ANNALS of the ORADEA UNIVERSITY.
Fascicle of Management and Technological Engineering, Volume VII (XVII), 2008

THE DESIGN OF A DC MOTOR SPEED CONTROLLER

Marius Sebastian RUSU[1], Lucian GRAMA[1]
[1]"Petru Maior" University of Tîrgu Mureş,
Technological and Managerial Engineering Department
e-mail: sebyrusu@yahoo.com, lgrama@upm.ro

Keywords: DC motor, speed controller, reference model, closed loop

Abstract: Modern manufacturing systems are automated machines that perform the required tasks. The electric motors are perhaps the most widely used energy converters in the modern machine-tools and robots. These motors require automatic control of their main parameters (position, speed, acceleration, currents). This paper presents a simple design method for a DC motor speed controller starting from a required reference model behavior.

1. INTRODUCTION

The modern machines, including the positioning systems, the robots, the flexible manufacturing systems and the application specific machine-tools, require a form of energy conversion toward mechanical energy. There are well known types of energy converters such as electric motors, pneumatic motors, hydraulic motors etc. Perhaps the most widely used motors are the electric ones because of their high flexibility and reliability as well as of their cost.

The common electric motors can be grouped in four major classes: DC motors, stepper motors, asynchronous motors and synchronous motors. The mechatronic systems, robots and low to medium power machine-tools often use DC motors to drive their work loads. These motors have rather simple functional and constructive models. There are several well known methods to control DC motors such as: PI, PID or bipositional. These can quite easily be implemented using analog electronics. However, modern digital computers provide an easy way of implementing very complex control algorithms.

This paper intends to present a simple technique for designing a DC motor speed controller based on system theory concepts. Even if the controller model may seem complex and its analog electronic implementation may actually be just that, its implementation using a digital computer is not difficult. However, this paper only presents the continuous time design.

With very little modifications, this system can be adapted to control the position or the acceleration of the motor. For more complex systems, additional current controllers can be used.

2. DC MOTOR MODEL

The design method uses the concepts of the system theory, such as signals and systems, transfer functions, direct and inverse Laplace transforms. This requires building the appropriate Laplace model for each component of the whole control system.

In order to build the DC motor's transfer function, its simplified mathematical model has been used. This model consists in differential equations of the electrical part, mechanical part and the interconnection between them.

Starting from the simplified equivalent electromechanical diagram of the motor, presented in fig. 1, the differential mathematical model has been presented in (1).

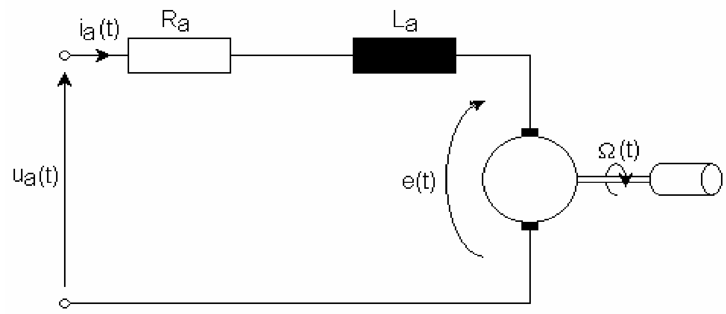

Fig. 1 – DC motor model

$$
\begin{aligned}
& u_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + e(t) \\
& e(t) = K_e \cdot \Omega(t) \\
& C_m(t) = K_m \cdot i_a(t) \\
& C_m(t) = J \cdot \frac{d\Omega(t)}{dt} + B \cdot \Omega(t)
\end{aligned}
\tag{1}
$$

where,

R_a – rotor circuit resistance (Ohm);
L_a – rotor circuit inductance (H);
u_a – input voltage (V);
e – electromotive voltage (V);
i_a – rotor circuit current (A);
Ω – rotor speed (rad/s);
C_m – motor torque (Nm);
J – rotor moment of inertia (kgm^2);
B – damping ratio (Nms);
K_e – electrical constant;
K_m – mechanical constant;

Rearranging the equations leads to the system in (2).

$$
\begin{aligned}
& u_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + K_e \cdot \Omega(t) \\
& K_m \cdot i_a(t) = J \cdot \frac{d\Omega(t)}{dt} + B \cdot \Omega(t)
\end{aligned}
\tag{2}
$$

After applying the Laplace transform to the equations in (2) the model becomes:

$$
\begin{aligned}
& U_a(s) = R_a \cdot I_a(s) + L_a \cdot I_a(s) \cdot s + K_e \cdot \Omega(s) \\
& K_m \cdot I_a(s) = J \cdot \Omega(s) \cdot s + B \cdot \Omega(s)
\end{aligned}
\tag{3}
$$

The next step is replacing the current from the second equation into the first one and rearranging the terms.

$$U_a(s) = \Omega(s) \cdot \frac{1}{K_m} \cdot \left[L_a \cdot J \cdot s^2 + \left(R_a \cdot J + L_a \cdot B \right) \cdot s + \left(R_a \cdot B + K_e \cdot K_m \right) \right] \tag{4}$$

Equation (4) is the basis for writing the motor's transfer function in (5).

$$H_m(s) = \frac{\Omega(s)}{U_a(s)} = \frac{K_m}{L_a \cdot J \cdot s^2 + \left(R_a \cdot J + L_a \cdot B \right) \cdot s + \left(R_a \cdot B + K_e \cdot K_m \right)} \tag{5}$$

This transfer function is build to allow the control of speed by the voltage input. This function also allows the simulation of motor behavior to various inputs.

Please note that the link between position, speed and acceleration transfer functions of the motor is given in (6).

$$H_m(s) = \frac{\Omega(s)}{U_a(s)} = \frac{s \cdot \theta(s)}{U_a(s)} = \frac{\frac{1}{s} \cdot \alpha(s)}{U_a(s)} \tag{6}$$

For the chosen motor the step response is shown in fig. 2.

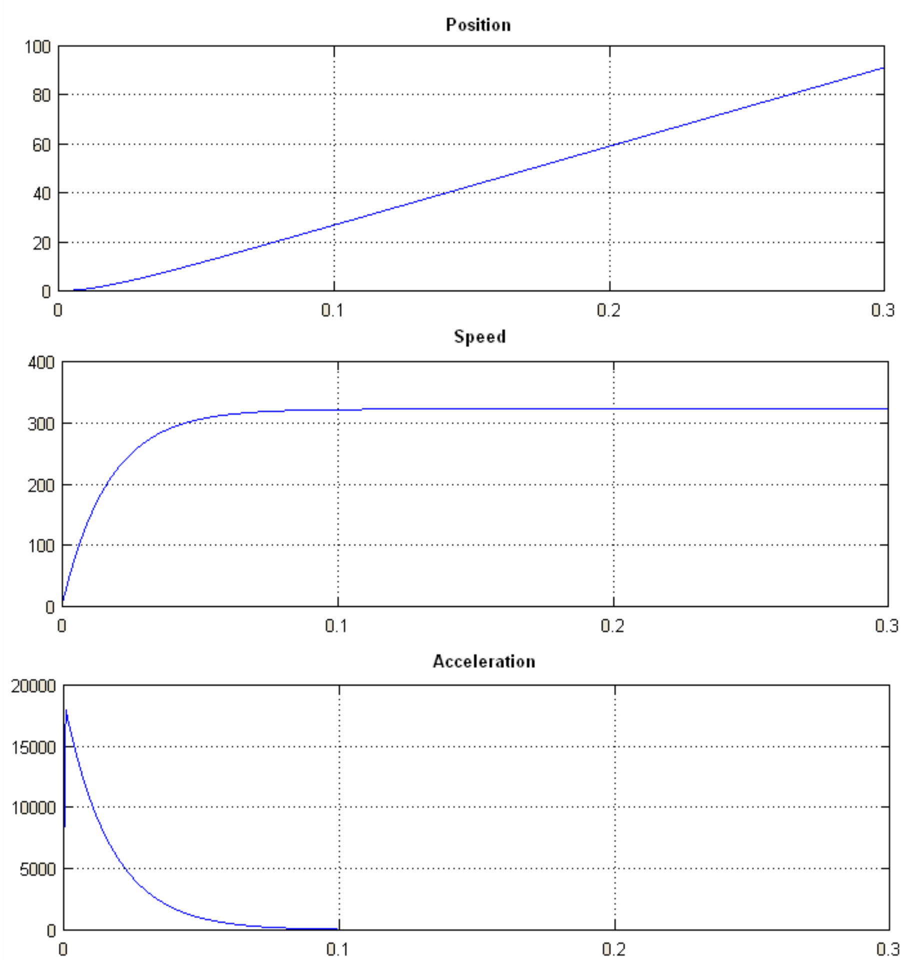

Fig. 2 – DC motor step response (position, speed and acceleration)

3. SPEED CONTROLLER DESIGN

The main automatic system loop is shown in fig. 3.

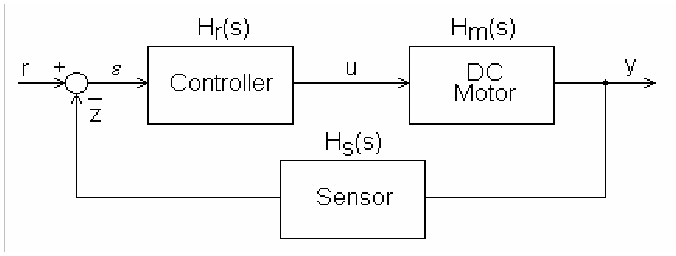

Fig. 3 – Closed loop DC motor speed control system

In fig. 3, the symbols have the following signification:

$H_r(s)$ – controller transfer function;
$H_m(s)$ – DC motor transfer function;
$H_s(s)$ – Sensor transfer function;
r – reference signal;
u – command signal;
y – output signal (speed);
z – sensor output signal;
ε – error signal ($\varepsilon = r - z$);

The sensor transfer function can be considered an integrator $H_s(s) = \frac{1}{s}$.

The next step implies building a reference model for the control system.

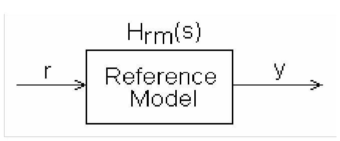

Fig. 4 – Reference model

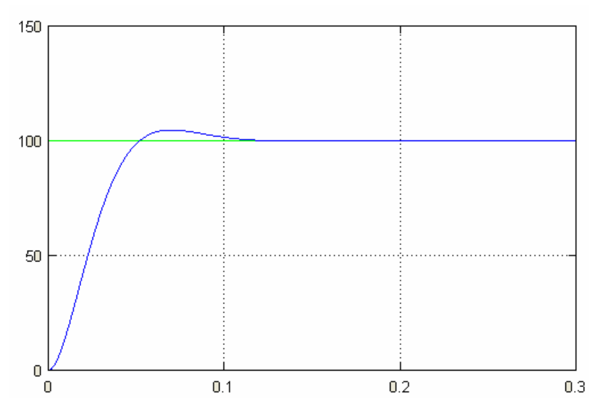

Fig. 5 – Reference model step response

A second order system has been chosen with the following transfer function:

$$H_{rm}(s) = \frac{K \cdot \omega_n^2}{s^2 + 2 \cdot \xi \cdot \omega_n \cdot s + \omega_n^2} \tag{7}$$

where, K – system gain; ω_n – natural angular velocity; ξ – damping ratio;

This reference model is chosen so that the performance requirements for the process are meet. The simulation of the reference model step response is shown in fig. 5.

After choosing the reference model, the closed loop model of the system should be built. To this result, transfer functions algebra elements must be used. The following formula will be obtained:

$$H_0(s) = \frac{H_r(s) \cdot H_m(s)}{1 + H_r(s) \cdot H_m(s) \cdot H_s(s)} = H_{rm}(s) \tag{8}$$

As this result must be equal to the reference model transfer function, the controller's transfer function can be determined using (9).

$$H_r(s) = \frac{H_{rm}(s)}{H_m(s) - H_{rm}(s) \cdot H_m(s) \cdot H_s(s)} \tag{9}$$

Replacing and rearranging the terms leads to the following controller algorithm:

$$H_r(s) = \frac{K \cdot \omega_n^2 \cdot s \cdot \left[L_a \cdot J \cdot s^2 + \left(R_a \cdot J + L_a \cdot B\right) \cdot s + \left(R_a \cdot B + K_e \cdot K_m\right)\right]}{K_m \cdot \left(s^3 + 2 \cdot \xi \cdot \omega_n \cdot s^2 + \omega_n^2 \cdot s - K \cdot \omega_n^2\right)} \tag{10}$$

This is the controller structure that assures the system behaves as required by the second order reference model.

4. CONTROL SYSTEM SIMULATION

After the design phase has been successfully finished, supplementary simulations are required to demonstrate that the actual system's behavior meets the requirements.

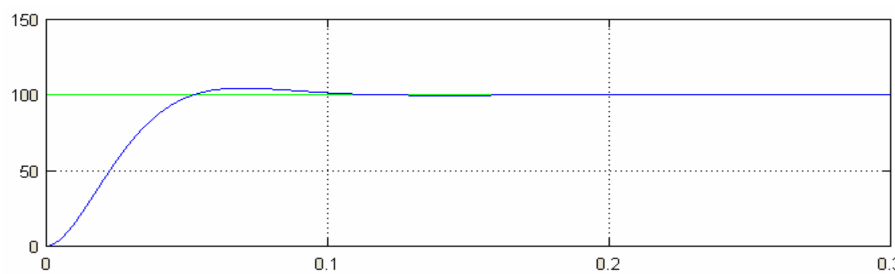

Fig. 6 – Control system output (DC motor speed)

The controller has a compensation command output shaped as in fig. 7.

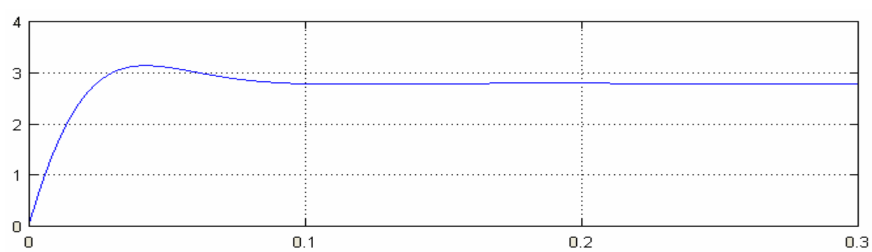

Fig. 7 – Speed controller output (DC motor input command)

Also, the difference between the desired speed and the actual speed can be seen in fig. 8.

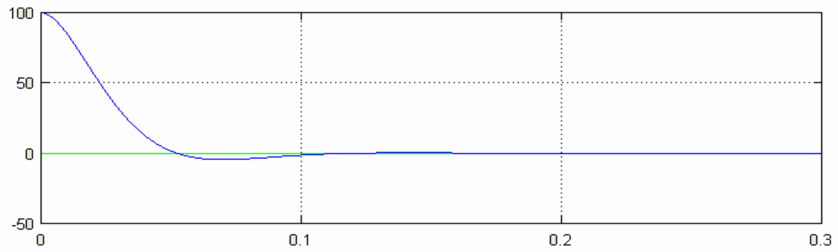

Fig. 8 – Control system error (r-y)

These simulations prove that the system meets its requirements and can be implemented in the real life. Implementations have been made using Octave, a free Matlab similar development environment.

5. CONCLUSIONS

There are many motor control systems that may be more or less appropriate to a specific type of application. The designer engineer must choose the best model for his project.

The model proposed in this paper can be implemented in a digital computer after discretization. With a few minor changes, suggested by the motor transfer function, the algorithm can be adapted for position or acceleration control.

BIBLIOGRAPHY

[1] Eaton, J, and others, “GNU Octave”, Octave software documentation
[2] Franklin, G., “Feedback control of dynamic systems”, Prentice Hall, New Jersey, 2002
[3] Vistrian, M, “Mecatronică”, Dacia, Cluj-Napoca, 1998
[4] Zărnescu, H., “Elemente de reglare automată. Ingineria reglării automate. Curs”, Editura Universităţii “Petru Maior”, Tîrgu Mureş, 1998