Report on Small Project: Trajectory Tracking in MuJoCo

1. Introduction

This report summarizes the work I conducted for a one-week assessment project, focusing on trajectory tracking in musculoskeletal models using MuJoCo. The goal was to implement forward dynamics trajectory tracking under different control methods, analyze the results, and reflect on possible improvements.

---

2. Problem Understanding

The main task was to design a control law that enables the simulated body to follow a given reference trajectory.

- I quickly realized that the provided reference trajectory was not always physically feasible. This meant that even with a perfect controller, the system might not be able to track it accurately.
- Therefore, I treated the problem as not only trajectory tracking, but also a retargeting challenge.
- My strategy was to start with a reasonable feedforward–feedback control scheme and then apply optimization techniques for parameter tuning.

---

3. Methods and Work Process

3.1 Computed Torque Control

I first implemented computed torque control using model-based inverse dynamics combined with PD feedback. While the method worked in principle, I found that it was highly sensitive to model accuracy and the quality of the reference trajectory.

3.2 Feedforward–Feedback Control (PD/PID)

Next, I developed a feedforward–feedback control structure. This combination improved robustness compared to pure computed torque control. However, I encountered significant difficulty in tuning the gains effectively.

3.3 SPSA Parameter Optimization

To address the tuning challenge, I introduced Simultaneous Perturbation Stochastic Approximation (SPSA). SPSA allowed me to search for better PID gains automatically. The results were promising but heavily dependent on the choice of initial parameters.

3.4 Multi-Start Global Optimization

To overcome the initialization problem, I tested a multi-start global optimization strategy. I generated multiple random initial points, simulated each, and applied SPSA to the best candidates. This method produced stronger results than single SPSA runs, though at the cost of longer simulation times.

3.5 Reinforcement Learning for Adaptive Gain Tuning

I also explored the idea of using reinforcement learning to adapt PID gains online. I defined a preliminary reward function, but I could not achieve a stable implementation within the limited time.

---

4. Results

- I successfully implemented computed torque control, feedforward–feedback control, SPSA optimization, and multi-start global optimization.

- The global optimization approach yielded the best control performance among the tested methods.
- However, I also observed instability during ground contacts and limitations due to the non-physical reference trajectory.
- Additional adjustments (e.g., contact parameters, gain scheduling) were attempted but remained at an exploratory stage.

---

5. Reflection and Insights

Through this project, I learned several important lessons:

1. Model-based feedforward is essential. For complex musculoskeletal systems, feedback alone is insufficient.
2. Model accuracy matters. Better physical modeling directly improves control quality.
3. Reference trajectories need to be realistic. Retargeting or generating feasible trajectories is as important as the controller design.
4. Optimization can help, but costs time. SPSA and multi-start searches improved performance, but computational demands became a bottleneck.

Looking ahead, I believe that improving model fidelity, designing robust reference trajectories, and exploring Model Predictive Control (MPC) would be valuable next steps.

---

6. Conclusion

Within one week, I progressed from learning the basics of Python and MuJoCo to implementing multiple control strategies, optimizing parameters, and reflecting on modeling and control challenges. Although not all attempts were fully successful (e.g., reinforcement learning and MPC), the project gave me both technical insights and confidence. I am eager to continue developing these ideas in future research.