

Adaptive Passivity-Based Control

Model-based control requires an accurate system model, but robotic systems often face the challenge of parameter variations. Parameters such as masses, inertias, and friction coefficients can change over time, introducing uncertainty and making precise control difficult.

Adaptive control offers a solution to address these uncertainties. By dynamically adjusting controller parameters in real-time based on the system's behavior, adaptive control ensures stability and performance. In this assignment, you will implement an **Adaptive Passivity-Based Controller**, specifically the **Slotine and Li controller**, which leverages the useful (though initially unintuitive) concept of **linearity in parameters**.

Linearity in Parameters

The equations of motion for robotic systems depend on various physical properties, such as link masses, moments of inertia, and similar inertial parameters. These physical parameters significantly influence the system's dynamics. Fortunately, the equations of motion exhibit a property called **linearity with respect to inertial parameters**, which simplifies their structure.

This linearity allows us to rewrite the Euler-Lagrange equations of motion in the following compact form:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = Y(q, \dot{q}, \ddot{q})\Theta$$

Here:

- $M(q)$: Mass (inertia) matrix.
- $C(q, \dot{q})$: Coriolis/centrifugal terms.
- $g(q)$: Gravity vector.
- $Y(q, \dot{q}, \ddot{q})$: **Regressor matrix**, a matrix-valued function of the system's kinematic variables q (position), \dot{q} (velocity), and \ddot{q} (acceleration).
- $\Theta \in \mathbb{R}^\ell$: **Parameter vector**, which encodes the system's inertial parameters.

The regressor $Y(q, \dot{q}, \ddot{q})$ effectively maps the kinematic variables to the inertial parameters Θ , enabling us to express the dynamics in a linearized form. This property is critical for adaptive control design.

Computing the Regressor Matrix

While deriving the regressor matrix manually can be complex, modern tools such as **Pinocchio** simplify the process. For instance, in Pinocchio, you can compute the regressor matrix with just a single line of code:

```
regressor = pin.computeJointTorqueRegressor(model, data, q, dq, ddq)
```

For an in-depth explanation on parameterization and regressor computation, refer to this [Colab link](#).

Assignment Tasks

In this assignment, you will implement a **passivity-based adaptive controller** (Slotine and Li controller) for a **6-DOF UR5 robot manipulator** with an **unknown additional mass** attached to its end effector. The steps are as follows:

1. Study the Slotine and Li Controller

Familiarize yourself with the controller design using the references below:

- [Slotine and Li's Original Paper](#)
- Textbooks such as [Robot Modeling and Control](#) by Mark W. Spong and [Robotics: Modelling, Planning and Control](#) by Bruno Siciliano et al.

2. Add Additional Mass to the UR5 Robot's End Effector

Simulate a scenario where an unknown mass is attached to the robot's end-effector.

3. Analyze the Default Inverse Dynamics Controller

Implement an inverse dynamics controller based on a **nominal model** (i.e., without considering the additional mass). Demonstrate that this controller fails to achieve accurate stabilization and tracking, resulting in significant error.

4. Implement the Slotine and Li Controller

Design and implement the Slotine and Li adaptive controller. Compare its performance with the inverse dynamics controller for both:

- **Tracking tasks:** Following a desired reference trajectory.
- **Regulation tasks:** Stabilizing the robot at a desired position.

5. Evaluate Parameter Convergence

Investigate how well the parameter estimation (e.g., the additional mass) converges to its true value over time.

6. Handle Unknown Joint Damping

Assume the robot joints have unknown damping effects represented as:

$$u_{\text{damping}} = -D\dot{q}$$

where D is a constant diagonal matrix. Modify the controller to account for these damping effects.

7. **Constrain Parameter Estimation**

Discuss how to modify the controller to ensure certain parameters remain within specific bounds.

For example, ensure all damping coefficients D remain positive.