ANNALS OF THE UNIVERSITY OF CRAIOVA
Series: Automation, Computers, Electronics and Mechatronics, Vol. 14 (41), No. 1, 2017

DC Motor Speed Control Using a Discrete PID Control Law

Teodor-Constantin C. Nichițelea*, Maria-Geanina Gh. Unguritu**

**Automation and Electronics Department, University of Craiova*
Craiova, Romania (e-mail: teodornichitelea@outlook.com).
*** Automation and Electronics Department, University of Craiova*
Craiova, Romania (e-mail: geany_unguritu@yahoo.com).

Abstract: This paper presents the speed control of the DC motors of an intelligent racing car by using a discrete-time Proportional-Integral-Derivative controller and tuning methods. Using wireless communication, data is sent from the intelligent car to a MATLAB interface which acquires the received data. The DC motors are therefore modeled using experimental identification.

Keywords: Discrete-time controller, DC motor control, experimental system identification, PID controller

1. INTRODUCTION

Building an intelligent car is one of the contemporary engineering topics, a topic that is supposed to come with solutions for autonomous driving of the car. All the sensors and actuators work together in a complex system to achieve this target. Therefore, many international contests have appeared in which a robot needs to follow a trajectory without human intervention.

One of these contests is the "NXP Cup - Intelligent Car Racing", in which different teams need to come up with different designs and developments, trying to discover the best hardware-software solution possible to attain the best speed (time) for completing an unspecified track, knowing only what kind of parts the track can consist of.

To be able to attain the lowest time for the race, some kind of control algorithm needs to be implemented and the feedbacks need to be used to obtain a closed-loop system. One of these feedbacks is the rotation speed of the motors based on which the motors can be controlled to achieve the desired speed within a time period.

Between the many existing control algorithms, such as adaptive control systems, fuzzy control logic systems and many others, the controller used in this paper is the well-known and well-used PID control law, with its discrete form, because of its simple implementation and good performance.

The article is structured as follows: Section 2 briefly presents a review of previous DC motor control studies. Section 3 provides an overview of the intelligent car used for the motor control studies. Section 4 presents the PID control law and the methods used for the controller tuning. In Section 5, the experimental results are described. Section 6 ends the paper with conclusions from the studies.

2. RELATED WORK

Several publications (Rambabu, 2007; Ivanov et al., 2015; Serrezuela et al., 2017) describe the mathematical modeling and control algorithms of the Brushless DC motors (BDCM), as those that were used for this application.

Regarding the command, hysteresis or pulse width-modulated (PWM) current controllers are typically used in order to maintain the currents flowing into the motor as close as possible to the rectangular reference values. Natural or phase variable models offer many advantages due to the trapezoidal back EMF, which avoids the transformation of the machine equations when sinusoidal variation of the motor inductances with rotor angle occurs (Ivanov et al., 2015).

S. Rambabu (2007) studied the speed control of a BDCM. He proposed a performance comparison between a fuzzy logic controller and a PI controller on the speed response of the motor.

Some authors (S. Satish Kumar, et al., 2013) developed a complete model of the BDCM and designed a controller using a PID control law adjusted first with Ziegler-Nichols method and afterwards with Genetic Algorithm in order to compare the performance.

R. Kandiban and R. Arulmozhiyal (2012) proposed an adaptive fuzzy PID controller to control the speed of the BDCM. The simulation results show that this non-conventional controller had better performance, especially when the motor was working at lower and higher speeds.

Many studies have been done on DC motor control. For the control of the speed of the intelligent car a discrete-time PID control law was used. While the intelligent car is accelerating the controller is actually a PI controller. The derivative component is taken into account only when the driving algorithm decides it is time to decelerate. This has been done to ensure a better stability while accelerating and a better braking system while decelerating.

3. OVERVIEW

The intelligent car was built using the following main hardware components, which can be identified in Fig. 1:

- A Standard High-Torque BB Servomotor Futaba S3010 that controls the direction of the car.
- Freescale Linescan Camera based on the sensor TSL1401CL from TAOS Inc. which is used to get information about the track. The data is essential for controlling the servomotor and the DC motors speed.
- Unipolar digital position Hall-effect sensors, Honeywell SS449a, used for the calculation of the rotation speed for each motor. The data from these sensors is indispensable for the speed control of the motors.
- Standard DC motors RN260C with winding 18130.
- The microcontroller NXP MKL25Z128VLK4 is the discrete controller and it is programmed with the algorithm for controlling the whole intelligent car.
- The FRDM-TFC Freescale Motor Shield that can drive up to two DC motors (5A per channel), two servomotors and Input/Output (I/O) for the Freescale Linescan Camera and Hall effect sensors.
- Bluetooth RN52 used for obtaining and transmitting data through the MATLAB interface. Based on information received from the Bluetooth, the experimental identification of the model of the DC motors is done.
- Reflectance sensors Polulu QTR-L-1A used for the implementation of the stop detection of the intelligent car.

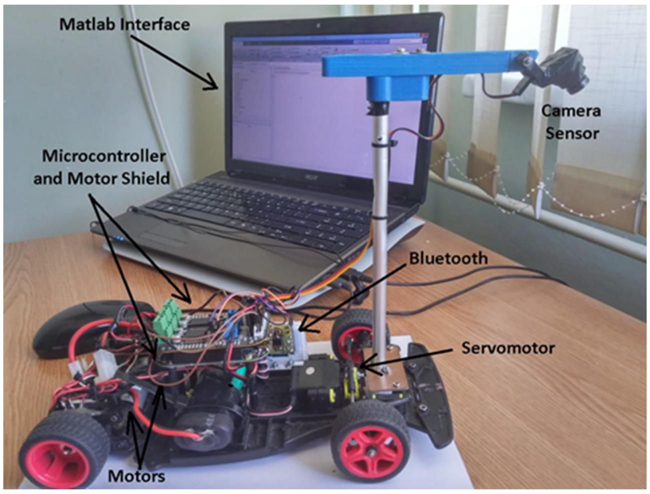

Fig.1: Intelligent Car

Figure 2 shows an example of a track the intelligent car must follow.

Fig. 2: Final track layout - Munich, Germany 2016

From the software point of view, the drivers for the microcontroller were implemented with regard to the AUTOSAR standard.

4. DC MOTOR CONTROL

4.1. Motor modelling

The first step for being able to control the DC motors of the intelligent car is to obtain the dynamical model of the motors. The motor modeling has been done by using experimental identification, via wireless technology, which in this case it is represented by the Bluetooth mounted on the car.

The software sends to the interface through Bluetooth communication the data regarding the rotation speed of each motor calculated in $\frac{m}{sec}$ using the encoder driver of the microcontroller every 4 msec. This rotation speed is the output of the motors to a step input. The input in this case is a constant value of the PWM applied from the microcontroller onto the H-Bridges.

At first a constant PWM of 26% duty cycle was applied, which is equal to about 2 Volts (7.8 Volts from Battery x 26% PWM = about 2 Volts) and the motor speed achieved steady state soon after. Immediately after, a constant PWM of 39% was applied, which is equal to about 3 Volts (7.8 Volts from Battery x 39% PWM = about 3 Volts) and the motor speed achieved steady state soon after. Therefore the difference between inputs is a step input, $\Delta u = 1$Volt.

Using the received data, the model of the DC motors has been approximated with a transfer function of first order:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot e^{-\tau \cdot s}}{T \cdot s + 1} \qquad (1)$$

where: K – gain factor, τ – dead-time constant, T – time constant.

The data received from the Bluetooth is shown in Fig. 3.

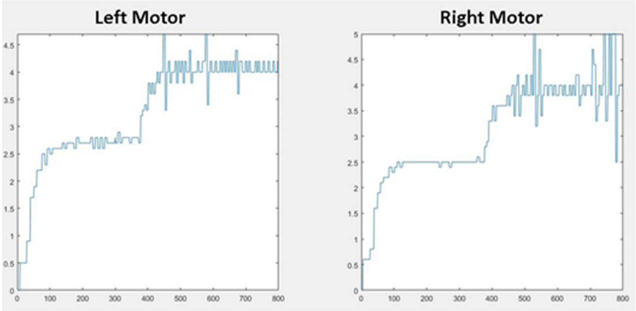

Fig. 3: Motor speed data acquired through Bluetooth

Since it is a fast process, no delay was considered:

$\tau = 0$

The following gain factors have been identified:

- For the left motor:

$\boldsymbol{y}(\infty) - \boldsymbol{y}_{st}(\boldsymbol{t_0}) = 4.1 - 2.75 = 1.35 \rightarrow K_{left} = 1.35 \, \frac{m}{sec \cdot V}$.

- For the right motor:

$\boldsymbol{y}(\infty) - \boldsymbol{y}_{st}(\boldsymbol{t_0}) = 3.9 - 2.5 = 1.40 \rightarrow K_{right} = 1.40 \, \frac{m}{sec \cdot V}$.

Using the tangent method, the following time constants have been identified:

- For the left motor:

$T_{left} = 435 - 375 = 60 \text{ msec} \cdot 4 = 240 \text{ msec}$.

- For the right motor:

$T_{right} = 445 - 375 = 70 \text{ msec} \cdot 4 = 280 \text{ msec}$.

Introducing the identified parameters in the structure of the transfer function of first order, the approximate models of the DC motors have been obtained:

- For the left motor:

$$H_{Left\ Motor}(s) = \frac{1.35}{0.24 \cdot s + 1} \tag{2}$$

- For the right motor:

$$H_{Right\ motor}(s) = \frac{1.40}{0.28 \cdot s + 1} \tag{3}$$

4.2. Discrete form of ideal PID controller

For the motor control, the discrete form of the ideal PID, which can be implemented on numerical systems, has been used.

An ideal PID element is represented by the transfer function:

$$\mathrm{H(s)} = \mathrm{K_P} \cdot \left[1 + \frac{1}{\mathrm{T_i} \cdot \mathrm{s}} + \mathrm{T_d} \cdot \mathrm{s}\right] \tag{4}$$

The discrete form of the PID control law is obtained by summing up the discrete forms of the three components, where T_e is the sampling time:

$$\begin{cases} u_{pid}(k) = u_p(k) + u_i(k) + u_d(k) \\ u_p(k) = K_P \cdot e(k) \\ u_i(k) = u_i(k-1) + T_e \cdot K_i \cdot e(k-1) \; with \; K_i = \dfrac{K_P}{T_i} \\ u_d(k) = \dfrac{K_d}{T_e} \cdot [e(k) - e(k-1)] with \; K_d = K_P \cdot T_d \end{cases} \tag{5}$$

4.3 Analytical calculation of PID controller parameters

Considering the transfer functions identified using the received data from the Bluetooth, the PID parameters have been calculated only for one transfer function.

The PID parameters have been determined by imposing the steady state error to be zero $\varepsilon_{0\infty}= 0$. Since the whole control algorithm for the intelligent car needs 4 msec to be processed, the time constant of the first order system in closed-loop has been chosen to have the value T=0.004 sec. In this way, the steady state will be achieved in one sampling period.

The system in closed-loop has been considered as a first order system described by the following transfer function:

$$H_O(s) = \frac{K}{T \cdot s + 1} \tag{6}$$

By imposing the steady-state error to be 0:

$$\varepsilon_{0\infty} = 0 \rightarrow H_O(0) = 1 \rightarrow K = 1 \tag{7}$$

Therefore, the closed-loop transfer function was obtained:

$$H_O(s) = \frac{1}{0.004 \cdot s + 1} \tag{8}$$

For calculating the transfer function of the control algorithm, relation (9) was used and the calculations give relation (10):

$$H_C(s) = \frac{1}{H_F(s)} \cdot \frac{H_O(s)}{1 - H_O(s)} \tag{9}$$

$$H_C(s) = 44.44 + \frac{1}{0.0054 \cdot s} \tag{10}$$

$$H_C(s) = 44.44 \cdot \left(1 + \frac{1}{0.24 \cdot s}\right) = K_P \cdot \left[1 + \frac{1}{T_i \cdot s}\right]$$

Therefore, the parameters of a PI controller have been determined:

$\mathrm{K_P} = 44.44$ and $\mathrm{T_i} = 0.24 \text{ sec}$

The simulation results with these parameters are shown in Fig. 4.

The PI controller is sufficient for the speed control of the DC motors used for the intelligent car but a complete PID formula was used, by introducing the derivative component to help the braking system of the car. In this way, the control algorithm is actually adaptive since it is a

PI when accelerating, and a PID when decelerating. The derivative component was experimentally determined, by trying to not affect the stability of the system but at the same time to see the presence of the component when braking, resulting a derivative component equal to the value $K_d = 0.05$.

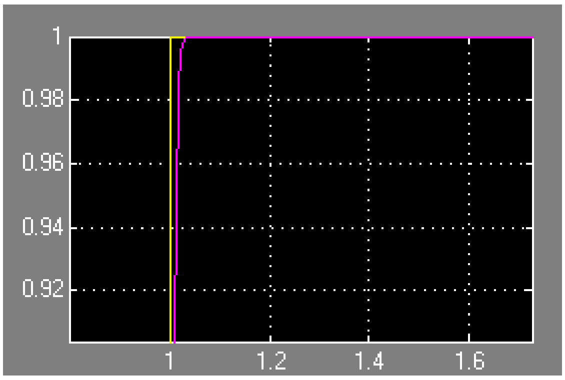

Fig. 4: Simulation results for analytical method

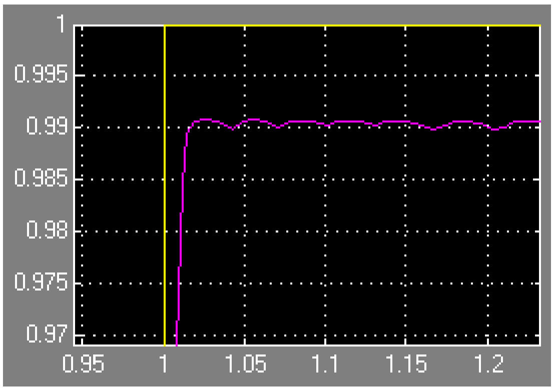

Fig. 5: Simulation to determine K_{Rlim} and T_{lim} for Nichols method

4.4 Parameters tuning using Nichols method

Nichols method can be used for experimental tuning of a P, PI or PID controller with the help of K_{Plim} and T_{lim} parameter.

For this, $T_i = \infty, T_d = 0$ and value of gain factor K_P is increased until maintained oscillations appear. This value of K_P is the value of K_{Plim} and the oscillations period is T_{lim}.

Using this method, experiments on the transfer function $H_{Left\ Motor}(s)$ have been performed. Taking into consideration that $H_{Left\ Motor}(s)$ is a first order transfer function, the maintained oscillations will not be obtained.

From the simulation shown in Fig. 5, the next values of the parameters are obtained:

$$K_{Plim} = 80$$

$$T_{lim} = 1.1011 - 1.0719 = 0.0292\ sec.$$

The PI controller parameters obtained using Nichols method:

$$K_P = 0.45 \cdot K_{Plim} = 36$$

$$T_i = 0.85 \cdot T_{lim} = 0.02482\ sec$$

Even though K_P is rather close as value to the one found using the analytical method, the T_i determined is 10 times smaller.

Simulation was done with the Simulink block diagram shown in Fig. 6 and the simulation results for the calculated parameters are shown in Fig. 7.

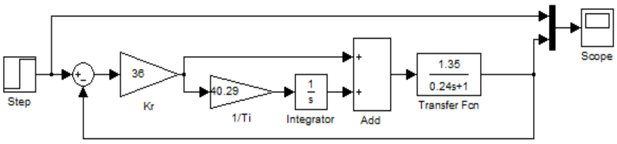

Fig. 6: Simulink simulation with Nichols parameters

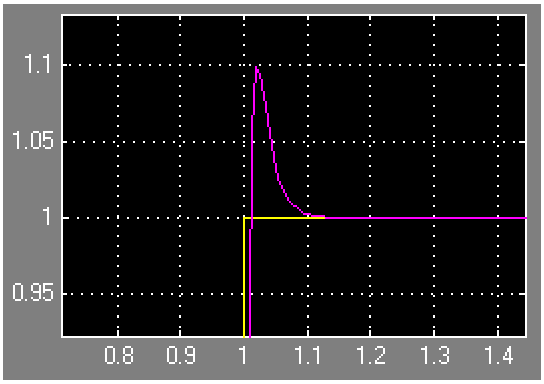

Fig. 7: Simulation results for Nichols method

4.5 Parameters tuning using experimental method

In order to determine the PI parameters for speed control of the intelligent car many tests were performed. These parameters were adjusted by monitoring the motors behavior to reach the reference speed.

For simplicity in experimental tuning, the following transfer function of the controller was used:

$$H(s) = K_P + K_i \cdot \frac{1}{s}, \text{ where } K_i = \frac{K_P}{T_i}$$

Values obtained using this trial and error method:

$$K_P = 27$$

$$K_i = 27 \rightarrow T_i = 1 \text{ sec}$$

A simulation of the system response considering the motor model obtained with the analytical method and using the PI parameters experimentally tuned was done in Simulink and is shown in Fig. 8. The simulation results using these parameters are shown in Fig. 9.

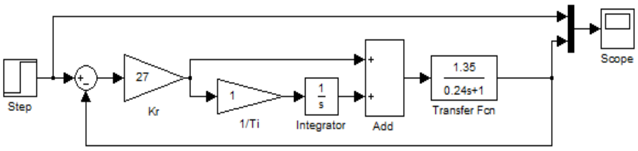

Fig. 8: Simulink simulation with experimental parameters

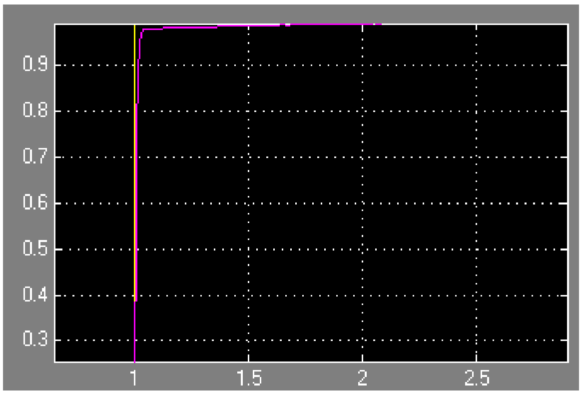

Fig. 9: Simulation results for experimental method

5. EXPERIMENTAL RESULTS

The parameters calculated and determined for the PID control law were tested by introducing them in the intelligent racing car's control algorithm, which is described by the block diagram in Fig. 10.

As a result of the tests, the expected simulations outputs performed in Simulink were attained.

The next step for the intelligent racing car was the participation at the contest where it won 2nd place in Europe in 2016 with a time of 16.4 seconds on the track presented in Fig. 2, with a speed reference of 3.4 $\frac{m}{sec}$. The track had about 40 meters but considering the layout and the control algorithm that takes into account the information from the camera sensor also, the average speed of 2.44 $\frac{m}{sec}$ is normal. In 2017, it won 2nd place in Europe with a time of 21.2 seconds on a track which had about 70 meters and with a reference speed of 3.8$\frac{m}{sec}$.

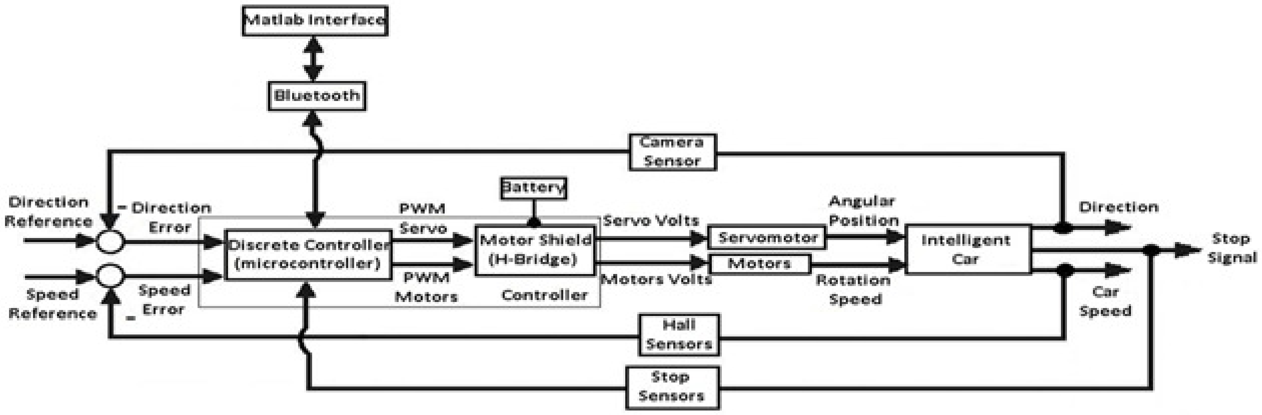

Fig. 10: Block diagram of the Intelligent Car's control algorithm

6. CONCLUSIONS

This paper presents the design of a speed controller for an intelligent car, using a discrete PID controller, always a good choice in finding a control solution. The control can be performed using the mathematical model of the system (if this can be obtained) or using tuning and experimental methods on the physical process.

The objects under control, the motors of the car, can be modeled if the response of these can be analyzed. Data acquisition of this process, in this case, is done using the Bluetooth communication between the car and a MATLAB interface.

The design of the PI controller can be done using one of the methods presented above.

The simulation results show the influence of every control design method in system's behavior. According to these, the smallest settling time is obtained with the analytical method using the motor model. On the other side, the other two methods can be used even if the motor model is unknown. The Nichols tuning method has the disadvantage of the overshoot of approximately 10%, compared to the other two methods. The experimental method can provide good results, but is more time consuming and needs a good knowledge of how the process should behave when adjusting the parameters.

REFERENCES

Ivanov, S., Ivanov, V., Selișteanu, D., Șendrescu, D. (2015). Compact Dynamic Model of the Brushless DC motor. *Annals of the University of Craiova*, Vol. 12 (39), No.1, pp. 13-16.

Kandiban, R,. Arulmozhiyal, R. (2012). Speed Control of BLDC motor using Adaptive Fuzzy PID controller, *International Conference on Modeling, Optimization and Computing*, Procedia Engineering 38, pp. 306-313

Rambabu, S. (2007). *Modeling and control of a brushless*

DC motor, PhD Thesis Satish, Kumar, S., Dr. Meenakumari, R., Ravikumar, P., Krishnaveni, V., Kiruthika, K. (2013). Design control and modeling of three phase BLDC motor applications using PID, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, Issue 8, pp. 3932-3941.

Serrezuela, R.R., Cardoso, M.A.T., Chavarro, A.F.C. (2017). *Design and implementation of a PID fuzzy control for the speed of a DC motor*, ARPN Journal of Engineering and Applied Sciences 12(8), pp. 2655-2660.

NXP Cup 2016 – EMEA Finals https://community.nxp.com/docs/DOC-330339

NXP Cup 2017 – EMEA Finals https://community.nxp.com/docs/DOC-334133