Models of Robotic Manipulators

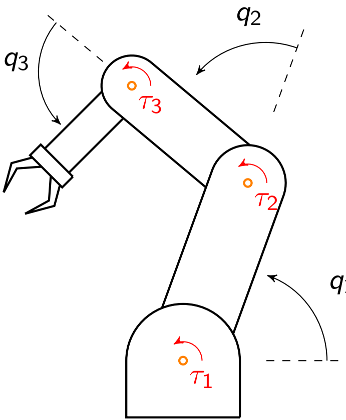
Spring 2020

Hasan Poonawala

Robot Configurations: Joint Variables

The configuration q of such robots will depend on the joints between rigid parts that make up the robot

Robot Configurations: Joint Variables



The configuration q of such robots will depend on the joints between rigid parts that make up the robot

Dynamics

The configuration q changes with time according to the second-order nonlinear ordinary differential equation

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_{friction} + \tau_e, \qquad (1)$$

where

- $D(q)$: inertia matrix (some formulations use $M(q)$)
- $C(q, \dot{q})$: Coriolis terms
- $G(q)$: the conservative forces (gradient of potential energy)
- τ is the force or torque vector generated by actuators at these joints
- $\tau_{friction}$ represents dissipative forces
- τ_e represents externally applied forces

State Space Models

How do these dynamics compare to the usual state-space system?

$$\dot{x} = f(x, u)$$
$$y = h(x, u)$$

The configuration $q \in \mathbb{R}^n$ defines a state

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}.$$

We may then use formulate a state space model

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x)$$

State Space Models

How do these dynamics compare to the usual state-space system?

$$\dot{x} = f(x, u)$$
$$y = h(x, u)$$

The configuration $q \in \mathbb{R}^n$ defines a state

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}.$$

We may then use formulate a state space model

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x)$$

The output y above corresponds to the task space coordinate of the robot.

WARNING: Roboticists do not use symbols x and y as above.

Goals

For the remainder of this course, we will

- Understand joint coordinates q
- Understand properties of the dynamics model (1)
- Understand the relationship $x = FK(q)$, where x is the task coordinates (output) and FK is the forward kinematics map
- Derive controllers that make $q(t) \to q_d(t)$
- Derive controllers that make $x(t) \to x_d(t)$
- Derive controllers that enforce relationships between external forces τ_e and joint velocity $\dot{q}$ (Energy-based control)
- Convert this learning into cool `Julia` simulations

Goals

At the end of this half of the course, you should

- understand how to derive equations and controllers in principle;
- know how to leverage software libraries to perform calculations involving robot models and controllers.