June 22, 2025

Project Overview

This project focuses on achieving accurate motion control of the manipulators, with support for challenging trajectory patterns such as circular, spiral, and fourier-based motions.

To ensure precise trajectory tracking, a **Computed Torque Controller (CTC)** integrated with a **PD feedback** algorithm has been implemented using ROS Noetic, Gazebo, RViz, and Pinocchio.

UR5e Robot Schematic & Overview

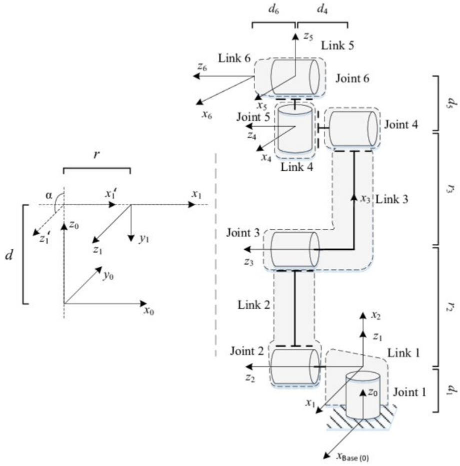
The DH parameters[a] of the UR5e:

Joint	θ [°]	θ [rad]	a [m]	d [m]	α [rad]
J1	302	5.2709	0	0.1625	1.5708
J2	190	3.3161	-0.425	0	0
J3	59	1.0297	-0.3922	0	0
J4	199	3.4732	0	0.1333	1.5708
J5	120	2.0944	0	0.0997	-1.5708
J6	90	1.5708	0	0.0996	0

Where:

- a_i: distance from z_{i-1} to z_i along x_i
- d_i: offset from x_{i-1} to x_i along z_{i-1}
- α_i: angle between z_{i-1} and z_i around x_i
- θ_i: angle between x_{i-1} and x_i around z_{i-1}

[a] UR DH Reference



Schematic and frames assignment of UR5e[a]

[a] Universal Robots Company

Dynamic Model

The Lagrangian L is defined as the difference between the total kinetic energy K and the total potential energy U of the system:

$$L = K - U \tag{1}$$

The dynamic model of the 6-DOF robotic manipulator is derived using the Euler–Lagrange formulation.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = \tau_j, \quad j = 1, \dots, 6 \tag{2}$$

Equations 2 can be compactly written in matrix form as:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) \tag{3}$$

where:

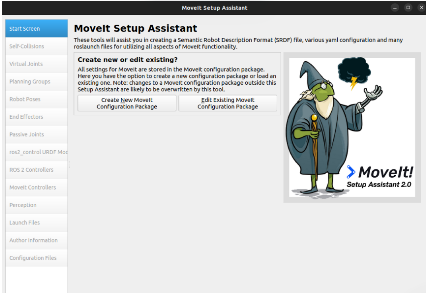
- $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^6$ are the joint position, velocity, and acceleration vectors.
- $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ is the symmetric, positive-definite inertia matrix.
- $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$ is the Coriolis and centrifugal matrix.
- $\boldsymbol{G}(\boldsymbol{q}) \in \mathbb{R}^6$ is the gravity vector.

URDF Model

In simulation environments, robot models are defined using **URDF** or **XACRO** files[a]. These files contain:

- Joint definitions
- Link dimensions
- Inertial parameters
- Actuator specifications

Although manufacturers provide standard files, we enhanced them using the **MoveIt Setup Assistant**[b] for improved motion planning and simulation accuracy.



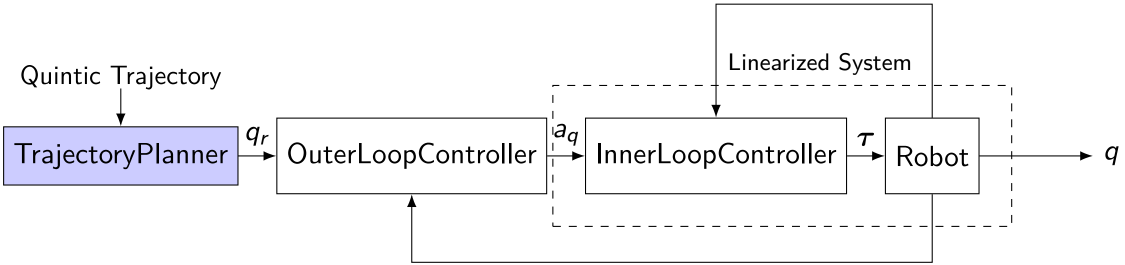
[a] Universal Robots GitHub
[b] MoveIt Setup Assistant

Computed Torque Controller + PD Feedback

Theoretically, the Computed Torque Controller with PD feedback is expressed as:

$$\boldsymbol{\tau} = M(q)\, a_q + C(q, \dot{q})\, \dot{q} + g(q) \tag{4}$$

And The computer architecture for performing these computations is shown below:



Computed Torque Controller + PD Feedback

In simulation environment,joint torques are computed using the Recursive Newton-Euler Algorithm (RNEA), implemented via the **Pinocchio** library[1]:

$$\tau = \text{rnea}(q, \dot{q}, a_{\text{ref}}) \tag{5}$$

The reference acceleration is defined as:

$$a_{\text{ref}} = a_{\text{desired}} - \mathbf{K}_p e - \mathbf{K}_d \dot{e} \tag{6}$$

where:

- a_{desired}: feedforward acceleration from the trajectory planner
- $e = q - q_{\text{des}}$: position error
- $\dot{e} = \dot{q} - \dot{q}_{\text{des}}$: velocity error
- $\mathbf{K}_p, thiswork, weusethe Exponential\ Moving\ Average\ (EMA) filter, whichwefindmoreeffectivethanothermeth$ proportional and derivative gain matrices

To ensure safety, the resulting torques are saturated to a maximum of **100 Nm**.

[1] Pinocchio Home Page

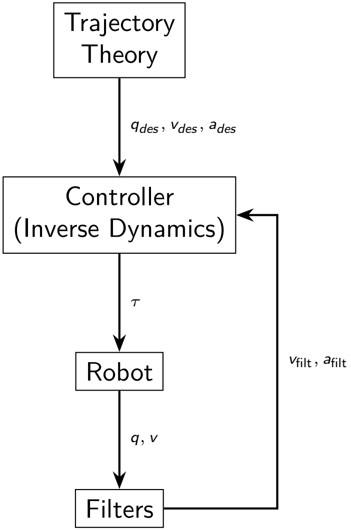
Joint State Filtering

Joint encoder signals are noisy. We apply exponential filtering to improve state estimation:

$$\text{Velocity:} \quad v_{\text{filt}}[k] = \alpha_v \, v_{\text{raw}}[k] + (1 - \alpha_v) \, v_{\text{filt}}[k-1] \tag{7}$$

$$\text{Acceleration:} \quad a_{\text{meas}}[k] = \frac{v_{\text{filt}}[k] - v_{\text{filt}}[k-1]}{\Delta t} \tag{8}$$

$$\text{Filtered Acc.:} \quad a_{\text{filt}}[k] = \alpha_a \, a_{\text{meas}}[k] + (1 - \alpha_a) \, a_{\text{filt}}[k-1] \tag{9}$$



Joint position tracking

This section presents the performance results of the controller through a plot illustrating position over time:

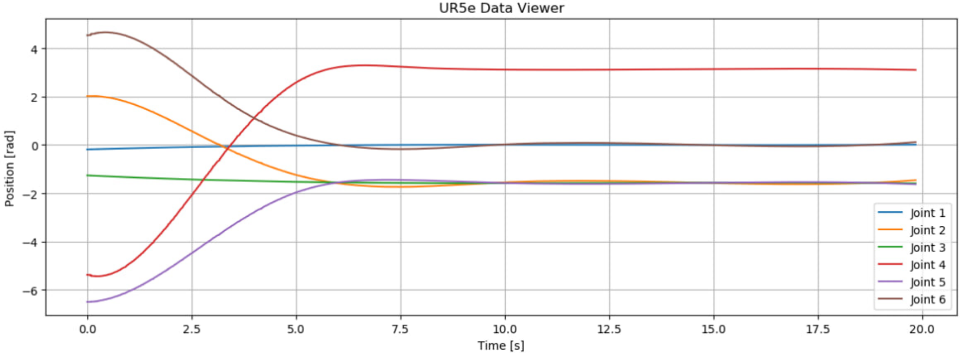


Figure: Joint position tracking over time

Circular Motion

To generate a circular end-effector trajectory in Cartesian space, we first introduce a pre-motion phase that employs Damped Least Squares (DLS) inverse kinematics. In this phase, desired joint positions are computed at each time step to track the planned Cartesian path accurately, ensuring smooth and feasible motion initiation before applying torque-based control.

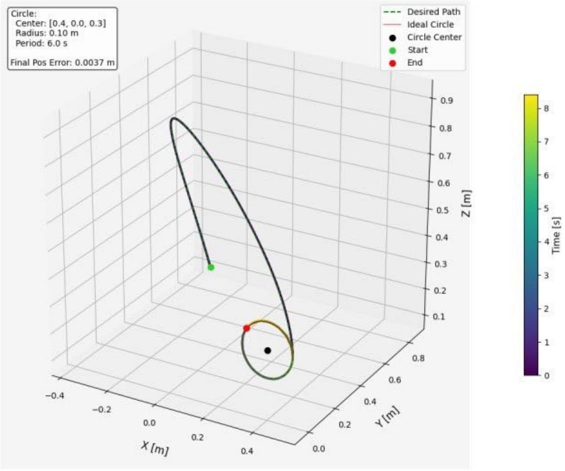


Figure: circular end-effector trajectory in Cartesian space

Spiral Motion

By using the same Damped Least Squares (DLS) inverse kinematics method, we extend the circular trajectory to a spiral motion by increasing the Z-coordinate linearly with each revolution. This generates smooth joint references that guide the end-effector along a helical path in Cartesian space during the pre-motion phase.

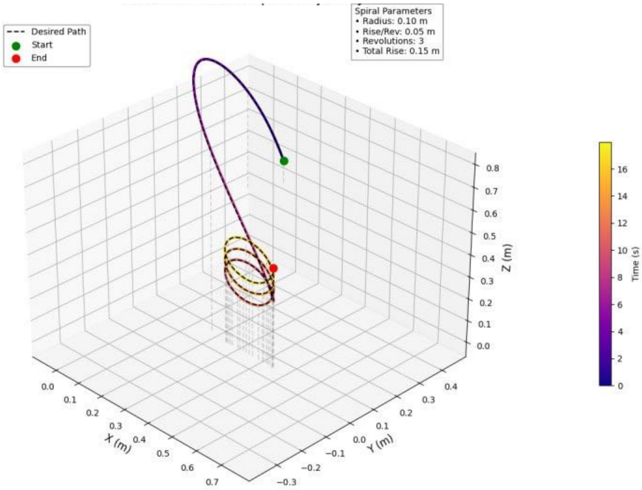


Figure: Spiral end-effector trajectory

Finite Fourier Series

A bandlimited periodic excitation signal can be represented using a finite Fourier series, which is a sum of sine and cosine terms. The excitation trajectory $q_i(t)$ is defined as : [2]

$$q_i(t) = q_{i,0} + \sum_{k=1}^{N} \left(a_{i,k}\cos(k\omega_f t) + b_{i,k}\sin(k\omega_f t)\right), \tag{10}$$

Where N is the number of harmonics, and ω_f is the fundamental frequency. For all following cases, $N = 5$ & $\omega_f = 2\pi f$, where f $= 0.1$ Hz.

[2] This test is based on the methodology presented in a paper published by IEEE.

Finite Fourier Series – Full

$$a_{5,k} = [0.2,\ 0.1,\ 0.5,\ 0.2,\ 0.1], \quad b_{5,k} = [0.1,\ 0.5,\ 0.2,\ 0.1,\ 0.5]$$
$$a_{6,k} = [0.1,\ 0.5,\ 0.3,\ 0.2,\ 0.1], \quad b_{6,k} = [0.5,\ 0.2,\ 0.1,\ 0.5,\ 0.2]$$

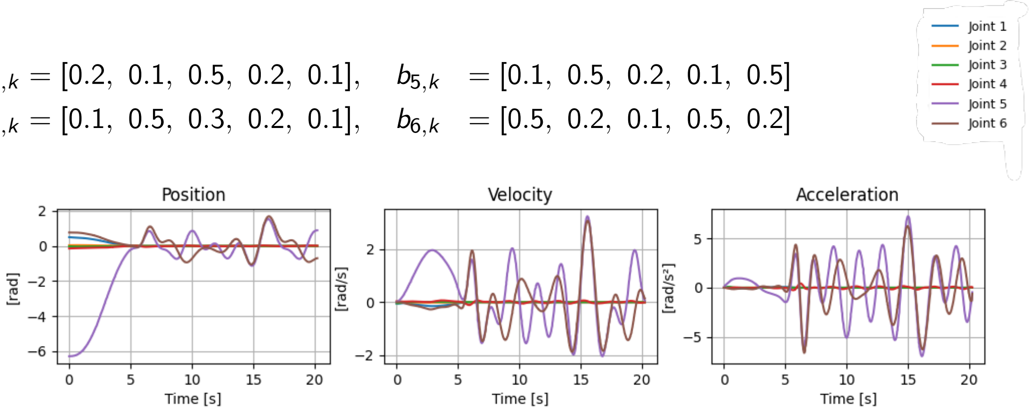


Figure: Fourier Series trajectory performance

Finite Fourier Series – $a_{i,k} = 0$

$$
\begin{aligned}
a_{i,k} &= 0,\\
b_{5,k} &= [0.5,\ 0.7,\ 0.9,\ 0.4,\ 0.5],\\
b_{6,k} &= [0.3,\ 0.9,\ 0.7,\ 0.5,\ 0.8]
\end{aligned}
$$

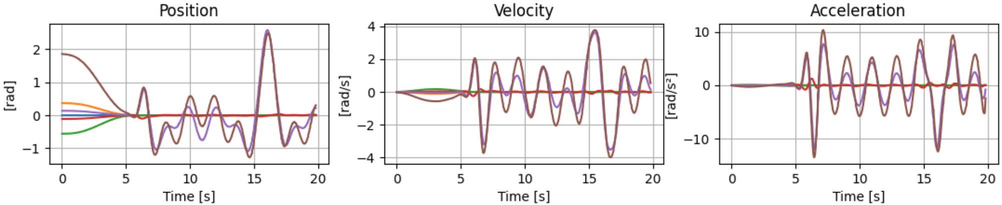


Figure: sinusoidal trajectory performance

Interactive Illustration

Figure: interactive plot trajectory

Conclusion

We developed a model-based control algorithm that can reliably reach arbitrary target positions and follow any sufficiently exciting trajectory, even in the presence of uncertainties or inaccuracies in the URDF model. This repository can be usable for all URDF robots by some adjustments.

GitHub Repository

The End

Thank you for your attentions!