

MATLAB/SIMULINK MODELING OF DC MOTOR SPEED

Gîlcă Gheorghe, Lecturer PhD, “Constantin Brâncuși” University from Tîrgu-Jiu, ROMANIA

Ionescu Marian, Engineer, “Constantin Brâncuși” University from Tîrgu-Jiu, ROMANIA

ABSTRACT: This paper describes the MATLAB/Simulink realization of the DC motor speed control methods, namely field resistance, armature voltage and armature resistance control methods, and open-loop control system modeling for DC motor drives.

KEY WORDS: DC motor, linear model, rotational speed, armature voltage

1. INTRODUCTION

Computer modeling and simulation tools have been extensively used to support and enhance electric machinery courses. MATLAB with its toolboxes such as Simulink [1] and SimPowerSystems [2] is one of the most popular software packages used by educators to enhance teaching the transient and steady-state characteristics of electric machines [3,4]. A permanent magnet DC motor can be thought of as an electrical system that consists of the rotor winding (ideal inductor and ideal resistor) and an electromotive force element. Back emf represents how the motor acts as a

generator/voltage source that works against the input voltage [5].

2. EXTRACTING A LINEAR MODEL INTO MATLAB

A linear model of the system can be extracted from the Simulink model into the MATLAB workspace. This can be accomplished employing the MATLAB command `linmod` or from directly within Simulink as we will do here. We will specifically use the base Simulink model developed from first principles shown below.

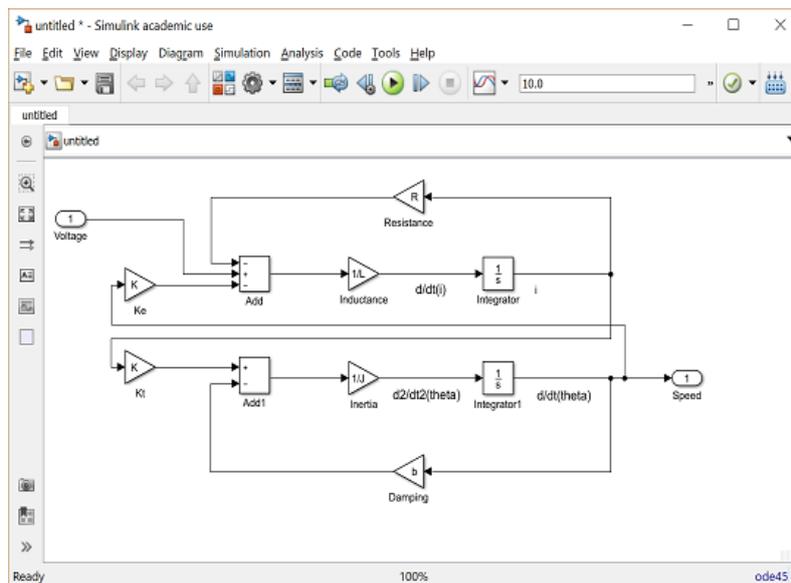


Figure 1. DC motor speed simulink modeling

The physical parameters for our example are: (J) moment of inertia of the rotor, (b) motor viscous friction constant, (Ke) electromotive force constant, (Kt) motor torque constant, (R) electric resistance, (L) electric inductance. The physical parameters have the following values:

$$J = 0.01;$$

$$b = 0.1;$$

$$K = 0.01;$$

$$R = 1;$$

$$L = 0.5;$$

We then need to identify the inputs and outputs of the model we wish to extract. First right-click on the signal representing the Voltage input in the Simulink model. Then choose **Linear Analysis Points > Open-loop Input** from the resulting menu. Similarly, right-click on the signal representing the Speed output and select **Linear Analysis Points > Open-loop Output** from the resulting menu. The input and output signals should now be identified on your model by arrow symbols as shown in the figure below.

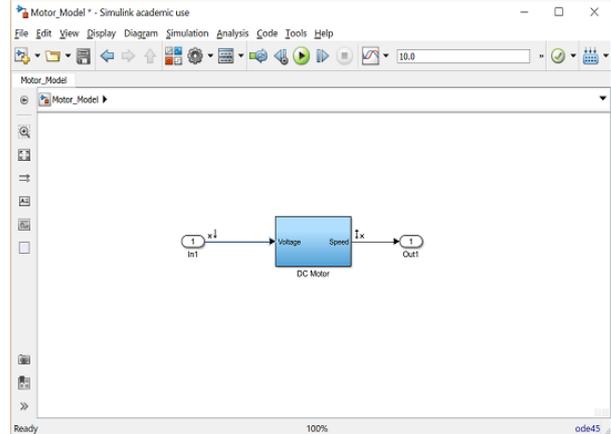


Figure 2. The dc motor model with input and output

In order to perform the extraction, select from the menus at the top of the model window **Analysis > Control Design > Linear Analysis**. This will cause the Linear Analysis Tool to open. Within the Linear Analysis Tool window, the Operating Point to be linearized about can remain the default, Model Initial Condition. In order to perform the linearization, next click the Step button identified by a small green triangle. The result of this linearization is the linsys1 object which will appear in the Linear Analysis Workspace as shown below. Furthermore, the open-loop step response of the linearized system also will be generated automatically.

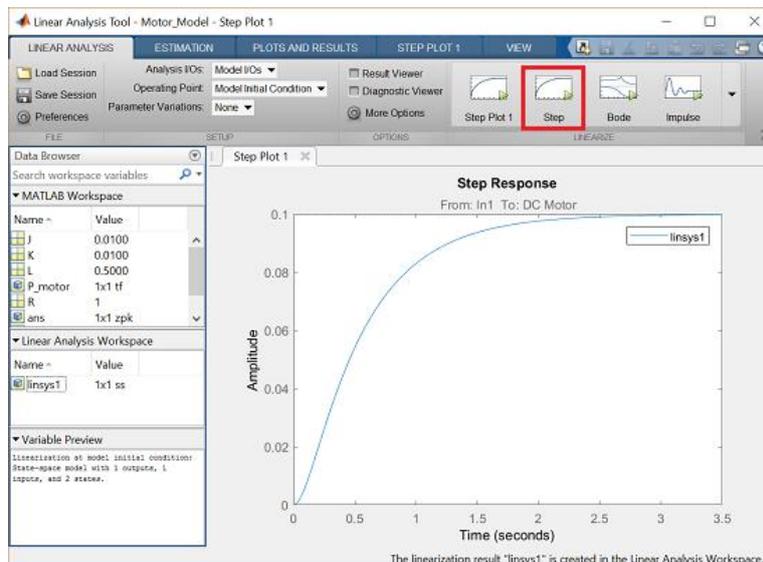


Figure 3. The open-loop step response of the linearized system

The open-loop step response above is consistent with the response generated in the DC Motor Speed: System Analysis page. The reason the responses match so closely is because this Simulink model uses only linear components. Note that this process can be used to extract linear approximations of models with nonlinear elements too.

We will further verify the model extraction by looking at the model itself. The linearized model can be exported by simply dragging the object into

```
s = tf('s');
P_motor = K / ((J*s+b) * (L*s+R) + K^2);
zpk(P_motor)

ans = 2
s+9.997) (s+2.003)
```

3. OPEN-LOOP RESPONSE

The open-loop step response can also be generated directly within Simulink, without extracting any models to the MATLAB workspace. In order to simulate the step response, the details of the simulation must first be set. This can be accomplished by selecting **Model Configuration Parameters** from the **Simulation** menu. Next we need to add an input signal and a means for displaying the output of our simulation. This is done by doing the following:

- Remove the In1 and Out1 blocks.
- Insert a Step block from the Simulink/Sources library and connect it with a line to the Voltage input of the motor subsystem.
- To view the Speed output, insert a Scope from the Simulink/Sinks library and connect it to the Speed output of the motor subsystem.
- To provide an appropriate unit step input at t=0, double-click the Step block and set the **Step time** to "0".

The final model should appear as shown in the following figure.

the **MATLAB Workspace**. This object can then be used within MATLAB in the same manner as an object created directly from the MATLAB command line. Specifically, entering the command `zpk(linsys1)` in the MATLAB command window demonstrates that the resulting model has the following form.

$$P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{2}{(s + 9.997)(s + 2.003)} \quad \left[\frac{\text{rad/sec}}{V} \right] \quad (1)$$

the **Simulation** menu. Within the resulting menu, define the length for which the simulation is to run in the **Stop time** field. We will enter "3" since 3 seconds will be long enough for the step response to reach steady state. Within this window you can also specify various aspects of the numerical solver, but we will just use the default values for this example.

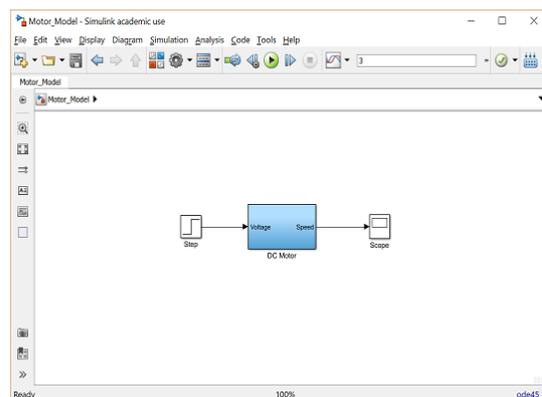


Figure 4. Block diagram for step function simulation

Then run the simulation (press **Ctrl-T** or select **Run** from the **Simulation** menu). When the simulation is finished, double-click on the scope and you should see the following output.

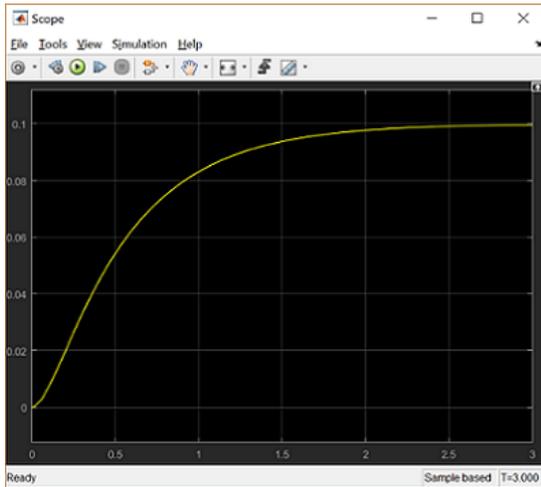


Figure 5. The response DC motor at step function

This response is identical to that obtained by MATLAB above using the extracted model. This is again to be expected

4. CONCLUSIONS

In armature controlled separately excited DC motor, rotational speed is proportional to armature voltage. However, due to load disturbances, mechanical wear and tear and

because this Simulink model includes only linear blocks.

other environmental factors DC motor response varies with time.

Simulation model of DC motor speed control method and open-loop control system for DC motor drives have been developed using MATLAB/Simulink.

REFERENCES

- [1] SIMULINK, *Model-based and system-based design, using Simulink*, MathWorks Inc., Natick, MA, 2000.
- [2] *SimPowerSystems for use with Simulink, user's guide*, MathWorks Inc., Natick, MA, 2002.
- [3] M. H. Nehrir, F. Fatehi, and V. Gerez, *Computer modeling for enhancing instruction of electric machinery*, IEEE Trans Educ 38 (1995), 166-170.
- [4] W. M. Daniels and A. R. Shaffer, *Re-inventing the electrical machines curriculum*, IEEE Trans Educ 41 (1998), 92-100.
- [5] Richardson, M. Back Emf of a DC Motor.[Referenced 28.06.2020]. Available at: <https://www.youtube.com/watch?v=a7oj0REQzpE>