Modelling and simulating a DC-motor using OpenModelica and Simulink

TNG022
Modelling and Simulation

Emil Alsbjer

Linköpings Universitet

emial133@student.liu.se

Instructor: Kjell Karlsson

Lab Performed: November 27, 2024

Submission Date: December 4, 2024

Table of Contents

1 Introduction

Modeling and simulation are essential for analyzing complex physical systems, allowing researchers to study behavior without relying on costly physical prototypes. Simulations can help identify potential problems early, saving time and resources.

This lab focuses on two common approaches to modeling: block diagram-based modeling and object-oriented modeling. Using a DC motor as the system of interest, the motor is modeled and simulated in Simulink and OpenModelica, respectively.

The objective of this report is to compare these two methods, analyzing their similarities, differences, and effectiveness in representing and simulating physical systems.

2 Method

2.1 Approach

The DC motor was modeled using its schematic representation shown in Figure 1.

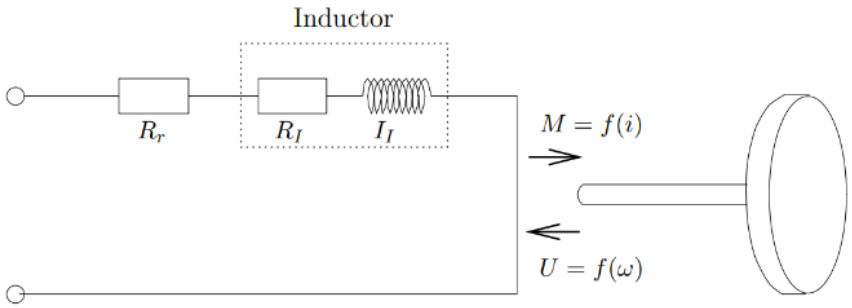

Figure 1: Schematic picture of DC-motor

The mathematical model underlying the two modelling approaches is given by equation 1.

$$\begin{aligned} u(t) &= (R_r + R_I)i(t) + I_I \frac{di(t)}{dt} - K\omega(t) \\ Ki(t) &= J\dot{\omega}(t) + b\omega(t) \end{aligned} \tag{1}$$

Where $u(t)$ is the applied voltage, $i(t)$ current, $\omega(t)$ angular velocity, $(R_r + R_I)$ is the terminal resistance, I_I is armature inductance, K is the conversion coefficient, J is the inertia and b is the friction coefficient.

2.2 Block scheme modelling

Before implementing the block-oriented model in Simulink, a bond graph was developed to represent the structure and interactions of the DC motor system. This bond graph served as the foundation for designing a block-based model using components such as gain, integrator, summation, constant, and scope blocks. These elements were carefully arranged to accurately replicate the motor's dynamics and operational behavior. The finalized model, providing a complete representation of the system, is illustrated in Figure 2.

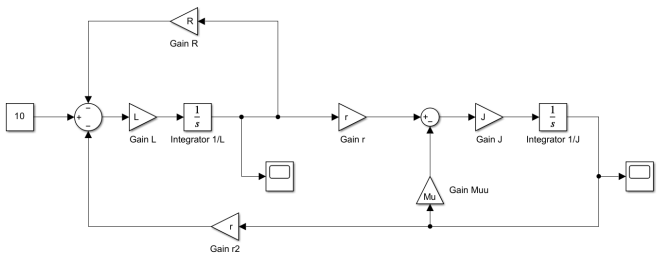

Figure 2: Implemented simulink block scheme

2.3 Object-oriented modelling

Object-oriented modeling relies on mathematical equations to to describe the components of the system. The model is acausal, meaning that the inputs and outputs of the systems are not predefined. This flexibility allows the simulation to determine the necessary relationships dynamically during execution. Additionally, object-oriented modeling supports graphical interfaces, allowing intuitive construction of the system, as shown in Figure 3.

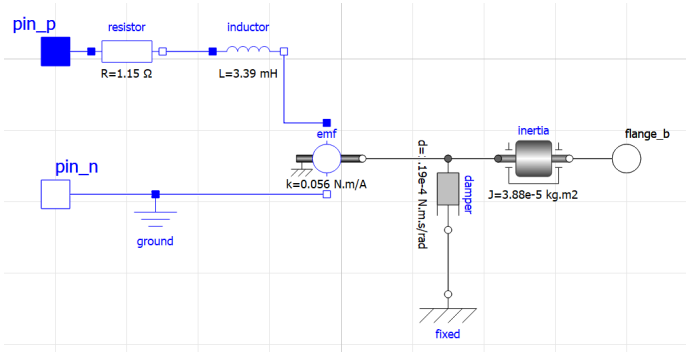

Figure 3: Implemented object-oriented model

3 Analysis

Both the block-oriented and the object-oriented approaches yield equivalent results for simulating the DC motor, despite the different approaches.
Simulink requires predefined input and output relationships, making it more abstract and reliant on system-specific connections. While **OpenModelica** uses an equation-based modeling approach, allowing for greater flexibility in modifying components.The object-oriented approach makes it easier to expand and modify the system because it uses mathematical equations instead of fixed input-output connections. On the other hand, block-oriented models work well when the input and output relationships are clearly defined.

Ultimately, the choice of modeling approach depends on the system's requirements. For larger or more flexible systems, the object-oriented method offers adaptability. For smaller, well-defined systems, the block-oriented approach may provide may provide a clearer structure. Both approaches demonstrate strengths that can be leveraged depending on the scenario.

4 Conclusion

Both modeling approaches produced similar results despite their differing methodologies. The block-oriented method (used in Simulink) is more abstract, relying on inputs and outputs, and is better suited for systems where these parameters are clearly defined. Meanwhile, the object-oriented approach (used in OpenModelica) is equation-based and acausal, making it more flexible for modeling nonlinear systems or systems with undefined inputs and outputs.

Each method has its strengths and weaknesses depending on the context. For systems like the DC motor studied in this lab, either approach is viable. However, determining the optimal method for continuing development would require further testing.

5 References