Path following, real-time, embedded fuzzy control of a mobile platform Pioneer 3-DX

A. FILIPESCU[1], I. SUSNEA[1], AL. STANCU[1], G. STAMATESCU[2]
[1]Department of Automation and Industrial Informatics
University "Dunarea de Jos" of Galati
Domneasca, 47, Galati, 800008
[2]Faculty of Automation and Computers
Polytechnic University of Bucharest
Splaiul Independentei, 313, Bucharest, 060042
ROMANIA

Abstract: - This paper presents a microcontroller implementation of a fuzzy logic control algorithm applied to the mobile platform. As mobile platform, Pioneer 3-DX has been chosen. A robotic manipulator, Pioneer 5-DOF Arm is on the platform. Pioneer 3-DX is a wheeled mobile robot (WMR) with two driving wheels and a caster wheel. The design and implementation of the fuzzy controller is described. Both speed and position are achieved using the same real-time fuzzy controller. The description starts with an example for a circular path following, then presents an implementation of the algorithms for following a generalized curve path.

Key-Words: - Pioneer 3-DX mobile platform, Embedded real-time fuzzy control, Wheeled mobile robot, Path following.

1 Why fuzzy control?

Fuzzy logic is now widely used to solve some of the most important problems in mobile robots and autonomous vehicle control, such as trajectory tracking, or obstacle avoidance. However, the vast majority of the existing implementations, count on powerful computing equipment, far beyond the capabilities of a simple microcontroller (see, for example, Filipescu, A.; Dugard, L.; Dion, J.-M., 2003 [1], or Thongchai S. and Kawamura K., 2000 [2]).

While the theoretical aspects of this topic are very well covered by the existing literature (Susnea I, Mitescu M. [3], or Ahmad I. [4]), relatively few papers describe practical fuzzy solutions for embedded systems, and these are mainly based on DSPs (see Baturone I, Moreno-Velo F.J., Sanchez-Solano S., Blanco V. [5]).

Therefore, we have focused on implementing advanced control algorithms on low-cost embedded devices, built around simple microcontroller structures. This algorithm described in this paper was actually implemented and tested on a simple, 8-bit microcontroller, attached to the Pioneer3-DX robot (fig.1).

The paper is structured as follows: Section 2 presents the structure of the Pioneer 3-DX mobile robot and additional hardware. It includes the kinematic variables associated with the WMR discrete. Section 3 describes the communication protocol. The kinematic model of the Pioneer 3-Dx is presented in Section 4. In Section 5, the fuzzy controller is presented. Closed-loop, real-time control result are presented in Section 6. The generalization of the solution is given in Section 7, while conclusions and some research directions are presented in Section 8.

2 The structure of the mobile robot Pioneer 3-DX and additional hardware

P3DX is a wheeled mobile robot (WMR), with two independent drive wheels, plus an additional castor wheel for stability, as shown in fig. 1. The kinematic variables associated with the WMR are presented in fig. 2.

An internal microcontroller monitors the position information provided by two encoders, and generates PWM commands, which are amplified and applied independently to the drive wheels, as shown in fig. 3. The two independent control loops are governed by the same PID algorithm.

Additional circuits (not shown in fig. 3) are related to the sonar system, which allows obstacle detection.
All the communication with the outside world is performed via a serial RS232 communication interface, according to a proprietary protocol.

Fig 1. Pioneer 3-DX WMR with Pioneer 5-DOF Arm

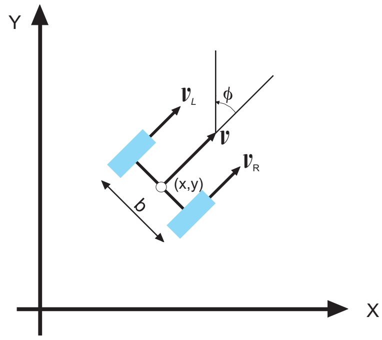

Fig. 2. Kinematic variables of the WMR mobile platform

For this reason, any implementation of a control algorithm for this robot must rely on an external device (either an on-board computer, or a remote PC with wireless capabilities) that communicates with the microcontroller core via the RS232 interface. We chose to design a small piece of hardware, installed on the WMR, consisting in a distinct microcontroller with two serial communication interfaces. (Fig 4)
The first serial interface is aimed to transmit command packets to the robot, and to receive SIPs (Status Information Packets), while the second RS232 interface is connected to a radio modem in order to provide a wireless link to a remote computer, which sends user commands, and adjusts parameter values.

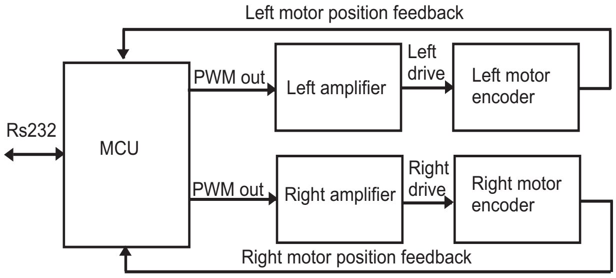

Fig. 3 The simplified structure of Pioneer3-DX mobile platform

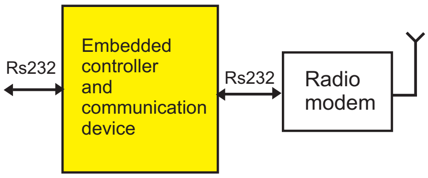

Fig. 4 Block diagram of the additional embedded controller

The control algorithm was implemented on this additional embedded controller, using ANSI C for maximum software portability.

3 Remarks on the communication protocol

Once the communication starts, the robot permanently sends SIPs, every 100 ms. Each SIP contains the current values of the coordinates (x, y, □), as well as the most recent readings of the sonar array, and status variables. The control application must parse the received packets and update the associated variables.
Pioneer 3-DX accepts a variety of command packets via the serial line, among which one is particularly significant for this application. The VEL2 command defines simultaneously individual values for the speeds of each drive wheel. By specifying different values for the speeds of the right wheel V_R and left wheel V_L, the robot is instructed to combine translation and rotation in a natural way.
Having the current coordinates, and a means to evaluate the distance from the current position to the target path, one can estimate the speeds of the right and left wheel, V_R and V_L, so that the distance to the target is minimized.

4 The kinematic model of the mobile robot Pioneer 3-DX

1) Assumption: The movement of the vehicle is on a circular trajectory on a flat plane, without obstacles,
2) Assumption: The WMR motion is assumed to be pure rolling, with no slipping.
If p is a point in a space with n generalized coordinates, and p' is the respective derivative of this vector, with the notations shown in fig. 5, the direct Jacobian model can be written:

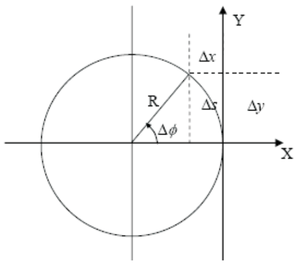

Fig. 5. WMR moving on a circular trajectory

$$p' = \begin{bmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{1}$$

where v is the linear velocity of the WMR and ω is the angular velocity.

This is equivalent to:

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 \\ \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

Considering the notations shown in fig. 2:

$$v = \frac{v_R + v_L}{2} = \frac{\omega_R + \omega_L}{2} r \tag{3}$$

$$\omega = \frac{v_R - v_L}{b} = \frac{\omega_R - \omega_L}{b} r \tag{4}$$

where *b* is the bias of the WMR (distance between the planes of the drive wheels), and *r* is the radius of the drive wheels.
Substituting (3) and (4) in (2):

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -r(\sin\phi)/2 & -r(\sin\phi)/2 \\ r(\cos\phi)/2 & r(\cos\phi)/2 \\ -r/b & r/b \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} x' \\ y' \\ \phi' \end{bmatrix} = \begin{bmatrix} -(\sin\phi)/2 & -(\sin\phi)/2 \\ (\cos\phi)/2 & (\cos\phi)/2 \\ -1/b & 1/b \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \tag{6}$$

5 The fuzzy controller

Assume that the circular target trajectory is defined by the coordinates of the center (x_C, y_C) and the radius R. The WMR must be controlled so that the distance d from the current position to the center of the circle equals R.
The position error *e* for the current position (x,y) is:

$$e = \sqrt{(x - x_C)^2 + (y - y_C)^2} - R \tag{7}$$

and the error dot *e'* can be approximated by:

$$e' = \frac{\Delta e}{\Delta t} \tag{8}$$

provided that Δt is small.

Remark: In case of an elliptic trajectory, defined by two focal points (x_{C1}, y_{C1}) ,(x_{C2}, y_{C2}), and the constant K (the sum of the distances from any point belonging to the ellipse to the focal points), the error is calculated with (9):

$$e = \sqrt{(x - x_{c1})^2 + (y - y_{c1})^2} + \sqrt{(x - x_{c2})^2 + (y - y_{c2})^2} - K \tag{9}$$

This leads to three obvious domains of variation for e(t): negative (N), zero (Z) and positive (P), which corresponds respectively to the situations when the WMR is inside the target circle, close to the circumference, and outside it.
Consider the membership functions defined in fig 6.

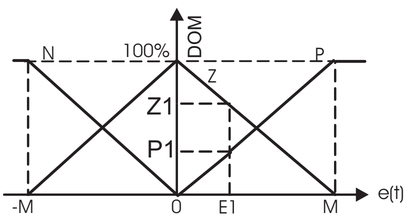

Fig. 6. Membership functions for e(t)

Note that, all the three membership functions associated with e(t) are fully determined by the parameter M, and the degree of membership (DOM) to a specific domain can be easily calculated by linear interpolation.

By defining a similar set of membership functions for the error dot (*e'(t)=□e/□*t), one gets a total of 9 rules (3x3) of the following type:

"If the error is positive and the error dot is positive, then v_R must be HIGH and v_L must be LOW."

The entire "rule base" describing the fuzzy controller is presented in Table 1.

Each cell of table 1 contains a logic sentence and should be read as: "If e(t) is Negative AND e'(t) is negative, THEN V_L must be HIGH and V_R must be LOW"

Table 1. Rule base for the Fuzzy controller

error dot e'(t)	error e(t)		
	N	**Z**	**P**
N	HL	LH	LM
Z	HL	MM	MH
P	ML	MH	LH

where H,M,L designate the singleton values for HIGH, MEDIUM and LOW fuzzy domains of the outputs v_L, v_R respectively.

The truth value of the antecedent of the above sentence is
min(N(e),N(e')), where N(e), N(e') are the degrees of membership of e(t) and e'(t) to the domain N.

The crisp output of the fuzzy controller is a combination of all the rules in the rule base as follows:

$$v_{out} = \frac{\sum_{i=1}^{K} z_i S_i}{\sum_{i=1}^{K} z_i} \tag{10}$$

where:

$$z_i = \min(E_i, E'_i) \tag{11}$$

Si is the corresponding singleton value of the fuzzy output, and K is the total number of rules in the rule base. E_i, E'_i are the degrees of membership of *e(t)* and *e'(t)* to the domain corresponding to the cell i.

The microcontroller implementation of the above algorithm relies on two background tasks (interrupt driven): one task receives and parses the SIPs sent by the P3DX WMR and updates the global variables x,y, and the other uses a timer to generate precise time intervals □t. Then, using (7) and (8) the MCU computes e(t) and e'(t) for the current position, and using (9) generates the values of the left and right wheel speed, v_L and v_R, for the next time interval. Finally, the computed values of v_L and v_R are sent to the robot via the VEL2 command packet, and the timer is restarted. This process continues indefinitely, until a STOP command is received from the human operator via the radio link.

6 Real-time fuzzy control results in path following

For testing the proposed discrete-time sliding-mode adaptive controller Pioneer 3-DX with on board PC and wireless adapter has been used in circular trajectory tracking. The basic Pioneer platform contains all of the components of an intelligent mobile robot for sensing and navigation in real world environment, including battery power, drive motors and wheels, linear encoders, and range-finding ultrasonic sonar transducers. The mobile platform is client-server architecture. The client is an embedded PC with Ethernet connection. The robot client software runs on the onboard PC and wireless Ethernet is used to monitor and control PC operations. The rugged P3-DX is 44cm x 38cm x 22cm aluminum body with 16.5cm diameter drive wheels. The two motors use 38.3:1 gear ratios and include 500-tick encoders. This differential drive platform is highly holonomic and can rotate in place moving both wheels, or it can swing around a stationery wheel in a circle of 32cm radius. A rear caster balances the robot.

A dedicated PC software simulation program was written to test and compare the performances of the embedded implementation, and to adjust the control parameters. This software uses (6) to compute the current position instead of reading it from the WMR. The results of the simulation are shown in fig. 7 and fig. 8.

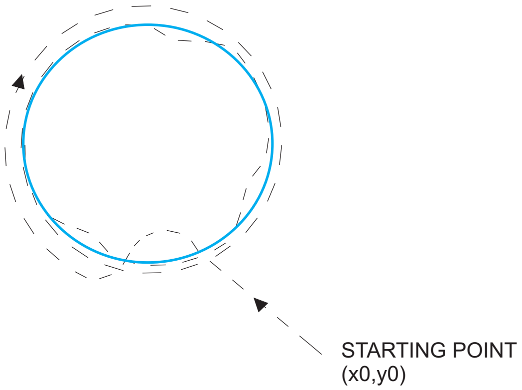

Fig. 7 WMR experimental circular trajectory

In these figures, the continuous line represents the target path, and the dashed line is the actual trajectory of the WMR. Fig. 7 shows the evolution of the WMR when the starting point was outside the target circular trajectory, while fig. 8 describes the evolution of the robot is inside the circle that defines the target.

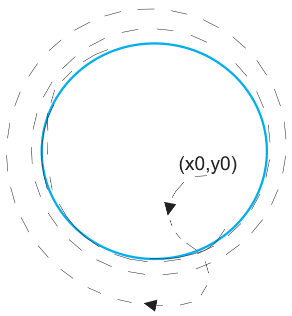

Fig. 8 Experimental circular trajectory

A variant of the algorithm, based on (9), produced the following results, for an elliptical trajectory:

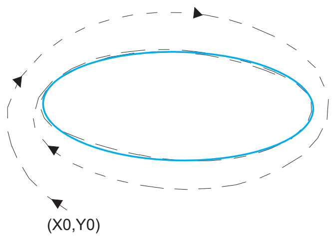

Fig. 9 Experimental elliptical trajectory

7 Generalization of the solution

Given a generalized two-dimensional curve, presented as a table of coordinate pairs (x_i,y_i), each segment of this curve, can be represented as a line, having the equation:

$$Ax+By+C=0 \tag{12}$$

where

$$\begin{cases} A = y_{(i+1)} - y_i) \\ B = (-x_{(i+1)} - x_i) \\ C = x_{(i+1)} y_i - x_i y_{(i+1)} \end{cases} \tag{13}$$

With these notations, the distance from the point (x_0,y_0) to the current segment of the curve, defined by two successive points (x_i,y_i), (x_{i+1},y_{i+1}) is determined by (14).

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}} \tag{14}$$

Since the distance, calculated with the equation (14) is always positive, the position error, in terms of the fuzzy algorithm described above, is:

$$e=sgn_i*d \tag{15}$$

where $sgn_i = +1$ when the motion on segment i of the curve is in trigonometric direction, and –1 otherwise.

The experimental results deriving from the above considerations are presented in figure 10.

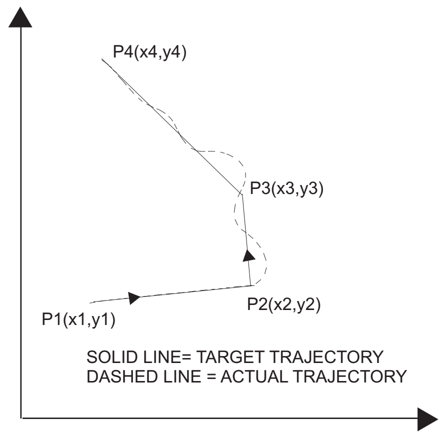

Fig. 10 WMR following an open, generalized path

8 Conclusion and further research

This research demonstrates that relatively complex real-time control algorithms can be implemented on small, microcontroller-based, devices.

Note that the fuzzy parameters used in this experiment have been arbitrarily selected, with virtually no tuning, and therefore significant improvements of the overall behavior of the WMR can be expected from applying a genetic algorithm for optimization.

Since the natural trajectory of a WMR with $v_R<>v_L$ is an arc of circle, complex spline calculations can be avoided by approximating the target path by a succession of arcs.

At any given moment, the current position point and the two consecutive points in the table that defines the generalized path determine a circle, defined by the equation (16).

$$\begin{vmatrix} (x^2+y^2) & x & y & 1 \\ ({x_1}^2+{y_1}^2) & x_1 & y_1 & 1 \\ ({x_2}^2+{y_2}^2) & x_2 & y_2 & 1 \\ ({x_3}^2+{y_3}^2) & x_3 & y_3 & 1 \end{vmatrix} = 0 \qquad (16)$$

Thus, any trajectory can be approximated by a succession of arcs, each determined by the current position and two successive points of the target trajectory.

Solving the equation (16) for each (x_i,y_i) point of the generalized trajectory to determine the coordinates of the center and the radius of the corresponding arc of circle that approximates the trajectory does not involve massive calculations, and it is feasible with an embedded device.

This approach could also be interesting for solving the problem of obstacle avoidance, since the current point, the target point behind the obstacle, and a lateral intermediary point in an obstacle free area determine a circle, where the simple algorithm described above is easily applicable.

Further research is also needed in the following directions:

Adding a mechanism for automatic adjustment of the control parameters, using a genetic algorithm;

Implementing a sliding mode control algorithm.

References:

[1] Filipescu, A.; Dugard, L.; Dion, J.-M, Adaptive gain sliding observer based sliding controller for uncertain parameters nonlinear systems. Application to flexible joint robots. Decision and Control, 2003. Proceedings. 42nd IEEE Conference, Volume 4, Issue , 9-12 Dec. 2003 Page(s): 3537 – 3542.

[2] Thongchai S. and Kawamura K. Application of Fuzzy Control to a Sonar-Based Obstacle Avoidance Mobile Robot. Proceedings of the 2000 IEEE International Conference on Control Applications Anchorage, Alaska, USA • September 25-27, 2000

[3] Susnea I. and Mitescu M. Microcontrollers in Practice, SPRINGER Verlag, 2005, ISBN 3540253017

[4] Ahmad I. Fuzzy Logic for Embedded System Applications, Newnes, 2003, ISBN 0750676051

[5] Baturone I, Moreno-Velo F.J, Sánchez-Solano S., Blanco V. Embedded fuzzy controllers on standard DSPs., J. Ferruz, Proc. IEEE Int. Symp. on Industrial Electronics, ISIE2005, Dubrovnik (Croacia), June 2005.