

FUNCTIONALITY EXTENSION FOR A D.C. MOTOR SPEED CONTROL DIDACTIC SYSTEM WITH PID CONTROLLER

SANDA DALE¹, C.R. COSTEA¹, S. ZSOLT²

¹University of Oradea, Control Systems Engineering and Management Department, str. Universității nr.1, 410186 Oradea, România, ² SC Connectronics Romania SRL, Str. Borsului, nr. 40, 410605, Oradea, România.
E-mail: sdale@uoradea.ro, ccostea@uoradea.ro.

Abstract. *The paper describes some functionality extensions for a D.C. motor speed control system with PID controller used as a didactic unit. The extensions are made in two directions: one at a physical-hardware level and the other one at software development of numerical algorithms and graphical interface. Both developments have the aim to extend and to get a more flexible paradigm for the control algorithms study on both continuous and discrete approach and to add the possibility to compare different control solutions.*

Keywords: extended functionality, PID controller, numeric algorithm, microcontroller.

1. INTRODUCTION

The real-time applications in engineering have a different perspective when we look at them from the didactic point of view [1], [5]. The processes to be controlled and the procedure of designing the controllers have to be at the beginning simple and transparent, easy to follow and to understand. Once understood this, in further steps the design process can become more and more complicated and is recommended to introduce new algorithms and to compare them step-by-step with the previous ones [3], [4], in order to achieve complete knowledge about their utility and advantages.

In this context, starting from an analogic PID speed control board for a D.C. motor system an extension is proposed to be made with the aim to offer more flexibility and the possibility to develop and to implement all kind of numerical algorithms easy to upload in a microcontroller [2], [6].

Around the microcontroller a development board was designed, with a USB interface for uploading the codes directly from the PC, without special programmers.

In section 2 the D.C. motor system is presented and in section 3 the development board and the graphical interface design are sketched.

In section 4 a MATLAB-SIMULINK approach is used to design and to verify a numerical PI algorithm, starting from a continuous PID structure, modeled from the real didactic board with D.C. motor.

In section 5 some experimental results from the MATLAB-SIMULINK simulated system and from the real one are presented, comparatively. Finally, some remarks conclude the study, in section 6.

2. D.C. MOTOR CONTROL SYSTEM WITH PID CONTROLLER

The initial system [7] where from the developments start is a D.C. motor drive with permanent magnets operating an inertial load through a gear with 8:1 transmission ratio. The load is an aluminium disc with inscribed indicator. The system is driven from the output of a power amplifier and the position is measured through a potentiometer with continuous rotation connected to the aluminium disc by a gear with 3:5 transmission ratio. The scheme of the system is shown in Figure 1.

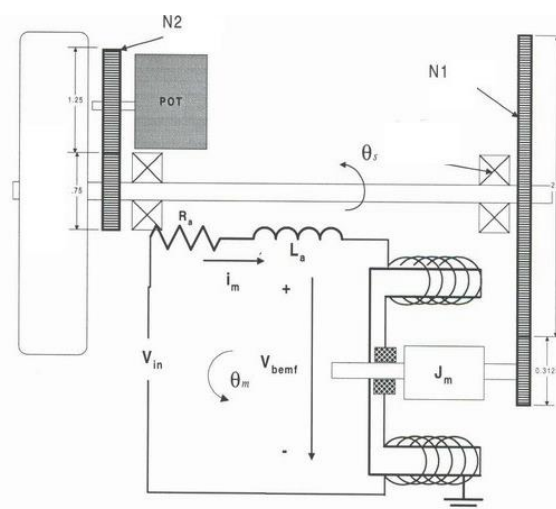


Figure 1. D.C. motor scheme

The constants of the system are the following: Inertia moment of the disc: $J_{ind}=4.09 \cdot 10^{-3}$; Inertia moment of the load $J_{load}=1.25 \cdot 10^{-2}$; Internal inertia moment $J_b=1.26 \cdot 10^{-3}$; Engine-load transmission ratio $N_1=0.125$; Load-potentiometer transmission ratio $N_2=0.6$. The engine system has a single input: the voltage applied to the motor input terminals and a single output: the voltage measured at the output of the potentiometer.

The ensemble motor - power amplifier – tachogenerator has the transfer function in equation (1).

$$H(s) = \frac{5403}{s^2 + 55.8 \cdot s + 774} \quad (1)$$

The aim of the system with PID board is to control the position of the inertial load through controlling the command voltage applied at the entrance of the power amplifier. The inertial load is intended to rotate with angles between 0°-75°, and the controlled variable variation is between 0-4V, with the possibility of reversing the rotation.

The command block has the structure in Figure 2 and the transfer function in (2).

$$\frac{C(s)}{E(s)} = K_p + \frac{1}{T_i \cdot s} + K_d \cdot s \quad (2)$$

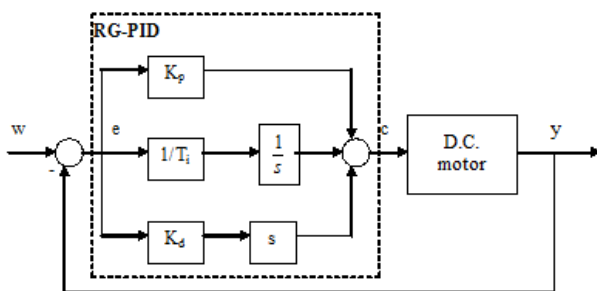


Figure 2. PID command unit for the D.C. motor system

The structure in Figure 2 is implemented with operational amplifiers and potentiometers in order to change the three parameters of the PID controller for the proper tuning of the controller.

3. COMPUTER MONITORED PID CONTROLLER. DEVELOPMENT BOARD WITH PIC CONTROLLER AND USB INTERFACE

From the point of view of building the monitored PID controller on the computer, it is necessary to make a communication between the controller and the personal computer. In recent decades it has become increasingly popular to use the USB connection in electronic devices. The role of this type of connection can be, for example, to change the program to a newer version via the USB connection or by monitoring the information received from the electronic device or by interacting between the personal computer and the electronic device. There exists nowadays integrated circuits dedicated to this purpose, but in our case for the PID controller a microcontroller from the PIC family was used - PIC18F2550. The advantage of using this microcontroller is that the PIC microcontroller also exists in the DIP encapsulated form, and in addition to communication via the USB interface it is possible to perform the calculations required to control a process. Another advantage of this integrated circuit is that it requires little additional knowledge if the programmer already knows how to use PIC family microcontrollers.

About the USB standard it is well known that there is a main device (Host) and a secondary device (Slave). The device titled "Host" has a superior order. In other words, in the main device we can connect the secondary devices. In the proposed application the personal computer works as a "Host" main device, and the PIC18F2550 is only used as a secondary device. For this reason, no components with USB connection are connected to circuits built with this microcontroller.

Related to the operation of the USB protocol, the devices containing this connection fall into the different classes. In the case of the PIC18F2550 microcontroller this class is defined by the program written in the integrated circuit and by the driver program installed on the computer. As a result, several classes of classes can be implement in an application with PIC18. Microchip Technology has already created for users all "firmware" programs to help the operator build a device with these classes. These class programs (firmware) are as follows: CDC - Communication Device Class – emulation of serial port; HID – Human Interface Device Class; MSC – Mass Storage Class; Custom Device Class; Bootloader.

3.1 Designing the development PID board and the graphical interface

In order to design the PID controller board, some requirements are to be followed:

- To connect with the computer using HID protocol
- To contain all the electronic devices for d.c. motor operating
- Force-circuit to be optically isolated from the command circuit
- To contain all the electronic components for tahogenerator signal processing
- To ensure the possibility of re-programing the device using only the USB interface – without an external ICSP programmer
- To be possible to connect different devices to PIC controller, for the circuit to perform as a development board.

The picture of the actual PID development board with USB interface is shown in Figure 3.

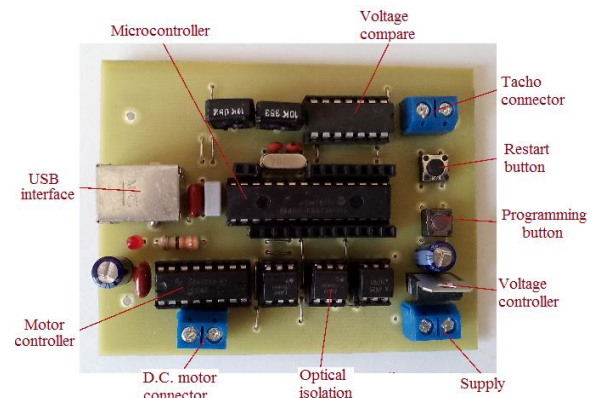


Figure 3. Development board with PIC18F2550 and USB interface

The central part of the board is the PIC18F2550 microcontroller. To control the D.C. motor the L293D circuit was chosen. The input of the power circuit is optically isolated from the microcontroller output via the circuit 4N25 and the LM324N comparator to delimit the voltage provided by the tacho was used. This circuit also detects the motor direction and inverts the tacho signal if the motor rotates in the reverse direction.

The next step is uploading the codes from the computer into the microcontroller via USB interface and programming the PID controller as a numerical algorithm.

A graphical interface was design in order to monitor the input and output signals in the system, using C# programming language. The interface is able to display the state of the reference set-point variable, the PWM duty cycle and the process variable status. This graphical interface allows also changing the Kp, Ki, and Kd parameters. As a result, the graphical behavior of the system with PID controller will be shown at parameters changes.

4. NUMERICAL PID-CONTROL ALGORITHM DESIGN VIA MATLAB SIMULATION

For the beginning, a continuous PI controller was tuned to meet the performances imposed to the D.C. motor system, modeled in MATLAB-SIMULINK with the exact parameters of the real one.

The closed-loop control structure for the D.C. motor drive with the PI-controller was implemented in SIMULINK as in Figure 4.

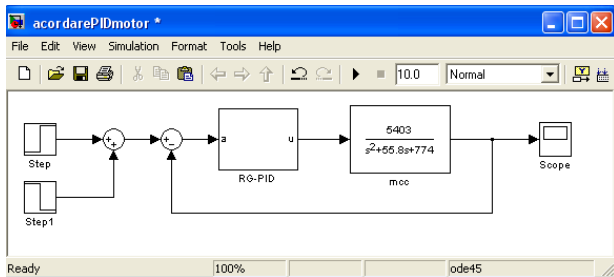


Figure 4. SIMULINK closed-loop control structure

The PID-algorithm is given by the time-expression (3) and the transfer function in (4).

$$c(t) = k_p \cdot e(t) + k_i \cdot \int e(t)dt + k_d \cdot e'(t) \quad (3)$$

$$H_c(s) = k_p + \frac{k_i}{s} + k_d \cdot s \quad (4)$$

where the tuned parameters are: $k_p = 0.039$, $k_i = 0.026$ and $k_d = 0$.

The derivative component was not significantly influencing the system performance and as a result, the value for it was chosen to be zero and the PID controller becomes a PI one.

In a second step the PI controller is sampled through Tustin method [8] with the sampling period $h = 0.01$ sec. The discrete transfer function obtained after this step, as in relation (5), is implemented by the control algorithm in (6).

$$H_c(z) = \frac{(k_p \cdot h + 2 \cdot k_i) \cdot z + k_i \cdot h - 2 \cdot k_i}{h \cdot (z - 1)} \quad (5)$$

$$c(t) = \frac{k_p h + 2k_i}{h} e(t) + \frac{k_p h - 2k_i}{h} e(t-1) - c(t-1) \quad (6)$$

In the SIMULINK structure in Figure 4, the RG-PID controller block was replaced by the numerical algorithm implemented in the SIMULINK scheme in Figure 5 and the sampling 'Zero-Order Hold' blocks was placed for the process.

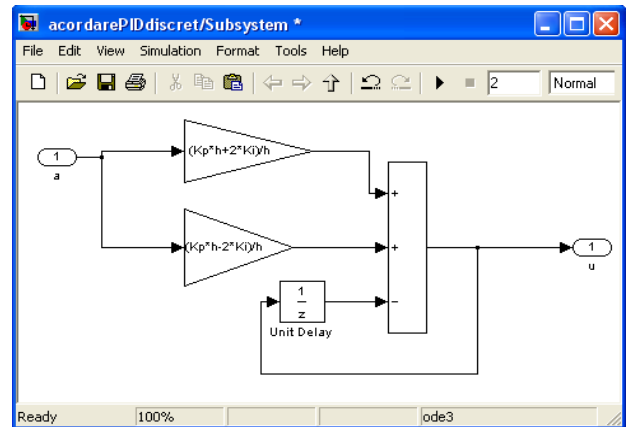


Figure 5. SIMULINK numeric PI controller structure

Both structures were implemented as a base for future comparisons with the microcontroller board performances.

In a third step, the numerical algorithm was converted in C language and uploaded via USB interface in the microcontroller on the development board, directly from the PC.

5. EXPERIMENTAL RESULTS

We have now the results from both closed-loop systems with numerical algorithm implemented in MATLAB-SIMULINK and through the development board with the PIC 18F2550 microcontroller. Some small changes in the values of the PID parameters were necessary: $k_p = 0.030$, $k_i = 0.015$ and $k_d = 0$.

The system with the D.C. motor was closed with the controller on the board and the results were registered

using the graphical interface. Both results are shown in Figure 6a) and 6b), respectively.

The reference signal used for simulation was a multiple step, as in equation (7):

$$w(t) = \begin{cases} 4, & 0 \leq t \leq 2 \\ 0, & t \geq 2 \end{cases} \quad (7)$$

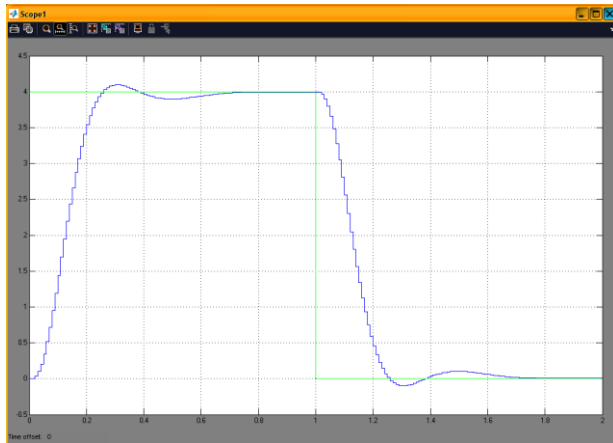


Figure 6a. SIMULINK numeric PI controller structure response at reference signal (7)

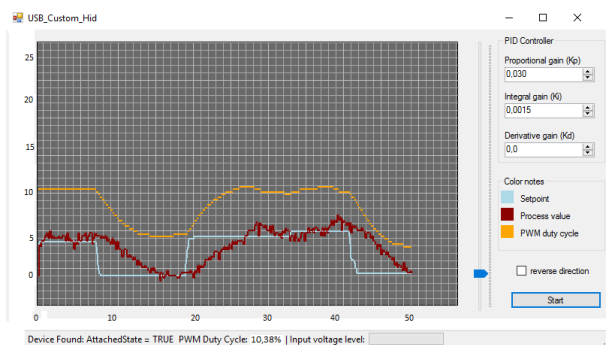


Figure 6b. Development board numeric PI controller structure response at reference signal (7)

As it can be seen, the results are comparable, having in mind the differences between a simulation process and the real-time experience. Both signals follow the reference with a small overshoot and an acceptable delay in response.

6. CONCLUSIONS

In the paper were presented some functionality extensions for a PID speed control system with D.C. motor.

The main aim was to enhance the usage of the didactic unit at numerical solutions for control algorithms and to upload them via an easy interface, without specific programmer devices.

A development board was proposed, built around the PIC18F2550 controller in order to ensure an easy access directly from the computer, via USB interface. A

graphical interface was also designed for a real-time results following and the possibility to see the changes in system performances at controller parameters changes.

A numerical algorithm was developed in MATLAB-SIMULINK based on the didactic unit analogic PID structure and the resulted algorithm was further implemented in the microcontroller. A comparative study was made on the two approaches (SIMULINK and real-time experiment) in order to show the utility of the development board.

In the end the students have the possibility to design and implement all kind of numerical algorithms for the D.C. motor system, using MATLAB-SIMULINK for design and validation procedure and with the possibility to implement the solution in both analogic and numerical alternative and to compare them in real-time application.

7. REFERENCES

- [1] C.R. Costea, S. Dale, D. Zmaranda, H. Silaghi, MATLAB-SIMULINK Modeling for the Well Station of a Geothermal Power Plant. In: Proceedings of 2017 14th International Conference on Engineering of Modern Electric Systems, ICEMES 2017, Oradea, 2017.
- [2] D. Trip, S. Dale, V. Popescu, Digital Control for Switched Mode DC-DC Buck Converters. In: Conference Proceedings of the 9th International Symposium on Electronics and Telecommunications ISET'2010, Timișoara, 2010.
- [3] Z. Nagy, A. Bara, S. Dale, Control algorithm of a BLDC Motor. In: Proceedings of the 20th International DAAAM Symposium Intelligent Manufacturing & Automation, Viena, 2009.
- [4] M.V. Garcia, M.F. Santos, C.H.S. de Vasconcelos, V.F. Vidal, M.F. Silva, Development of an electromechanical positioner with PID controller embedded in a PIC18F4550 microcontroller: A practical validation case study. In: Proceedings of the 19th International Carpathian Control Conference, Miskolc, 2018.
- [5] S.E. Oltean, M. Dulău, A.V. Duka, Didactic System For The Experimental Study Of Digital Pid Control Structures, Scientific Bulletin of the Petru Maior University of Tîrgu Mureș 2016; 13(2): 22-28.
- [6] M. Gavran, M. Fruk, G. Vujisić, PI controller for DC motor speed realized with Arduino and Simulink. In: Proceedings of 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, 2017.
- [7] D.C. Motor unit – User's Guide, Driver CKT26-T5-3VC.
- [8] K.J. Åström, B. Wittenmark. Computer controlled systems, Englewood Cliffs, NJ, 3rd edition: Prentice Hall, 1997.