

Learning Module on Electric Motors Modeling, Control, and Testing (LM-EMMCT)

Prof. Ana Djuric P.E., Wayne State University

Dr. Ana Djuric is an Assistant Professor of Engineering Technology in the College of Engineering at Wayne State University, Detroit, Michigan. Dr. Djuric research areas are industrial robots, kinematics, dynamics, control, and advanced manufacturing systems. She supervises multiple undergraduate and graduate students in their research and is a member of Council on Undergraduate Research (CUR). Her Dipl.-Ing. degree is in the area of mechanical engineering from the University of Belgrade, Serbia, focusing in Control Systems. Her M.A.Sc. degree is in Industrial and Manufacturing Systems Engineering from University of Windsor, Canada, area of Industrial Robotics, and a Ph.D. in Mechanical Engineering from University of Windsor, Canada in the area of Reconfigurable Robotics. Prior to her arrival at WSU, Dr. Djuric worked in the industry as a machine and tool designer first and then as a Robotics software Analyst for five years. Prior to joining WSU, Dr. Djuric was an Instructor at the Mechanical, Automotive and Materials Engineering, and Industrial and Manufacturing and Systems Engineering departments at the University of Windsor.

Dr. Vukica M. Jovanovic, Old Dominion University

Dr. Vukica Jovanovic is an Assistant Professor of Engineering Technology in Mechanical Engineering Technology Program. She holds a Ph.D. from Purdue University in Mechanical Engineering Technology, focus on Digital Manufacturing. Her research is focused on mechatronics, digital manufacturing, digital thread, cyber physical systems, broadening participation, and engineering education. She is a Co-Director of Mechatronics and Digital Manufacturing Lab at ODU and a lead of Area of Specialization Mechatronics Systems Design. She worked as a Visiting Researcher at Commonwealth Center for Advanced Manufacturing in Disputanta, VA on projects focusing on digital thread and cyber security of manufacturing systems. She has funded research in broadening participation efforts of underrepresented students in STEM funded by Office of Naval Research, focusing on mechatronic pathways. She is part of the ONR project related to the additive manufacturing training of active military. She is also part of the research team that leads the summer camp to nine graders that focus on broadening participation of underrepresented students into STEM (ODU BLAST).

Dr. Ece Yaprak, Wayne State University

Dr. Ece Yaprak is a Professor of Engineering Technology in the College of Engineering at Wayne State University. She received her Ph.D. in computer engineering from Wayne State University in 1989. She has held research fellowships at NASA research centers (John Glenn Laboratory at Case Western, Jet Propulsion Laboratory at Cal Tech, Ames Research Center at Stanford, and the Johnson Space Center at Texas A&M Universities) and the U.S. Navy (at its SPAWAR Research Center in San Diego). She has also served as a Fulbright scholar at the Nokia Wireless Communications Research Center at the University of Oulu in Finland in 2009. She is a senior member of the IEEE. Dr. Yaprak serves as an ABET IEEE/ETAC Commissioner since 2012. Currently, she serves as Program Director of Division of Undergraduate Education (DUE) at the National Science Foundation (NSF). She manages a large and diverse portfolio of awards on STEM education and workforce development.

Prof. Wen Chen, Wayne State University

Learning Module on Electric Motors Modeling, Control and Testing (LM-EMMCT)

Abstract

The objective of this paper is to develop and integrate a learning module on Electric Motors Modeling, Control and Testing (EMMCT) into the Electrical Engineering Technology (EET) and Mechanical Engineering Technology (MET) programs. Preparing future engineers to work in highly automated production requires proper education and training in mechatronics theory and applications. Although Engineering Technology programs at various universities offer various courses related to the controls, electrical motors, and automation, they are not including the same methods when it comes to the selection of appropriate electrical motor for a specific application in mechatronic system. MET student do have exposure to the electrical systems in the various courses that are offered at their lower division level courses. However, these methods have to be further emphasized and applied in the upper level courses as well. This paper will present one such application and a learning module that is focused on the Electric Motors Modeling, Control and Testing (EMMCT). This module can be integrated in various controls, mechatronics, robotics, senior design and capstone courses.

Introduction

The precise control of mechatronics systems is based on their mathematical models. Hence, an accurate modelling of mechatronic systems is highly important. Mechatronics systems are made of different mechanical and electrical parts, connected in the system where actuators (pneumatic, hydraulic, or electrical motors) play a key role in determining the performance, and the dynamic characteristics. Even though, multiple courses in Engineering Technology cover topics related to the electrical technology, it is still hard for students to understand the concept of optimal motor selection and determination of its performance over load, specifically for students in Mechanical Engineering Technology.¹ Therefore, the goal of this paper is to prepare a specific learning module on Electric Motors Modeling, Control and Testing (LM-EMMCT). Direct current (DC) motors modeling involves mathematical models of electrical and mechanical parts of the motor. It also include their integration in the MATLAB/Simulink tools to practice analysis of the basic open loop DC motor system. This analysis can be used in mechatronics motion control applications, mostly in robot arm modeling and control, mobile robotics, and electric power steering control.²⁻⁴

The education module presented in this paper can be implemented in the following seven courses which require knowledge of DC motors selection, modeling and control at the Wayne State University (WSU), Detroit, Michigan, and Old Dominion University (ODU), Norfolk, Virginia:

- Control Systems (WSU) EET4200
- Special Topics on Industrial Robots Dynamics and Control (WSU) ET 7995
- Senior Project (WSU) ET 5999
- Introduction to Mechatronics (ODU) MET 426
- Modeling and Simulation of Mechatronics Systems (ODU) MET 495
- Senior Design Project (ODU) ENGT 435

Currently, this module has been implemented in the following courses: Control Systems as a project, Special Topics on Industrial Robots Dynamics and Control as Assignment # 2, and

in Senior Project at WSU – student used this module for analysis of what kind of actuator will be used for a rotary table design. This module will be implemented in the Mechatronics related courses at ODU in the Fall 2017.

Electric Motors Modeling, Control and Testing (EMMCT) + ROTATIONAL Joint

Assignment Example: The electro-mechanical system in Figure 1 represents a Single Joint robot model with flexible link. The robot has an armature-controlled DC permanent magnet torque motor for a joint.⁵

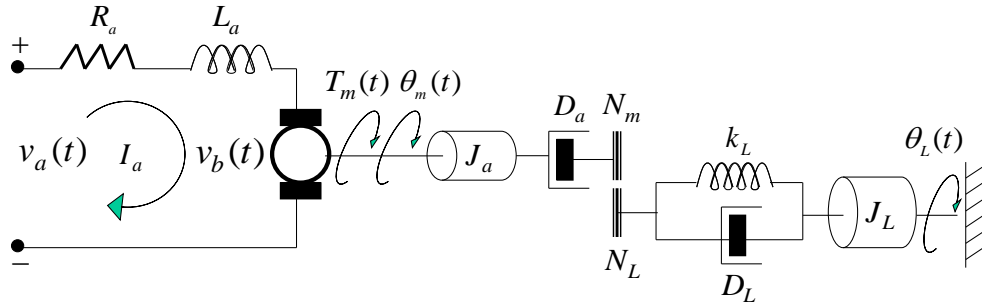


Figure 1: Single Joint Robot Model

Basic mechanical and electrical properties are given in the Table 1.

Table 1: Information for the Electro-Mechanical System

| One Degree of Freedom (1DOF) Robot Parameters | | |
|---|--|---------|
| Armature Voltage | V_a [V] | 45 |
| Armature Inertia | J_a [kgm^2] | 1.140 |
| Armature Damping | D_a [$\frac{Nms}{rad}$] | 0.00148 |
| Armature Resistance | R_a [ohm] | 2.1 |
| Armature Inductance | L_a [$\frac{ohm}{s}$] | 0.0048 |
| Armature Current | I_a [A] | 8.25 |
| Stall Torque | T_{stall} [Nm] | 97.2 |
| No-load angular velocity | $\omega_{no-load}$ [$\frac{rad}{s}$] | 150 |
| Load Inertia | J_L [kgm^2] | 5 |
| Load Damping Coefficient | D_L [$\frac{Nms}{rad}$] | 30 |
| Spring Coefficient | k_L [$\frac{N}{m}$] | 55 |
| Number of teeth of the input gear | N_m | 100 |
| Number of teeth of the output gear | N_L | 150 |

For the given set of parameters in the Table 1, determine the following:

- Develop the electrical and mechanical equations for the electro-mechanical system.
- Develop the transfer function for the electro-mechanical system.
- Draw the diagram for the unity feedback control system, that is, the electro-mechanical system with controller and feedback using general parameters.
- Calculate K_m and K_b using the Torque-Speed curve, and given electro-mechanical system information.
- Create an .m file (Matlab) with all given and calculated parameters for the electro-mechanical system.
- Using the given information develop the transfer function and select the appropriate Ziegler-Nichols method.
- Design a PID controller using the proper Ziegler-Nichols method. For the simulation use the step input.
- Do some fine-tuning for the controller parameters so that you can get better performance and report all your results and conclusions.
- Write conclusions.

Example of Student Work:

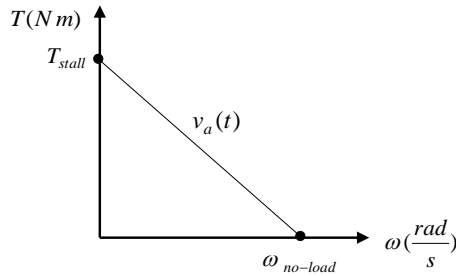


Figure 2: Torque-Speed Curve for DC motor

Develop the electrical and mechanical equations for the electro-mechanical system.

$$\text{Motor torque: } T_m(s) = K_m I_a(s), \quad T_m(s) = T_L(s) + T_d(s), \quad T_d(s) = 0 \quad (1)$$

$$\text{Armature voltage: } V_a(s) = R_a I_a(s) + sL_a I_a(s) + V_b(s) \quad (2)$$

$$\text{BMF voltage: } V_b(s) = K_b \omega(s), \quad \omega(s) = s\theta_m(s) \quad (3)$$

$$\text{Load torque: } T_L(s) = (J_L s^2 + D_L s + k_L)\theta_L(s) \quad (4)$$

Develop the transfer function $G_1(s) = \frac{\theta_m(s)}{V_a(s)}$ and $G_2(s) = \frac{\theta_L(s)}{V_a(s)}$ for the electro-mechanical system.

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_m}{J_m L_a s^3 + (J_m R_a + D_m L_a) s^2 + (D_m R_a + K_b K_m + k_m L_a) s + k_m R_a} \quad (5)$$

$$J_m = J_a + \left(\frac{N_m}{N_L}\right)^2 J_L \quad (6)$$

$$D_m = D_a + \left(\frac{N_m}{N_L}\right)^2 D_L \quad (7)$$

$$k_m = \left(\frac{N_m}{N_L}\right)^2 k_L \quad (8)$$

$$\theta_m = \theta_L \left(\frac{N_L}{N_m}\right) \quad (9)$$

$$\frac{\theta_L(s)}{V_a(s)} = \frac{K_m \frac{N_m}{N_L} \frac{1}{J_m L_a}}{\left[s^3 + \frac{(J_m R_a + D_m L_a)}{J_m L_a} s^2 + \frac{(D_m R_a + K_b K_m + k_m L_a)}{J_m L_a} s + \frac{k_m R_a}{J_m L_a} \right]} \quad (10)$$

The following task is to draw the diagram for the unity feedback control system, that is, the electro-mechanical system with controller and feedback using general parameters.

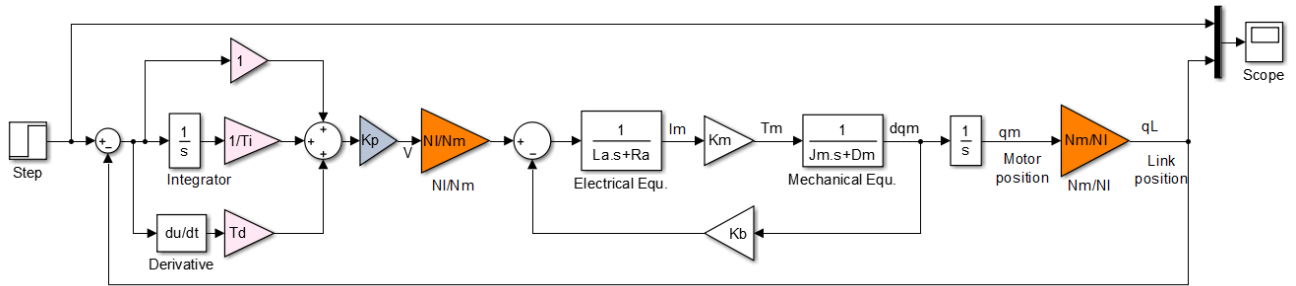


Figure 3: DC Motor block diagram with Simulink

After that, the task is to calculate K_m and K_b using the Torque-Speed curve (Figure 2), and information from Table 1. Assume that L_a is much smaller than R_a (only for calculating these two parameters).

$$T_m(s) = -\frac{K_m}{L_a s + R_a} K_b \omega(s) + \frac{K_m}{L_a s + R_a} V_a(s) \approx -\frac{K_m}{R_a} K_b \omega(s) + \frac{K_m}{R_a} V_a(s) \quad (11)$$

$$\frac{T_{stall}(s)}{V_a(s)} = \frac{K_m}{R_a} \quad (12)$$

$$K_b = \frac{V_a(s)}{\omega_{no-load}} \quad (13)$$

The following task is to create an .m file with all given and calculated parameters for the electro-mechanical system. See attached Appendix A.

Using the information from Table 1, develop the transfer function $G_2(s) = \frac{\theta_L(s)}{V_a(s)}$ for the electro-mechanical system and select spring coefficient k_L such that the plant has integrator(s) and/or dominant complex-conjugate poles (Use Matlab for fast calculation).

$$\frac{\theta_L(s)}{V_a(s)} = \left[\frac{K_m \frac{N_m}{N_L}}{J_m L_a s^3 + (J_m R_a + D_m L_a) s^2 + (D_m R_a + K_b K_m + k_m L_a) s + k_m R_a} \right] \quad (14)$$

Transfer function:

$$\frac{\theta_L(s)}{V_a(s)} = \frac{3.0240}{0.01614s^3 + 7.125s^2 + 29.48s + 51.33} \quad (15)$$

The System poles from equation (15) are: $s_1 = -437.31$; $s_2 = -2.08 + 1.72i$, $s_3 = -2.08 - 1.72i$. System has dominant complex-conjugate poles and according to that we will use second Ziegler-Nichols method.⁶

Design a *PID* controller using the proper Ziegler-Nichols method. For the simulation use the step input.

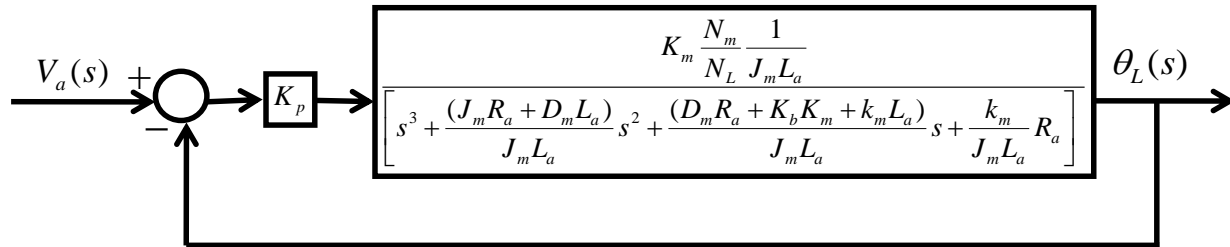


Figure 4: Block diagram for the system with PID controller designed by use of the Ziegler-Nichols tuning rule

The Routh–Hurwitz array has been found from the block diagram in Figure 4, see equation 16.

$$A(s) = s^3 + \frac{(J_m R_a + D_m L_a)}{J_m L_a} s^2 + \frac{(D_m R_a + K_b K_m + k_m L_a)}{J_m L_a} s + \frac{k_m}{J_m L_a} R_a + K_m \frac{N_m}{N_L} \frac{1}{J_m L_a} K_p \quad (16)$$

By applying parameters from Table 1 to the equation 16, the final Routh–Hurwitz array has been found. See equation (17).

$$A(s) = 0.0164s^3 + 7.125s^2 + 29.48s + (51.33 + 3.024K_p) \quad (17)$$

Using the Routh–Hurwitz stability criterion presented in Table 2, the critical gain $K_{cr} = 418.53$ and the corresponding period $P_{cr} = 0.145$ sec have been found. See Figure 5.

Table 2: Routh–Hurwitz stability criterion

| | | |
|-------|------------------------------------|---------------------|
| s^3 | 0.01614 | 29.48 |
| s^2 | 7.125 | $51.33 + 3.024 K_p$ |
| s^1 | $\frac{504.003 - 0.05 K_p}{7.125}$ | 0 |
| s^0 | $51.33 + 3.024 K_p$ | 0 |

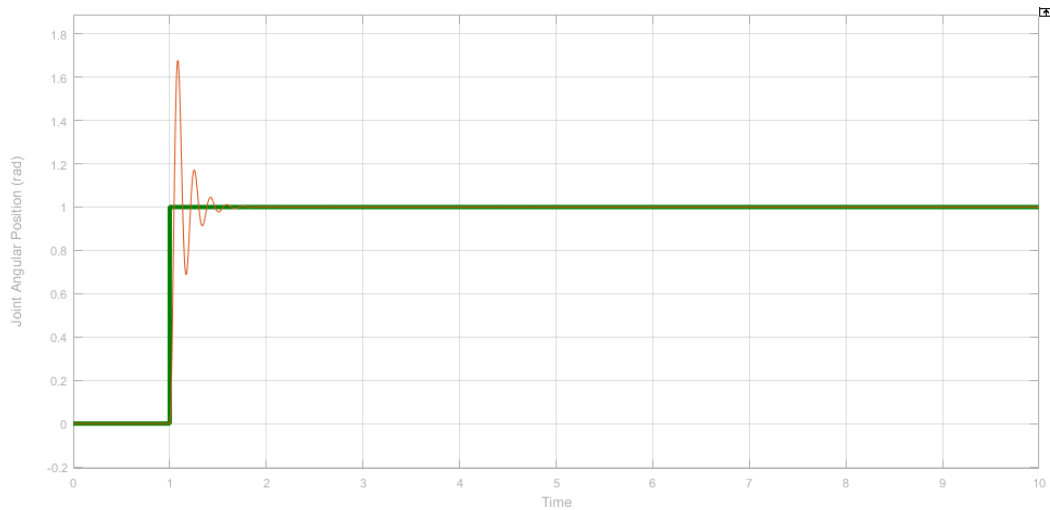


Figure 5: System response for calculated initial $K_{cr} = 4184.153$ controller and $P_{cr} = 0.145$ sec.

Some fine-tuning is necessary for the controller parameters so that the better performance can be achieved: by increasing the period $P_{cr} = 2.5$ ses, the system poles become real and negative. In this case the response is better. See Figure 6.

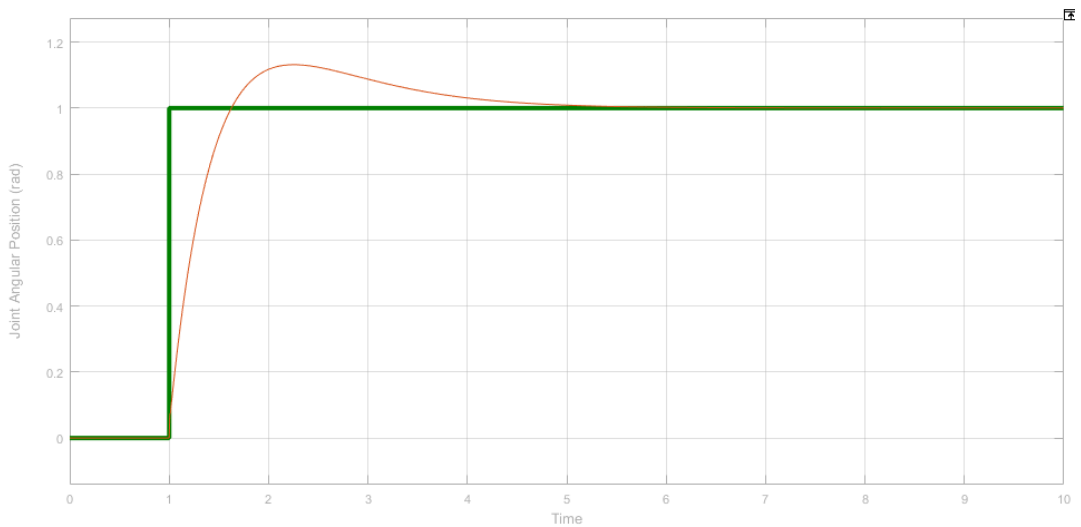


Figure 6: System response after tuning $K_{cr} = 4184.153$ controller and $P_{cr} = 0.145$ sec.

From the literature⁷, by applying constant voltage V to the motor terminal, the motor shaft will accelerate according to Equations 2 and 4, and achieve a final steady state velocity. Under steady state conditions, the current is constant, and the motor equations become:

$$V_a(s) = R_a I_a(s) + K_b \omega(s) \quad (18)$$

$$T_m(s) = K_m I_a(s) \quad (19)$$

Combining Equations 18 and 19, the relation between the velocity and generated torque at steady state is given in Equation 20.

$$V_a(s) = R_a \frac{T_m(s)}{K_m} + K_b \omega(s) \quad (20)$$

From the Equation 20, the maximum speed can be calculated if there is no load, (torque is equal to zero).

$$\omega_{\max}(s) = \frac{V_a(s)}{K_b} \quad (21)$$

Maximum RPM is calculated in Equation 22:

$$RPM_{\max} = \omega_{\max}(s) \frac{60}{2\pi} \quad (22)$$

The generated torque at stall is given from Table 1. The relation between the speed and the torque is expressed in Equation 23.

$$RPM = RPM_{\max} - Slope_{RPM} T_M \quad (23)$$

Slope is calculated from the relation in Equation 24:

$$Slope_{RPM} = \frac{RPM_{\max}}{T_{Stall}} \quad (24)$$

Figure 7 shows the speed-torque curve for the DC motor and Torque lines for different voltages.

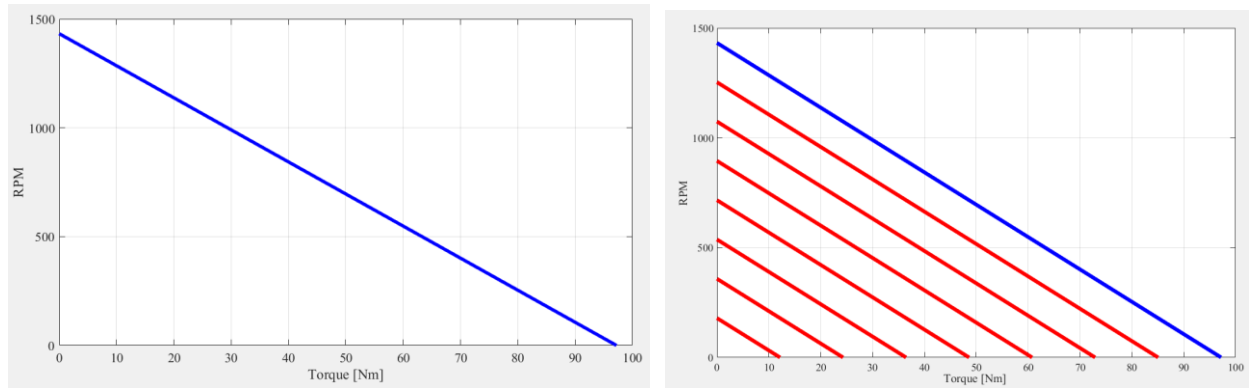


Figure 7. DC motor speed-torque curve and related DC motor torque lines for different voltages

The current line, presented in Figure 8, can be constructed from the known maximum current in Table 1.

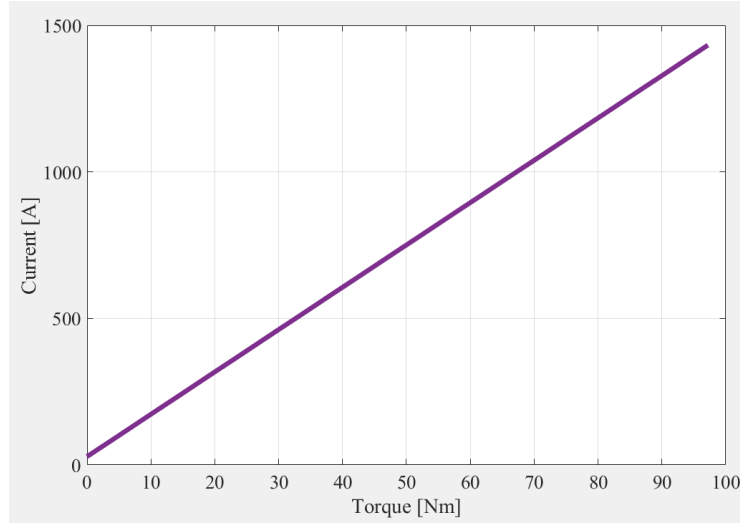


Figure 8. DC motor Current Line

The input power to the motor is given in Equation 25. The maximum voltage is $V_{\max} = 45V$. The current line in Figure 8 is scaled by the value of $\frac{RPM_{\max}}{I_{\max}}$, where I_{\max} is maximum current given in Table 1. This is applied later on all motor steady state characteristic diagrams.

$$P_{IN} = V_{\max} I_m \quad (25)$$

The output power is the power that is delivered to the load. See Equation 26.

$$P_{OUT} = RPM \frac{2\pi}{60} T_m \quad (26)$$

The power curve is shown on Figure 9.

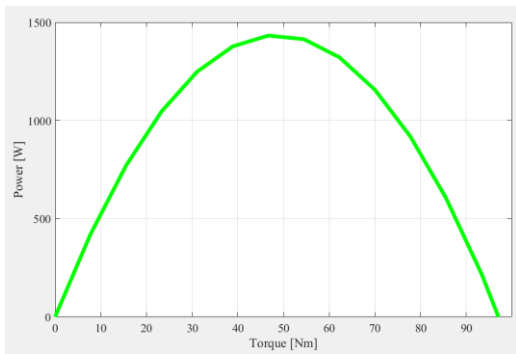


Figure 9. DC motor Power curve

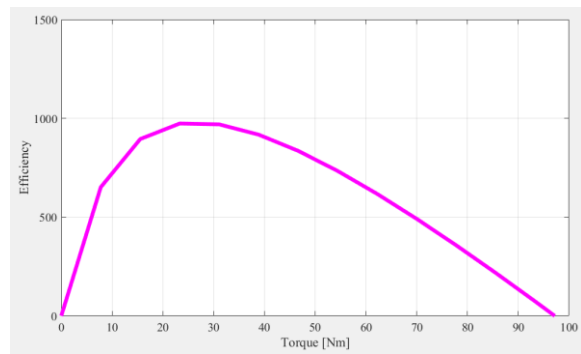


Figure 10. DC motor Efficiency curve

The efficiency can be calculated as a relation between input and output power, Equation 27. The efficiency curve in Figure 10 is scaled by the value of RPM_{\max} . This is applied later on motor steady state characteristic diagram.

$$e = \frac{P_{OUT}}{P_{IN}} \quad (27)$$

All the motor sizing data are presented graphically in Figures 11.

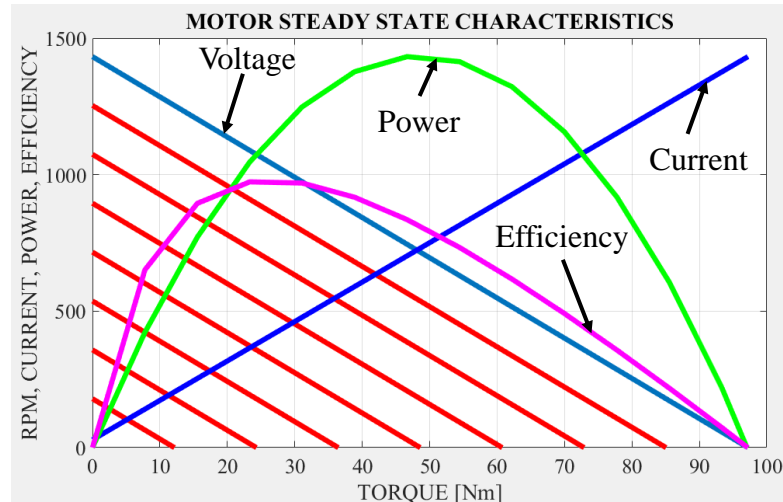


Figure 11. DC motor Characteristics

Conclusions

EMMCT learning module presented in this paper covers following important topics: electric circuits, DC motor, gears, translational and/or rotational links, applications to the first principles (Ohm's law Kirchoff's laws Newton's laws), Transfer Functions, Block diagrams, Matlab/Simulink modeling, simulation and analysis, etc. Students are expected to select a DC motor and identify all needed parameters, choose a gear and load (Rotational or Translational), model the electro-mechanical system, generate transfer functions, and create Matlab/Simulink models with the appropriate controller; then, students will simulate and analyze the whole system. They are also required to do some fine-tuning for the controller parameters so that they can achieve better performance. At the end, students submit comprehensive engineering reports to document all requirements.

References

1. Jovanovic V., Goris T., Djuric A. M., Katsioloudis P., Luetke N., Moustafa M., Matrood B., (2015). Integration of Mechatronics Design Approach into Teaching of Modeling Practices", *122nd ASEE Annual Conference & Exposition*, June 14-17, Seattle, WA
2. Mahfouz A. A. Mohammed M. K. Salem F. A. (2013). Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink, *I. J. Intelligent Systems and Applications*, 05, pp. 39-57, Retrieved on January 4, 2017 from: <http://www.mecspress.org/>
3. Tarn T., Bejczy A. K., Yun X., and Li Z., (1991). Effect of Motor Dynamics on Nonlinear Feedback Robot Arm Control, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 1, pp. 114-122.
4. Tewari A., (2002). Modern Control Design With Matlab and Simulink, *John Wiley & Sons, Ltd.*

5. Djuric A. M. (2007). Reconfigurable Kinematics, Dynamics and Control Process for Industrial Robots, PhD dissertation, University of Windsor (Canada)
6. Ziegler J. G. and Ziegler N. B. Nichols, (1942). Optimum Settings for Automatic Controllers, *Trans. ASME*, Vol. 64, pp. 759-768
7. Electro-Craft Corp., (1973). DC Motors, Speed Controls, Servo Systems, Second Edition.

Appendix A

```

%=====
% DC MOTOR PARAMETERS
%=====
%Input data
Va = 45                % Armature Voltage [V]
Ja = 1.140            % Armature Inertia [Kgm^2]
Da = 0.00148          % Armature damping coefficient [Nms/rad]
Ra = 2.1              % Armature Resistant [om]
La = 0.0048           % Armature Inductance [ohm/s]
Ia = 8.25             % Armature current [A]
Tstall = 97.2         % Stall Torque [Nm]
omega_noload = 150    % omega no load [rad/s]
Jl = 5                % Load Inertia [Kgm^2]
Dl = 30               % Load damping coefficient [Nms/rad]
kl = 55               % Spring coefficient [N/m]
Nm = 100              % Number of teeth of the input gear (motor gear)
Nl = 150              % Number of teeth of the output gear (load gear)
%Calculated data
Dm = Da+Dl*(Nm/Nl)^2  % Combined damping coefficient (motor + load)
Jm = Ja+Jl*(Nm/Nl)^2  % Combined Moment of Inertia (motor + load)
km = kl*(Nm/Nl)^2     % Combined spring coefficient
Km = Ra*(Tstall/Va)    % Motor Tortue Constant [Nm/A]
Kb = Va/omega_noload  % Back EMF constant [Nm/A]
%Plant Transfer Function
num = [0 0 0 Km*(Nm/Nl)]
den = [Jm*La (Jm*Ra+Dm*La) (Dm*Ra+Kb*Km+Km*La) Km*Ra]
plant = tf(num,den)
[z,p,K] = tf2zp(num,den)
%PID controller
Kcr = 4184.153
Pcr = 2.5
Kp = 0.6*Kcr
Ti = 0.5*Pcr
Td = 0.125*Pcr
%=====
Vmax = Va
OM1max = Vmax/Kb
RPMmax = OM1max*60/(2*pi)
SlopeRPM = RPMmax/Tstall
Torque = [0:Tstall/100:Tstall]';
%=====
%Determine RPM line
RPM = RPMmax-SlopeRPM*Torque;
%=====
%Plot RPM-Torque Lines for 40 V
plot(Torque, RPM)
hold on

```

```

grid on;
%=====
%Plot Torque Lines for different voltages
hold on
for V=Vmax-Vmax/8:-Vmax/8:Vmax/8
OMEGA = V/Kb*ones(length(Torque), 1) -(SlopeRPM*(2*pi/60))*Torque;
RPM =OMEGA*60/2/pi;
plot(Torque,RPM, 'r')
hold on
end
grid on;
%=====
%Current Line
Imax = Ia
Imin = 0.02*Imax
Iline=[Imin:(Imax-Imin)/100:Imax]';
plot(Torque, (Iline*RPMmax/Imax), 'b')
axis([0 100 0 1500]);
grid on;
%=====
%Plot Power Line
Pin = Vmax*Iline;           %Input Power
Pout = (RPM*2*pi/60).*Torque; %Output Poer
Pout_max = max(Pout);
plot(Torque*8,Pout*RPMmax/Pout_max,'g');
axis([0 100 0 1500]);
grid on;
%=====
%Plot Efficiency line
Effline = (Pout./Pin)/5;
Emax = max(Effline);
plot(Torque*8, Effline*RPMmax*1.5, 'm');
axis([0 100 0 1500]);
grid on;
xlabel('TORQUE [Nm]')
ylabel('RPM, CURRENT, POWER, EFFICIENCY')
title('MOTOR STEADY STATE CHARACTERISTICS')

```