ArmPy

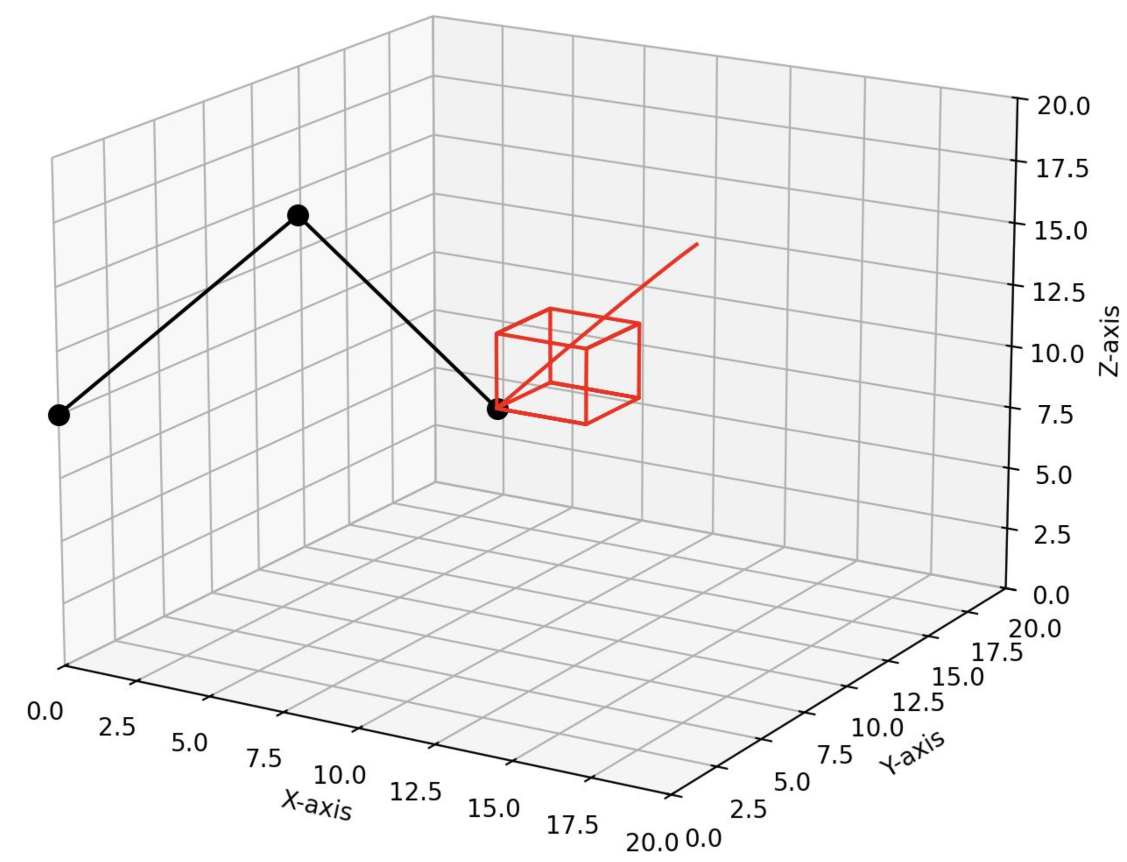
Open-source library for simulating and controlling multi-DOF robotic manipulators.

Boston University
Manuel Morteo

ArmPy Github and References

Purpose and Key Features of the Library

- **Scalability**: Supports robotic systems from 2-DOF to n-DOF with modular and expandable design.
- **Comprehensive Dynamics**: Simulate kinematics, inertia, Coriolis forces, and gravity for precise motion modeling.
- **Advanced Control**: Offers versatile strategies, including PID, log-scale PID, for tailored performance.
- **Educational Utility**: Serve as a teaching tool for control theory and multi-DOF robotics.
- **Research**: Facilitates advanced studies in dynamics, control, and collaborative multi-robot systems.
- **Visualization**: Real-time plotting and animation for debugging and performance analysis.



3 D.O.F Jacobian Animation

ArmPy vs Matlab

- **Open-Source and Cost-Effective:**
- **Scalability and Flexibility:** incorporating additional controllers and n-DOF configurations.
- **Educational and Research Utility**
- **Customization and Growth**
- **Community Collaboration**
- **ROS and Hardware Compatible**

Dynamics Modeling

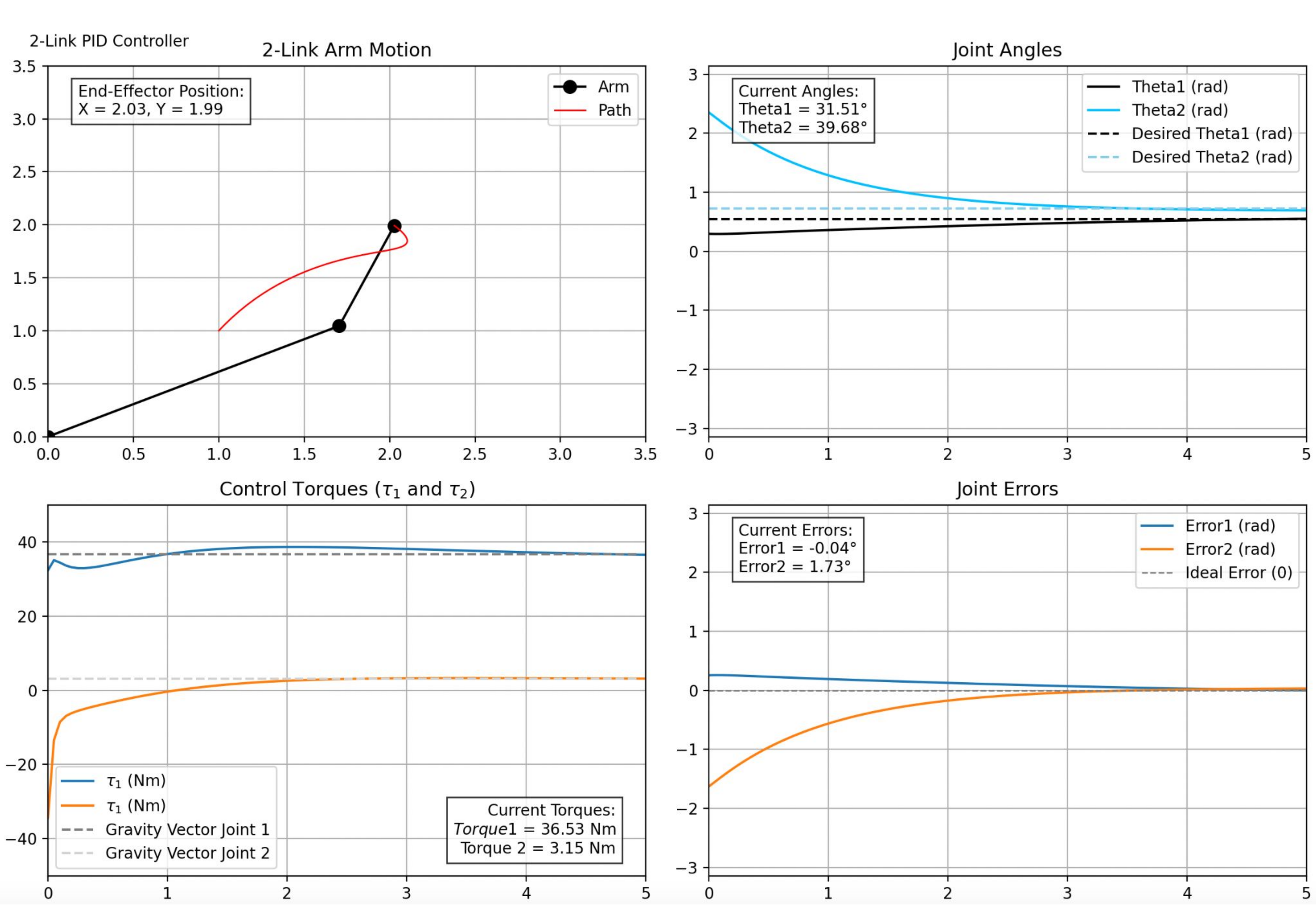
General Dynamics Equation for n-link manipulator:

$$F = M(\Theta)\ddot{\Theta} + C(\Theta, \dot{\Theta}) + G(\Theta)$$

- Models the full motion dynamics of the system, including inertia, Coriolis/centrifugal effects, and gravity.
- **Control**: Enables torque control, critical for smooth and robust motion under varying conditions.
- **Simulation:** Forms the foundation for simulating dynamics and designing controllers (e.g., PID, log-scale PID).
- **Optimization**: Improves precision and minimizes energy consumption.

Simulation and Example Results

PID Controller Simulation

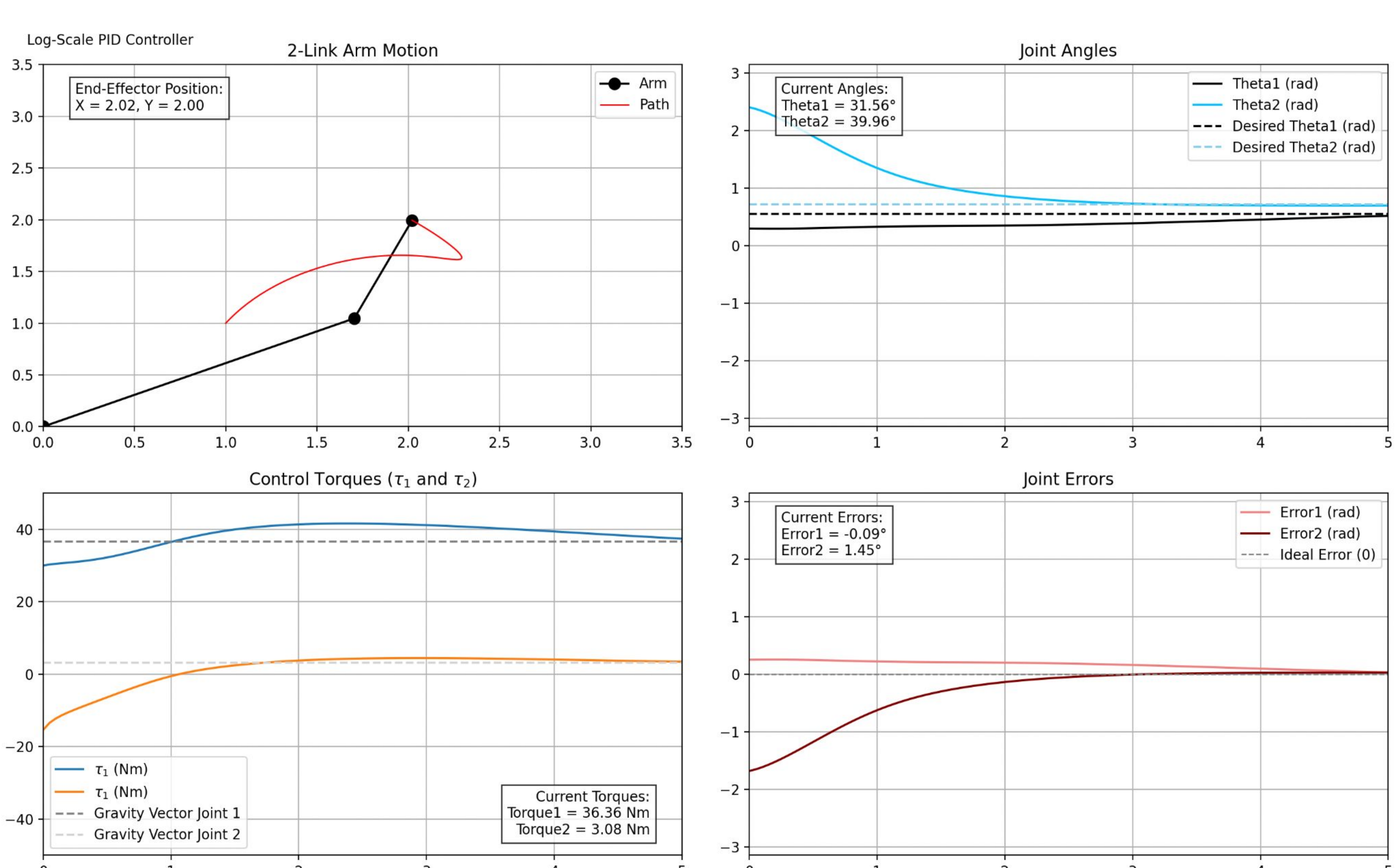


Dynamic Control with a PID Controller

$$K_p \cdot e(t) + K_i \cdot \int e(t)\,dt + K_d \cdot \frac{de(t)}{dt}$$

- Energy Consumption is not a priority.
- Standard Torque Adjustments.
- Works well for general robotic operations.
- Realistic approach compared to using the Jacobian.

Log-Scale PID Controller Simulation



Dynamic Control with a Log Based PID Controller

$$K_p \cdot \log_b(|e|+1) \cdot \mathrm{sgn}(e) + K_i \cdot \int \log_b(|e|+1) \cdot \mathrm{sgn}(e)\,dt + K_d \cdot \log_b\left(\left|\frac{de}{dt}\right| + 1\right) \cdot \mathrm{sgn}\left(\frac{de}{dt}\right)$$

- Reduces Aggressive Corrections.
- Energy-efficient; avoids torque spikes and improves stability.
- Provides smooth motion by minimizing oscillations or overshoot in joint movements.
- Reduces mechanical stress on actuators, minimizing wear and tear.