Mobile Robot embedded Architecture Based on CAN

M. Wargui, S. Bentalba, M. Ouladsine, A. Rachid and A. El Hajjaji
Laboratoire des systèmes Automatiques, University of Picardie - Jules Verne
7, Rue du Moulin Neuf, 80000 Amiens, FRANCE

***Abstract**. This paper deals with the analysis of real time systems subject to the time delay in the communication networks. Different approaches to handle the effect of communication delays are discussed. To improve the robustness of the control system, a fuzzy controller is used. The use of the fuzzy technique is motivated by the fact that time delays give rise to phase lag, which often degenerate system stability and performances The case of the Controller Area Network bus is particularly discussed.*

Keywords : Distributed systems, Time varying delay, CAN bus, Fuzzy control, mobile robot

INTRODUCTION

Generally, the mobile robot architecture used in industry is a centralized one. This type of architecture is easily built because all actuators and sensors are connected to a central processing unit . However, this architecture presents many drawbacks such as the difficulty of fault detection and its high cost.
Decentralized architecture is an interesting alternative which progressively replaces the old one. Its basic concept is to provide autonomy to each specialized unit in the mobile robot. In this way the sensors units compute all information locally and communicate only the useful one [1], involving a reduced number of bytes transmitted. This is also the case for the actuators. Since the information transmitted by sensors or received by actuators don't have the same sizes as in the centralized architecture, a real time network can be used to connect all the units of the mobile robot. However, this choice must take into account two fundamental conditions: first, each unit can transmit its information in a bounded time; second, all the interested units should be able to receive this information [2].

Consider the continuous time systems described by the following equations

$$\begin{cases} \dot{X}(t) = A X(t) + BU(t) + v(t) \\ Y(t) = CX(t) + e(t) \end{cases} \qquad (1)$$

where $X \in R^n$ is the process state, $U \in R^m$ is the control signal, $Y \in R^p$ is the output $v \; and \; e$ are process and measurement noises. Matrices A, B, C are of appropriate dimensions. The above system is controlled via the control loop shown in figure 1. This scheme has been adopted with respect to the Controller Area Network bus which will be discussed furthermore. It has a time driven sensor with constant sampling period *h,* event driven controller, and event driven actuator node. Sensor node measure process values and transmit these over the communication network. Actuator node receive new values for the process inputs over the communication network and apply these on the process input. Controller node read process values from sensor nodes. Using a control algorithm control signals are calculated and sent to the actuator nodes. This system reduces cost of cabling, and offers modularity and flexibility in system design. However, there is a communication delay τ_k^{sc} between the sensor and the controller, and a communication delay τ_k^{ca} between the controller and the actuator.

Through out the paper we will assume that :

A1. The sum of τ_k^{sc} and τ_k^{ca} is always less than one sampling period.

A2. The sizes of the past time delays are known to the controller. This can actually be realized using the time stamp of each signal transferred in the system.

The timing of signals in our control system is shown in figure 2. The first diagram illustrates the process output and the sampling instants, the second diagram illustrates the signal into the controller node, the third diagram illustrates the signal into the actuator node, and the fourth diagram illustrates the process input. It is seen from figure 2 that the control system is changed at irregular times making the system time varying. Furthermore, if we have a larger time delay than the sampling period h, samples may arrive in a non-chronological order at the actuator node. This make both implementation of algorithms and system analysis much harder, which justify the assumption A1.

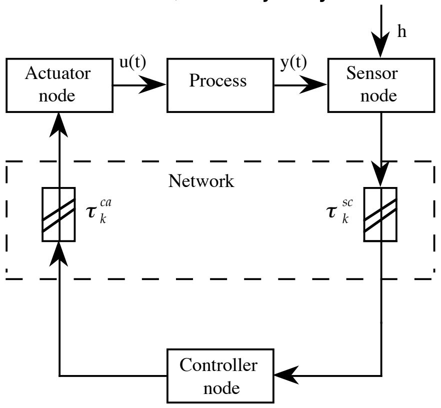

Fig. 1. Control system loop.

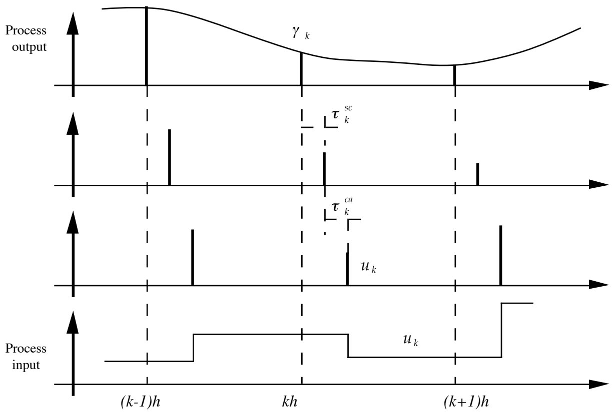

Fig. 2. Timing signals.

Discretizing the process (1) in the sampling instants *kh,* and taking into account the effect of the time delays τ_k^{sc} and τ_k^{ca}, see Figure 2, gives the discrete time model.

$$X(k+1) = \Phi X(k) + \Gamma_0\ (\tau_k^{sc},\ \tau_k^{ca}) U(k) + \Gamma_1\ (\tau_k^{sc},\ \tau_k^{ca}) U(k-1) + v(k) \tag{2}$$
$$Y(k) = CX(k) + e(k) \tag{3}$$

where

$$\Phi = e^{Ah} \tag{5}$$
$$\Gamma_0\ (\tau_k^{sc},\ \tau_k^{ca}) = \int_0^{h-\tau_k^{sc}-\tau_k^{ca}} e^{As} ds B \tag{6}$$
$$\Gamma_1\ (\tau_k^{sc},\ \tau_k^{ca}) = \int_{h-\tau_k^{sc}-\tau_k^{ca}}^{h} e^{As} ds B \tag{7}$$

and $v(k)$ and $e(k)$ are uncorrelated white noise with zero mean and covariance matrices R_1 and R_2, respectively. Denote the information available when control signal $U(k)$ is calculated by $y(k)$. This has the structure

$$y(k) = \{Y_k, Y_{k-1}, \ldots, \tau_k^{sc}, \tau_{k-1}^{sc}, \ldots, \tau_k^{ca}, \tau_{k-1}^{ca}, \ldots, u_{k-1}, u_{k-2}, \ldots\}. \tag{8}$$

Notice that the sensor to controller delay τ^{sc} at time kh and older are available, and that the controller to actuator delays τ^{ca} are assumed known up to time $(k-1)h$. The control signal is a function of all information available when it is calculated, i.e. $U(k) = f(y(k))$.

If there were no time delays the system will be described by

$$X(k+1) = \Phi X(k) + \Gamma_0\ (0{,}0) U(k) + v(k) \text{ and } Y(k) = CX(k) + e(k) \tag{9}$$

THE COMMUNICATION NETWORK

The concept used in this section relies on a distributed computer system, which is embedded in the mobile robot, and whose nodes are spatially close to actuators, controller and sensors. Each node incorporates one or more processing element(s), a communication interface and an actuator/sensor interface see figure 3.
The CAN protocol used is carrier sense multi access and collision detect (CSMA/CD). All the nodes in the network are sensing carrier and when medium is free they can begin to transmit When the node is transmitting, it monitor the data signal during its transmission [3]. Collisions are generally detected by performing a bit-by-bit comparison between transmitted and received data. If the transmitted and monitored signals are different, a collision is assumed to have occurred. The difference between CMSA/CD and CAN is that in the last protocol there is also properties of arbitration on message. It means that when a collision occur the message with higher priorities can go through and the message with lower priorities will lose. It is important to see that this propriety is not valid with conventional CMSA/CD in which both messages are destroyed. The CAN node could be event-driven, or time-driven, since it can send its message periodically or sporadically.

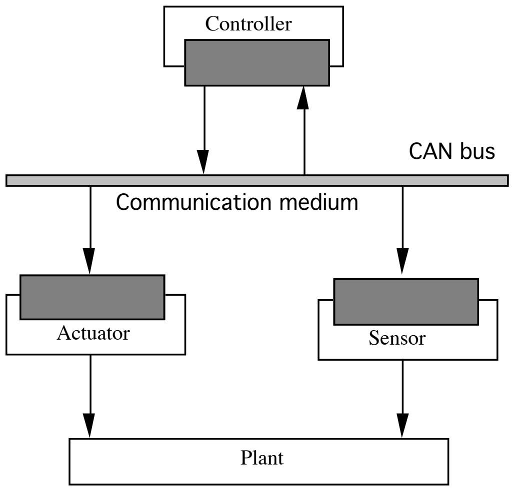

Fig. 3.

The mobile robot application will contain a mixture a tasks to be performed. Also tasks not directly related to control, coexist with time-critical ones (which we deals in this paper). Such tasks would typically be concerned with man-machine interaction, mobile robot diagnostics. On the other hand the time critical tasks as planning and navigation are concerned by the closed loop system depicted in figure 1.

THE MOBILE ROBOT MODEL

1. KINEMATICS MODEL OF THE ROBOT

The positioning of a mobile robot is generally described by 3 state variables ϕ, x, and y where ϕ is the angle of the robot with respect to the horizontal, x and y are the coordinates of the robot's center. the control of the robot is the steering angle θ. The simulated vehicle is shown in Fig. 1.
The task here is to design a control system which allows the robot to converge to a desired configuration (x_f,ϕ_f), starting from any one. From this, we have to determine the steering angle θ at each point of the robot trajectory.
The used kinematics model of the vehicle proposed in [4] is:

$$
\begin{aligned}
x(t+1) &= x(t) + \cos([\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)] \\
y(t+1) &= y(t) + \sin([\phi(t) + \theta(t)] - \sin[\theta(t)]\cos[\phi(t)] \\
\phi(t+1) &= \phi(t) - \sin^{-1}[\frac{2\sin[\theta(t)]}{b}]
\end{aligned}
\quad (10)
$$

where b is the length of the robot.

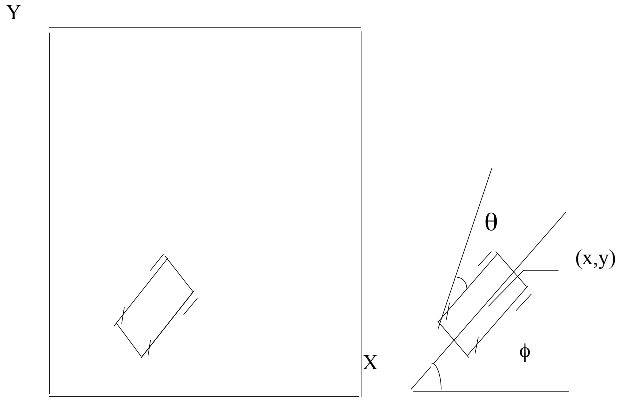

Fig. 4. Diagram of simulated mobile robot

2. FUZZY LOGIC CONTROLLER

A. Basics of fuzzy controller

The steering angle has to be controlled in such a way that the robot can roll toward the desired configuration. The idea is to reduce the error dx between the reference position xr and the current position xc, and the difference angle dϕ between the reference direction ϕr and the current heading direction ϕc, until the two errors converge to 0.
The architecture of the developed control system is shown in Fig 2-a.

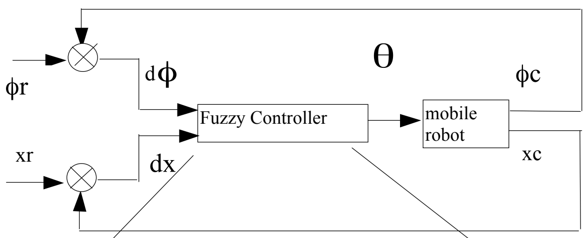

Fig 5-a

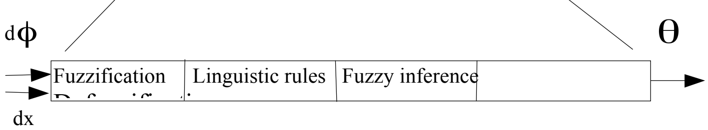

Fig 5-b

Fig. 5-b shows the internal configuration of the fuzzy controller, decomposed into four parts:
Fuzzification: this block computes membership values for dx (position error) and dϕ (orientation error) of fuzzy sets qualifying each linguistic variable, respectively.
Fuzzy rules: the rules translate relations between a set of causes and their effects in the following form:

IF condition(s) THEN action(s).

Fuzzy inference: the mechanism of inference determines the output fuzzy set, using linguistic rules and inference operator. The most famous one is Max-Min inference [5].
From the set of rules of the following form:

if dx is Ai, and dϕ is Bi then is Ci

we deduce the following membership function:

$$m_\theta(\theta) = Max(m_{\theta 1}(\theta), \cdots m_{\theta i}(\theta), \cdots) \quad (11)$$

where $m_{\theta i}(\theta) = \mathrm{Min}(\mathrm{Min}(m_{Ai}(dx), m_{Bi}(d\phi)), m_{Ci}(\theta))$

$m_{Ai}(dx)$ membership values for dx of fuzzy set Ai.

$m_{Ci}(\theta)$: membership function of fuzzy set Ci

Defuzzification: the defuzzification consists in selecting the most representative value of the output fuzzy set.
Among defuzzification methods, we choose the center average defuzzification defined as :

$$\theta_0 = \frac{\int_\theta m_\theta(\theta)\,\theta\, d\theta}{\int_\theta m_\theta(\theta)\, d\theta} \quad (12)$$

B.Fuzzy controller for the robot

In this section, we present a fuzzy controller based on rules generated from intuition and common sense[6][7].
The steering angle θ has to be controlled in such a way that the robot can roll toward the desired configuration. The main idea is to reduce the position error dx and the orientation error dϕ, until both errors converge to 0. Therefore, the fuzzy rules have to be generated in such a way that the steering angle θ can be big when the robot is at a point far away from the desired posture and small when the robot reaches the destination.
For example, let's consider the case when the robot is located in situation with an orientation such that dϕ is positive middle and dx is zero (Fig. 5-a.). In this situation, the steering angle θ has to be negative middle to decrease dx. Therefore, we obtain the following rule :

if dϕ is positive middle and dx is zero then θ is negative middle.

Now, let's consider the case when dϕ is zero and dx is positive middle (Fig. 5-b.). In this situation, a negative middle of the steering angle θ is required to reduce dx. The corresponding rule is then :

if dϕ is zero and dx is positive middle then θ is negative middle

Based on the above qualitative analysis, we are able to design the control output allowing the convergence of the robot to the desired configuration.

The resulting control rules are shown in table 1.

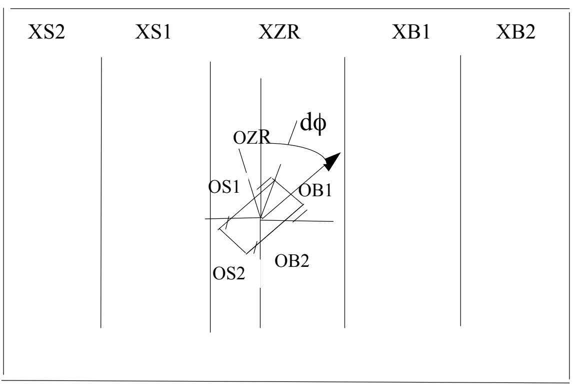

fig. 6-a

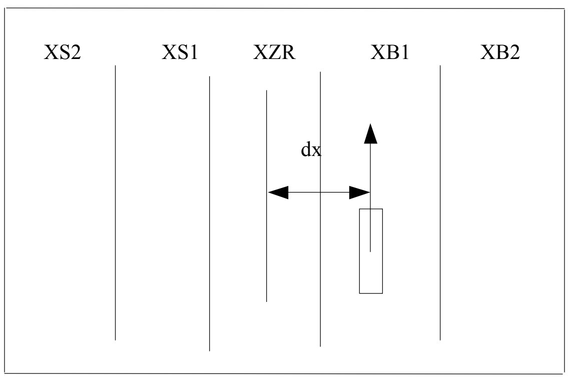

fig. 6-b

Each linguistic variable takes five fuzzy set (ç, S2, S1, ZR, B1, B2). Membership functions m(dx), m(dϕ) and m(θ) are defined as shown in Fig. 4.
For finer adjustments near the destination, the fuzzy region relevant to the desired configuration are designed to be narrower (i.e. the regions XZR, OZR, ZR). The wider fuzzy sets allows rough control far from the destination. The universe of discourse of the position error dx is normalized to [-1,1], while the orientation error dϕ $\in$ [-180,180], and the steering angle $\theta \in$[-40,40].

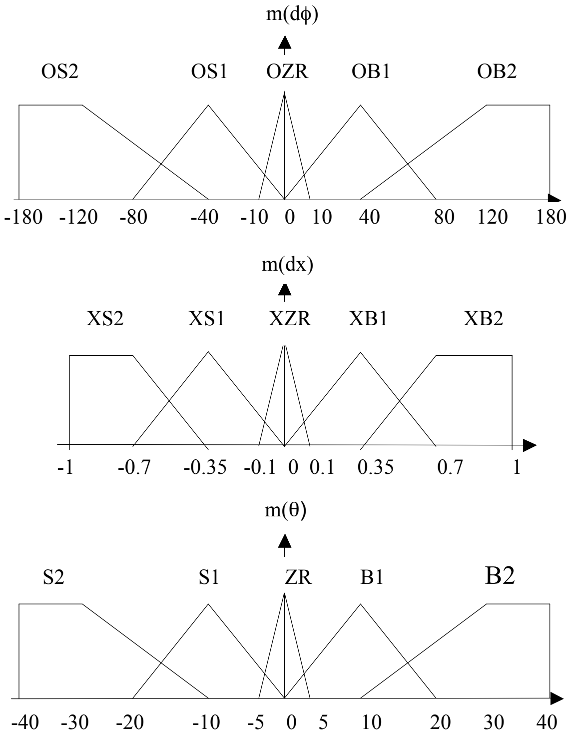

Fig. 7. Fuzzy membership for the robot control problem

dϕ \ dx	XS2	XS1	XZR	XB1	XB2
OS2	*B2*	*B2*	*B2*	*B1*	*B1*
OS1	*B1*	*B1*	*B1*	*B1*	*ZR*
OZR	*B2*	*B1*	*ZR*	*S1*	*S2*
OB1	*ZR*	*S1*	*S1*	*S1*	*S1*
OB2	*S1*	*S1*	*S2*	*S2*	*S2*

Table 1.

C. Related timing problems.

The sensor reads the coordinate $y(k)$ of the mobile robot and send it through the communication network to the controller. The controller utilizes the available data to compute the new control input as soon as it arrives. Figure 4, shows the timing possibilities.
The delays τ_k^{sc} and τ_k^{ca} can considerably change from sample to sample. They can not be assumed deterministic. Hence, the discrete time mobile robot model becomes time varying delayed model.

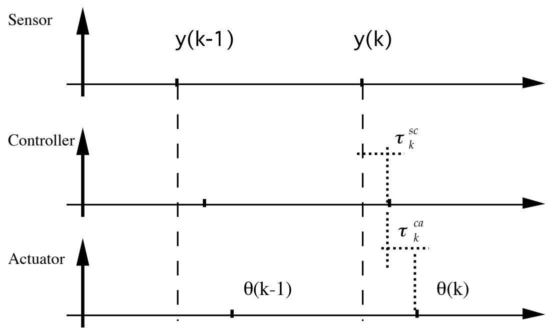

Fig. 8. Timing problems in the controlled system.

D. Simulation results

In order to illustrate the effectiveness of this fuzzy controller, simulations have been performed. We used the Max-Min inference and the center average defuzzification. The used membership functions are shown in Fig. 4. Table 1 gives the linguistic rules.

The objective is to reach a given position and orientation, starting from any configuration. Four arbitrarily chosen initial states, (x0,ϕ0)=(3,0°), (20,30°), (24,220°), (25,-30°) and three desired configurations (xf,ϕf) = (10,90°), (20,90°), (30,90°), were used to test the fuzzy controller.

Fig. 5 shows the robot trajectories using the 25 fuzzy rules. We see that the fuzzy controller successfully control the vehicle to the desired configurations starting from all four initial states.

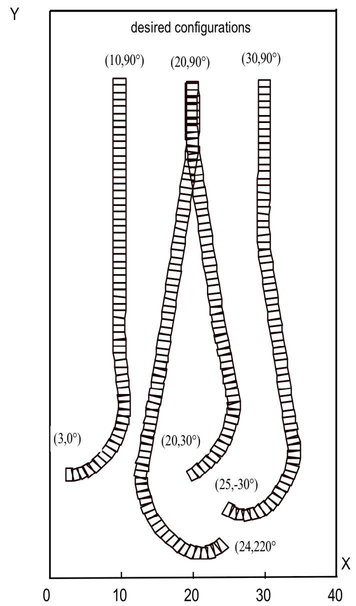

Fig. 10. Robot trajectories using 25 fuzzy rules

REFERENCES

[1] H. Chan, Ü, Özgüner 'Closed-loop control of systems over a communication network with delays,' *Int. J. Control, Vol. 62, No 3,* pp. 493-510.

[2] L. Liou and A. Ray, 'A stochastic regulator for Integrated Communication and Control Systems Part - I - formulation of control law,' *ASME J. Dyn. Sys., Meas. & Contr.,* vol. 113, pp. 604-611, Dec. 1990.

[3] K. Tindell, A. Burns and A. Wellings, 'Calculating Controller Area Network message response times', *Proc. the 10th IFAC Workshop on Distributed Computer Control Systems,* Toledo, Spain, September 1994.

[4] L.Wang and J. Mendel, 'Generating fuzzy rules by learning from examples' ,' *IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, No. 6, Nov./Dec* ' 92

[5] E.H. Mandani, 'Application of fuzzy algorithms for dontrol of simple dynamic plant'*, proc IEEE 121 (12), 1585-1588'*, 74

[6] U.S.K.Alahakoom, J.V. Wijayakulasooriya, B.A Kithsiri, B.A.A.P. Balasooriya and P.R.P. Hoole, 'Dual-motor controller for laboratory model auto guided vehicle' , ' *IEEE/KTH Stocholm Power Tech. Conference, Stocholm, Sweden, June 18-22* ' 95

[7] S.Ishikawa, 'A method of indoor mobil robot navigation by using fuzzy control', ' *IEEE International Workshop on Intelligent Robots and Systems*, IROS '90.