

Non-linear Control of an Autonomous Ground Vehicle

Wilmar Hernandez
Universidad Politécnica de Madrid
Dep. de Ingeniería de Circuitos y Sistemas
EUI de Telecomunicación
Ctra. De Valencia Km. 7
Madrid 28031. SPAIN
E-mail: whernan@ics.upm.es

Norberto Cañas
Universidad Politécnica de Madrid
Dep. Informática Aplicada
EU de Informática
Ctra. de Valencia Km 7
Madrid 28031. SPAIN
E-mail: norberto@eui.upm.es

Abstract- In this paper, in order to select a speed controller for a specific non-linear autonomous ground vehicle, proportional-integral-derivative (PID), Fuzzy, and linear quadratic regulator (LQR) controllers were designed. Here, in order to carry out the tuning of the above controllers, a multicomputer genetic algorithm (MGA) was designed. Then, the results of the MGA were used to parameterize the PID, Fuzzy and LQR controllers and to test them under laboratory conditions. Finally, a comparative analysis of the performance of the three controllers was conducted.

I. INTRODUCTION

In this paper, three multivariable speed controllers were designed to control the autonomous ground vehicle (AGV) shown in Fig. 1. Here the AGV was modeled as a non-linear time varying multivariable system. First an MGA was used to find the parameters for a PID multivariable controller. In this case the controller was a fixed one consisting on three diagonal matrixes with the parameters of the proportional, integral, and derivatives parts. Several tests were carried out under disturbances such wavy surfaces. Then as the PID was a diagonal linear controller used to control a non-linear plant it was decided to test the AGV but using a non linear controller that could take into account the non-linear dynamics of the plant.

To this end a fuzzy controller was designed. In this controller the membership functions were also parameterized using the genetic algorithm (GA) and the disturbances affecting the performance of this new feedback-controlled system were the same as the ones used for the design of the PID controller. The results of the comparison of both controllers were satisfactory and very similar to each other. However, the steady state performance of the fuzzy controller was better than the one of the PID controller. Finally, it was decided to design and advance controller based on a LQR algorithm. Hence, this algorithm was modified to build a non-linear multivariable adaptive controller, based on a finite horizon LQR algorithm.

The results of this last controller was the best among the three controllers tested. Moreover, as its performance does not depend on the past history of the state vector, it could be a good candidate for the design of multivariable, non-linear gain scheduling controllers.

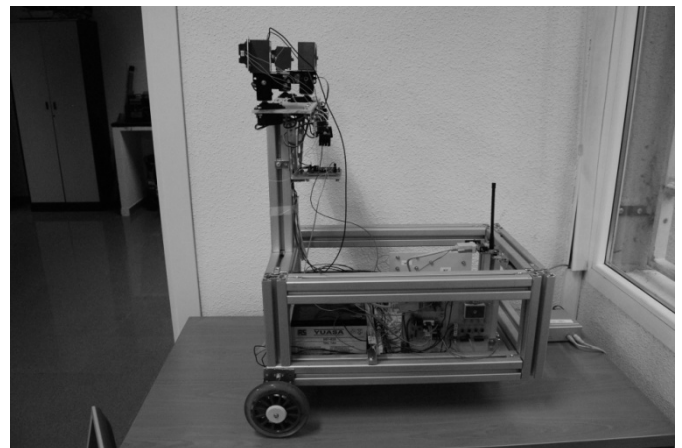


Fig. 1. Three wheeled AGV prototype for research and teaching purposes. In the current configuration it has three mobile ultrasonic sensors, two mobile infrared sensors, three dsPIC-30F4012 microcontrollers, two Maxon RE-36 DC main motors (with optical encoders), an accelerometer of two axes (used as tilt sensor) and a RF transmitter receiver.

The outline of this paper is the following: In Section II the model of the AGV is presented along with some highlights of its animation process; in Section III three multivariable controllers were built; Section IV is devoted to carry out the comparison among the previous controllers; and the conclusion of this research are given in section V.

II. AGV MODEL

At this point, it is important to mention that in the scientific literature there are several well-known references on control of autonomous mobile robots [1-3] that propose AGV models for teaching purposes. However, those models must be adapted to be used in real-life systems.

On the other hand, for scientific research, there are also many examples of very detailed, advanced AGV models [4-9] that have been built for control purposes.

In this paper, the basic model of the parallelepiped was used as a starting point for modeling the AGV shown in Fig.1. Nevertheless, such a model was adapted to the problem at hand by applying some constraints that made it more realistic.

A. General model of the AGV

To manage the position and orientation of an AGV is often used an Euclidean affine space in which two reference systems are defined (see Fig. 2). The first one is a global reference frame that is fixed with respect to a known position of the space in which the AGV can operate and, for simplicity this reference is considered to be an inertial reference frame.

The second reference system is a frame located on the AGV. This is obviously non-inertial.

In this communication the global reference frame is called $M_0(O_0, x_0, y_0, z_0)$ and the local reference frame is called $M_1(O_1, x_1, y_1, z_1)$.

The nomenclature used for positions, velocities, moments and forces (1) is based on the one provided by SNAME (Society of Naval Architects and Marine Engineers) in 1950.

η_1 is the vector that indicates in M_0 the coordinates of the origin O_1 of M_1 .

η_2 is the vector of angles (the Tait Bryan angles) that indicates the AGV orientation in M_0 .

v_1 is the vector of longitudinal speeds of the AGV in M_1 .

v_2 is the vector of angular speeds of the AGV in M_1 .

τ_1 is the vector of forces that acts on the AGV in M_1 .

τ_2 is the vector of angular moments that acts on the AGV in M_1 .

$$\begin{aligned} \eta_1 &= [x, y, z]^T & \eta_2 &= [\phi, \theta, \psi]^T & v_1 &= [u, v, w]^T \\ v_2 &= [p, q, r]^T & \tau_1 &= [X, Y, Z]^T & \tau_2 &= [K, M, N]^T \\ \eta &= [\eta_1^T, \eta_2^T]^T & v &= [v_1^T, v_2^T]^T & \tau &= [\tau_1^T, \tau_2^T]^T \end{aligned} \quad (1)$$

First, in order to develop the dynamic model of the AGV (Fig. 1) a solid parallelepiped with six degrees of freedom (imagine the vehicle of Fig.2 without wheels) suspended in space was considered. The six degrees of freedom (6dof) of the above parallelepiped corresponds to its longitudinal movement along the x , y and z axes, and its roll, pitch and yaw speeds.

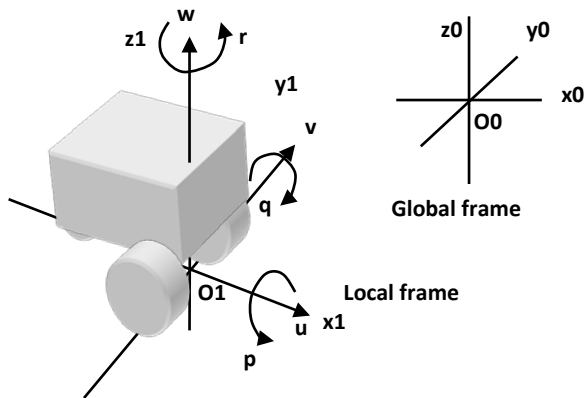


Fig. 2. Representation of the AGV with global and local reference frames. Longitudinal and angular speeds are included.

The above statements are summarized in the following equations [10], where m is the mass of the solid, r_{CG} is the

position of its gravity center in M_1 , and I is its inertia tensor, also in M_1 .

$$\begin{aligned} \tau_1 &= m[v_1' + v_2 \wedge v_1 + v_2' \wedge r_{CG} + v_2 \wedge v_2 \wedge r_{CG}] \\ \tau_2 &= m[r_{CG} \wedge (v_1' + v_2 \wedge v_1)] + I v_2' + v_2 \wedge (I v_2) \end{aligned} \quad (2)$$

If equations (2) are expanded with the notation given in (1), then, six expressions that could be grouped in different ways to reach a single matrix equation will be obtained.

Again, some frequently referenced examples about this could be found in [10]. Nevertheless, a three-wheeled AGV is not a parallelepiped and it does not have 6dof. Even so, the dynamic behavior of this kind of AGV could be obtained from the previous equations by adding the following constraints.

The proposed AGV (Fig.1) has two motorized non-steering front wheels and a free rear wheel. Then, it can rotate thanks to the differential traction but it cannot move transversally. More formally, the AGV has a non-holonomic constraint (a differential constraint on a position variable [3]) that is $dy/dt = v(t) = 0$.

It is also considered that the proposed AGV does not jump and does not go under the floor level either. Consequently, there is another constraint of the same kind $dz/dt = w(t) = 0$. In addition, there are some simplifications motivated by the following facts: $y_G = 0$, $I_{xy} = 0$ and $I_{yz} = 0$.

Finally, the angular speeds p and q are mainly imposed by the change of floor tilt, which might suggest not using them as input reference to the system. Putting it all together, the state equation (3) of the AGV dynamic behavior can be obtained. In (3), τ_M is the force vector generated by DC motors, τ_G is the force vector due to gravity, and τ_{fric} is the force vector produced by frictions.

$$\begin{aligned} v' &= -H^{-1}C(v)v - H^{-1}\Omega\rho' - H^{-1}\Xi\rho + \\ &+ H^{-1}\tau_M + H^{-1}\tau_G - H^{-1}\tau_{fric} \\ H &= \begin{bmatrix} m & 0 \\ 0 & I_z \end{bmatrix} & C &= \begin{bmatrix} 0 & -mx_G r \\ mx_G r & 0 \end{bmatrix} & v &= \begin{bmatrix} u \\ r \end{bmatrix} \\ \Omega &= \begin{bmatrix} 0 & mz_G \\ I_{xz} & 0 \end{bmatrix} & \Xi &= \begin{bmatrix} mz_G r & -mx_G q \\ I_y q & -I_{xz} r - I_x p \end{bmatrix} & \rho &= \begin{bmatrix} p \\ q \end{bmatrix} \end{aligned} \quad (3)$$

B. AGV orientation and gravity force vector (τ_G)

In short, to specify the orientation of an AGV is the same as establishing the degree of alignment between the local and global reference frameworks. To do that, a Tait Bryan alternative was used in this paper, which consists of three rotations around the axes Z_1 , Y_1 and finally X_1 , as indicated by the angles ψ , θ and ϕ , respectively, which are η_2 components. These three rotations can be carried out by using the matrix J_1 (4).

$$J_1(\eta_2) = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (4)$$

Post-multiplying matrix (4) by the coordinates of any vector in the AGV local frame gives the coordinates of that vector in the global reference frame. Fortunately, J_I is an orthonormal matrix ($J_I^{-1} = J_I^T$), then the opposite operation could be easily performed as well. That is, if there exists a vector whose coordinates are known in the global reference frame, post-multiplying the matrix J_I^T by this vector gives the coordinates of the same vector in the AGV local reference frame. This is useful to know τ_G components.

First of all, it is needed to know the AGV weight vector in the local reference frame. Considering that m is the mass of the vehicle and g is the gravity acceleration, then the local frame weight vector can be given by (5).

$$\begin{bmatrix} P_{x_1} \\ P_{y_1} \\ P_{z_1} \end{bmatrix} = J_I^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = J_I^T(\eta_2) \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (5)$$

Considering again the AGV non-holonomic restrictions and the influence of the floor tilt, the AGV weight vector contributes only to the X force component of τ_1 and the N moment component of τ_2 . These contributions can be calculated as in (6).

$$\tau_G = \begin{bmatrix} X_G \\ N_G \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & x_G \end{bmatrix} \begin{bmatrix} P_{x_1} \\ P_{y_1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & x_G \end{bmatrix} \begin{bmatrix} -\sin(\theta) \\ \sin(\phi)\cos(\theta) \end{bmatrix} [-mg] \quad (6)$$

C. Friction force vector (τ_{fric})

In this research, it was considered that the main friction force was the one yield in each wheel axle. Those forces are non-linear, and to model them it is usual to consider wheel speeds (V_R and V_L) in order to isolate the scenery in which a wheel is either stopped or not. This is necessary because friction forces have different behavior in each of these situations.

When a wheel is stopped a static resistance force (F_{RES_STA}) must be used and on the other hand when it is not stopped a dynamic resistance force (F_{RES_DYN}) must be used. These forces are calculated by using the static friction coefficient μ_S and the dynamic friction coefficient μ_D . These coefficients multiply the modulus of the normal force to the ground opposed the weight of the AGV as given by (7).

$$|F_{RES_EST}| = \mu_s |P_z| \quad |F_{RES_DIN}| = \mu_d |P_z| \quad (7)$$

Finally, to obtain the friction force in each wheel (F_{FRIC_R} and F_{FRIC_L}) the protocol defined in (8) was follow. In (8) the total forces of each wheel (F_{R_TOTAL} , F_{L_TOTAL}) due to DC motors and gravity were taken into account.

$$\begin{aligned} (V_R = 0) \wedge (|F_{R_total}| \geq F_{RES_EST}) &\Rightarrow F_{FRIC_R} = -\text{signo}(F_{R_total}) F_{RES_EST} \\ (V_R = 0) \wedge (|F_{R_total}| < F_{RES_EST}) &\Rightarrow F_{FRIC_R} = -\text{signo}(F_{R_total}) F_{R_total} \\ (|V_R| > 0) &\Rightarrow F_{FRIC_R} = -\text{signo}(V_R) F_{RES_DIN} \\ (V_L = 0) \wedge (|F_{L_total}| \geq F_{RES_EST}) &\Rightarrow F_{FRIC_L} = -\text{signo}(F_{L_total}) F_{RES_EST} \\ (V_L = 0) \wedge (|F_{L_total}| < F_{RES_EST}) &\Rightarrow F_{FRIC_L} = -\text{signo}(F_{L_total}) F_{L_total} \\ (|V_L| > 0) &\Rightarrow F_{FRIC_L} = -\text{signo}(V_L) F_{RES_DIN} \end{aligned} \quad (8)$$

To place F_{FRIC_R} and F_{FRIC_L} into τ_{fric} of (3) it is necessary to join each wheel contribution to the X force and N moment as in (9), where b is the wide of the AGV.

$$\tau_{fric} = \begin{bmatrix} F_{FRIC_R} + F_{FRIC_L} \\ \frac{b}{2} F_{FRIC_R} - \frac{b}{2} F_{FRIC_L} \end{bmatrix} \quad (9)$$

D. Motor vector τ_M

The vector force τ_M of (3) is obtained from the force contributions generated by the DC motors at each wheel position. These forces contributions are F_R and F_L , and are given by (10). Where τ_R and τ_L are the right and left generated-engine-moment respectively, and R_{rad} the wheel radius.

A straightforward but good enough model of a DC motor can be found in [11]. In such a model, the moment generated by a DC motor is proportional (by a constant frequently called A) to the current through its right coil (i_R) and left coil (i_L). Moreover, if the motor has a mechanical reduction stage and δ is the inverse reduction ratio, the formulas that allow us to compute the forces in each wheel position are given by (10).

$$F_R = \frac{\tau_R}{R_{rad}} = \frac{\Lambda \delta i_R}{R_{rad}} \quad F_L = \frac{\tau_L}{R_{rad}} = \frac{\Lambda \delta i_L}{R_{rad}} \quad (10)$$

The electric equation of a DC motor is given by (11) [11], where L is the motor winding inductance, R is the resistance, e_b is the EMF voltage and e_a is the input motor voltage.

$$L \frac{di}{dt} + Ri + e_b = e_a \quad (11)$$

Solving (11) for $i(t)$ and discretizing the result carries us to (12), where e_b was substituted by the expression that originates it based on the wheel speed (w_{wheel})

$$i(t_{k+1}) = e^{-\frac{R}{L}T} i(t_k) + \frac{1}{R} \left(1 - e^{-\frac{R}{L}T} \right) (e_a(t_{k+1}) - \Lambda \delta w_{wheel}(t_{k+1})) \quad (12)$$

Finally (10) is transformed into (13) and τ_M can be obtained as in (14) by an equation similar to (9).

$$\begin{aligned} F_R(t_k) &= \frac{\Lambda \delta}{R_{rad}} \left(e^{-\frac{R}{L}T} i_R(t_{k-1}) + \frac{1}{R} \left(1 - e^{-\frac{R}{L}T} \right) (e_{aR}(t_k) - \Lambda \delta w_{wheel_R}(t_k)) \right) \\ F_L(t_k) &= \frac{\Lambda \delta}{R_{rad}} \left(e^{-\frac{R}{L}T} i_L(t_{k-1}) + \frac{1}{R} \left(1 - e^{-\frac{R}{L}T} \right) (e_{aL}(t_k) - \Lambda \delta w_{wheel_L}(t_k)) \right) \end{aligned} \quad (13)$$

$$\tau_M = \begin{bmatrix} F_R + F_L \\ \frac{b}{2} F_R - \frac{b}{2} F_L \end{bmatrix} \quad (14)$$

E. Discrete AGV model

In order to build the software that executes the GA that is used to parameterized the controller, it was necessary to discretize the continuous time model of the AGV given by (3). In this paper a procedure based on the exact solution was used. This procedure was inspired by the ones in chapter 9 in [12] and [13, 14].

$$v(t_{k+1}) = \Phi(t_k)v(t_k) + \Gamma_{\rho'}(t_k)\rho'(t_k) + \Gamma_{\rho}(t_k)\rho(t_k) + \Gamma_M(t_k)\tau_M(t_k) + \Gamma_G(t_k)\tau_G(t_k) + \Gamma_{fric}(t_k)(-\tau_{fric}(t_k)) \quad (15)$$

$$\begin{aligned} \Phi &= e^{-H^{-1}C(v(t_k))T} \\ \Gamma_{\rho'} &= \left[\int_{t_k}^{t_{k+1}} e^{-H^{-1}C(v(t_k))(t_{k+1}-\lambda)} d\lambda \right] (-H^{-1}\Omega) \\ \Gamma_{\rho} &= \left[\int_{t_k}^{t_{k+1}} e^{-H^{-1}C(v(t_k))(t_{k+1}-\lambda)} d\lambda \right] (-H^{-1}\Xi) \\ \Gamma_M &= \left[\int_{t_k}^{t_{k+1}} e^{-H^{-1}C(v(t_k))(t_{k+1}-\lambda)} d\lambda \right] (H^{-1}) \\ \Gamma_M &= \Gamma_G = \Gamma_{fric} \end{aligned} \quad (16)$$

To compute matrices Φ and Γ and obtain approximate results of them, the following procedure was used [14], where A is the state matrix and B the input matrix.

$$\begin{aligned} \Phi(t_k) &= I + A(t_k)T\Upsilon \\ \Upsilon &= I + \frac{A(t_k)T}{2} \left(I + \frac{A(t_k)T}{3} \left(I + \frac{A(t_k)T}{N-1} \left(I + \frac{A(t_k)T}{N} \right) \right) \right) \\ \Gamma(t_k) &= \Upsilon TB(t_k) \end{aligned} \quad (17)$$

III. AGV SPEED CONTROLLERS

There are several issues that make it difficult to design speed controllers for AGV systems. Some of these issues are the following: The state equation (3) is non-linear; C and Ξ in (3) are time varying matrices; and rounding errors introduced by optical speed sensors, as their output is a natural number of pulses per motor axle revolution instead of a real number, then those parts of the rotations that do not generate a pulse are not taking into account. Therefore, if the current speed of the wheel is w_{wheel} , and P is the number of pulses per motor axle revolution, then the speed measured by the optical sensor will be (18).

$$w_{trough_sensor} = \left\lfloor \frac{\delta w_{wheel} P}{2\pi} \right\rfloor \quad (18)$$

A. PID controller

In order to adjust a PID controller, the discrete equation (19) of the one-step-prediction speed error was considered.

$$e(t_{k+1}) = -K_p e(t_k) - K_d \frac{e(t_k) - e(t_{k-1})}{T} - K_I T \sum_{n=1}^k e(t_n) \quad (19)$$

The current speed error vector can be calculated as in (20), where v_D is the desired speed vector and v_R is the current speed vector.

$$e(t_k) = v_D(t_k) - v_R(t_k) \quad (20)$$

Now, (21) is the result of solving (3) for τ_M .

$$\begin{aligned} \tau_M(t_k) &= [\Gamma_M(t_k)]^{-1} [v_R(t_{k+1}) - \Phi(t_k)v_R(t_k) - \Gamma_{\rho'}(t_k)\rho'(t_k) \\ &\quad - \Gamma_{\rho}(t_k)\rho(t_k) - \Gamma_G(t_k)\tau_G(t_k) - \Gamma_{fric}(t_k)(-\tau_{fric}(t_k))] \end{aligned} \quad (21)$$

Then, working with (19), (20) and (21), the controller equation (22) is obtained.

$$\begin{aligned} \tau_M(t_k) &= [\Gamma_M(t_k)]^{-1} \\ &\quad \left[v_D(t_{k+1}) + K_p e(t_k) + K_d \frac{e(t_k) - e(t_{k-1})}{T} + K_I T \sum_{n=1}^k e(t_n) \right. \\ &\quad \left. - \Phi(t_k)v_R(t_k) - \Gamma_{\rho'}(t_k)\rho'(t_k) - \Gamma_{\rho}(t_k)\rho(t_k) \right. \\ &\quad \left. - \Gamma_G(t_k)\tau_G(t_k) - \Gamma_{fric}(t_k)(-\tau_{fric}(t_k)) \right] \end{aligned} \quad (22)$$

Now, the input voltages to the DC motors are obtained by substituting the result of (22) into (14), and then substituting the result of this into (13).

B. Fuzzy controller

In this case, a decentralized kinematic approach of the problem was considered, instead of the centralized and dynamic version of the preceding PID controller.

Once the desired longitudinal and angular speeds of the AGV are known, the system computes the AGV speed at each front wheel. Then, these speeds are used for two independent fuzzy controllers that follow a Mamdani model and a PI architecture.

Therefore, each controller has two inputs (i.e., the current and the accumulated errors) and one output (i.e., the input voltage to the associated DC motor).

The fuzzification system for the input and output values uses five labels (negative big - NB, negative small - NS, zero - Z, positive small - PS, positive big - PB) and each label has a membership function with a triangle shape (see Fig. 3), located in a specific place of the abscissa.

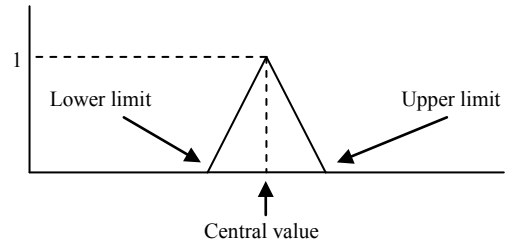


Fig. 3. Membership function with a triangular shape. There are specified the membership input values which qualified the triangular figure.

In this paper, 15 membership functions were designed: 5 functions used the current error as their input parameter (NB_e, NS_e, Z_e, PS_e, PB_e); 5 functions used the accumulated error as input parameter (NB_{Σe}, NS_{Σe}, Z_{Σe}, PS_{Σe}, PB_{Σe}); and 5 functions used the controller output as input parameter (NB_o, NS_o, Z_o, PS_o, PB_o).

The input and output values were normalized between -1 and 1. These limits were the NB central value and the PB central value of the membership functions, respectively.

In short, all the membership functions had the shape of Fig. 4, and the inference rules are summarized in Table 1.

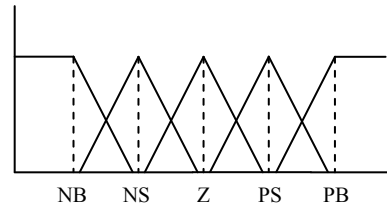


Fig. 4. Pattern of all labels of an input or output membership function.

TABLE I
PI FUZZY CONTROLLER INFERENCE RULES

		e				
		NB	NS	Z	PS	PB
Σe	NB	NB	NB	NS	NS	Z
	NS	NB	NS	NS	Z	PS
	Z	NS	NS	Z	PS	PS
	PS	NS	Z	PS	PS	PB
	PB	Z	PS	PS	PB	PB

Column labels refer to current error. Row labels refer to accumulative error. Values of the table are labels of the controller output.

The antecedents of the rules were obtained by applying a fuzzy AND operator to the inputs ($e \wedge \Sigma e$).

An example of using Table 1 is given by (23). Also, in (23), the math operators “ \wedge ” and “ \vee ” are fuzzy operators.

$$z_o = [PB_o(e) \wedge NB_{\Sigma e}(\Sigma e)] \vee [PS_o(e) \wedge NS_{\Sigma e}(\Sigma e)] \vee [Z_o(e) \wedge Z_{\Sigma e}(\Sigma e)] \vee [NS_o(e) \wedge PS_{\Sigma e}(\Sigma e)] \vee [NB_o(e) \wedge PB_{\Sigma e}(\Sigma e)] \quad (23)$$

In this paper, the result of the operation “ \wedge ” is the minimum of its operands, and the result of the operation “ \vee ” is the maximum of its operands.

Mamdani fuzzy controllers calculate their outputs following the next steps:

1. To compute the truncated triangle area of each output membership function (Fig. 5).
2. To calculate the controller output by using (24).

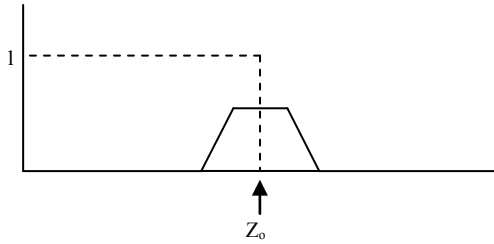


Fig. 5. Possible truncated output membership function Z_o , computed as in (22).

$$C_o = \sum_{MF_o} \frac{Area(MF_o) CV(MF_o)}{Area(MF_o)} \quad (24)$$

where C_o is the controller output; each MF_o -element takes the value of the output membership functions NB_o , NS_o , Z_o , PS_o and PB_o ; CV is the central value in Fig. 3; and $Area(MF_o)$ stands for the area of the truncated triangle shown in Fig. 5 and its height is calculated by using (23).

C. Modified adaptive LQR algorithm

In Control Theory, the LQR assumes that the system to be controlled must be linear and time invariant. However, the proposed AGV does not complain these requirements. Therefore, a novel adaptive LQR algorithm is proposed here.

Equation (25) shows the conventional LQR cost function. In this equation, $X(k)$ is the k sequence element of the system state vector in, $U(k)$ is the k sequence element of the controller output, S is the cost matrix of the final state, Q is

the cost matrix of the current state, and R is the cost matrix of the current controller output.

$$J = \frac{1}{2} X^T(N_e) S X(N_e) + \frac{1}{2} \sum_{k=0}^{N_e-1} [X^T(k) Q X(k) + U^T(k) R U(k)] \quad (25)$$

where N_e is the number of steps needed to reach the goal.

Equation (26) represents the state-space equation of the dynamic system and (27) is obtained by applying the Bellman dynamic programming procedure (chapter 14 in [12] and appendix D in [15]) to (25) and (26).

$$X(k+1) = AX(k) + BU(k) \quad (26)$$

$$U(k-1) = -[R + B^T S_k B]^{-1} B^T S_k A X(k-1) = -K_G(k-1) X(k-1) \quad (27)$$

$$S_{k-1} = Q + A^T S_k A - A^T S_k B [R + B^T S_k B]^{-1} B^T S_k A$$

where S_{N_e} is the identity, $X(N_e)$ is the goal state, and $X(0)$ is the initial state.

Equations (27) give the procedure to build the controller output sequence $U(k)$, where k ranges from 0 to N_e . If we start from the final state $X(N_e)$, then the algorithm allows us to obtain each $U(k)$ as a function of the unknown state $X(k)$. Hence, when the initial state $X(0)$ is reached, which is only known state, then we obtain each $U(k)$ value repeating the algorithm again but this time in a forward manner.

This procedure solves the regulator problem. Nevertheless, the AGV problem is a servo one. Hence, in this case the change of variable (28) is proposed and it leads to the controller output (29).

$$\tilde{X}(k) = X(k) - desired_X(k) \quad (28)$$

$$U(k) = -K(k) \tilde{X}(k) = -K_G(k) [X(k) - desired_X(k)] \quad (29)$$

At this point, it is considered that the model of the AGV is time-invariant for the period of time corresponding to one time iteration (i.e., one time-step). Then, the controller output is applied to the plant and a new final horizon is established along with new initial state conditions. These new initial state conditions are the current output of the plant for the applied controller, and the new final state conditions are updated with the final horizon that we want to reach.

Afterwards, the LQR is calculated with the new initial and final conditions but with the same S , Q and R matrices, and the whole process is repeated again. Therefore, a new LQR gain matrix is obtained.

IV. RESULTS

In the previous section the controllers for the AGV were designed, however it is well known that the feedback-controlled AGV system does not perform well if it uses the controllers without a supervision navigation algorithm.

In this research, in order to make the errors of the controllers to come out to the light, a blind navigation algorithm was used. This algorithm sent command references to the controllers depending on the trajectories that we wanted the AGV to follow. However, during the performing of each trajectory no corrections were introduced in the

command references, so that the deviations from the expected trajectories gave an idea of the quality of the controllers.

Several tests were carried out with the following parameters for the AGV shown in Fig. 1: $x_G = -0.11\text{m}$, $z_G = 0.12\text{m}$, $a = 0.55\text{m}$ (AGV length), $b = 0.36\text{m}$, $I_x = 0.1598\text{Kg}\cdot\text{m}^2$, $I_y = 0.1887\text{Kg}\cdot\text{m}^2$, $I_z = 0.178\text{Kg}\cdot\text{m}^2$, $I_{zx} = 0.0417\text{Kg}\cdot\text{m}^2$, $I_{xz} = I_{zx}$, $K = 0.0364$ $L = 0.0002\text{H}$, $R = 1.11\Omega$, $\delta = 26$, $R_{rad} = 0.062\text{m}$, $\mu_s = 0.013$, $\mu_k = 0.0011$, $m = 19.36\text{Kg}$.

The results of three tests consisting of performing circular trajectories on a wavy surface are shown in Fig. 6-8. In these figures the command references were the following: longitudinal speed equal to 0.2m/s and angular speed equal to 0.02rad/s . It is important to point out that Fig. 6-8 represent AGV trajectories for the above command references, which are not represented in those figures because it is unnecessary.

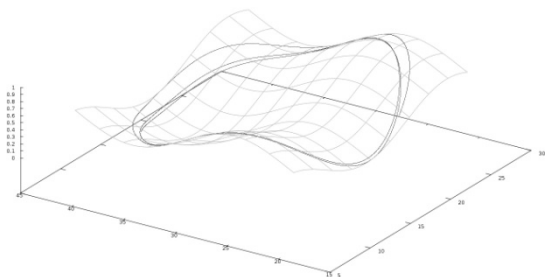


Fig. 6. PID feedback-controlled AGV system: The projection of the surface trajectory on the plane represents the x-y coordinates and the height represents the z coordinate.

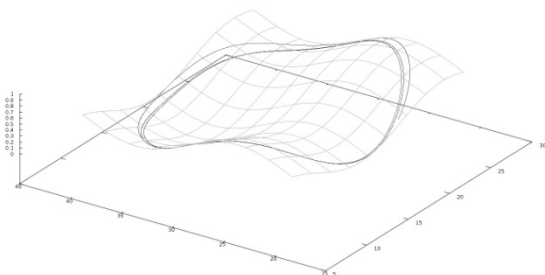


Fig. 7. Fuzzy feedback-controlled AGV system: The projection of the surface trajectory on the plane represents the x-y coordinates and the height represents the z coordinate.

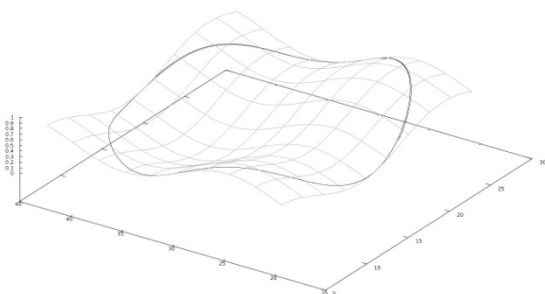


Fig. 8. AGV feedback-controlled system by using the modified adaptive LQR: The projection of the surface trajectory on the plane represents the x-y coordinates and the height represents the z coordinate.

From Fig. 6, 7 and 8 it can be seen that the fuzzy controller performed better than the PID controller, and that the performance of the modified adaptive LQR performance was the best. However, this result cannot confirm that it will always happen this way for any kind of surfaces and trajectories, but it only shows that for the specific test carried out the LQR alternative was the best.

V. CONCLUSIONS

In this paper, an AGV system was modeled as a non-linear time-varying plant, three different controllers were designed to control it and tested. A novel version of the LQR algorithm was proposed and tested. The results shown that for the kind of tests carried out here the fuzzy controlled performed better than the PID controller but that the modified adaptive LQR had the best performance of them all.

Future research works will be aimed at testing these controllers on different surfaces and different trajectories.

ACKNOWLEDGMENT

This work has been partially supported by the Ministry of Science and Innovation (MICINN) of Spain under the research project TEC2010-17429, and the Universidad Politécnicade Madrid, Spain.

REFERENCES

- [1] O.D.I. Nwokah and Y. Hurmuzlu, *The Mechanical Systems Design Handbook*, CRC Press, 2002, ch 26.
- [2] S.M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006, pp. 718-776.
- [3] R. Siegwart and I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT Press, 2004, pp. 47-82.
- [4] T.I. Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, 1999, pp. 5-28.
- [5] J. Velagic, B. Lacevic, and b. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, issue 12, pp. 989-1004, December 2006.
- [6] T. Fukao, H. Nakagama, and N. Adachi, "Adaptive Tracking Control of a Nonholonomic Mobile Robot," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 609-615, October 2000.
- [7] M. Kim, J. Shin, S. Hong, and J. Lee, "Designing a robust adaptive dynamic controller for nonholonomic mobile robots under modeling uncertainty and disturbances," *Mechatronics*. Vol. 13, issue 5, pp. 507-519, June 2003.
- [8] M. Minor, B. Albiston, and C. Schwensen, "Simplified Motion Control of a Two-Axle Compliant Framed Wheeled Mobile Robot," *IEEE Transactions on Robotics*. Vol. 22, no. 3, pp. 491-506, June 2006.
- [9] P. Tsai, L. Wang, and F. Chang, "Modeling and Hierarchical Tracking Control of Tri-Wheeled Mobile Robots," *IEEE Transactions on Robotics*. Vol. 22, no. 5, pp. 1055-1062, October 2006.
- [10] M. Eghtesad and D.S. Neculescu, "Study of the internal dynamics of an autonomous mobile robot," *Robotics and Autonomous Systems*. Vol 54, issue 4, pp. 342-349, April 2006.
- [11] K. Ogata, *Modern Control Engineering*, Prentice Hall, 1998.
- [12] W. Brogan, *Modern Control Theory*, Prentice Hall. 1991.
- [13] K. Astrom and B. Wittenmark, *Systemas Controlados por Computador*, Ed. Paraninfo, 1988, ch 3.
- [14] G. Franklin, J. Powell and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley, 1990, ch 2.
- [15] P. Albertos and A. Sala, *Multivariable Control Systems: An Engineering Approach*, Springer, 2004.