Control of Manipulators

EE366/CE366/CS380: Introduction to Robotics

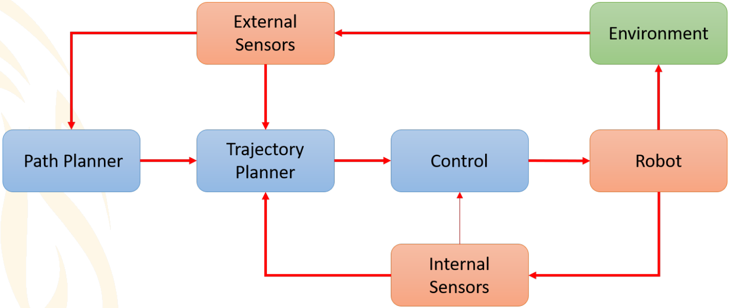
Dr. Basit Memon

Electrical and Computer Engineering
Habib University

April 15, 17, 2024

Table of Contents

Table of Contents



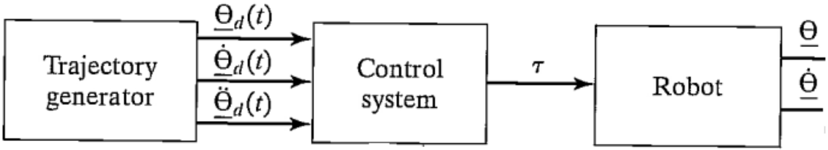
- successfully completing a task (high-level)
- accurate execution of motion trajectory (intermediate-level)
- zeroing in on a position (low-level)

Examples of low-level control

- Moving an object from one place to another (Motion Control)
- Tracing trajectory for spray painting gun (Motion Control)
- Applying right force, e.g. polishing a workpiece (Force Control)
- Writing on chalkboard - applying force in one direction and moving in another (Hybrid Motion-Force Control)
- Haptic pen receiving force input from users (Impedance Control)

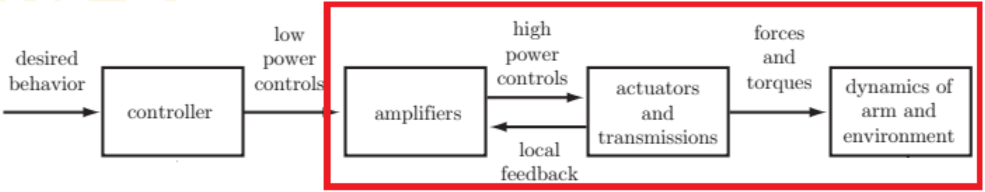
Table of Contents

What is the objective of controller?



- Find controller such that error between desired trajectory $(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$ and actual trajectory is minimized

Zooming into the robot system:

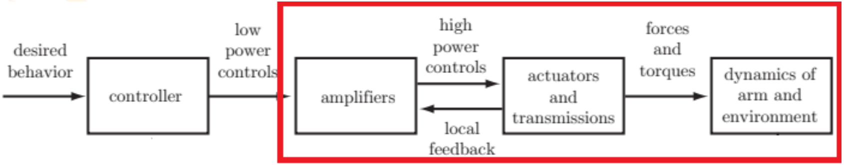


- If the relationship between actuator input (motor voltage) and joint angle is determined, then we could determine input function that will achieve desired motion.

- This idea is called **open-loop control**.

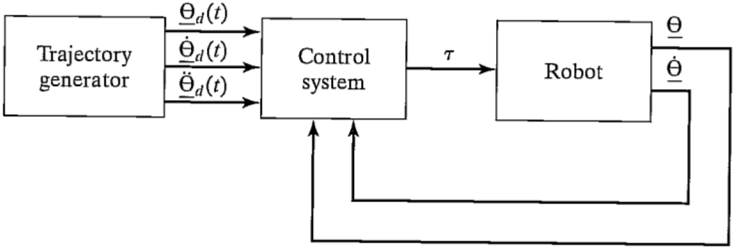
Open-loop control is not used, because

Need to determine mathematical model for this



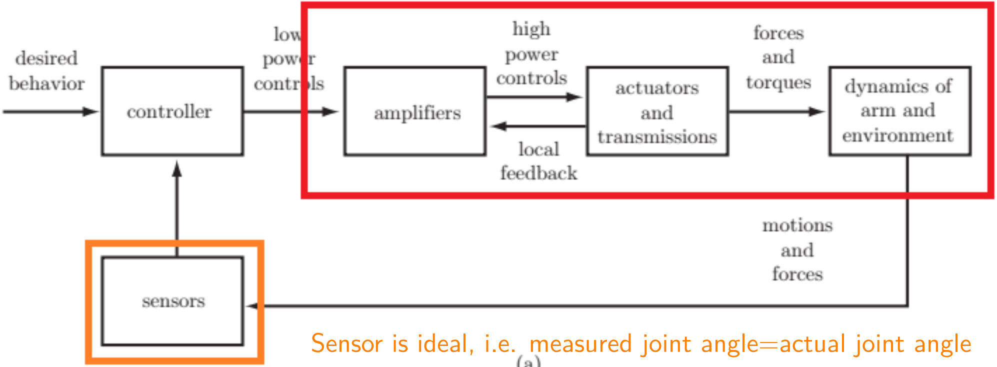
- Relationship or *Dynamics of the system* are complex;
- Uncertainties in determining the parameters in dynamical equation, e.g. masses, etc;
- Unstructured uncertainties such as flexibilities, sensor noise, unknown environment, etc;
- Wear and tear over time will change values of parameters;

Closed-loop control is the preferable strategy.



- It is preferable to use feedback.
- Control strategy could have significant impact on performance of manipulator.
- Mechanical design and drive system influence control strategy.

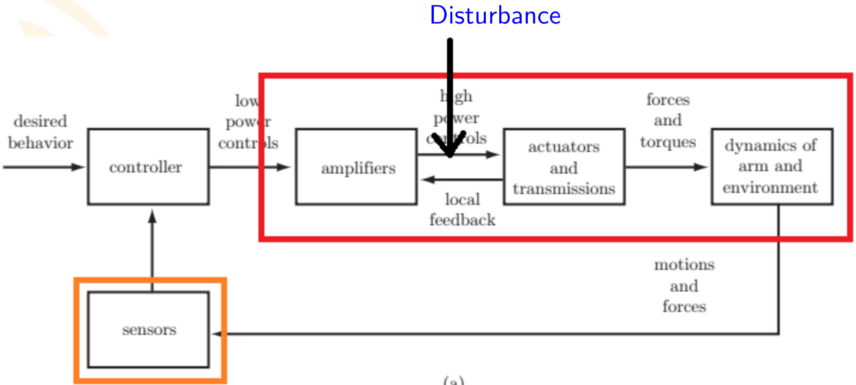
Need to determine mathematical model for this



Sensor is ideal, i.e. measured joint angle=actual joint angle

(a)

We can use a simplified model by adjusting our controller objective.



(a)

- Find controller such that error between desired trajectory $(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$ and actual trajectory is minimized, and effects of disturbance on trajectory are minimized.

- If desired joint position is $\theta_d(t)$ and actual joint position is $\theta(t)$, then

$$\theta_e(t) = \theta_d(t) - \theta(t).$$

- Our goal is for $\theta_e(t) \to 0$, i.e steady-state error is zero.

- Since the error is time-evolving trajectory as well, then we could write a differential equation governing the error. For linear systems,

$$a_p\theta_e^{(p)} + a_{p-1}\theta_e^{(p-1)} + \cdots + a_1\dot{\theta}_e + a_o\theta_e = c.$$

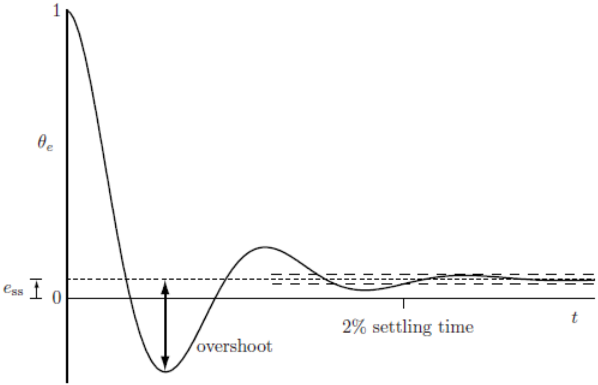


Figure: Possible error profile

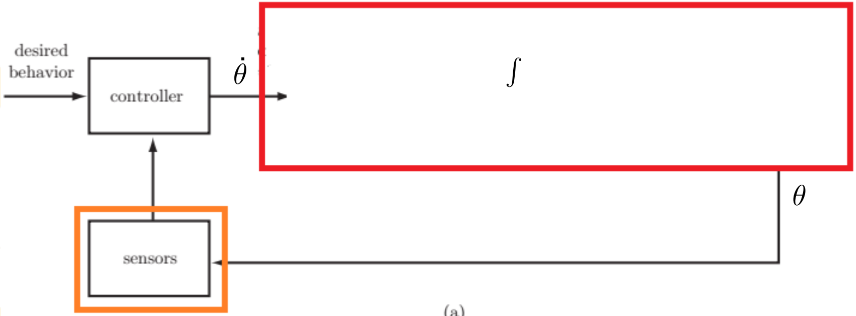
- little or no overshoot;
- a short settling time.

Table of Contents

Kinematic Control of robot

- For each joint, controllers needs to generate a command for the system; this command is usually motor torque.
- It is possible in some cases to consider a kinematic command, e.g. velocity.
- This is the case for stepper motors that can directly receive velocity commands
- And for DC motors equipped with low-level controllers accepting velocity commands.
- Performance is satisfactory if motion is not too fast or doesn't require high accelerations.

Kinematic Control abstracts the robot system as an integrator.



(a)

Feedback velocity control of single joint

- Simplest feedback controllers is *proportional* controller:

$$\dot{\theta}(t) = K_p(\theta_d(t) - \theta(t)) = K_p\theta_e(t).$$

- **Setpoint Control:** Special case. $\theta_d(t)$ is constant.

$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$$
$$\dot{\theta}_e(t) = -K_p\theta_e(t)$$

This is first order ODE, so

$$\theta_e(t) = \theta_e(0)e^{-K_p t}$$

Steady-state error for setpoint control is zero with P Controller

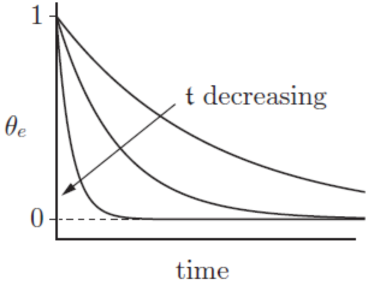


Figure: First order error response,
$\theta_e(t) = \theta_e(0)e^{-K_p t}$

- Evident that $\theta_e(t) \to 0$.
- Larger the K_p, the faster the error goes to zero.

What if θ_d is not constant?

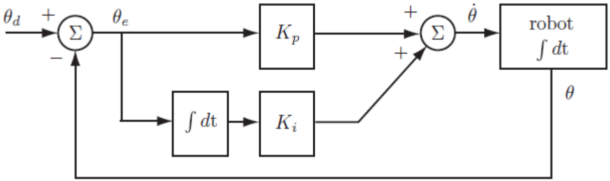
- Say $\theta_d(t)$ is a line trajectory, i.e. $\dot{\theta}_d(t)$ is constant.

$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t) = c - K_p\theta_e(t)$$
$$\dot{\theta}_e(t) + K_p\theta_e(t) = c$$

first-order nonhomogeneous ODE

$$\Rightarrow \theta_e(t) = \frac{c}{K_p} + \left(\theta_e(0) - \frac{c}{K_p}\right) e^{-K_p t}$$

- We can see that $\theta_e(t) \rightarrow c/K_p$. Steady-state error is nonzero.
- Increasing K_p reduces error, but K_p cannot be made arbitrarily large
  - Practical constraints on the maximum velocity.



- Add term proportional to integral of error:

$$\dot{\theta}(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt.$$

- Error dynamics for the same line trajectory are:

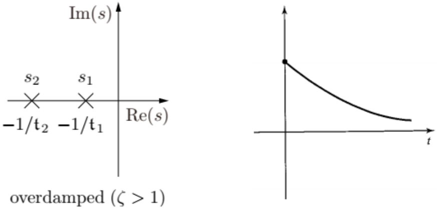
$$\dot{\theta}_e(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$$

$$c = \dot{\theta}_e(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt$$

Differentiate:

$$0 = \ddot{\theta}_e(t) + K_p\dot{\theta}_e(t) + K_i\theta_e(t)$$

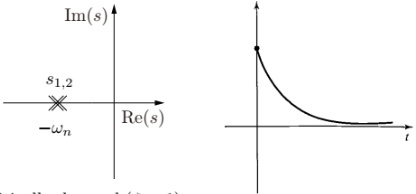
What is the error response for $\ddot{\theta}_e(t) + K_p\dot{\theta}_e(t) + K_i\theta_e(t) = 0$?



- Second order error response.
- Standard 2nd order ODE:
  $\ddot{e}(t) + 2\zeta\omega_n\dot{e}(t) + \omega_n^2 e(t) = 0$
- Shape of $e(t)$ depends on roots of $s^2 + 2\zeta\omega_n + \omega_n^2$.

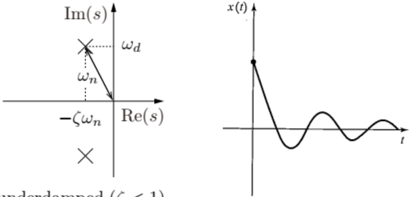
$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

- Overdamped case: $\zeta > 1$
  - Roots are real and distinct.
  - $e(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$



critically damped ($\zeta = 1$)

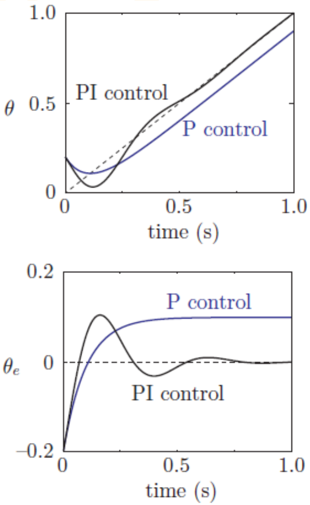
- Critically damped case: $\zeta = 1$
  - Roots are real and equal.
  - $e(t) = (c_1 + c_2 t)e^{-\omega_n t}$



underdamped ($\zeta < 1$)

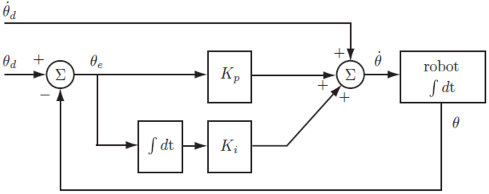
- Underdamped case: $\zeta < 1$
  - Roots are complex conjugates.
  - $e(t) =$ $(c_1 \cos \omega_d t + c_2 \sin \omega_d t)e^{-\zeta\omega_n t}$, where $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

Steady-state error is zero for ramp trajectory with PI controller



- If $\zeta > 0$ and $\omega_n > 0$, i.e. $K_p > 0$ and $K_i > 0$ then $\theta_e(t) \to 0$ for all possible responses.
- How to choose K_p and K_i?
- We don't want an overshoot in the error. $\zeta = 1$ gives the fastest decaying response. So choose K_p and K_i such that $\zeta = 1$.
- We'll still get an error for any θ_d other than ramp.

Feedforward plus Feedback Control tracks any trajectory



- Feedback control requires non-zero error before joint moves.
- Use feedforward to initiate motion before error accumulates.

$$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt$$

- The error is zero for any desired trajectory. Feedback brings the error to zero in case of disturbances.

Control of multiple joints is simple extension of single joint case.

- Generalize single-joint controller to n joints.
- $\theta_d(t)$ and $\theta(t)$ are $n \times 1$ vectors.
- Gains $K_p = k_p I$ and $K_i = k_i I$, where k_p and k_i are scalars and I is $n \times n$ identity matrix.
- Performance analysis remains same for each joint.

Table of Contents

Kinematic control for given trajectory in task space:

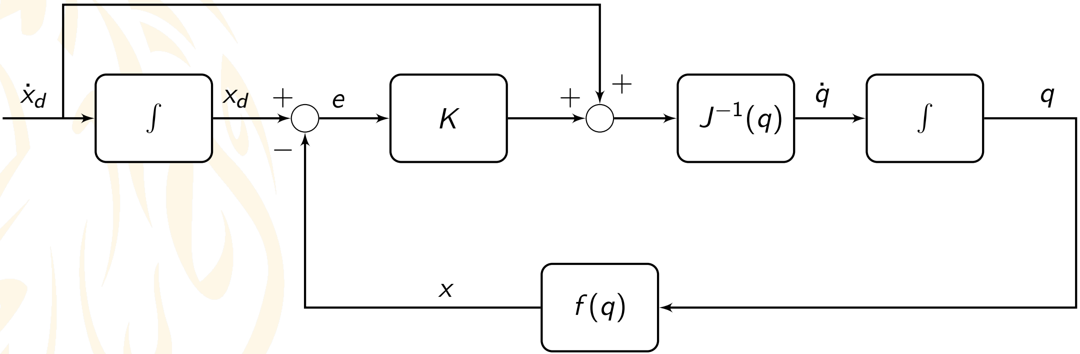


Figure: Block diagram of kinematic control in task space

$$e_x = x_d - x \quad \Rightarrow \dot{e}_x = \dot{x}_d - J(q)J^{-1}(q)\left[\dot{x}_d + K(x_d - x)\right] = -Ke_x$$

Should we formulate control problem in task space or joint space?

- Desired motion is specified in the task space, but interaction between actuator and joint is in joint space.
- Control problem is easier in joint space, as kinematics becomes embedded in the control problem in task space.
- For problems such as grasping, where there is huge uncertainty in the desired position it's better to work in task space.

Table of Contents