Lecture 28, Nov 17, 2025

Control Design

- As we've previously seen, for control design we have the decentralized view (i.e. independent joint control, where robot dynamics are treated as a disturbance) or the centralized view (considers nonlinear coupling of robot dynamics)
- For control design with robot dynamics, we can use 2 classes of models:
  - Ignore motor dynamics: $D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$
    - τ are torques (generalized forces) applied to the joints
  - Include motor dynamics: $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + B(q)\dot{q} + G(q) = u$
    - u are the motor voltages
    - $B(q)$ models back-EMF
    - $M(q) = D(q) + J$ where J is a diagonal matrix containing the J_m of each motor (see previous lecture with motor model); note $M(q)$ is still symmetric positive definite
    - This is known as the *augmented model*
  - In the end, the algorithms we use for control aren't very different
- Several methods have been developed over the years:
  - *Feedback linearization* (aka *computed torque method*)
  - PD control with gravity compensation
  - *Passivity-based control*
  - Passivity-based control with *parameter adaptation*
    - Unlike the previous methods, we don't actually need to know the system parameters
    - This algorithm uses an adaptive controller where the system parameters are learned
    - This results in a controller that can operate without system parameters, but results in bad transient behaviour

Feedback Linearization

- Consider the augmented model (with motor dynamics): $M(q)\ddot{q} + C(q, \dot{q})\dot{q} + B(q)\dot{q} + G(q) = u$, and assume a *fully-actuated* system, i.e. a motor at each joint (as opposed to an *underactuated* system)
- Given the model with all (nominal) parameters known, and a twice-differentiable $r(t) \in \mathbb{R}^n$ (note this is in q-space), the control problem involves finding a control input $u = f(q, \dot{q}, r, \dot{r}, \ddot{r})$ such that $q(t) \to r(t)$ as $t \to \infty$, for all initial conditions
  - We are designing a state feedback controller
  - Note u is not a function of $\ddot{q}$; practically accelerometers are too noisy for us to get useful measurements
- *Feedback linearization* involves using a u that cancels out all the nonlinear terms, so we end up with a linear model
  - Choose $u = M(q)v + C(q, \dot{q})\dot{q} + B(q)\dot{q} + G(q)$, where $v(t) \in \mathbb{R}^n$ is an exogenous input (i.e. a "new input" which we will determine)
  - This cancels out most of the dynamics, so we end up with simply $M(q)\ddot{q} = M(q)v$, and since $M(q)$ is invertible, $\ddot{q} = v$
  - Now let the states $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ \dot{q}_1 \end{bmatrix}$, so $\dot{x} = \begin{bmatrix} x_2 \\ v_1 \end{bmatrix}$, so now we can design a controller $v_1 = k_1 x_1 + k_2 x_2 = k_1 q_1 + k_2 \dot{q}_1$ to stabilize the system – which is a PD controller on this joint
  - Now do this for every joint to track the reference, and we end up with a system with $2n$ states
- Feedback linearization essentially constructs u such that nonlinear dynamics are cancelled out, so the resulting linear system is decoupled and we can control it by independently controlling each joint
- For tracking the reference $r(t)$, define the tracking errors $e_i(t) = r_i(t) - q_i(t), i = 1, \dots, n$ which we will drive to zero
  - Differentiate the error until we get the input: $\dot{e}_i = \dot{r}_i - \dot{q}_i, \ddot{e}_i = \ddot{r}_i - \ddot{q}_i = \ddot{r}_i - v_i$
  - Choose $v_i = \ddot{r}_i + k_{p,i} e_i + k_{d,i} \dot{e}_i$, i.e. a PD controller with a feedforward term
  - This results in the dynamics $\ddot{e}_i + k_{d,i} \dot{e}_i + k_{p,i} e_i = 0$
    - Any choice of a positive $k_{d,i}, k_{p,i}$ results in poles in the open left half plane, so the error

converges to 0

- This is the simplest and worst performing of the centralized methods we will talk about, since we need to know all the system parameters exactly