A Simple PID Controller with Adaptive Parameter in a dsPIC; Case of Study

João Chaínho, Pedro Pereira, Silviano Rafael[1] and A.J. Pires[1]

[1] LabSEI - Escola Superior de Tecnologia de Setúbal
Instituto Politécnico de Setúbal/Campus do IPS, Estefanilha,
2914-508 Setúbal, Portugal
Telef:+351 265 790 000, e-mail: srafael@est.ips.pt

Abstract: The main goal of this work consists in the development and implementation of a discrete PID controller with fast response and parameters adaptation capability, in an automatic way. This controller is based on a classic PID where a parameters adaptation algorithm was associated in order to control a process. This PID do not require any kind of adjustment or calibration from the operator. For the parameters adaptation one fuzzy system with a Takagi-Sugeno inference mechanism was chosen and some simplification of this system algorithm was implemented. These simplifications had the goal of decreasing the processing time and the controller response (250μs), in order to control fast processes without losing stability. The developed algorithm was implemented in a recent dsPIC30F.

Keywords

PID Controller, Adaptive Parameter, Fuzzy logic system, control systems.

1. Introduction

The PID controller is much used in the control loops of industrial processes. Its parameters are need to be adjusted in function of the control process and remain unchanged during its regular activity.

The start up of the PID controller requires a not always simple work in the parameters adjustment, besides the existence of some methodologies, described in [1]. Despite the helpful of these methodologies in the approached parameters values calculation, is however necessary an observation period to survey with greater certainty the controller performance, which requires, in some cases, a substantial amount of time. This is interpreted as a disadvantage or a difficulty in the controller start-up service.

Other more complex cases exist due to their particularities, where there are small procedures changes that compromise the PID controller performance. These situations are observed by the trends maps analyzed by the process operators, occurring the necessity of a controller parameters readjustment. The reason is difficult to define or explain, being, most of the time, from diverse procedures aspects.

This work presents a proposal that contributes to reduce or even to prevent the already referred problems. The imposed requirements are: fast response, do not need any type of previous adjustment and universality of communication with the sensors, automates (PLC) or scada systems (DCS).

2. PID Controller

The used discrete PID controller is characterized by the following equation (1).

$$u_{(t)} = K_{p(t)} e_{(t)} + K_{i(t)} T_s \sum_{k=0}^{t} e_k + K_d \frac{e_{(t)} - e_{(t-1)}}{T_s} \quad (1)$$

Where $e_{(t)}$ is the error of the system response in the t instant , T_s is the signal sampling period and K_p, K_i and K_d are the proportional, integral and derivative controller gains, respectively.

This algorithm, associated with the error calculation, is of very fast execution, however its parameters should be previously and appropriately adjusted.

Actually, there are various calculation and parameters adjustment methods for PID controllers (K_p, K_i and K_d). From static parameters adjustment methods, like Ziegler – Nichols and Kitamori methods, to methods where the parameters are dynamic, depending on the system response, as, for example, the ones based on Fuzzy Logic systems [2, 3,4], Neural Network systems or Neuro-Fuzzy systems[5]. The disadvantage of these last ones is the need of too many processing resources, being therefore usually slower.

3. Adaptive Algorithm

The considered adaptive algorithm intends to have the advantage of simplicity and to be implemented with few hardware resources and simultaneously to obtain a reduced implementation time (processing cycle time). The question related with the processing time is very important because it limits the quickness of the control signal, the quickness of the controller parameters adaptation and consequently it limits the set performance and behaviour in the reference signal tracking.

The adaptive algorithm is inspired in a Tagaki-Sugeno fuzzy system [6] to which some simplifications were applied. In this type of fuzzy system, the condition part uses linguistic variables and the conclusion part is represented by a mathematical function. The proposed system has four conditions and two distinct conclusions and it can be represented as in figure 1. It is characterized by one universe of discourse that is the error percentage.

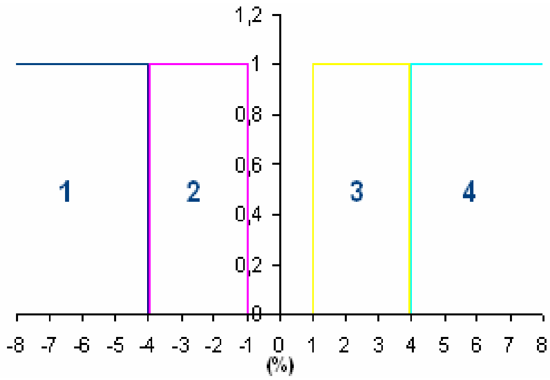

Fig. 1. Distribution and type of membership functions

The error percentage, *Pe*, is defined by (2) and the equation selection way and activation system of the adaptation expressions is made by four rectangular membership functions distributed like is showed in figure 1.

$$Pe = \frac{Y_{ref} - Y_{t(process)}}{Y_{ref}} * 100 \tag{2}$$

The membership functions $n = \{1, 4\}$ activate expression (3) when the error percentage is lower than -4 or greater than 4, actualizing the K_p parameter value.

$$K_{p(t)} = K_{p(t-1)} + \vartheta . e_{(t)} . \mu^n \tag{3}$$

The membership functions $n = \{2, 3\}$ activate expression (4) when the error percentage is in the interval]-4, -1] or [1, 4[, actualizing the K_i parameter value.

$$K_{i(t)} = K_{i(t-1)} + \vartheta . e_{(t)} . \mu^n \tag{4}$$

In expressions (3) and (4) ϑ represents the adaptation factor, e(t) the error value in the instant t and μ_n the strength activation of the adaptation function of the membership function n.

The adaptation factor is loaded initially with a value between 0 and 1 and will remain constant to the long one of the system functioning. The value of this factor is important because it will influence the adaptation speed of the K_p and K_i parameters.

The error value in the instant t is defined by (5). Depending on the activation of the membership function and the error signal value the parameter value K_p (or K_i) is incremented or decremented. The increment or decrement value will depend on the magnitude of the error due the remaining elements are constant.

$$e_t = Y_{ref} - Y_{t(process)} \tag{5}$$

The strength activation, μ_n, of the membership function assumes the value 0 or 1 due to the used rectangular membership function, and this implies the activation of (3) or (4) in order to maintain or change the parameter value.

4. Simulation

This controller with parameter adaptation was first simulated in Matlab Simulink. The block diagram of the model is shown in figure 2. The named control block in figure 2, implements the Tagaki-Sugeno fuzzy system presented in figure 1. The Ki control and Kp control blocks implement the parameter adaptation algorithm of K_p and K_i with a sampling frequency, Fs. This defines therefore the cycle time that will go to exist in the practical implementation of the control system. The PID block represents the PID algorithm presented in (1). The saturation block implements the physical limitation of the output values of the real controllers.

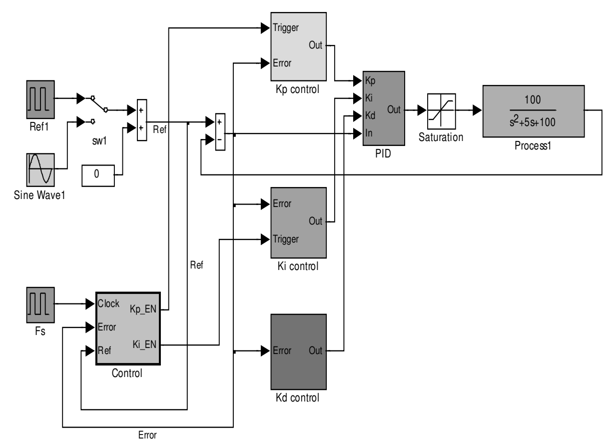

Fig. 2. Blocks diagram of the system model simulation

5. Implementation

The described system was implemented in connection with a processor logic controller (PLC) replacing the PID controller in the PLC due the speed response requirement of the control loop. The block diagram of this system is presented in figure 3.

Related to this system interface with the PLC and the process, it was intended that it was simultaneously efficient and universal, in order to allow the linking with any one PLC. The use of the PLC communication port was studied initially, with which it would be possible to implement advanced monitor and parameter operations. However the communication protocols of this kind of interface are many times specific of each manufacturer who represents a drawback regarding the principal objective. So the control system connected to the PLC was made hardwiring by two analogical signals, the PLC reference signal and the control system output signal, and two binary signals that represent the functioning status set.

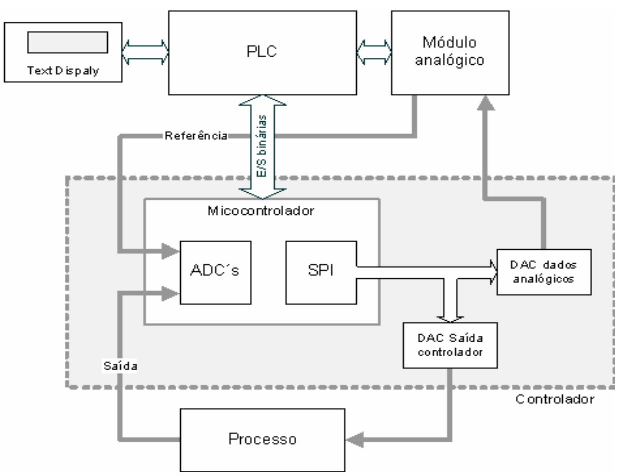

Fig. 3. Blocks diagram of the system

For monitoring the state of the variable value of this controller system, after development, is used a text display module.

A. Software

The principal routine is showed in figure 4 which schematizes the main flows of the algorithm codification.

In a first place the configuration and definition instructions of the initial microcontroller state are executed. Next, if the PLC sends the RUN binary command signal, the circuit start-up mode will activate the RUN_STATUS binary signal and will execute the next sub routine algorithm. In contrary the microcontroller places zero in the output process controller and enters in SLEEP mode, of which will only return to start up when the PLC activates the RUN command signal.

The cycle goes on with the "Data acquisition" block processing the reference data (RefADC) and the output process (ProcOutADC), proceeding from the ADC's. These data had previously been processed by the routine of the interruptions attendance of the ADC's. The results values (changeable Ref and ProcOut) as well as many others are normalized values between 0 and 1.

The sub-routine "Calc Error Values" calculate the variable associates to the error, for example the proper error, its percentage, its derivative, integral and absolute values. An important detail is the entrance of the cycle time, T_s (or the sampling frequency, F_s) for the integral and the derivative calculation.

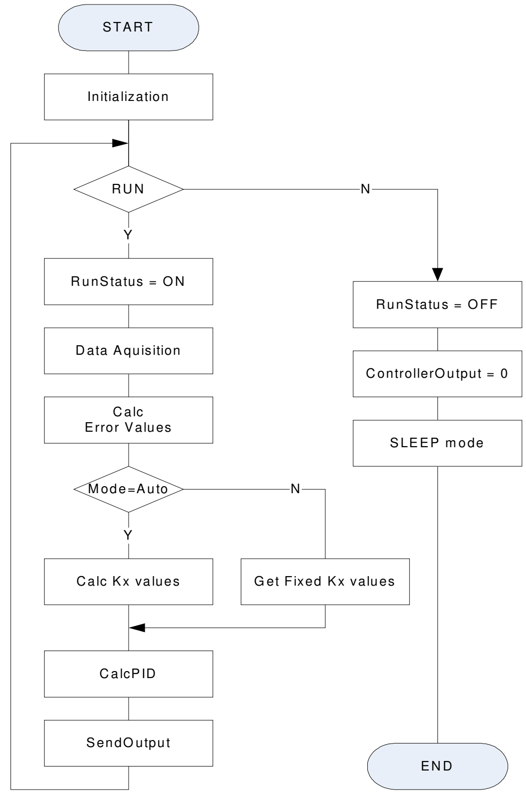

Fig. 4. Blocks diagram of the system

The sub-routine "Calc Kxx values" puts in practice the adaptive algorithm previously explained in figure 1.

The sub-routine "CalcPID" calculates the output signal that will be applied to the process on the basis of equation (1). The necessary variable had already been calculated.

The "SendOutput" routine sets up the two output DAC and counts the cycle time.

This controller was developed in integrated MPLAB^R IDE software tools allowing the programming and debugging functions, presented in figure 5. Another great advantage is related with another tool, the Visual Device Initializer, that is integrated in the development software and supplies a visual way for configuration of the diverse internal modules of the controller. This one prevents the hard and difficult task of manual configuration of all registers.

B. Hardware

The criteria for the choice of the microcontroller DSPIC was the possibility's of incorporate a lot of peripherals (like ADC's, DAC's), communications peripherals (like UART, SPI, I2C, etc) and DSP functionalities (like multipliers and accumulators blocks) too. On the other hand, they present high processing speed and superior architectures than the normal µC of 8 bits.

The clock signal applied to the microcontroller is supplied by a crystal whose frequency later is multiplied 16 times with an internal PLL block. To obtain the maximum frequency of 120MHz praised by the manufacturer, it is necessary to use a crystal of 7,5MHz. So each machine cycle is executed in four clock cycles obtaining a maximum of 30 million cycles per second (30MIPS).

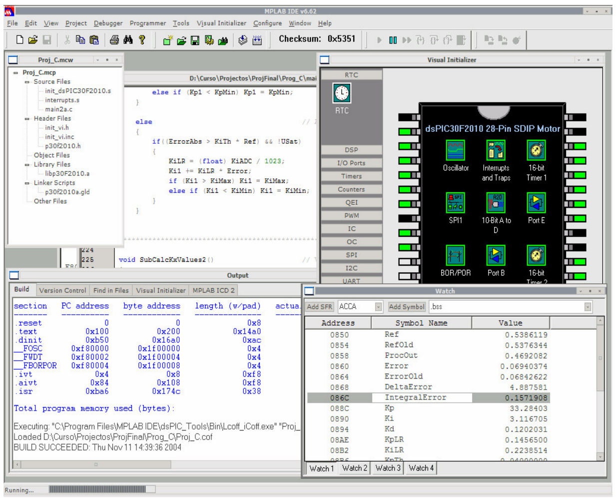

Fig. 5. Environment of MPLAB[R] IDE development

6. Experimental Results

The necessity to test the practical implementation of the circuit and observe its performance implied the connection of this one to some well known processes.

First it was tested the behaviour of the controller with a first-order process characterized by the transfer function in (6).

$$H_{(s)} = \frac{200}{s+200} \tag{6}$$

In figure 6 it is observed the tracking of a rectangular reference signal function between 20% and 80% of the maximum process value and the response when the system is of first order.

The used adaptation factor (ϑ) was 0.1 for the K_p and K_i parameters calculation.

The figure 6 evidences the control system adaptation that starts from zero in tracking the rectangular signal reference. In the third step it is already not observed a static error.

The same behaviour of this system was observed in tracking the sinusoidal and triangular reference signals with the same time period of the rectangular reference signal.

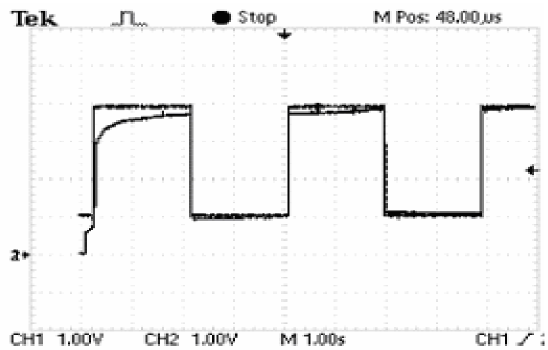

Fig. 6. Tracking of a rectangular reference signal with a process of first order

Next it was tested the behaviour of the controller with a second-order process characterized by the transfer function in (7).

$$H_{(s)} = \frac{210^2}{s^2+969s+210^2} \tag{7}$$

The K_p and K_i adaptation depends on the evolution of the error value and the adaptation factor value (in figure 7 the adaptation factor value is always the same and equal to 0.1 and in figure 8 the adaptation factor value is 0.5). Comparing figure 7 with 8 it is easy to observe that last one converges more quickly and the process output was also entered more quickly in the K_i adjustment zone diminishing the static error.

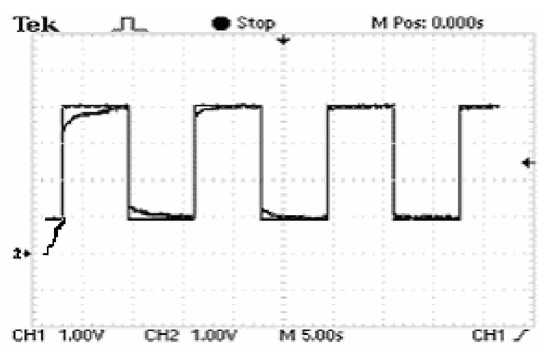

Fig. 7. Tracking of a rectangular reference signal with a process of second order and $\vartheta = 0.1$

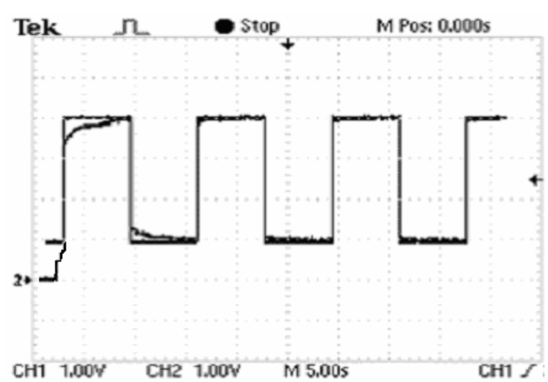

Fig. 8. Tracking of a rectangular reference signal with a process of second order and $\vartheta = 0.5$

There exists some relation between the adaptation factor value and the frequency of the signal reference. Figure 9 show the system control with a second order process with almost the double frequency of the signal reference that is in figure 7. The used adaptation factor was 0.1. It show that the system needs more time to reach acceptable values due to the increase of the signal reference frequency. This implies a minor number of adaptation cycles between constant values of reference signal steps. One solution could be to increase the adaptation factor value.

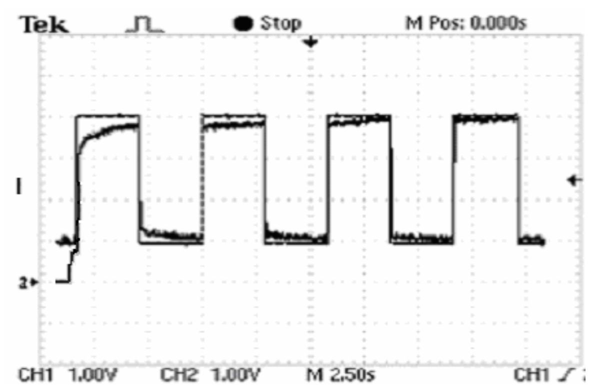

Fig. 9. Tracking of a rectangular reference signal with a process of second order and $\vartheta = 0.1$

7. Conclusions

This work presents a PID with a parameter adaptive algorithm and its performance with a first and second order processes. For the parameters adaptation one fuzzy system with a Takagi-Sugeno inference mechanism was chosen and some simplification of this algorithm system was implemented.

The main advantage of the presented system is that it does not need any kind of adjustment or PID calibration. It has the advantage of the adaptive systems, quickly compensating the disturbances that can appear in the system control functioning.

The K_p and K_i adaptive algorithm that is demonstrated in this work, is quite simple, robust and converge quickly. The limitation of the membership in +4%,-4% and +1%, -1% had given the best experimental response results. It is observed that the PI parameters depend on the evolution of the error value and the adaptation factor value.

One limitation appears in the third order process systems, which is in phase of study and implementation.
The applications are the most diverse in the industry, due to its simplicity, use of usual electrical measures and fast start up application and use.

References

[1] Aström K.J., Hägglund T., "PID Controllers – Theory, Design and Tunning", Instrument Society of América,67 Alexandre Drive,POBox 12277,Research Triangle Park, North Carolina 27709,USA, 1995.

[2] J.H. Kim and S.J. Oh, "A fuzzy PID controller for nonlinear and uncertain system", in Soft Computing, 4,Springler Verlag, pp. 123-129, 2000.

[3] Pauli Viljama and Heikki Koivo, "Fuzzy logic in PID gain scheduling" *Third European Congress on Fuzzy and Intelligent Technologies EUFIT'95*., ELITE-foundation, vol. 2, pp. 927-931, Aachen, Germany, August 28 – 31.

[4] JJ. Buckley, Sugeno type controllers are universal controllers, Fuzzy Sets Syst. 53, 1993), pp 299-303.

[5] C.T.Lin, " Neural Fuzzy Control Systems with Structure and Parameter Learning", Wordl Scientific Publishing Co. Pte. Ltd. ISBN 981-02-1613-0, 1994.

[6] P.Liu and H.Li, "Hierarchical TS fuzzy system and its universal approximation",. Information Sciences 169 (3-4): 279-303 Feb 1 2005