Arab Academy for Science, Technology and Maritime Transport

College of Engineering and Technology

Mechatronics Department

Design of lag compensator for two degree of freedom robotic arm

Presented By:

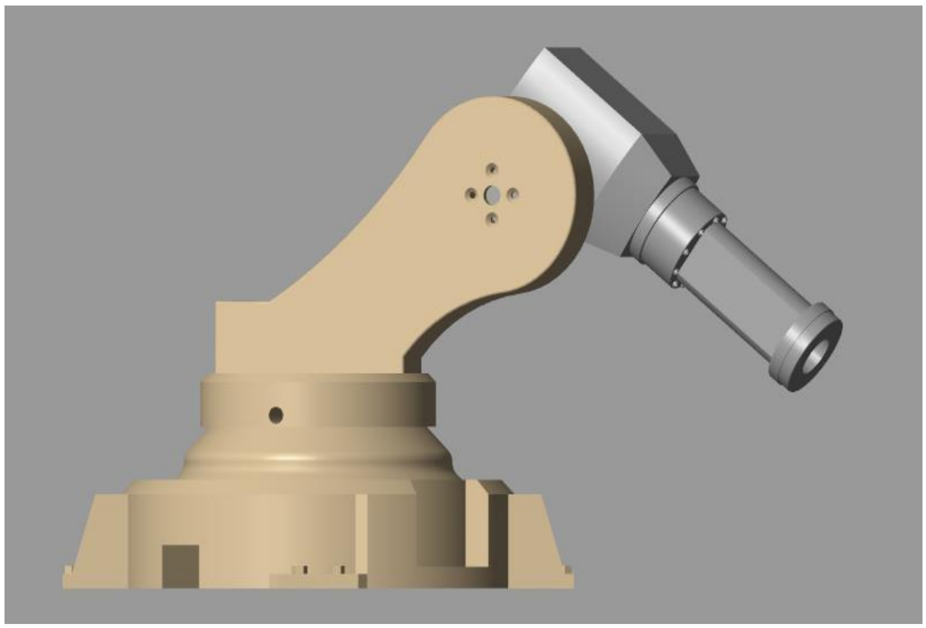
Hamza Hussein Mohamed

CONTENTS

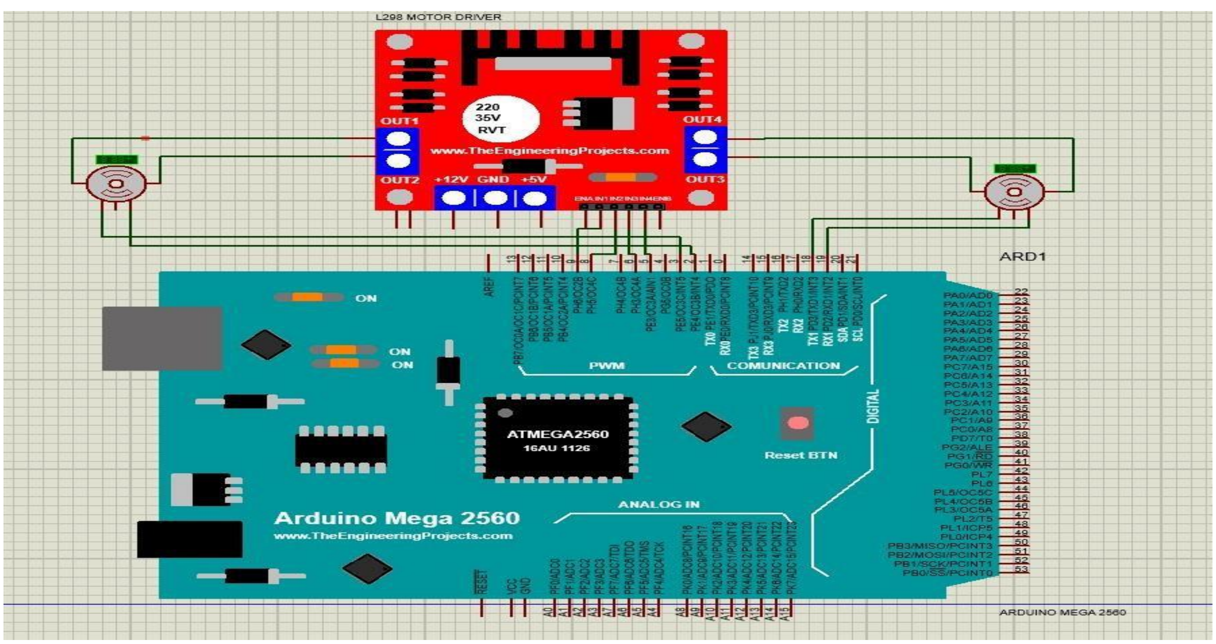
s

1 Construction

Robot model consists of 2 degrees of freedom, the main actuator of each joint is dc motor (jga25_370 210 rpm) with encoder. All the model was designed by inventor application, more over it was 3d printed with PLA plus material. The encoder act as the main feedback of position and speed. As it consists of 2 channel encoders to feedback the direction also. The robot model is shown in the below figure 1.1.



1.1 cad model.



1.2 circuit design on proteus software 1

2 Theory of Operation

A 2-degree-of-freedom (2-DOF) robot typically refers to a robotic manipulator or arm with two joints that can move independently, providing two degrees of freedom. The joints could be revolute (rotational) or prismatic (linear), allowing for various configurations and movements. Simulink control system design where a lead-lag compensator is employed to stabilize and control the motion of the robot. Simulink is a simulation and model-based design environment provided by MATLAB, commonly used for dynamic system modeling and simulation. Here's a basic outline of how such a system might work:

Modeling the Robot Dynamics: First, you would create a dynamic model of the robot in Simulink. This model would describe how the position, velocity, and acceleration of each joint relate to the control inputs (usually voltages or currents applied to motors) and external forces or torques acting on the robot.

Designing the Lead-Lag Compensator: The lead-lag compensator is a type of controller designed to improve the stability and performance of the control system. In this context, it would be designed to control the motion of the robot joints, ensuring accurate and smooth movement while minimizing overshoot and settling time.

Closed-Loop Control System: The lead-lag compensator is integrated into a closed-loop control system with feedback from sensors measuring the position or velocity of each joint. This feedback allows the controller to adjust the control signals in real-time to track desired trajectories or commands.

Simulation and Tuning: With the Simulink model set up, you would simulate the behavior of the closed-loop control system under various conditions, such as different trajectories, disturbances, or controller parameters. Through simulation, you can evaluate the performance of the control system and tune the parameters of the lead-lag compensator to achieve the desired response.
Implementation and Testing: Once satisfied with the simulation results, the control system can be implemented on the actual hardware of the robot. Testing would be conducted to validate the performance of the control system in real-world conditions and make any necessary adjustments.

3 Designing steps.

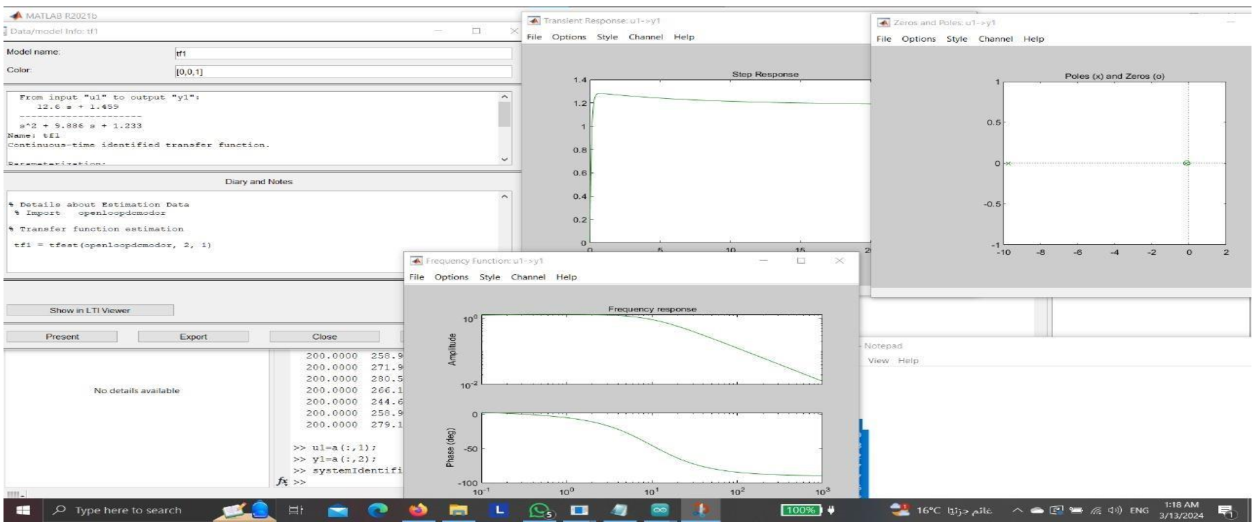
Data Collection:

- **Programming the Motor:** Wrote code in Arduino IDE to control the motor and generate pulses according to the desired specifications.
- **Serial Communication Tool:** Utilized a serial communication tool to establish communication between the Arduino board and the computer.
- **Excel Data Logging:** Employed the serial communication tool to log both input and output pulse data into an Excel sheet in real-time.
- **Data Preprocessing:** Conducted necessary preprocessing steps such as filtering and noise removal on the collected data to ensure its quality for further analysis.

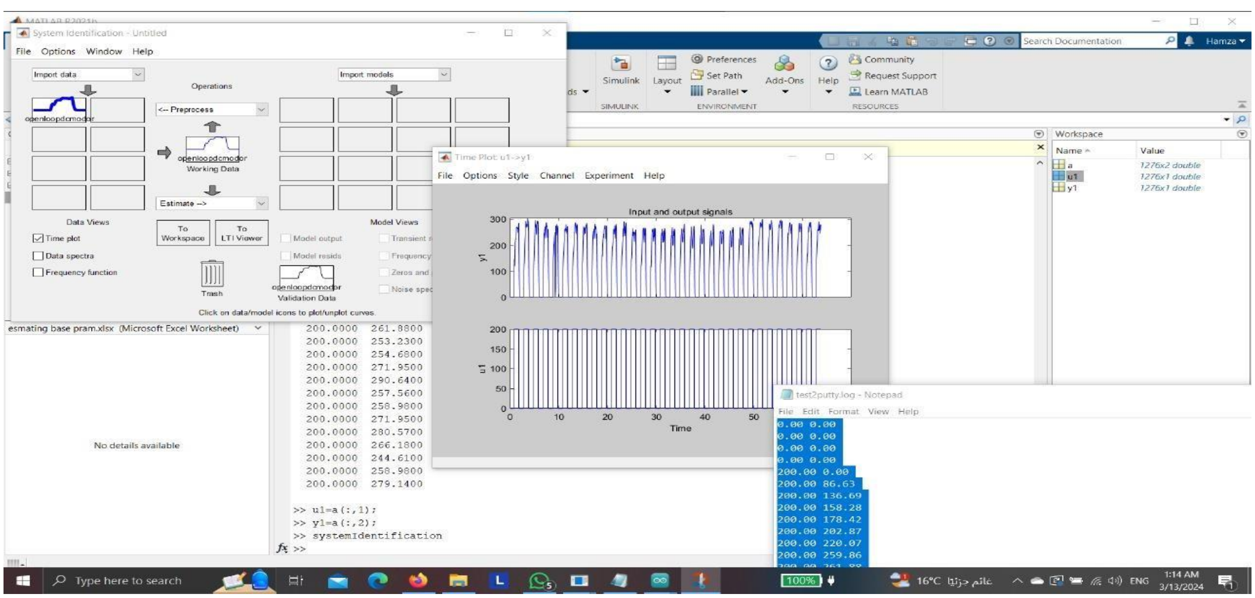
3.1 arduino ide sample collection

System Identification:

- **Data Analysis:** Imported the collected Excel data into a system identification tool, such as MATLAB's System Identification Toolbox.
- **Model Estimation:** Employed system identification techniques, such as time-domain or frequency-domain analysis, to estimate a mathematical model (transfer function) that represents the dynamic behavior of the motor system.
- **Model Validation:** Validated the identified model by comparing its response with the experimental data, ensuring its accuracy in representing the actual system dynamics.



3.2 system identification toolbox

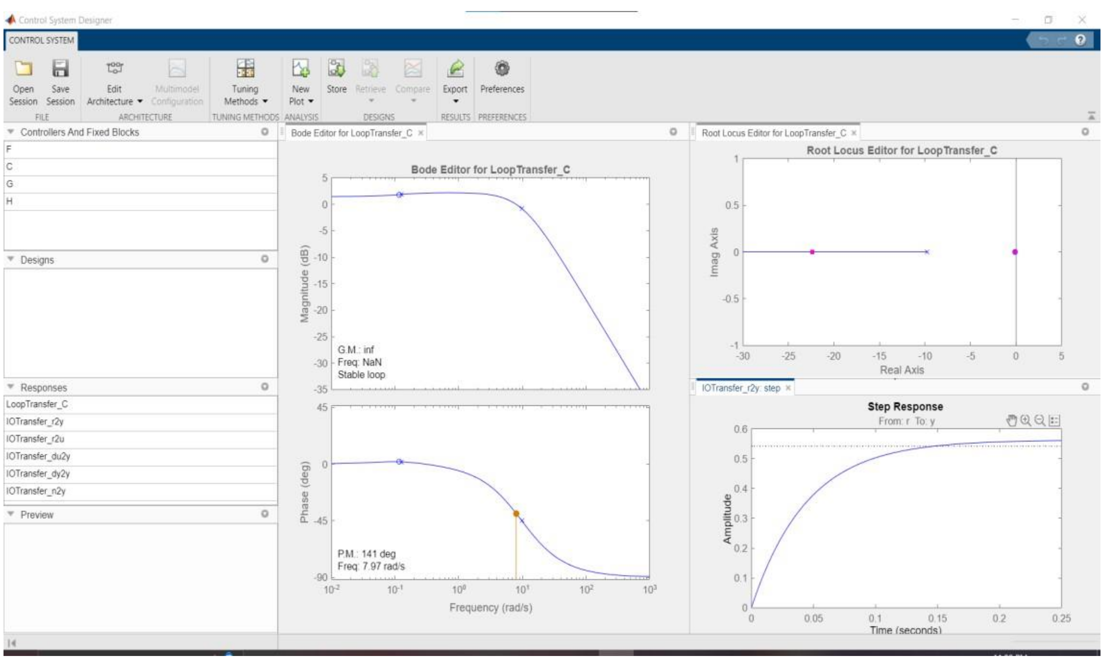


3.2 system identification toolbox

Lag Compensator Design:

- **SISO Tool Utilization:** Utilized MATLAB's SISO Design Tool for designing the lag compensator.
- **Controller Tuning:** Tuned the parameters of the lag compensator to meet the desired performance specifications, such as settling time, overshoot, and stability margins.
- **Stability Analysis:** Conducted stability analysis to ensure that the designed lag compensator stabilizes the system and does not introduce instability or oscillations.

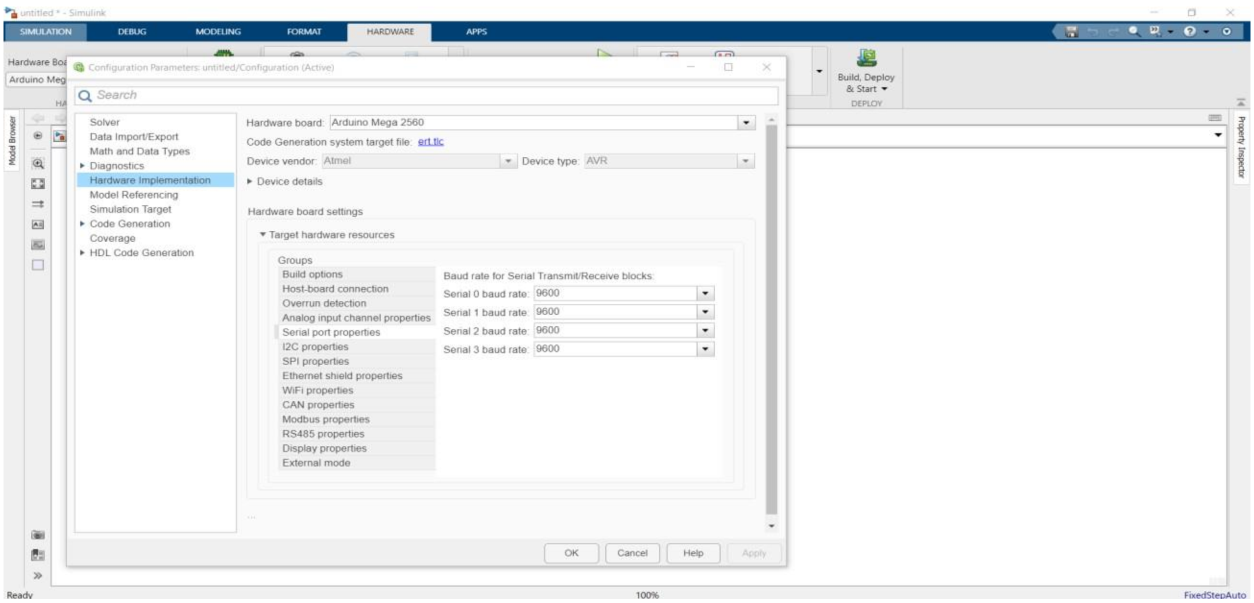
3.3 lag compensator design 1



3.4 lag compensator design 2

Implementation in Simulink:

- **Model Integration:** Transferred the designed lag compensator into a Simulink model representing the motor control system.
- **Arduino Toolbox Integration:** Integrated the Simulink model with the Arduino toolbox to establish communication between the Simulink model and the physical motor system.
- **Hardware-in-the-Loop (HIL) Simulation:** Conducted Hardware-in-the-Loop (HIL) simulations to verify the performance of the designed lag compensator in real-time hardware environment, setting the sampling time to 0.001 seconds also bud rate at 9600.



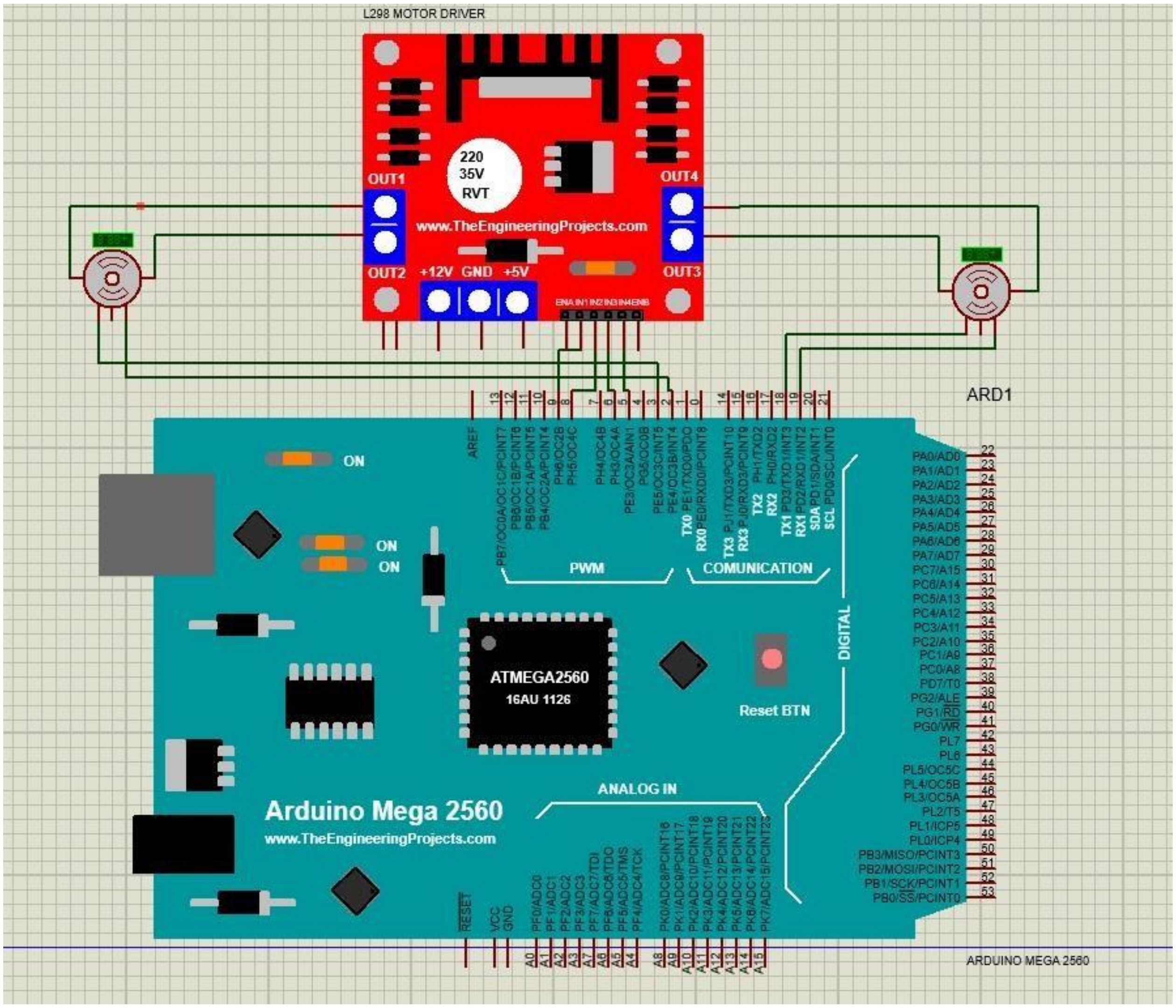
3.5 setting bud rate to 9600

Simulation and Validation:

- **Simulation Setup:** Configured simulation scenarios in Simulink to represent various operating conditions and disturbances that the motor system may encounter.
- **Performance Evaluation:** Conducted simulation experiments to evaluate the performance of the motor control system with the implemented lag compensator.
- **Comparative Analysis:** Compared the simulated responses with the desired performance criteria and experimental data to validate the effectiveness of the designed lag compensator in achieving the control objectives.

4 Implementation

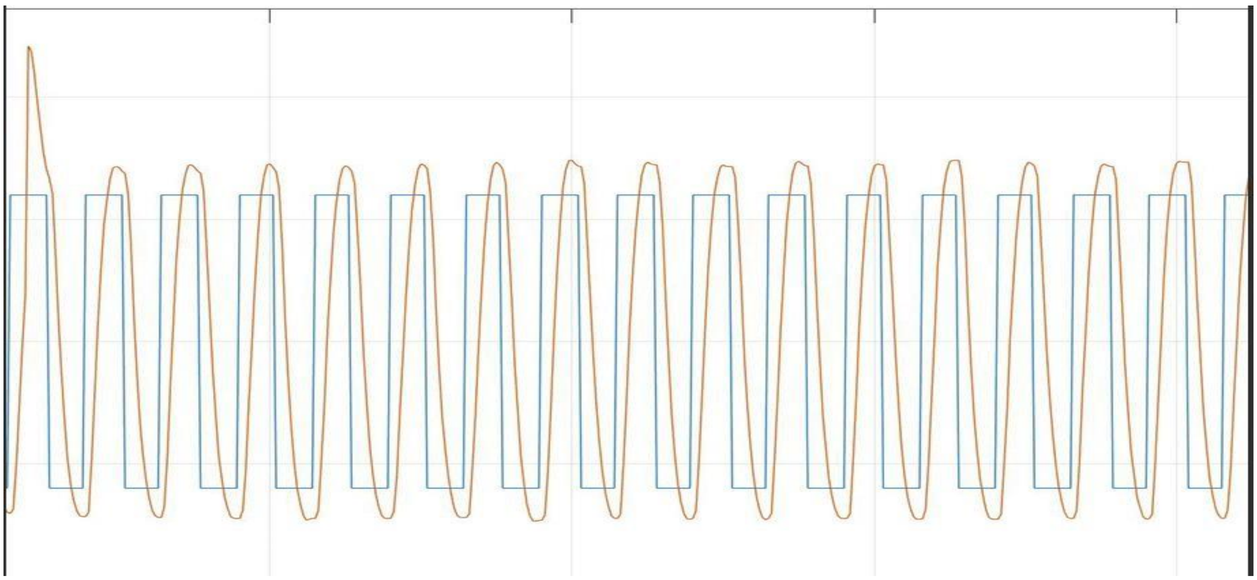
The circuit design is consisting of 2 dc motors with a built-in hall effect sensor encoder, moreover a dc motor driver of type l298n, and an Arduino mega is used as a data acquisition card to receive and transmit the data from the pc program (SIMULINK) via a wired serial communication. The circuit with designed and simulated by proteus software, more over the model was tested and simulated on simscape before implementation.



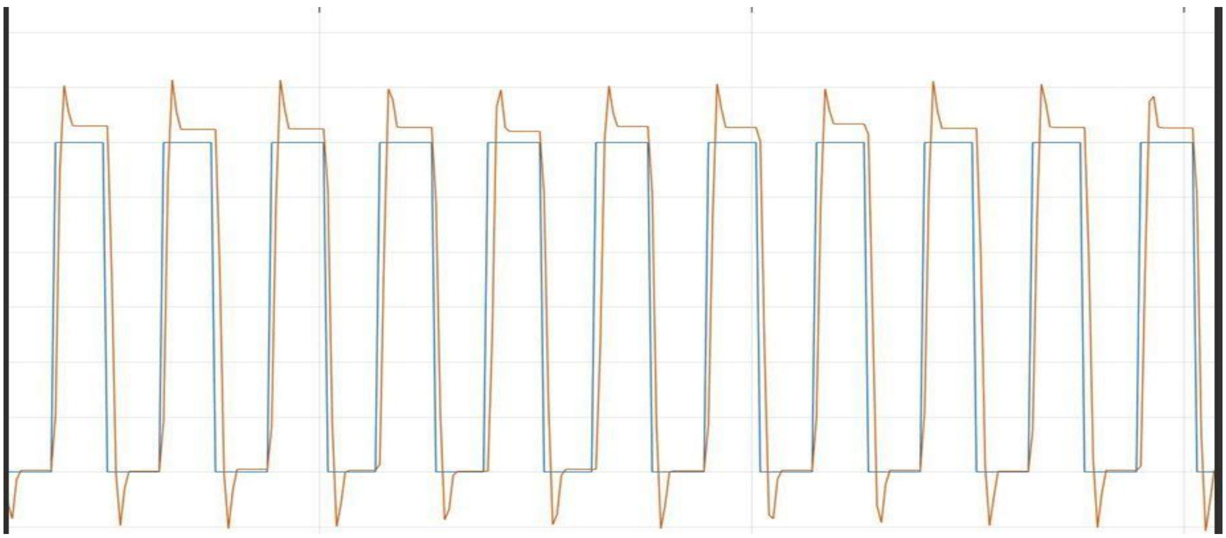
4 circuit design on proteus software

5 results.

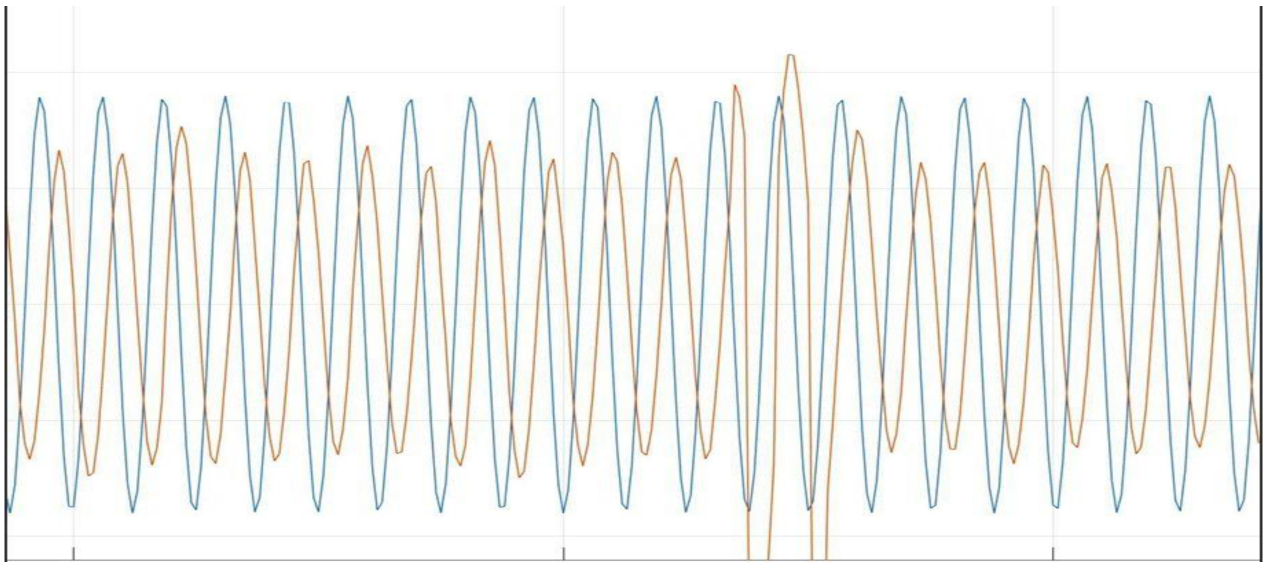
Without compensating end effector with square wave.



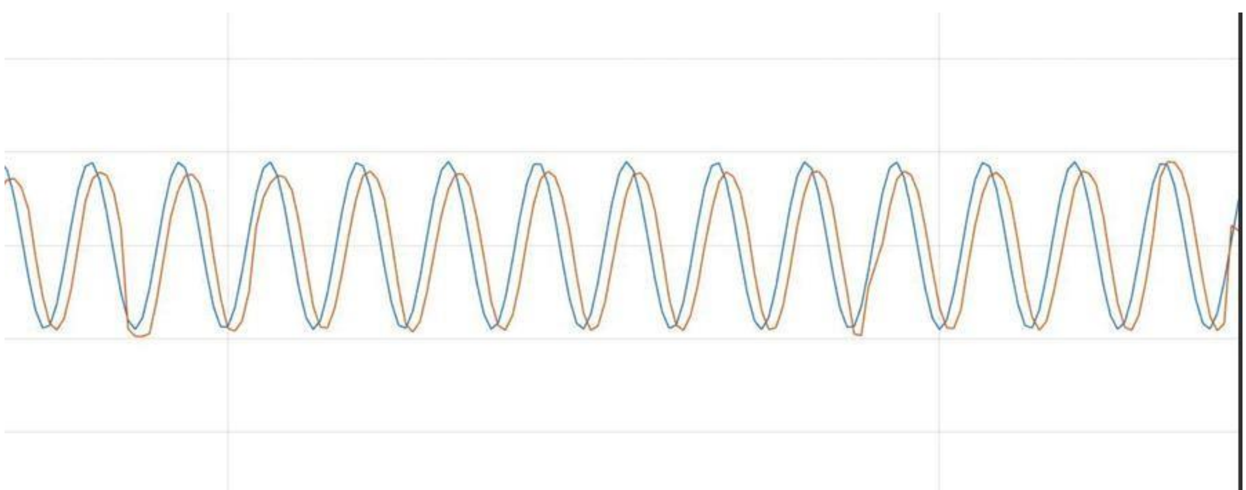
With compensating end effector with square wave.



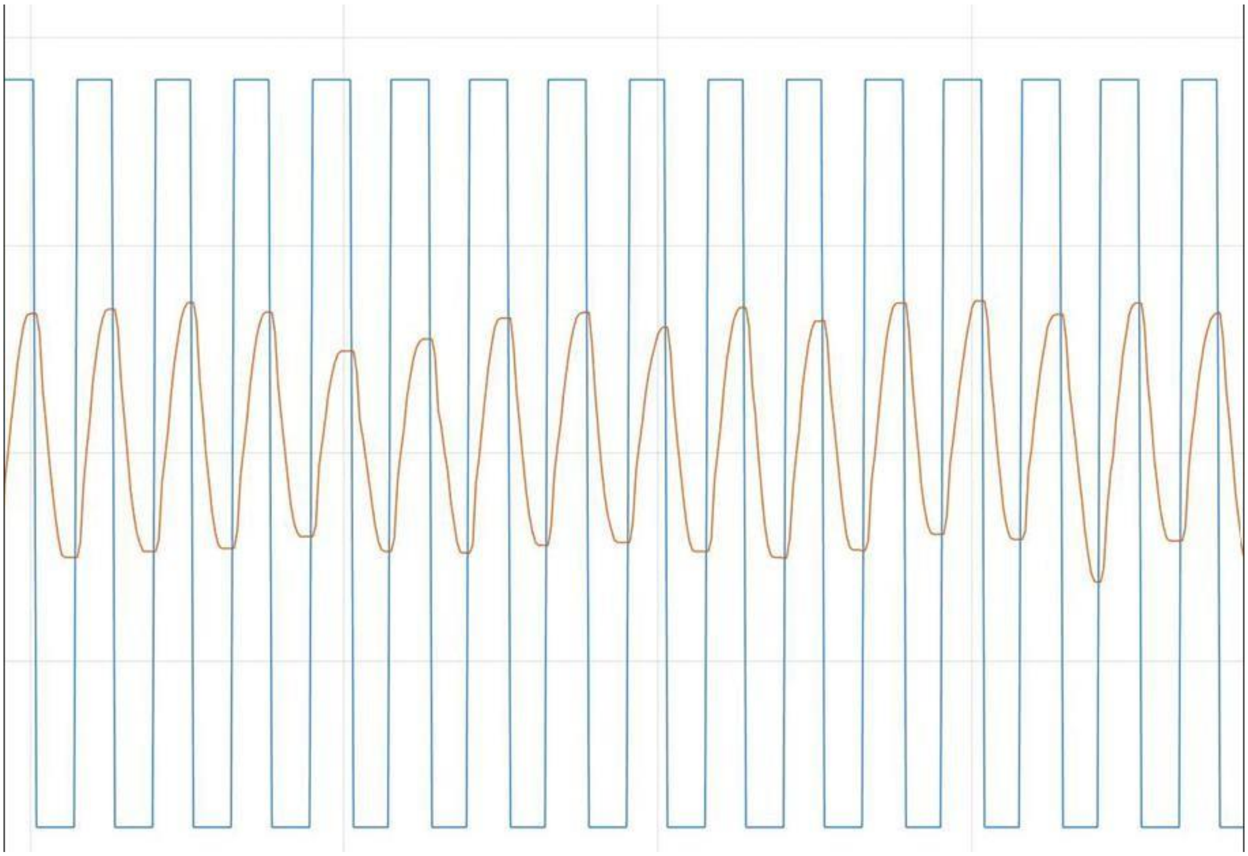
Without compensating end effector with sine wave.



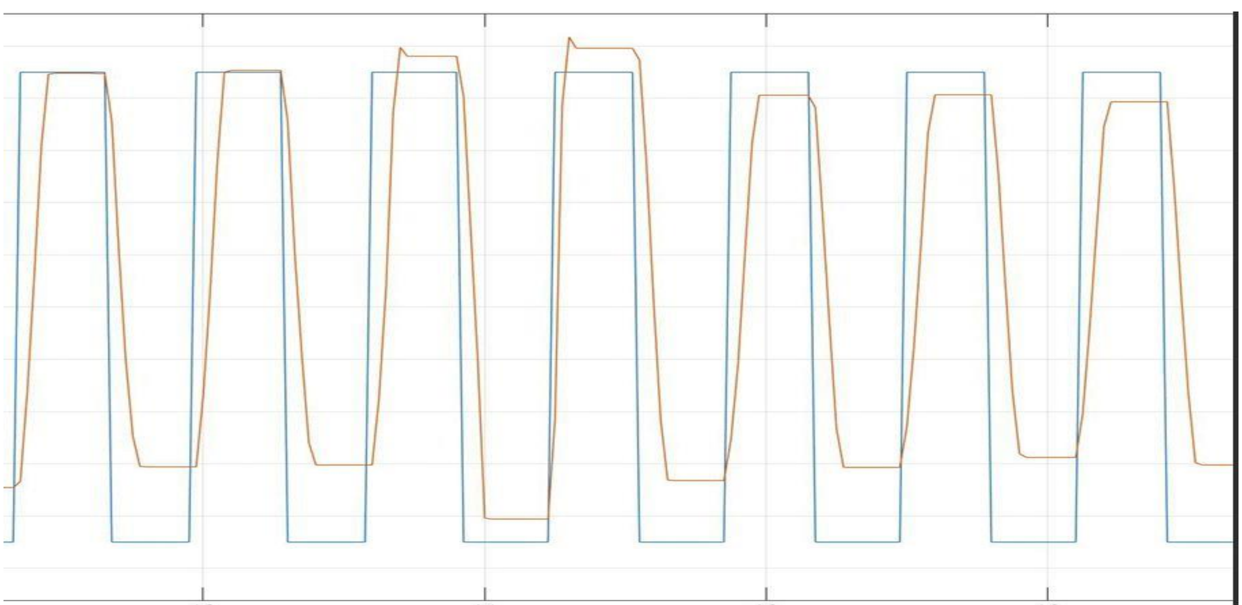
With compensating end effector with sine wave.



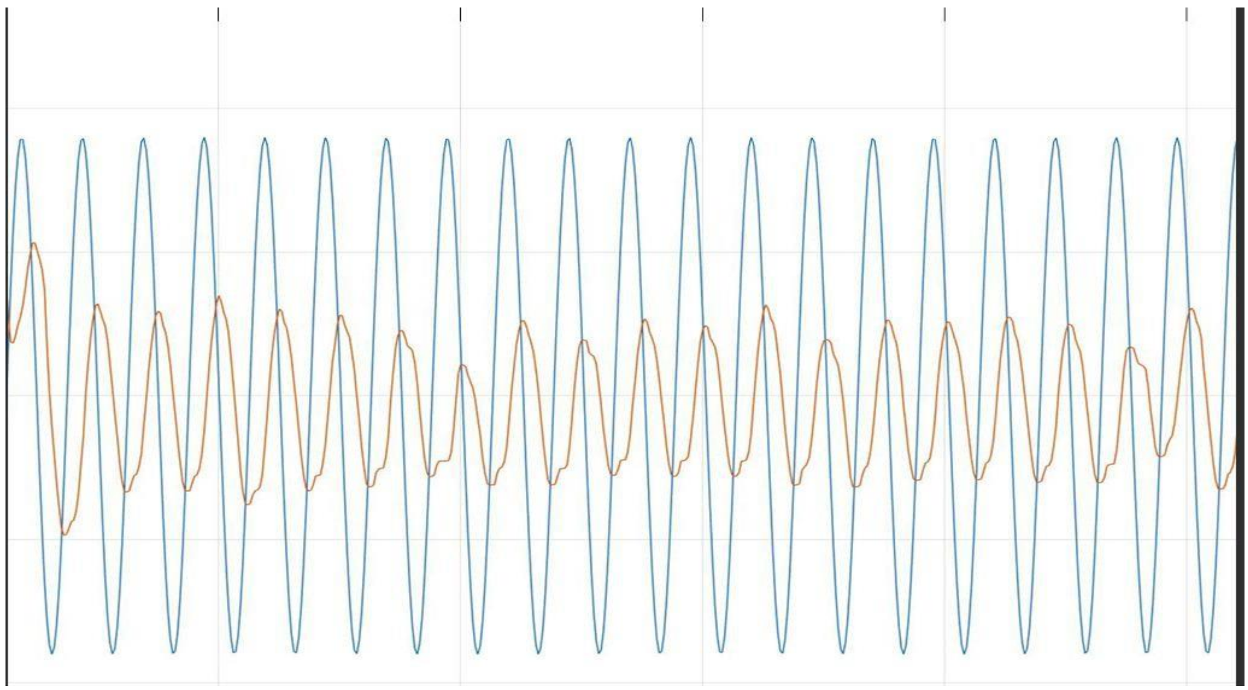
Without compensating base with square wave



With compensating base with square wave.



Without compensating base with sine wave.



With compensating base with sine wave.