Project
ME 5245 Mechatronic Systems

PART C - Speed Control

Author: Tirth Patel

1 Objective

The objective of this phase was to design and implement a closed-loop feedback control system for the DC motor to achieve Speed control objectives. This report details the transition from manually setting the PWM value for motor speed to potentiometer controlled motor speed system using a Proportional (P) controller architecture, the simulation of this system in Simscape, and the validation of results through hardware experimentation using Arduino and Simulink Real-Time.

2 Control System Design

In accordance with the project requirements, the control law was implemented using the proportional gain (K_p) acting on the error signal. The Simscape model has a modified control feedback loop showing the Error Calculation, Proportional Gain, and Saturation blocks replacing the PID Controller block.

- Control Law: $u(t) = K_p \cdot e(t)$
- Error Signal: $e(t) = Reference - Feedback$

Design Modifications done to replicate the physical constraints of the hardware in the simulation:

1. Gain Block: This gain value of 6, which is set with The Potentiometer voltage (0-5V) is mapped to a target speed (-0 RPM to +30 RPM).
2. Sum Block: This calculates the error signal.
3. Gain Block: This is the proportional gain that is set after iteratively testing the system response. This value of 0.2 gives a relatively perfect behavior.

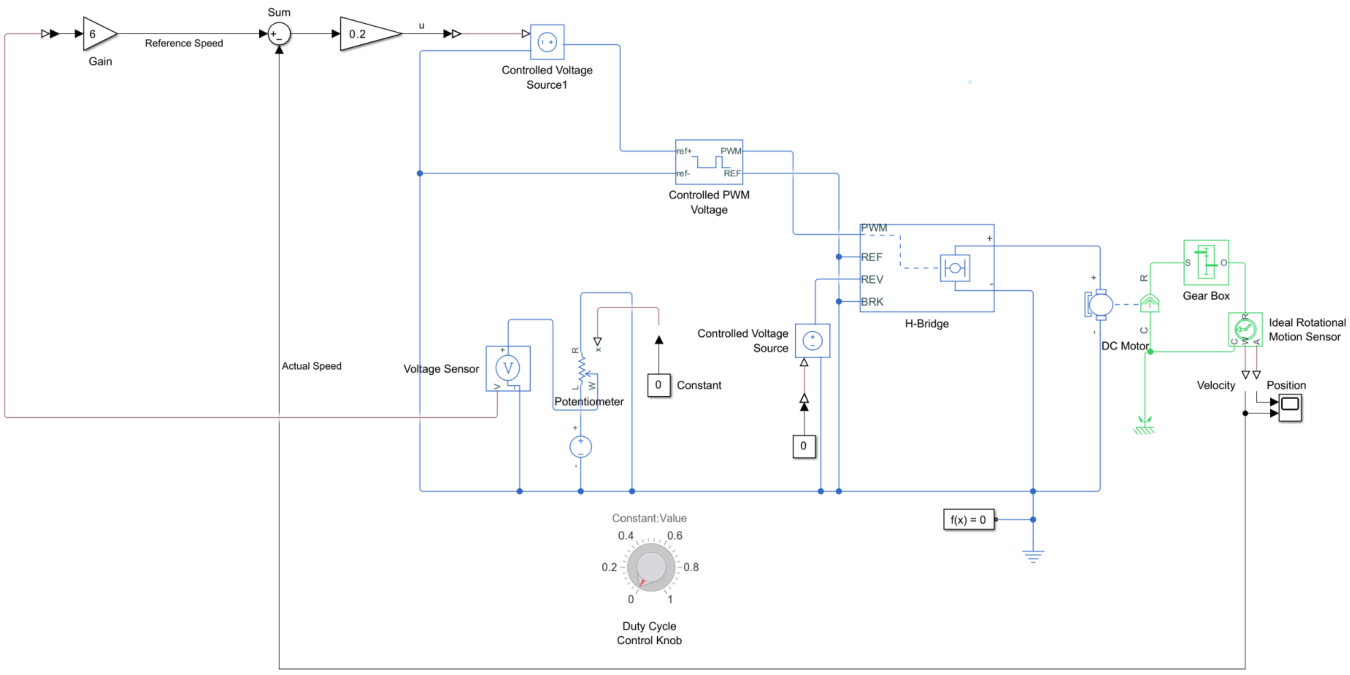

Figure 1: Simscape Model modified for control loop showing P control

3 Simulation Results (Simscape)

The system was simulated with a step input reference to observe the transient response and steady-state behavior.

3.1 Tuning Methodology

The proportional gain K_p was tuned iteratively.

- Low Gain(<0.2): Resulted in a sluggish response with significant steady-state error.
- High Gain(>0.2): Reduced steady-state error but introduced oscillation (overshoot).
- Final K_p Value: 0.2 was selected as a compromise between speed and stability.

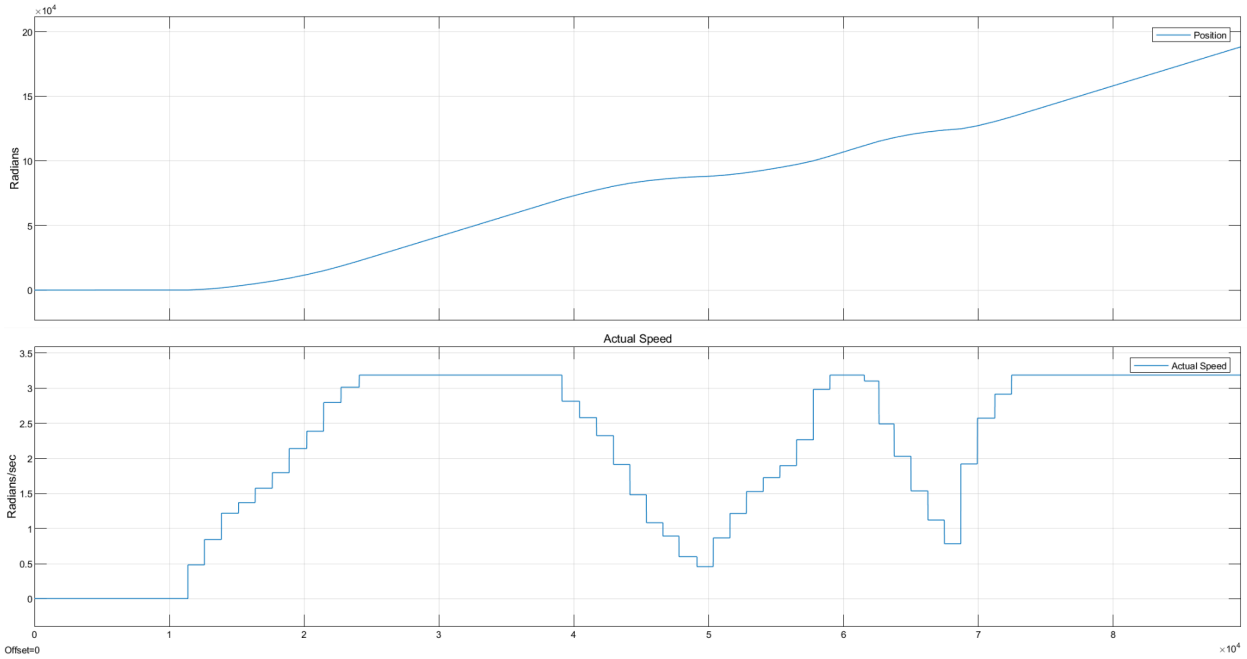

Figure 2: Simulation response showing Reference Signal vs. Measured Output at variable potentiometer reference

3.2 Analysis of Simulation

The motor reached 100% of the target value in 0.3 seconds at full value of potentiometer. It also showed controllable response at the full wiper rotation from 0-1 with the Simscape knob. At different values the motor speed changed accordingly as shown in the fig. 2, where lower speed maps to 0V and full resistance and 3.14 rad/sec (reference speed) at zero resistance.

4. Hardware Implementation (Simulink Model Design)

4.1 Experimental Setup

The hardware setup utilized the Arduino Uno, Pololu metal gearmotor with encoder, and L293D H-Bridge3. The control algorithm was deployed to the Arduino using Simulink Real-Time.

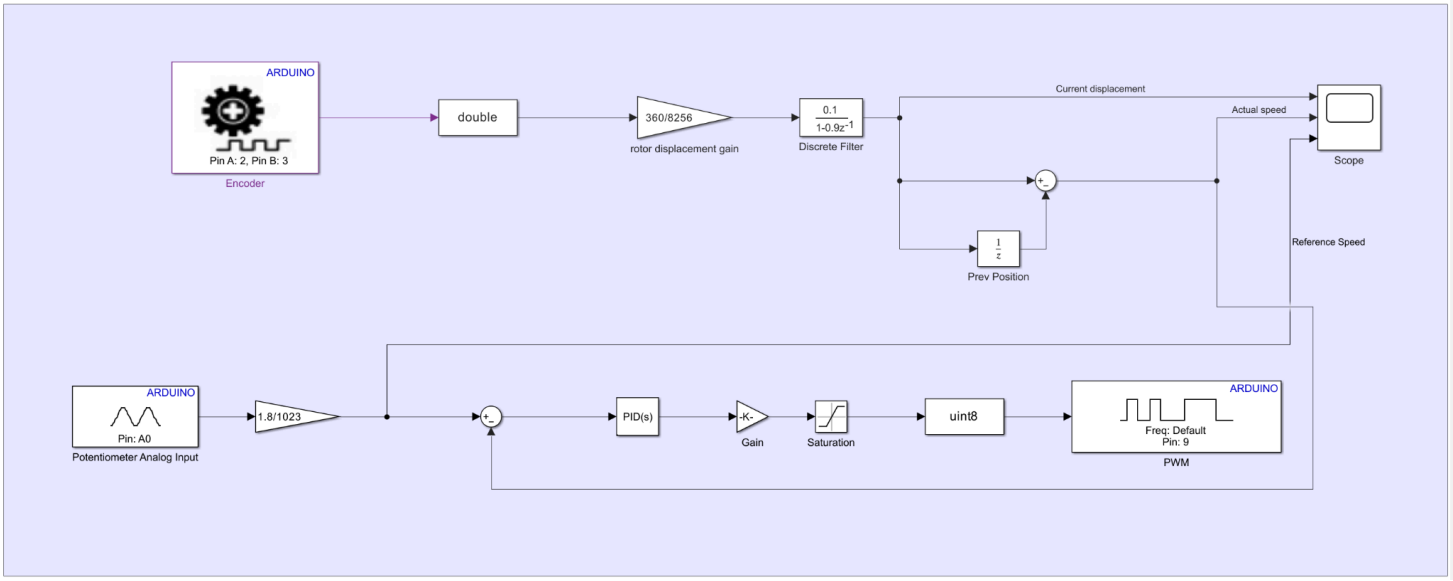

Figure 3: Simulink diagram including Motor,

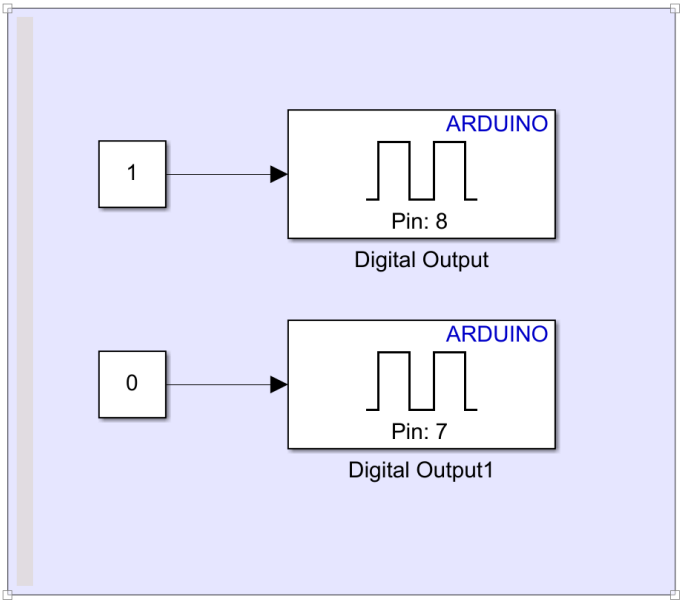

Figure 4: Digital set for motor direction

In the above Simulink setup the top section is the same as the previous part where it takes input from the motor encoder and calculates the distance and the velocity by storing the previous position and taking the derivative. The output of this is the actual speed that is used in the feedback loop to be subtracted from the reference speed and hence calculating the error.

The potentiometer analog input has a range (0-1023), this value from the same is multiplied by the gain (1.8/1023) where 1.8 rad/sec is the maximum shaft speed. This output is our reference speed. After the sum block, the output error is inserted into the PID block where the proportional gain Kp is set to 10 (which is calculated after iterative experimentation). Any value above 10 gives the same result and values below it gives a sluggish response to the potentiometer signal.

The PID signal output is multiplied by the gain value of 255 (maximum PWM value) because the max output value from the PID is 1 unit. Now, this value is passed through a saturation block where the value doesn't exceed (< 0 & > 255). This value is the absolute value that is fed through the PWM arduino input block to the motor.

4.2 Experimental Data

The following plot compares the commanded reference (potentiometer input) with the actual motor response (encoder feedback) along with the variable reference speed (0 - 1.8 rad/sec).

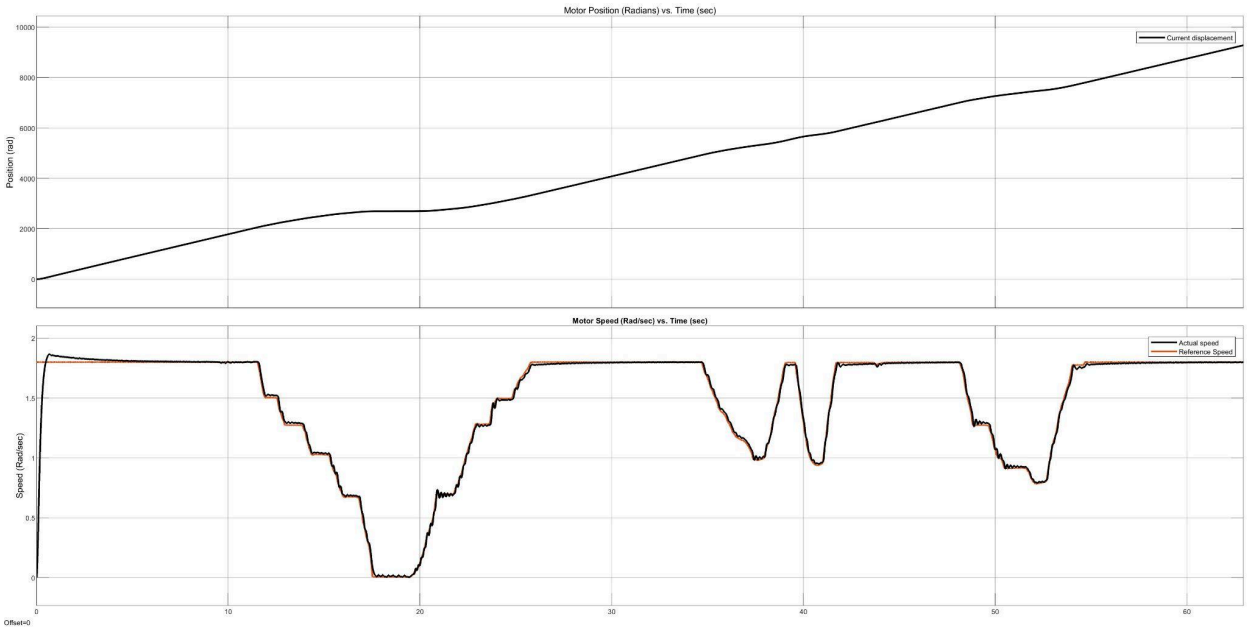

Figure 5: Real-time hardware response showing Reference vs. Actual Position/Speed.

5 Results & Analysis

5.1 Simulation vs. Hardware

The experimental results followed the general trend of the simulation but exhibited specific discrepancies:

- **Friction Effects:** The hardware demonstrated a larger "dead zone" than the simulation. The motor stopped short of the target once the error (and thus the PWM signal) dropped below the threshold required to overcome static friction.
- **Noise:** The encoder feedback signal contained quantization noise not present in the ideal simulation.

5.2 Evaluation of P-Control

Using only proportional gain achieved the basic tracking objective but highlighted the need for Integral action (to remove steady-state error) and Derivative action (to dampen oscillations at higher gains). The tracking error was minimized by increasing K_p, but was ultimately limited by the stability margin of the physical system.

The following table summarizes the key performance metrics and parameters observed in both the simscape simulation and the hardware experiment.

Metric / Parameter	Simscape Simulation	Hardware Experiment
Proportional Gain (K_p)	**0.2**	**10**
Steady-State Error	0 (Zero)	Small / Non-zero (Due to Dead Zone)
Friction Effects	None (Ideal Linear Model)	Significant (Static/Coulomb Friction)

Response Behavior	Smooth, ideal tracking	Slower start (inertia), minor noise
Control Signal Units	Ideal Voltage / Duty Cycle (0–1)	PWM Integer (0–255)
Overshoot	None (Overdamped/Critical)	Minimal (with tuned gain)

6 Conclusion

The Proportional controller successfully regulated the motor [Position/Speed], satisfying the Part C requirements. The exercise demonstrated the trade-offs inherent in P-only control: simplicity of implementation versus the inability to eliminate steady-state error entirely. The Simscape model served as a reasonably accurate predictor of performance, the hardware deviations largely attributable to unmodeled non-linear friction and voltage drops in the H-bridge.