

$$z_r(t) = Rot^{-1}(\theta_b)\cdot z_p + z_b \quad (14)$$

The position-heading vector $z_r(t)$ and the programmed speed $v_r(t)$ form the new reference state, which with the new reference steering angle (α_r) and the reference drive speed (ω_r) comprise the data output by the Reference State Generator to the mobile robot controller in each control update cycle.

MOBILE ROBOT CONTROLLER

The coordinates by which the mobile robot is controlled are the front wheel steering angle α and the front wheel drive speed ω. If the location of the mobile robot is measured with respect to the characteristic point CP (see Figure 5) located at the mid-point between the rear odometry wheels RW, the cinematic equations are:

$$v = R\cdot\omega\cdot\cos\alpha$$
$$\dot{x} = v\cdot\cos\theta$$
$$\dot{y} = v\cdot\sin\theta$$
$$\dot{\theta} = (R/l_B)\cdot\omega\cdot\sin\alpha \quad (15)$$

Since any odometry system has cumulative errors, a regular calibration point in the workspace is needed. The general block diagram of the onboard mobile robot control structure is shown in Figure 7.

Considering the case included in Figure 8, the errors between the reference state and the measured state for the driven wheel mobile robot can be grouped in four components (Koh 1994):

- the normal error (e_n);
- the tangential error (e_t);
- the heading error (e_h), $e_h = \theta_r - \theta_m$;
- the velocity error (e_v), $e_v = v_r - v_m$.

These errors can be evaluated from the next equations:

$$e_n = -(x_r - x_m)\cdot\sin\theta_r + (y_r - y_m)\cdot\cos\theta_r$$
$$e_t = (x_r - x_m)\cdot\cos\theta_r + (y_r - y_m)\cdot\sin\theta_r$$

$$e_h = \theta_r - \theta_m$$
$$e_v = v_r - v_m \qquad (16)$$

The measured values $(x_m, y_m, \theta_m, v_m)$ are obtained by odometry (Nishizawa 1995) and the reference path generator gives the reference values $(x_r, y_r, \theta_r, v_r)$. The path controller generates finally the steering α_c and driven ω_c signals for Steering Motor Control Unit and respectively for Drive Motor Control Unit.

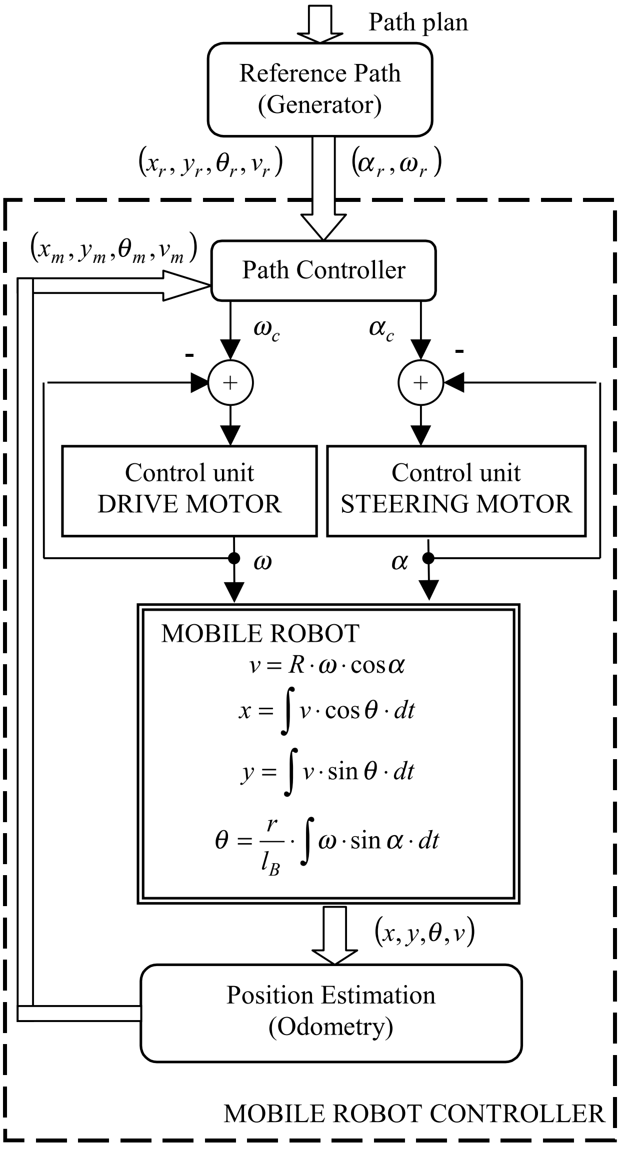

Figure 7. Mobile robot controller, the blocks diagram.

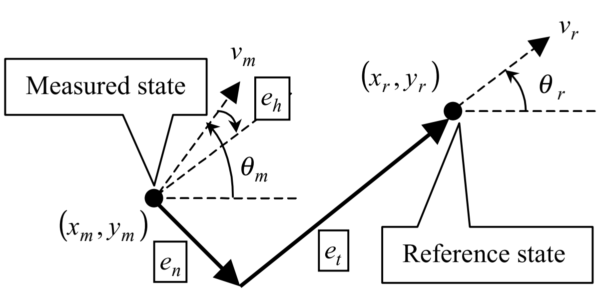

Figure 8. Error components in path control.

Because the Reference Path Generator gives the reference values and the errors can be determinate by the equations (16), the command steering and drive signals generated by the path controller in each update cycle are:

$$\alpha_c = \alpha_r + K_1 \cdot e_n + K_2 \cdot e_h$$
$$\omega_c = \omega_r + K_3 \cdot e_t + K_4 \cdot e_v \qquad (17)$$

where the constants set (K_1, K_2, K_3, K_4) is chosen to optimize the overall path-tracking performance of the driven wheel mobile robot (Lee 1993). Note that when the mobile robot is following the reference path without error, the command steering and speed are equal to their reference values.

A STUDY CASE

Figure 9 shows a possible path plan for the driven wheel mobile robot between two stations included in a flexible manufacturing system. (IP) represents the initial position of the mobile robot, near the first station noted PROC. AUT. (a station for an automatic processed operation) and (FP) represents the final position of the mobile robot, near the second station noted ASBY. AUT. (a station for an automatic assembly operation).

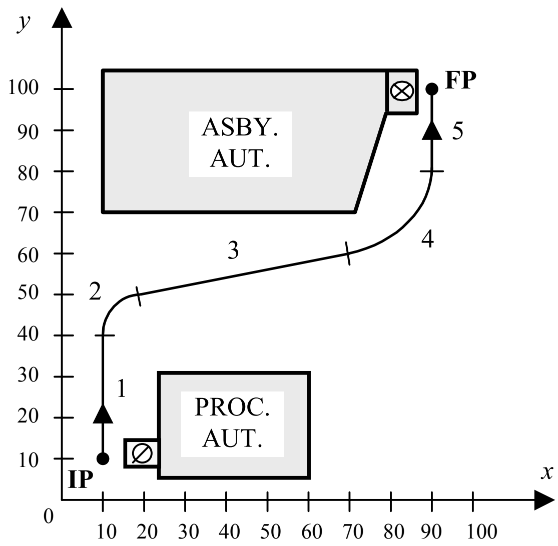

Figure 9. The trajectory composed by a concatenation of different path segments.

In this study case, the path plan contains five segments: tree line segments separated by two arc segments. The corresponding path data plan is presented in Figure 10 and it contains five lines of numbers.

Seg. No.	Segment type	Desired End-State			
		$x_d\,[m]$	$y_d\,[m]$	$\theta_d\,[grd]$	$v_d\,[m/s]$
1	1 L (line)	10	10	+ 90	0,5
2	2 CR (circular arc)	10	40	+ $\tan^{-1}(1/5)$	0,25
3	1 L	70	60	+ $\tan^{-1}(1/5)$	0,5
4	2 CR	90	80	+ 90	0,35
5	1 L	90	100	+ 90	0

Figure 10. The Path plan data, output gives by the external Global Path Planner.

Note that segments 2, 3 and 4 of the path shown in this study case could be represented by one spline segment, which can replace an arc-line-arc sequence. In consequence, implementing spline segments gives facilities for robot maneuvers.

An illustration of the reference points generated for the path shown in Figure 9 is given in Figure 11. The Path Control System computes at each control update cycle the reference state on the path, the reference steering angle and drive speed, and the distance to the preceding reference point. When the distance to the end point of the segment become less than the distance between the preceding reference point, the Path Control System requests the next segment in the path and begins generating reference states along the next segment.

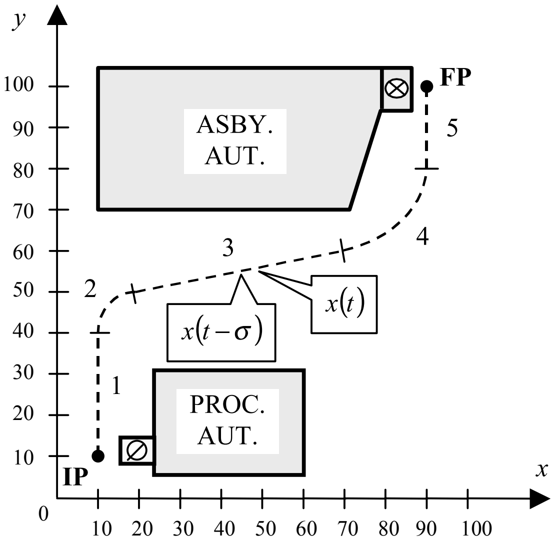

Figure 11. Reference points generated along the global trajectory.

CONCLUSIONS

The main advantage of the reference path guidance scheme over point-to-point guidance schemes (Sousa 1996) is that the onboard system is always providing a current point on the path where the mobile robot should be, with smooth transitions across path segment boundaries. Because of measurement errors, a point-to-point control scheme may not detect the point at which the control algorithm should switch to a new target point, and hence may fail to make proper transition from one path segment to the next.

The reference path guidance scheme also provides a means for onboard monitoring of the operating status of the mobile robot. 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 exceed some established limits, then the mobile robot must be functioning abnormally. The controller can testy for abnormally errors and signal the local path planner to take remedial action. In consequence, system malfunctions such as failure of motors or odometry sensors can be detected in time for safely stopped.

An other advantage of the reference path guidance scheme is that it provides a convenient separation between the path-guidance logic and the controller logic, which must be tuned to the dynamic characteristics of the driven wheel mobile robot. In consequence, once properly adjusted, the controller design remains fixed throughout all the different operating situations that may arise along the trajectory.

REFERENCES

Andrea, B. and G. Bastin (1991). Modeling and control of non-holonomic wheeled mobile robots. In: *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, SUA.

Arkin R. and Murphy R. (1990). Autonomous navigation in a manufacturing environments, In: *IEEE Transactions on Robotics and Automation*, Vol. 6, No. 4.

Biekert R. (1993). *CIM technology: Fundamentals and applications*. Goodheart-Willcox Company, USA.

Groover M. (1987). *Automation, production systems and computer integrated manufacturing*. Part V, Prentice Hall International Inc., USA.

Hong S. and S. Kim (1996). Local motion planner for non-holonomic mobile robot in the presence of unknown obstacles. In: *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, SUA.

Koh K. (1994). A path tracking control system for autonomous mobile robots. *Mechatronics,* No. 4.

Kriegman D. (1987). A mobile robot: sensing, planning and locomotion. In: *Proceedings of IEEE Transaction. on Robotics and Automation*, Detroit, USA.

Krogh B. and C. Thorpe (1986). Integrated path planning and dynamic steering control for autonomous vehicles, In: *Proceedings of IEEE International Conference on Robotics and Aut.,* San Francisco, USA.

Lee S. and Williams J. (1993). An automatic guidance system of an autonomous vehicle, the trajectory generation and the control algorithm, *Robotica*, Vol. 11.

Nishizawa T. and A. Ohia (1995). An implementation of on board position estimation for a mobile robot, In: *Proceedings of IEEE International Conference on Robotics and Automation,* Nagoya, Japan.

Nițulescu M. (1995). Odometric measurements and navigation following of the reference trajectory. In: *Proceedings of 10th International Conference on Control Systems*, Bucharest, Romania.

Nițulescu M. (1999). Modelling and path-tracking control of a differential wheeled mobile robot, In: *Proceedings of 18th IASTED International Conference on Modelling, Identification and Control,* Innsbruck, Austria.

Nițulescu M. (2003). Sisteme robotice cu capacitate de navigatie, Editura Universitaria, Craiova, Romania.

Sousa J., Pereira F. (1996). On the design and implementation of a control architecture for a mobile robotic system. In: *Proceedings of IEEE International Conference on Robotics and Aut.*, Minneapolis, SUA.

Tournassound P., Jehl O. (1988). Motion planning for a mobile robot with kinematic constraint, In: *Proceedings of IEEE International Conference on Robotics and Automation*, Lyon, France.