Lecture notes: Trajectory tracking control for robot manipulators

Jie Fu

Department of Electrical and Computer Engineering
Robotics Engineering Program
Worcester Polytechnic Institute

RBE502, 2018

Outline

This lecture note is based on

- Chapter 8 in M. Spong **Robot modeling and control.**
- Chapter 5: Position Control and Trajectory Tracking of Murray et. al. **A Mathematical Introduction to Robotic Manipulation**

Exponential tracking with PD control

- Less demanding PD with gravity compensation realizes asymptotic stability for **a setpoint control.**

$$\tau = N(q, \cancel{\dot{q}}) - K_D\dot{e} - K_P e.$$

$q^d = \text{const.}$ $\left| \begin{array}{l} e = q - q^d \\ \dot{e} = \dot{q} - \dot{q}^d = \dot{q} \end{array} \right.$

$= N(q) - K_D\dot{q} - K_P e$

Exponential tracking with PD control

- Less demanding PD with gravity compensation realizes asymptotic stability for **a setpoint control.**

$$\tau = N(q, \dot{q}) - K_D \dot{e} - K_P e.$$

- Whileas computed Torque control

$$\tau = \underbrace{M(q)(-K_P e - K_D \dot{e})}_{\text{Feedback control}} + \underbrace{C(q, \dot{q})\dot{q} + N(q, \cancel{\dot{q}}) + M(q)\ddot{q}_d}_{\text{Feedforward compensation}}$$

Δ

realizes exponential trajectory tracking.

$\ddot{q} = a_q$

$\hookrightarrow q(t) \to q^d(t)$

Exponential tracking with PD control

- Less demanding PD with gravity compensation realizes asymptotic stability for **a setpoint control**.

$$\tau = N(q,\dot{q}) - K_D\dot{e} - K_P e.$$

- Whileas computed Torque control

$$\tau = \underbrace{M(q)(-K_P e - K_D\dot{e})}_{\text{Feedback control}} + \underbrace{C(q,\dot{q})\dot{q} + N(q,\dot{q}) + M(q)\ddot{q}_d}_{\text{Feedforward compensation}}$$

realizes exponential trajectory tracking.

Augmented PD control law

$$\tau = \underbrace{(-K_P e - K_D\dot{e} + N(q,\dot{q}))}_{\text{PD control w gravity compensation}} + \underbrace{M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d}_{\text{Partial feedforward}}$$

Realizes **exponential** trajectory tracking (when $\dot{q}_d = 0$, setpoint stabilization).

Proof

$m: M(q)$
$C: C(q,\dot{q})$

The error dynamics with the control input:

$$\tau = \underbrace{(-K_P e - K_D \dot{e} + N(q, \dot{q}))}_{\text{PD control w gravity compensation}} + \underbrace{M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d}_{\text{Partial feedforward}}$$

$$M\ddot{q} + C\dot{q} + N(q) = \tau$$

$$M\ddot{q} + C\dot{q} + \cancel{N(q)} = -K_p e - K_D \dot{e} + \cancel{N(q)} + M\ddot{q}_d + C\dot{q}_d$$

$$M(\ddot{q} - \ddot{q}_d) + C(\dot{q} - \dot{q}^d) + K_p e + K_D \dot{e} = 0$$

$q - q_d = e$: $\quad \underset{M(q)}{M}\ddot{e} + \underset{C(q,\dot{q})}{C}\dot{e} + K_p e + K_D \dot{e} = 0$

↳ nonlinear system

Exponential stability of augmented PD

Proof: Hint:

$\varepsilon > 0$

① V pos. def
② $\dot{V}$ neg def

$$V = \underbrace{\frac{1}{2}\dot{e}^T M(q)\dot{e} + \frac{1}{2}e^T K_P e}_{V_0} + \underbrace{\varepsilon e^T M(q)\dot{e}}_{V_c}.$$

cross term

$V = V_0 + V_c$

V_0 pos. def.

①: (i) $e^T M \dot{e} > 0$ $\quad \varepsilon > 0$ $\quad V$ pos. def.

(ii) $e^T M \dot{e} < 0$; $\quad \varepsilon > 0$, $\quad \varepsilon$ arbitrarily small.

large $K_p > 0$ $\quad \underline{\frac{1}{2} e^T K_p e + \varepsilon e^T M \dot{e}} > 0$

②: $\dot{V} = \dot{V}_0 + \dot{V}_c$

$\dot{V}_0 = e^T K_p \dot{e} + \dot{e}^T \underline{M(q)\ddot{e}} + \frac{1}{2}\dot{e}^T \dot{M}(q)\dot{e}$

$$= e^T \cancel{K_p} \dot{e} + \dot{e}^T(-\cancel{K_p} e - K_D \dot{e} - C\dot{e})$$

$$+ \frac{1}{2} \dot{e}^T \dot{M} \dot{e}$$

K_p symmetric $$= -\dot{e}^T K_D \dot{e} - \dot{e}^T (C - \tfrac{1}{2} \dot{M}) \dot{e}$$

$$= -\dot{e}^T K_D \dot{e} + \underbrace{\tfrac{1}{2} \dot{e}^T (\dot{M} - 2C) \dot{e}}_{0}$$

$\dot{M} - 2C$ skew sym.

$$\dot{V}_c = \varepsilon \Big(\dot{e}^T M(q) \dot{e} + e^T (\dot{M}(q) - C(q,\dot{q}) - K_D) \dot{e}$$

$$\underbrace{- e^T K_p e}_{A_1} \Big) = A_1 + A_2 + A_3$$

$$A_1 = -\varepsilon e^T K_p e \qquad < 0$$

$$A_2 = \varepsilon \dot{e}^T M(q) \dot{e}$$

$$A_3 = \varepsilon e^T (\dot{M} - C - K_D) \dot{e}$$

$$A_2 + \underset{\sim}{\dot{V}_0} = -\dot{e}^T(\underset{\sim}{K_D} - \varepsilon M(q))\dot{e} < 0$$

K_D large $K_D - \varepsilon M$ pos. def.

ε small

$$A_3 = \varepsilon e^T(\dot{M} - C - K_D)\dot{e}$$

To show A_3 is bounded

$$= \underbrace{\varepsilon e^T(\dot{M} - C)\dot{e}} - \varepsilon e^T K_D \dot{e}$$

$$\| e^T(\dot{M} - C)\dot{e} \|$$

A_3 bounded: $\| A_3 \| \leq \underset{\sim}{\text{const}}$ pick K_D large to make $\dot{V} < 0$

To shown A_3 is bounded:

$$|A_3| = |\varepsilon e^T(\dot{M} - C)\dot{e} - \varepsilon e^T K_D \dot{e}|$$

$$\leq \varepsilon \underbrace{|e^T(\dot{M} - C)\dot{e}|}_{①} + \varepsilon \underbrace{|e^T K_D \dot{e}|}_{②}$$

$1+\frac{3}{2}=\left(\frac{5}{2}\right)$

① $\le \|e^T\| \, \|(\dot M - C)\| \|\dot e\|$

$$e^T \dot M \dot e = \frac{\partial M_{ij}}{\partial q_k} \dot q_k e_i \dot e_j$$

$$e^T C(q,\dot q)\dot e = \Gamma_{ijk} \dot q_k e_i \dot e_j$$

$$= \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot q_k e_i \dot e_j \qquad \frac{3}{2}$$

① $\le (\qquad) \le \frac{5}{2} \left\| \frac{\partial M}{\partial q} \right\| \|\dot q\| \|e\| \|\dot e\|$ const upper bound

$\max\limits_{i,j,k}\left(\frac{\partial M_{ij}}{\partial q_k}\right)$ $\quad \max\limits_{k}(\dot q_k)$ $\max\limits_{i}(e_i)$ $\max\limits_{j}(\dot e_j)$

② $= e^T K_D \dot e < \|\dot e^T\| \|K_D\| \|\dot e\|$ bounded const.

pick K_D large ε small s.t. $\dot V_0 + \dot V_c < 0$

$\dot V_0 + \dot V_c = 0$

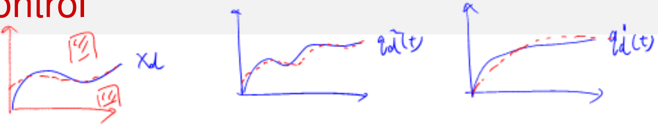
Workspace control

q_d

Suppose given a path $x_d(t) \in SE(3)$: desirable end-effector configuration as a function of time.
To apply joint space control (centralized or decentralized)

1. Solve $q_d(t)$ from $x_d(t)$ using inverse kinematics.
2. Design joint space controller.

Workspace control



Suppose given a path $x_d(t) \in SE(3)$: desirable end-effector configuration as a function of time.

To apply joint space control (centralized or decentralized) $X_d \to q_d$

1. Solve $q_d(t)$ from $x_d(t)$ using inverse kinematics.
   - time consuming.
   - hard to reason about the trajectory of end-effector with the trajectory of q — may cause undesirable behavior.
2. Design joint space controller.

Workspace control

Suppose given a path $x_d(t) \in SE(3)$: desirable end-effector configuration as a function of time.

To apply joint space control (centralized or decentralized)

1. Solve $q_d(t)$ from $x_d(t)$ using inverse kinematics.
    - time consuming.
    - hard to reason about the trajectory of end-effector with the trajectory of q — may cause undesirable behavior.
2. Design joint space controller.

To overcome these,

Direct workspace control.

Recap

Forward kinematics: A smooth and invertible mapping f from the joint variables to the workspace variables

$$x = f(q)$$

The Jacobian

$$\dot{x} = J(q)\dot{q}, \quad J(q) = \frac{\partial f}{\partial q}$$

$$\ddot{x} = \dot{J}(q)\dot{q} + J(q)\ddot{q}$$

$$\Rightarrow \underbrace{\ddot{q}}_{a_q} = J(q)^{-1}(\underbrace{\ddot{x}}_{a_x} - \dot{J}(q)\dot{q})$$

$$\ddot{q} = a_q$$

$$\boxed{\ddot{x} = a_x}$$

$$x^d(t)$$

$$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \qquad \dot{z} = Az + Ba_x$$

Recap: $\ddot{q} = a_q \rightarrow a_q = -K_p(q - q^d) - K_D(\dot{q} - \dot{q}^d) + \ddot{q}^d$

substitute q with x

$$\boxed{a_x = -K_p(x - x^d) - K_D(\dot{x} - \dot{x}^d) + \ddot{x}^d} \quad (\star)$$

Relate a_x with τ.

$a_q = J^{-1}(q)(a_x - \dot{J}\dot{q})$ ✓ a_q ✓ $\tau = M(q)\,a_q + C(q,\dot{q})\dot{q} + N(q)$

$$\boxed{\tau = M(q)\left[J^{-1}(q)(\underset{(\star)}{a_x} - \dot{J}\dot{q})\right] + C(q,\dot{q})\dot{q} + N(q)}$$

Recap

Forward kinematics: A smooth and invertible mapping f from the joint variables to the workspace variables

$$x = f(q)$$

The Jacobian

$$\dot{x} = J(q)\dot{q}, \quad J(q) = \frac{\partial f}{\partial q}$$

Given f is smooth and invertible,

$$\dot{q} = J^{-1}\dot{x}, \quad \ddot{q} = J^{-1}\ddot{x} + \frac{d}{dt}(J^{-1})\dot{x}.$$

The dynamic of robot in the joint space

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q,\dot{q}) = \tau$$

From joint space to task space

State variables in

- Joint space: q
- Work space: X.

X is a function of q, we have

$$\dot{X} = J(q)\dot{q}$$

where $J(q)$ is the **analytical** Jacobian. Using inverse dynamics

control:

Summary

- Show Lyapunov stability of decentralized PD control for plannar manipulators.
- PD+Gravity compensation for set-point tracking.
- Inverse dynamics control for trajectory tracking.
- Inverse dynamics control in task space.

Cons:

- Inverse dynamics requires exact knowledge of model dynamics.
- PERFORMANCE is **NOT GUARANTEED** when
  - parameter becomes uncertain.
  - robot picking up an unknown load.

Next: Robust and adaptive control.