**Proportional-Derivative Control of an Arm Model using OpenSim**

**CSCI 4620 - Computational Motor Control**

*Christian Mandujano Borjas*

**Assignment: #2**

Date: April 19, 2025

Abstract:

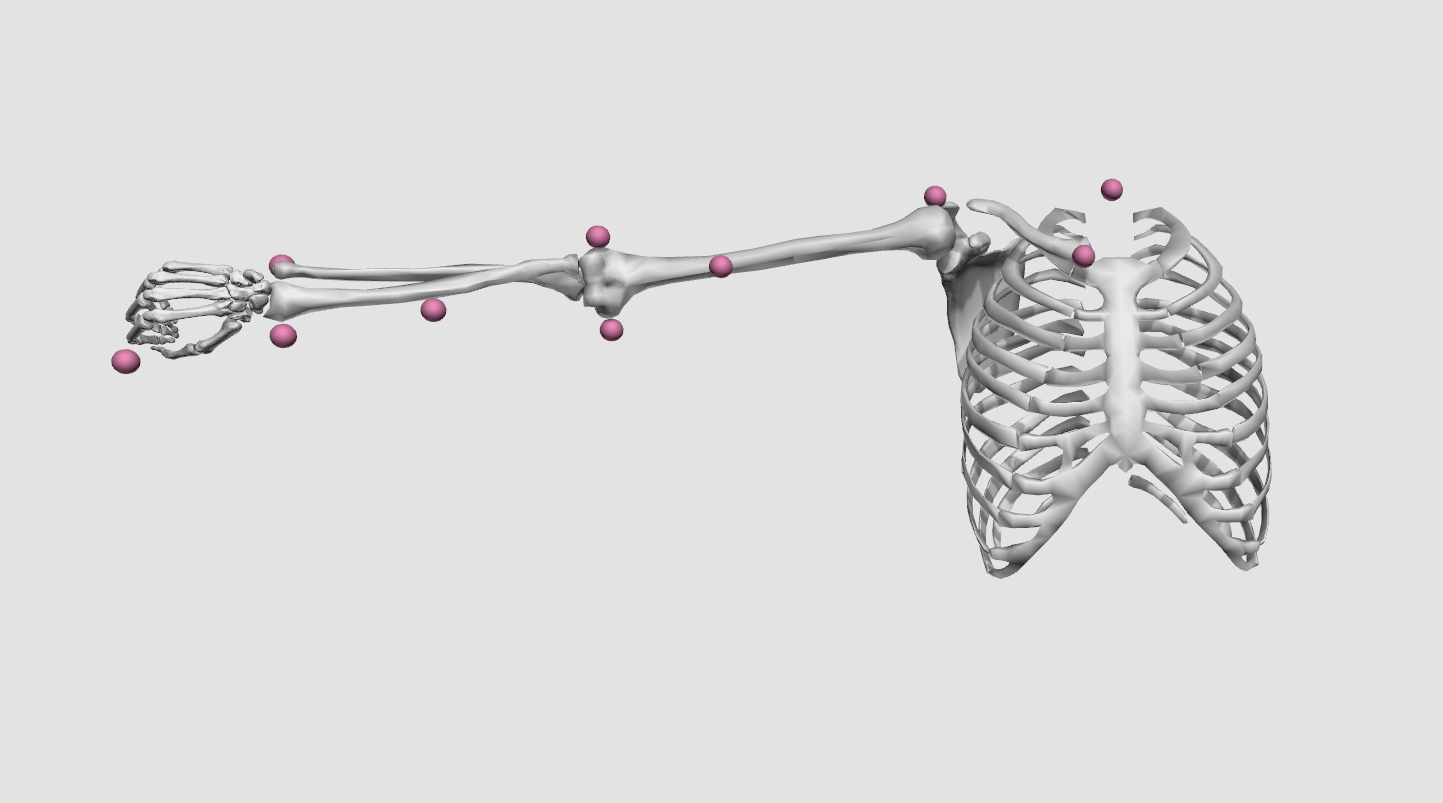
In this project I was set put to explore the application of Proportional-Derivative(PD) control. This control was used to simulate the arm movement using OpenSim. The simulation applied torque to the shoulder and the elbow joints to move the arm to the target position. Through the PD the research was able to affect overshoot, undershoot and response times.

In this assignment we were able to implement and analyze the PD controller that manipulates the “Shoulder\_elv” and the “elbow\_flexion”. This was able to move the coordinates of an OpenSim mode that was provided by the assignment. This assignment was able to allow the user to understand the biomechanical model through PD control. This allowed the user to evaluate the behavior of the arm under the different gain settings. Both of these allowed the user to visualize the resulting motion and assess the control quality.

To be able to assess the control quality we need to create an environment for the open sim python file to be compiled correctly. For the software to be able to compile correctly we can use OpenSim library, Python 3.X and matlotilb to be able to visualize the simulation. For the model we can use the zip file model from the assignment supplied files. Model that was used was Arm263.osim model. This mode was the right arm of human skeleton model. This was all created as the environment for the python script.

The script was modified with shoulder elevation and velocity to move the arm. For the elbow flex, the elbow velocity, we retrieved the coordinates, then we retrieved the value and the speed value. Respectively. From the range loop we added a function to set the value. In the mathematical for we can break it down to .

The τ is the control torque being applied. θ is the current joint angle. is the desired or target joint angle. is the angular velocity of the joint. it is the proportional gain, and is the derivate gain. This controls the PD controller by applying a torque that increases the angular position error. Adding a dampening through the velocity term to prevent overshooting and oscillation. This condensed to . In the code that was submitted we used a target\_angle variable to make sure that the arm extends correctly to a parallel the arm.



For the assignment we tested three sets of PD gains for the shoulder joint and observed the results as shown below:

| **Experiment** | **P Gain** | **D Gain** | **Expected Outcome** |
| --- | --- | --- | --- |
| A | 50 | 1 | Overshoot |
| B | 5 | 1 | Undershoot |
| C | 30 | 5 | Fast convergence |

The elbow joint remained constant for all tests (P = 15, D = 0.5).

With the A experiment we were able to have a high and the system was able to aggressively reach the target but with low damping (), not able to slow down fast enough. It caused the target to overshoot and oscillate before settling. With experiment B we were able to have a low and the system was not able to push the target to the target. It was not able to reach it fully or be too slow to reach on time. It would also settle too early causing undershooting. The C experiment was able to have a good balance and the system was able to dampen the oscillation. This result was a smooth and fast approach to the target with very little overshoot or undershoot.

In the experiment for A this trial, a high proportional gain was applied to the control system. As a result, the joint moved past the target position, exhibiting significant overshoot and oscillations before settling. This behavior highlights how excessive P gain can lead to instability in precision tasks.

A skeleton with multiple points

AI-generated content may be incorrect.

This experiment involved using a low proportional gain. The reduced gain was not sufficient to fully drive the arm to the desired target, resulting in noticeable undershoot. The movement stalled before reaching the goal, emphasizing the need for adequate gain to ensure complete and timely positioning.

A skeleton with pink dots

AI-generated content may be incorrect.

A carefully balanced set of proportional and derivative gains was used in this case. The result was a quick and accurate convergence to the target position, with minimal overshoot and no oscillations. This demonstrates the effectiveness of tuning PD gains to optimize both speed and stability in joint control.

A skeleton with a long arm

AI-generated content may be incorrect.

The results of this simulation confirm the expected behavior of a Proportional-Derivative (PD) controller. A high proportional gain (P) was observed to cause the system to respond quickly to errors in position but resulted in overshoot due to the lack of sufficient damping. This overshoot was typical in the results and reflected the aggressive nature of a high P gain attempting to eliminate error quickly. On the other hand, a low proportional gain led to a sluggish response that failed to bring the system fully to the target position, resulting in undershoot. The system did not have enough force and resistance. When the derivative gain (D) was increased, the results stability and less oscillations, it accounted for the rate of change in error and damped accordingly. These simulations demonstrated the importance of PD gains, balancing speed of convergence with system stability. Choosing gains involves trade-offs. Increasing responsiveness compromises stability, and vice versa.

This assignment successfully demonstrated the implementation of a PD controller in a simulation environment using OpenSim creator. Applying torques at the shoulder and elbow joint; the simulation was able to control human-like arm movements. The variation of PD gain values, the effects on dynamic response, such as overshooting, undershooting, and convergence speed, were clearly felt in the process results. This assignment reinforced theoretical concepts from control systems but also highlighted the importance of experimentation and simulation in computational motor control. The simulations were also less of a waiting experience and a more quick gratification for the simulation.