Assignment 2 - MechEng 705

Ivan Santiago

May 2025

1 Task 1

1.1 Model Identification

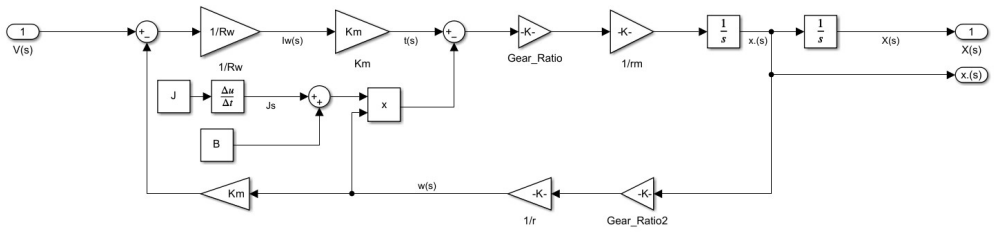

Figure 1: Plant Model

System Transfer Function

$$T_f(s) = \frac{3r_{\text{pulley}}K_m}{(R_w r_{\text{pulley}}^2 \text{mass} + 9JR_w)s + 9(R_w B + K_m^2)}$$

1.2 Design for Simulation

1.2.1 Pole Placement

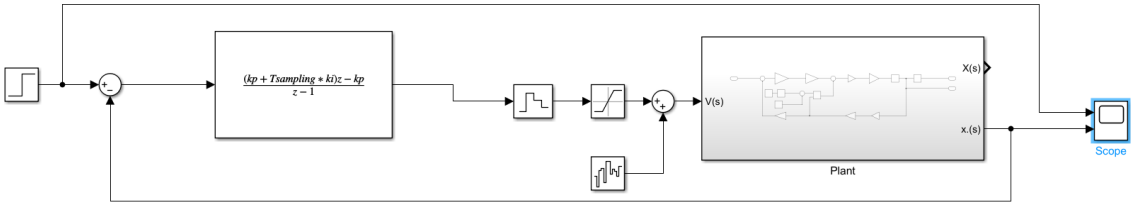

Figure 2: Pole Placing Controller

The following parameters were derived from pole placing.

- Kp = -10.5350
- Ki = 10.7404
- Tsampling = 0.0415 s

The controller parameters were selected based on a 5% overshoot and 2-second settling time. In the discretized system, poles were placed inside the unit circle, near the origin, to ensure fast, stable response without excessive control effort or noise sensitivity. The sampling frequency was chosen as 10 times the system bandwidth, following standard digital control guidelines. Noise as artificially added to better simulate a realistic model. Voltage is saturated between 0V and 32V, reflecting the motor's realistic operating limits.

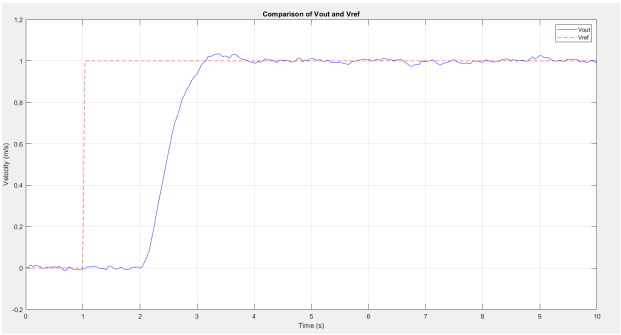

Figure 3: Pole Placement Control Output with Noise (1500g)

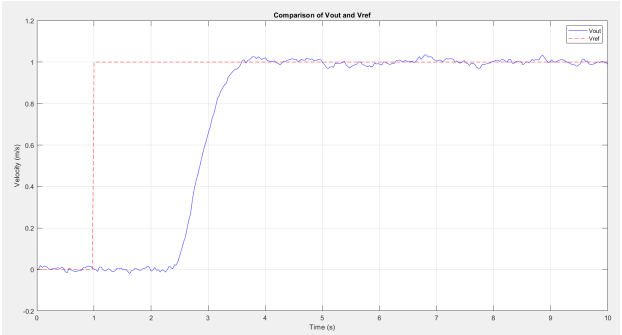

Figure 4: Pole Placement Control Output with Noise (1000g)

1.2.2 Deadbeat

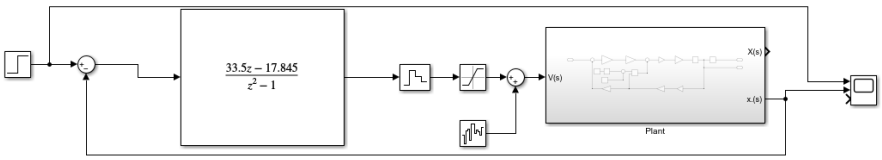

Figure 5: Ripple Free Deadbeat Controller

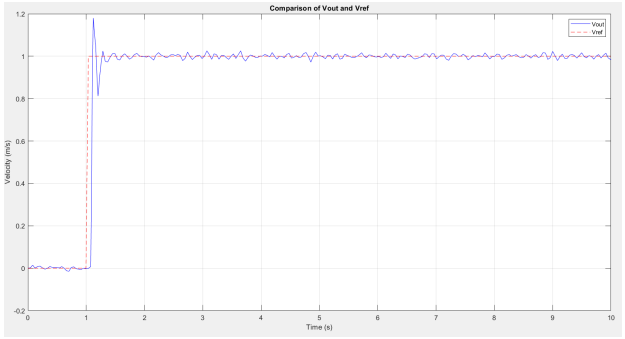

Figure 6: Deadbeat Controller Output with Noise (1500g)

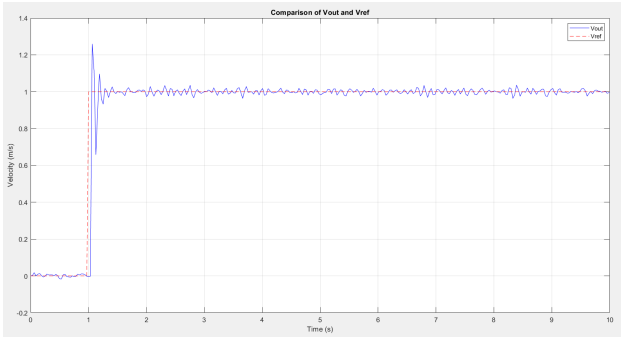

Figure 7: Deadbeat Controller Output with Noise (1000g)

The deadbeat controller achieves a faster rise time than the pole placement method but at the cost of increased overshoot and rippling. A reduction in system mass from 1500g to 1000g improves the pole-placement controller's response by slightly decreasing rise time and reducing overshoot, while degrading deadbeat controller performance through increased ripple and overshoot, likely due to plant-model mismatch and changes in motor inertia dynamics at the lower mass.

1.3 Implementation on the Realistic Model

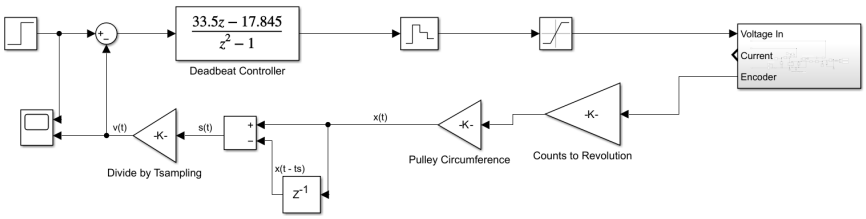

Figure 8: Deadbeat Controller on Real Model

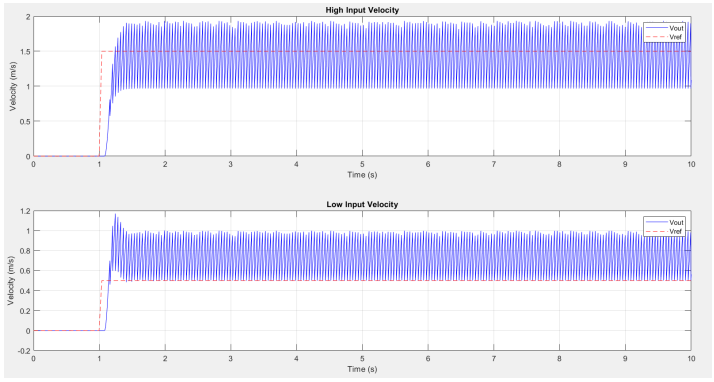

Figure 9: Output Plot (Velocity Comparison)

The realistic system shows much more oscillations in its output, which might suggest a marginally stable system, however it still essentially hovers around the target setpoint. As seen from the plot at low velocities the accuracy of the setpoint tracking decreases while at a relatively large velocity it remains fairly consistent. This output is roughly equivalent between using a deadbeat and pole placement controller and thus it was decided that only showing the deadbeat controller is sufficient.

Some uncertainties can lie within this model such as not accounting for friction of the system and potential external disturbances like noise,load changes and/or vibrations.

2 Task 2

2.1 System Simulation

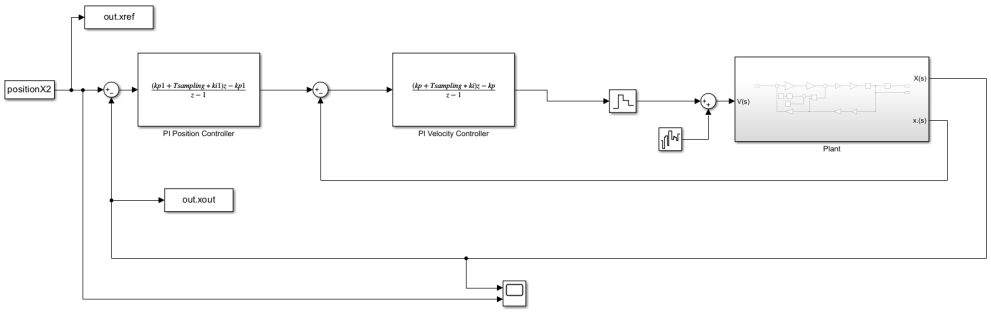

Figure 10: Pole Placement Simulink Model

In Figure 10, a Simulink model implementing the approximated plant is presented. The transfer functions for both pole placement and deadbeat control remain identical to those derived in the previous analysis. Position control is achieved by cascading a PI controller with the velocity control subsystem. The resulting outputs for both control strategies are compared in the subsequent analysis. The PI controller parameters were empirically tuned to $K_p = 10$ and $K_i = 0.1$ through iterative testing, providing optimal setpoint tracking performance for the position control system.

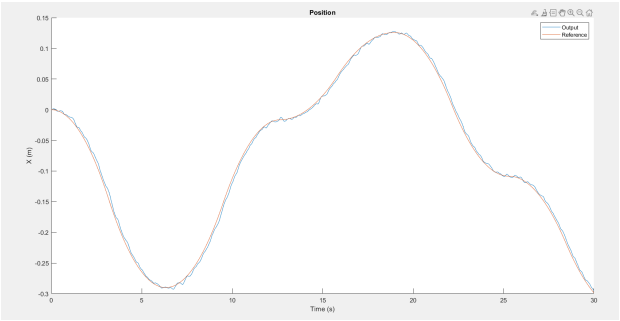

Figure 11: Pole placement position control (Trajectory 2)

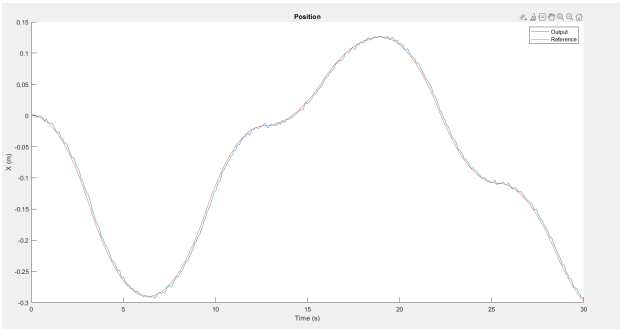

Figure 12: Deadbeat position control (Trajectory 2)

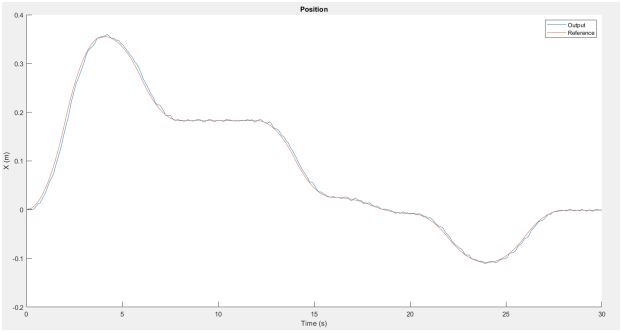

Figure 13: Pole placement position control (Trajectory 1)

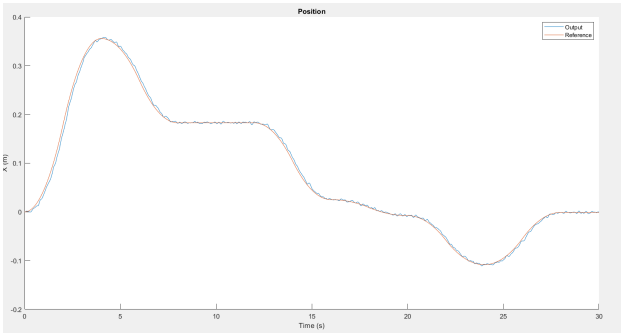

Figure 14: Deadbeat position control (Trajectory 1)

It is evident that both control strategies yield highly similar, if not identical, position responses. This confirms the validity of the design and demonstrates the robustness of both controllers.

2.2 Realistic Implementation

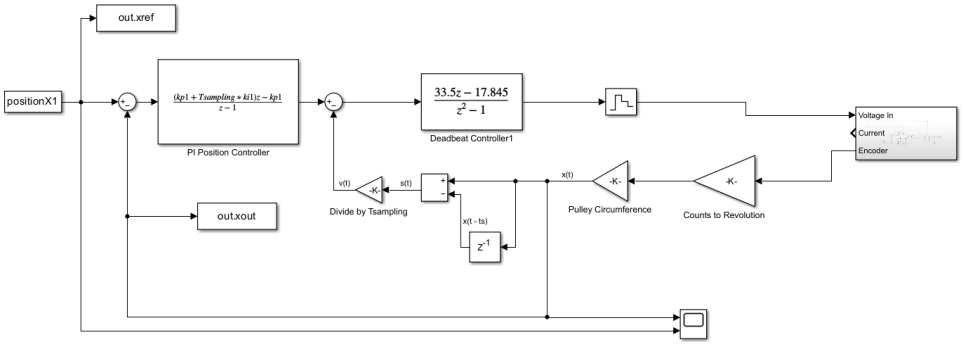

Figure 15: Realistic Position Control Deadbeat Controller

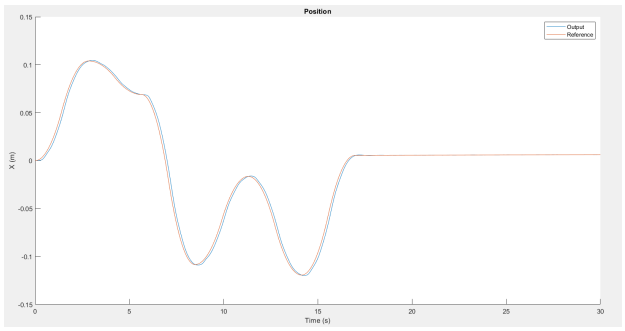

Figure 16: Pole Placement Realistic Position Control (Trajectory 3)

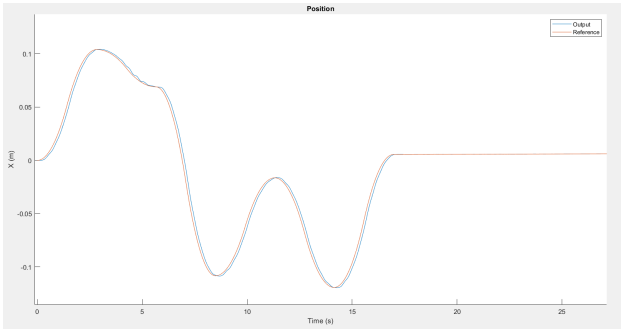

Figure 17: Deadbeat Realistic Position Control (Trajectory 3)

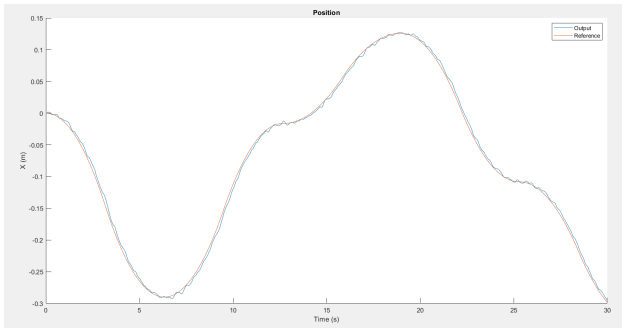

Figure 18: Pole Placement Realistic Position Control (Trajectory 2)

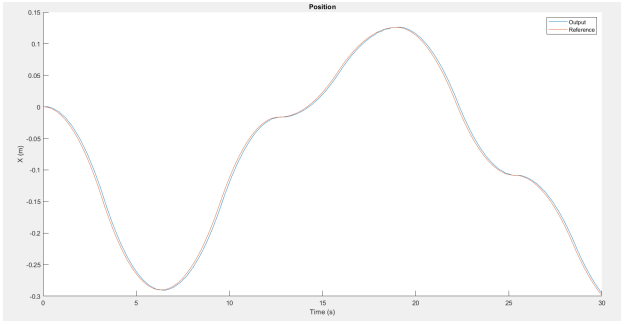

Figure 19: Deadbeat Realistic Position Control (Trajectory 2)

As seen in the output figures from above, using the realistic model gives very similar outputs for both types of controlers with good setpoint tracking and responsiveness. Compared to the simulation these real system outputs are very similar in that its accuracy is about the same. A cascaded PI structure was chosen for position and velocity control due to its simplicity in implementation, as it avoids complex calculations required by deadbeat or pole placement methods, relying instead on straightforward iterative tuning. Furthermore, a cascaded control system offers significantly improved disturbance rejection as opposed to a single control loop. This is because disturbances affect the system's velocity first, allowing the controller to correct the velocity before significant position deviation. Additionally, this design makes tuning a lot easier as we are able to tune the velocity control and thus position control sequentially.