Fuzzy Self-Tuning PI Control of PM Synchronous Motor Drives

E. Chiricozzi, F. Parasiliti, M.Tursini, D.Q.Zhang

Department of Electrical Engineering, University of L'Aquila
67040 - Poggio di Roio, L'Aquila, Italy

Abstract

In this paper, a new gain self-tuning method for PI controllers based on fuzzy inference mechanism has been proposed and implemented. The essential idea is: 1). Define a dynamically changed reference trajectory in the error and error derivative plain, where we call it "sliding trajectory". 2). Compute the error area between the sliding trajectory and the effective one with a special algorithm, use this quantity as a performance index to evaluate the system response. 3) Draw a fuzzy relationship between a PI gain correction parameter and the performance index. 4) Based on the fuzzy inference mechanism, PI gains are calculated and tuned. The proposed method has been verified through simulations, using the speed control of a PM synchronous motor drive system as testbed. The obtained results demonstrate the effectiveness of this novel method.

1 Introduction

Despite the advent of a lot of new control algorithms, nowadays PI control is still one of the most popularly used control strategies in industrial applications. The key issue in designing the PI controller is to settle the gains so that the controller works well in every condition. Unfortunately for only a set of fixed gains it is very difficult to suit for a great range of working conditions. The obvious solution is either to auto-tune the gains on-line according to the environment so as to adapt to it or to assign gains from an off-line computed look-up table based on a case study of the possible working conditions.

In recent years, a lot of studies have been made in the area of gain self-tuning of PID control, with both model-based analytical method and model-free fuzzy method [1][2][3]. For the model-based analytical method, the self-tuning law is based on an assumed model and its effectiveness depends on the accurateness of the model. Because of the system non linearity, disturbance, noises etc., very often a lot of assumptions in deriving the model are violated in real applications. When the estimated model is inaccurate in some conditions, the model-based method doesn't sound solid, and thus leading to degraded controller performance. Although there don't exist quantitative mathematical formulas which can express the relationship between controller gains and system parameters, yet engineers or operators do have some experiences or intuitive rules in mind in tuning or choosing gains for their controller. Fuzzy logic is an effective tool in conveying qualitative information and expressing human's experience, by means of fuzzy inference it can approximate any non-linear relation to any extent [4]. In this sense the fuzzy inference method maybe an ideal alternative to tackle the non-linear gain tuning problem.

For gain on-line self-tuning, there are two main methods: step-based gain tuning method and cycle-based gain tuning method.

- step-based gain tuning: using the information derived from each sampling step, tune the gains accordingly. In this case, the concerned step information include the error, the error derivative, the sum of the precedent errors, etc.
- cycle-based gain tuning: using the information got from each step reference change, tune the gains so as to improve the performance. In this case, the common used cycle parameters include overshoot, rising time, etc.

In this paper, a new fuzzy PI gain self-tuning method to suit load inertia variations has been proposed and implemented; the method is based on the cycle information and derived from a detailed analysis of the speed control transient responses of a PM Synchronous Motor (PMSM) drive. The essential idea of the fuzzy gain auto-tuning mechanism is:

- For each cycle, starting from a reference variation, define a dynamically changed reference trajectory in terms of the error and the error derivative plain, where we call it "sliding trajectory".
- Compute the area of the defined strip between the sliding trajectory and the effective one with a special algorithm, use this quantity as a performance index to evaluate the system response.
- Draw a fuzzy relationship between an intermediate PI gain parameter and the performance index.
- Based on the fuzzy inference mechanism, PI gains are calculated and tuned in each cycle with the computed strip area as fuzzy input variable so that the effective response can be improved.

The paper is organised in six parts. Sec.2 begins with the description of the PI controller and the system, then it presents the three typical responses with the system modelled by a transfer function. Simulations of the real system in the same operating conditions are shown in Sec.3. After putting the speed responses in the error and error derivative plain, some remarkable characteristics of the responses are analysed. In order to tune the PI controller parameters accordingly, a dynamic sliding trajectory is first defined, then an algorithm for computing the defined strip area (performance index) is proposed. Sec.4 draws a fuzzy relation between the PI controller parameters and the performance index. Based on the simulation results, the number of fuzzy labels, membership functions and fuzzy gains are specified subjectively, the fuzzy inference mechanism is also described. In Sec.5 simulation results and comments with the fuzzy gain self-tuning

IEEE Catalogue No. 95TH8025
0-7803-2423-4/95/$4.00©1995 IEEE

mechanism for speed control of a PMSM drive are presented. Finally, some conclusions are given.

2 PI control: general description and ideal cases

Fig.1 shows the typical speed control scheme of an electrical drive using standard PI controller [5].

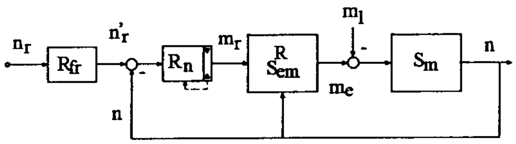

Fig.1 Speed control scheme of an electrical drive

In the figure R_n refers to the speed controller and the cascade part (S^R_{em} + S_m) represents the controlled system, where S^R_{em} is the internal torque control loop (electromagnetic controlled system) and S_m the mechanical system. The scheme also shows the load torque disturbance m_l. Controller design is then carried out using the Laplace approach, all the blocks are represented by transfer functions. According to the designing technique for nested control loops, the internal torque loop (S^R_{em}) is simplified as a first order transfer function

$$G^R_{em}(s) = \frac{1}{1+sT_{pe}} \tag{1}$$

where T_{pe} is the equivalent time constant of the electromagnetic controlled system, whose value depends on the dynamic characteristics of the torque controller.
The mechanical system (S_m) is represented by

$$G_m(s) = \frac{1}{sT_m} \tag{2}$$

where T_m is the mechanical time constant, proportional to the inertia of the rotating mass of the whole system.
The PI speed controller (R_n) takes the form of

$$G_{Rn}(s) = \frac{1}{sT_i} + \frac{T_n}{T_i} = \frac{k_i}{s} + k_p \tag{3}$$

where k_i =1/T_i and k_p =T_n/T_i are the integral and proportional gains, respectively.
In order to improve the responses to both load and reference variations [6], a speed reference filter (R_{fr}) is also introduced, with the form of

$$G_{fr}(s) = \frac{1}{1+sT_n} \tag{4}$$

The design of the PI controller consists in the calculation of the parameters T_n and T_i, supposing the time constants of the control system T_{pe} and T_m are known.
According to [6] the optimal choice is as follows:

$$T_i = \frac{8T_{pe}^2}{T_m}; \qquad T_n = 4T_{pe} \tag{5}$$

For the sake of analysing the effect of load inertia variation, we suppose that the controller is designed for an inertia different from the actual one.
With such an assumption and considering (1÷5) the transfer function of the speed closed loop is given by

$$\frac{n}{n_r} = \frac{1}{1+s4T_{pe}+s^2 8T_{pe}^2 \frac{J_m}{J_c}+s^3 8T_{pe}^3 \frac{J_m}{J_c}} \tag{6}$$

where J_c and J_m are the p.u. inertia assumed in the controller and in the system, respectively.
Eq. (6) describes the dynamics of the speed control system. In the case J_c=J_m the optimal response is obtained, with a rising time of about 7.5T_{pe} and an overshoot of about 7.5%. The response is illustrated in fig.2 with the label (J_c=J_m=1), The system is assumed to have T_{pe}=5.5 ms and T_m=34.9 ms.

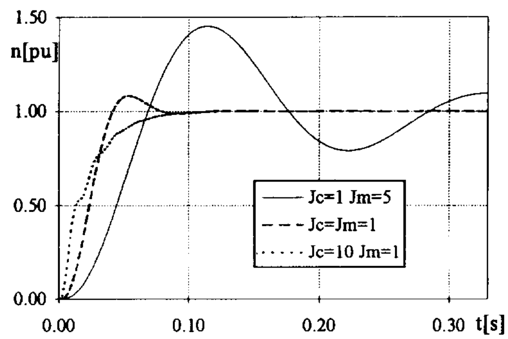

Fig.2 Speed responses (ideal system)

If the load inertia is varied and the corresponding controller inertia remains unchanged, that is $J_c \neq J_m$, then the optimal response doesn't hold. Fig.2 also demonstrates the responses obtained with a controller designed for an inertia smaller (J_c=1; J_m=5) or greater (J_c=10; J_m=1) than that of the actual system. In the case (J_c=1; J_m=5), the response has an increased overshoot compared with the optimum design. In the opposite case (J_c=10; J_m=1), the response has a damped behaviour with superimposed oscillation. Obviously the difference between the inertia imposed in the controller and that inherent in the system influences the system responses greatly, so a possible solution to improving the system responses is to adjust the inertia imposed in the controller so that the difference is minimised. In this work, the strategy of only tuning the parameter J_c is adopted, with the idea that J_c is the predominant parameter affecting the PI gain and responses. Fig.3 shows the trajectories of Fig.2 in the plane (x1, x2), where x1=n_r-n is the speed error, and x2=K· dx1/dt is the error derivative.

3 PI control of a PMSM drive: practical consideration and analysis

Fig.4 shows the block diagram of a PMSM drive. The power supply is characterised by a voltage source inverter modulated by hysteresis current regulators. The motor is controlled according to the principle of the orientation on the rotor flux (control in the rotor d,q reference frame).

As it is known, a linear relationship between current amplitude and torque is guaranteed by setting at zero the reference of the direct axis current. With such an assumption the reference of the quadrature current can be obtained by the speed controller. A d,q $\Rightarrow$ abc transformation gives the current references in the natural three phase reference system, needed for the hysteresis regulators.

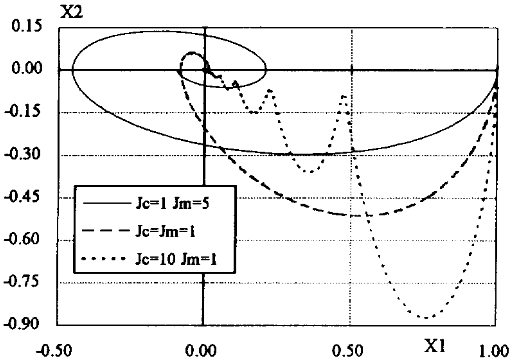

Fig.3 Trajectories in the error and error derivative plain (ideal system)

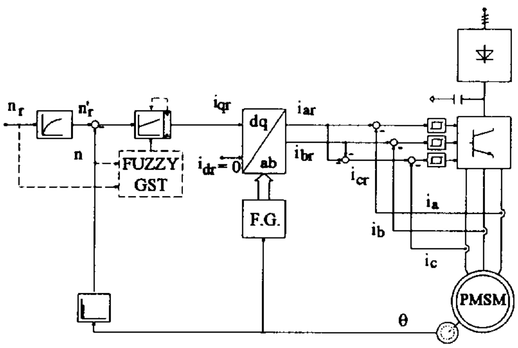

Fig.4 Block diagram of the PMSM drive

The speed controller is provided with current saturation. A correction of the integral component during saturated transients is also provided, in order to improve system dynamics. Those aspects introduce so many non-linearities that the actual system differs from the ideal one as described by (6).

In order to analyse the whole system, a simulation program has been developed, where each part of the drive has been taken into consideration. Fig.5 and fig.6 show the responses to the speed step variation in same conditions as in fig.2 and fig.3 respectively. It can be noted that in spite of the presence of the above mentioned non-linearities, the responses maintain the same characteristics as in the ideal case.

Obviously, the responses can be grouped in three categories. That is,

- $J_c=J_m$, "optimal" response with target overshoot;
- $J_c<J_m$, with overshoot greater than target one.
- $J_c>J_m$, with overshoot smaller than target one or without overshoot.

Interesting enough, from Fig. 3 and Fig.6 we can see that:

- all three trajectories start from the point in x_1-axis which corresponds to the step change of reference, and end at the origin of coordinates which corresponds to the steady state with both error and derivation of error equal to zero.
- In case $J_c<J_m$, we can find a point where the trajectory crosses over the "optimal" one (corresponding to $J_c=J_m$). After that point the trajectory approaches the steady state point remaining external to the ideal one and forming an irregular strip with it. In order to mark the sign of this error area, we define it "positive".
- Similarly, in case $J_c>J_m$, there is a point after which the trajectory moves towards the steady state point remaining internal to the ideal one. In this case the formed irregular strip area is defined "negative".

From the above observation it is easy to have the idea of using the sign and quantity of the area of the strip as the Jc tuning parameter. But in real applications an "optimal" reference trajectory can not be obtained on-line. Moreover, with different parameters such as speed reference, load torque etc., the quantity of the area also varies. In fact the real situation is so complex that one cannot quantitatively analyse all the factors which can influence the trajectory. Fortunately, some qualitative and quantitative information about the relative difference between the effective trajectory and the optimal one do give some indication of tuning the related PI gain parameters in the direction of targeting the optimal response gradually.

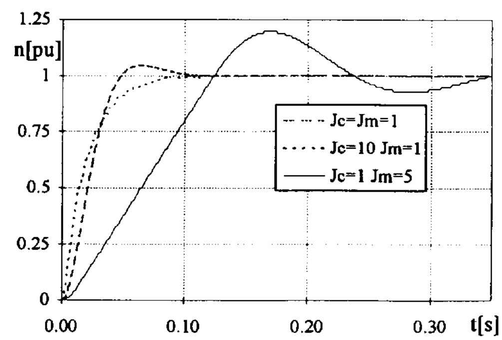

Fig.5 Speed responses (actual system)

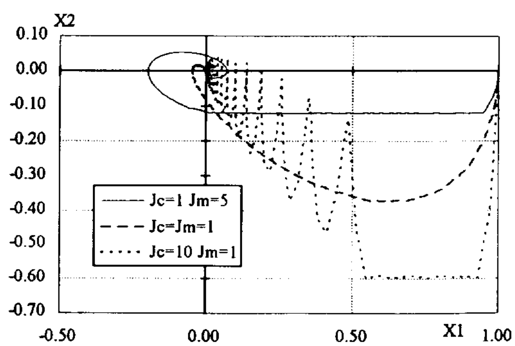

Fig.6 Trajectories in the error and error derivative plain (actual system)

Theoretically we can define a desirable target overshoot for a given reference change, e.g. in the x_1-x_2 plain, a key point of the desirable trajectory is known. With the idea of tuning

J_c using the cycling strip area, a dynamic reference trajectory which can reflect the feature of "optimal" trajectory has to be defined first, then a general-purpose computing algorithm which can represent both the adjusting direction (sign of the strip) and the degree of the correction (area of the strip) for all conditions are needed. In this research we have defined a dynamic reference trajectory called "sliding trajectory" as shown in fig.7. The aim is just to generate a simple but effective reference on-line with which correct area information can be computed. The sliding trajectory for a positive change of reference is defined as follows:

*1) **IF** (current point = starting point) **OR** (x1=OV)*
*   **THEN** add this point to the sliding trajectory*

2) for current point (x1, x2):
* **IF** (|x1| > |OV|) **AND** (x1<0)*
*   **THEN** add (OV,x2) to the old trajectory*

3) Connect old trajectory point (x1', x2') with point (OV,0) to form reference trajectory, then:
* **IF** (current point (x1, x2) is external to the trajectory)*
*   **THEN** add current point to the sliding trajectory*
*   **ELSE** keep the old trajectory.*

As an example, when at point 1 the sliding trajectory is A→1→OVp, but at point 2, since point 2 is external to the line A→1→OVp, so the trajectory is modified as A→1→2→OVp. For point 3, because point 3 is internal to the trajectory A1→1→2→OVp, so the trajectory remains unchanged. With the above sliding trajectory generation method, the feature of the "optimal" trajectory is maintained. Fig.8 shows the generated reference trajectories corresponding to the two extreme cases considered in fig.3: curve A→B→OVp refers to the final sliding trajectory for J_c=1, J_m=5, and curve A→C →OVp to the case J_c=10, J_m=1.

The next step is to specify the strip area computation algorithm. At each sampling time the trajectory error is calculated by

$$E(k) = x1_s(k) - x1(k)$$

where x1 is the speed error and $x1_s$ is its counterpart on the sliding trajectory. Then the strip area is computed by

$$IE(k) = IE(k-1) + k_E\, E(k)$$

where k_E is a scaling factor depending on the speed reference and the target overshoot values, so as to assure that the computed area has a proportional relationship with the PI gain correction factor. The defined strip areas from instants t_0 to t_1 are marked in fig.8 for both the considered extreme cases.

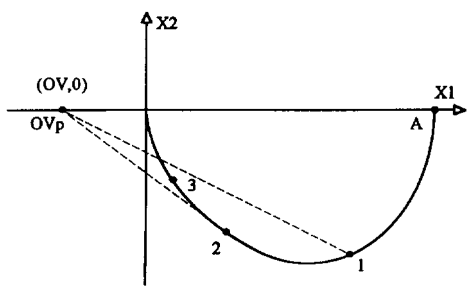

Fig.7 Definition of the reference trajectory

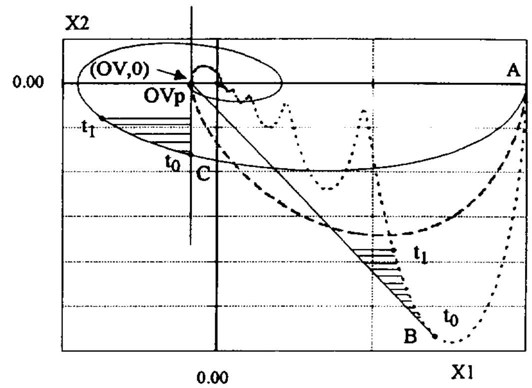

Fig.8 Definition of the computed error area

4 Fuzzy gain tuning method description

With the above analysis and definition, we have formalised the gain tuning idea and drawn some qualitative relation as follows:

- when $J_c \approx J_m$, IE≈0;
- when $J_c < J_m$, IE >0;
- when $J_c > J_m$, IE <0;

where IE stands for the computed strip area. Apparently, the bigger the difference between J_c and J_m, the larger the absolute value of IE. In this way we can easily generate some fuzzy rules as follows:

IF *(IE is PB)* ***THEN*** *(ΔJ_c is PB)*
IF *(IE is PS)* ***THEN*** *(ΔJ_c is PS)*
IF *(IE is ZE)* ***THEN*** *(ΔJ_c is ZE)*
IF *(IE is NS)* ***THEN*** *(ΔJ_c is NS)*
IF *(IE is NB)* ***THEN*** *(ΔJ_c is NB)*

where ΔJ_c refers to the correction needed for the controller inertia J_c, PB, PS, ZE, NS, NB stand for, respectively, Positive Big, Positive Small, Zero, Negative Small, Negative Big. For the sake of simplicity, the fuzzy rules have taken the form proposed by Bernard:

IF *(IE is A_i)* ***THEN*** *(ΔJ_c is SC_i)*   *i=1,2..N*

where IE and ΔJ_c are the fuzzy input and output variables, A_i are fuzzy sets and SC_i are singletons.

Because the defined strip varies from case to case, and may take many irregular shapes, so the relationship between ΔJc and IE is highly non-linear, IE distribution also shows great asymmetry with regards to positive and negative values. To guarantee smooth and bounded adaptation of the PI gains, fuzzy gain is set to 300 subjectively and the range of interest IE is covered by thirteen fuzzy labels, that is (N8, N7, N6, N5, N4, N3, N2, N1, ZE, P1, P2, P3, P4), where the meanings of the acronyms are, respectively, from N1 to N8 for negative terms, ZE for zero, from P1 to P4 for positive terms. Based on the simulation results, the membership function of each label for input and output variable has been determined with trail-and-error method as shown in fig.9 and fig.10. Correspondingly, the fuzzy rules accumulate to 13.

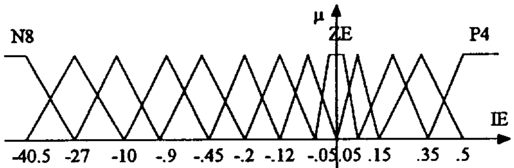

Fig.9 Membership function of IE

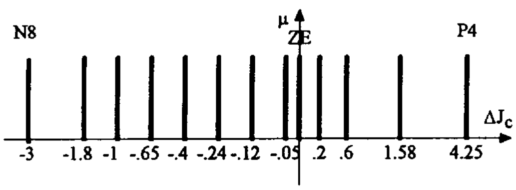

Fig.10 Membership function of ΔJc

The fuzzy reasoning has been done using the max-prod compositional rule of inference. The output fuzzy set is obtained by the union of the N output fuzzy sets produced by the single rules, whose membership functions are:

$$\mu_{SC_{i0}}(\Delta J_c) = \alpha_i \, \mu_{SC_i}(\Delta J_c) \quad i=1,2\ldots N$$

where α_i is the firing strength of the i-th rule:

$$\alpha_i = \mu_{Ai}(IE)$$

The defuzzified crisp value obtained by the centre of gravity method has been simply calculated as:

$$\Delta J_c = \frac{\sum_{i=1}^{N} \alpha_i \cdot SC_i}{\sum_{i=1}^{N} \alpha_i}$$

With the preceding design details of the new gain self-tuning scheme, a rough account of how it actually works can be summarised as follows:

- At each sampling step, the sliding trajectory is revised and a specially defined strip area between it and the effective response trajectory is computed.
- At each variation of reference, which marks the beginning of another cycle, the parameters of the controller are tuned based on the computed area and the fuzzy inference mechanism. So new gains are got and applied in the new cycle.

5 Fuzzy self-tuning PI control for PMSM drives: simulations and comments

The developed fuzzy Gain Self-Tuning (GST) mechanism has been integrated in the PMSM speed control scheme as shown in fig.4 (dashed line).

The aim of the method is to improve the step change response gradually and to assure a certain system overshoot target reached with a reasonable rising time. In contrast to the simulations shown before, typical tests have been done with different set-points and extreme initial conditions, all with an overshoot target 7.5%.

Fig.11 shows the case with J_{c0}=1 and J_m=5, where J_{c0} is the initial value of the inertia assumed in the controller. With the proposed mechanism, the speed response is improved quite rapidly and only after a few cycles, the target overshoot is obtained. Fig.12 demonstrates the case with J_{c0}=10 and J_m=1, in this extreme case the initial response starts without overshoot. Clearly, same conclusions can be drawn as its counterpart.

Fig.13 presents the remarkable speed responses in the case of J_{c0}=1 and J_m=5 with four different speed reference values. Apparently, all the responses "converge" quickly and the satisfactory results prove the effectiveness and applicability of the proposed method to the change of working point and load inertia.

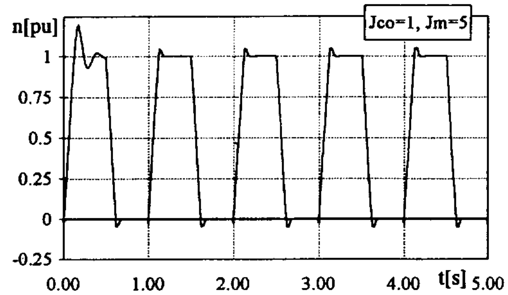

Fig.11 Speed response with initial overshoot

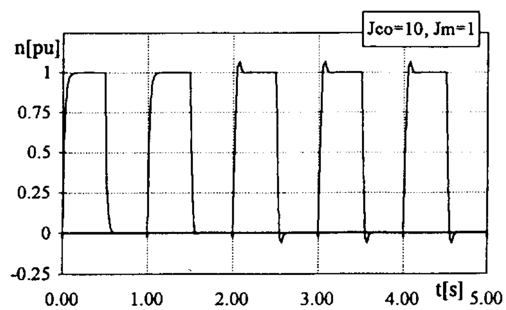

Fig.12 Speed response without initial overshoot

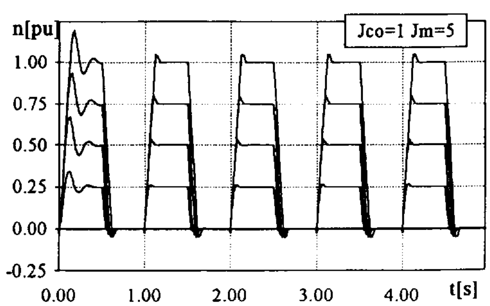

Fig.13 Speed response with different set points

6 Conclusions

In this paper, a new fuzzy gain self-tuning method based on the cycle information for PI controller has been proposed and implemented, its basic principle, design procedure and simulation analysis has been described in detail. The presented method has been illustrated by means of the design of the speed PI controller of a PMSM drive system, the simulation results demonstrate the effectiveness of the novel method.

Fuzzy logic can be directly applied in the design of a fuzzy logic controller. In this case it serves as a direct controller in the sense that it generates the control command directly. In fact, there is a great potential in using the fuzzy logic as part of a control scheme and here it works as an indirect controller. Fuzzy self-tuning PI controller is one example of the fuzzy indirect controllers; it incorporates the adaptive ability to the traditional PI controller so that the resulting hybrid controller may adapt itself to varying working conditions and thus gives very high performance.

References

[1] K.J. Aström and B. Wittenmark, *Adaptive Control*, Addison-Wesley, Reading, MA.,1989.

[2] S.Z. He, S. Tan, F.L. Xu and P.Z. Wang, "Fuzzy self-tuning of PID controllers", *Fuzzy Sets and Systems 56*, pp37-46, 1993.

[3] K. Izumi, M. Tsuji, J. Oyama and E. Yamada, "Improvement of fuzzy auto-tuning method of DC chopper system using manipulated values", *IECON'93*, Hawaii, USA.

[4] L.X. Wang, "Fuzzy systems are universal approximators", *Proc. IEEE Int. Conf. on Fuzzy Systems*, San Diego, USA, pp. 1163-1170,1992.

[5] F. Parasiliti and M. Tursini, "A comparative analysis of different speed control methods for a PM synchronous motor drive", *EDS'90*, Capri, Italy, pp. 181-186, 1990.

[6] H. Bühler, *Élecronique de réglage et de commande*, Traité d'Électricité, Edition Georgi, 1979.

This work is supported by CNR "Progetto Finalizzato Robotica", Contract n° 9300891PF67.