# Building a physical system:

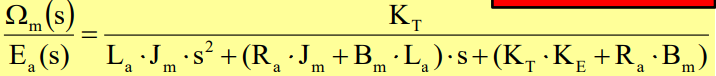
To build a physical system of a DC motor, you will need the following components:

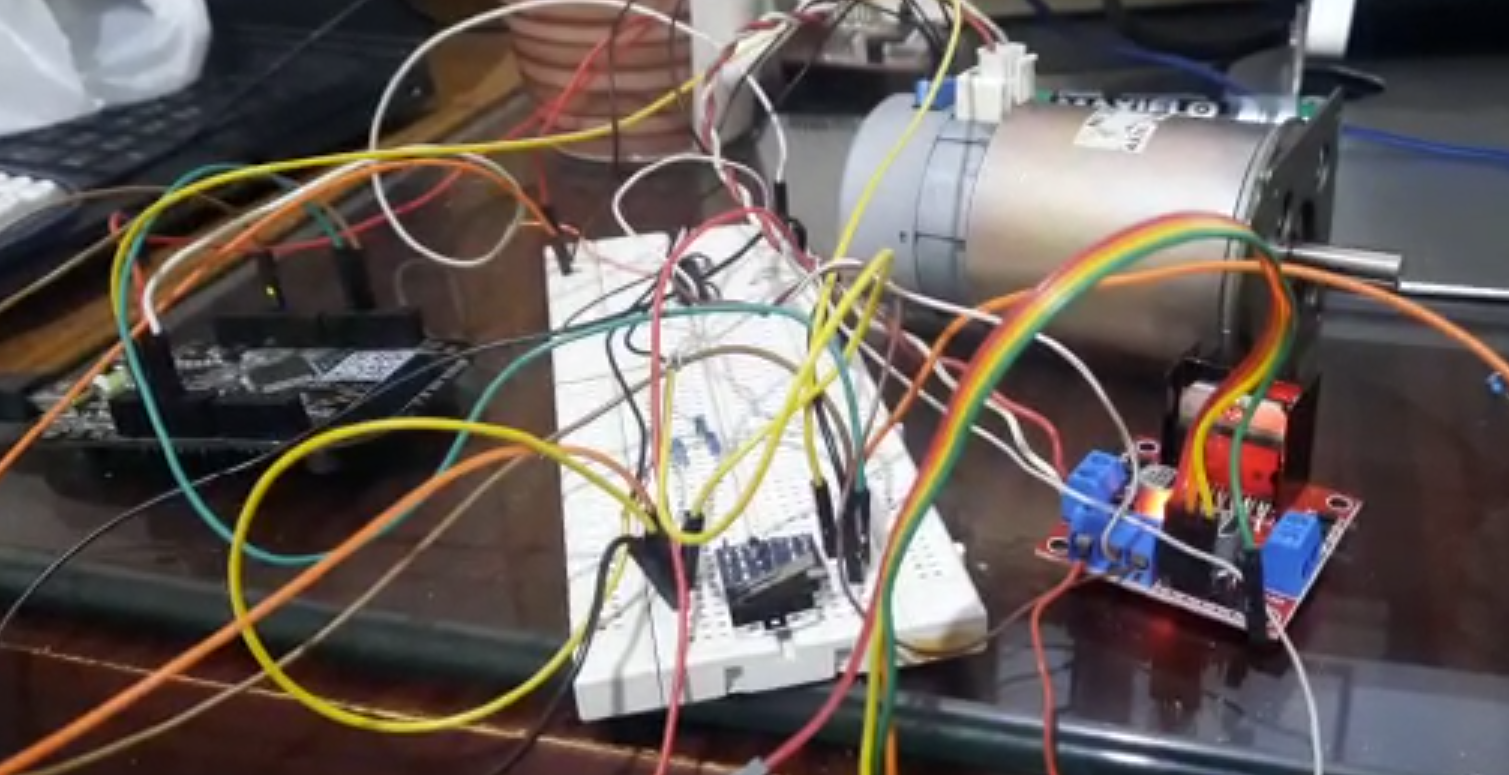
* DC motor
* Power supply (DC)
* Motor driver (l298d)
* Encoder
* Microcontroller (in my case NXP KL25Z4)
* Breadboard
* Jumper wires
* Multimeter
* PC with MATLAB and Simulink installed.
* Serial port to read measurement.

Here is a step-by-step guide to building the physical system:

1. Mount the DC motor on a stable base or chassis. Make sure it is securely fixed and can rotate freely.
2. Connect the power supply to the motor driver. The voltage and current ratings of the power supply should be compatible with the motor and driver specifications.
3. Connect the motor driver to the DC motor. Make sure the polarity of the connections is correct.
4. Connect the encoder to the motor shaft. The encoder will provide feedback on the motor position and speed.
5. Connect the motor driver and encoder to the microcontroller or Arduino board. The board will control the motor driver and read the encoder signals.
6. Write a program in the microcontroller to control the motor speed and direction based on the encoder feedback. You can use PWM (Pulse Width Modulation) to control the motor speed.
7. Test the system by running the program and observing the motor behavior. You can use a multimeter or oscilloscope to measure the voltage and current ofthe motor and the encoder signals.

Transfer function of the system





**The details and source code of the implemented system is documented here: (Check readme.md)**

<https://github.com/AbdallahAwdalla/DC_Motor_Param_Estimation/tree/main/DAQ>

# Preparing a set of experiments:

To acquire data for the model identification, you can perform the following experiments:

Step response: Apply a step input to the motor and measure the motor speed and current response. Vary the input voltage and observe the response.

# Modeling and simulating the system:

To model and simulate the system on Simulink/Simscape, you can follow these steps:

1. Create a Simulink model of the DC motor system, including the motor, driver, encoder, and microcontroller.
2. Define the system parameters, such as the motor parameters (resistance, inductance, back EMF), the driver parameters (voltage and current limits), and the encoder parameters (resolution, accuracy).
3. Implement the control algorithm in Simulink, based on the program you wrote for the microcontroller or Arduino board.
4. Simulate the system by running the model in Simulink. Apply the same inputs as the experiments you performed on the physical system and observe the simulated response.

A picture containing sketch, screenshot, jack, design

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

# Obtaining the values of the system parameters:

To obtain the values of the system parameters using the Parameter Estimation tool in MATLAB Simulink, you can follow these steps:

1. Create a MATLAB script or Simulink model that defines the system model and the experimental data.
2. Use the Parameter Estimation tool in Simulink to estimate the values of the system parameters based on the experimental data.
3. Refine the parameter estimates by running additional simulations and comparing the simulated and experimental data.

A screenshot of a computer

Description automatically generated with medium confidence

## Comparing the responses:

To compare the responses of both the physical system and the simulated one, you can follow these steps:

1. Run the same experiments on the physical system and the simulated one.
2. Record the experimental data for both systems.
3. Plot the experimental data for both systems and compare the responses.
4. Refine the system model and parameter estimates if there are significant differences between the experimental and simulated data.

By following these steps, you can build and test a physical system of a DC motor, model and simulate the system on Simulink/Simscape, and obtain the values of the system parameters using the Parameter Estimation tool in MATLAB Simulink. Comparing the responses of both the physical system and the simulated one will help you validate the accuracy of your model and parameter estimates.

A screen shot of a graph

Description automatically generated with low confidence

# Comment about the accuracy of parameters estimation tool.

As Observed it’s needed for the system parameters to initialized with reasonable value other wise the iterative solution may not be able to get the correct parameters.