Content

- Introduction
- Control of Second Order Linear Systems
- Control-Law Partitioning
- Trajectory Following
- Nonlinear and Time-Varying Systems
- Nonlinear Control of Multi-Input-Multi-Output System
- Matlab Simulink Simulation

Control-Law Partitioning

- We know want a controller which is largely independent of the actual dynamics

Model-based controllr

- Control-Law Partitioning: The controller is partitioned into:
  - I Model based portion – system parameters appear here
  - II Servo portion – Independent of the system parameters

Control of only one joint/link

- For the system: $\Rightarrow$ $J\ddot{\theta} + b\dot{\theta} + k\theta = \tau$ 5
- We design: Step1- $\tau = J\alpha + b\dot{\theta} + k\theta$ $\ddot{\theta} \leftarrow \alpha$ 6

Model-based compensation

Step2-

- And we can design $\alpha = \ddot{\theta}_d + k_D(\dot{\theta}_d - \dot{\theta}) + k_P\,(\,\theta_d - \theta\,)$ $\rightarrow$ Servo portion 7
  - Setting a desired stiffness k_p which is now independent of J.
  - Then we calculate $k_D = 2\sqrt{k_p}$ which is also independent of J.

Control of only one joint/link

- To analyse the closed loop system

6 & 7 into 5

$$\Rightarrow \quad J\ddot{\theta} + b\dot{\theta} + k\theta = J\underline{\alpha} + b\dot{\theta} + k\theta = \tau$$

$$\Rightarrow \quad J\ddot{\theta} + \cancel{b\dot{\theta}} + \cancel{k\theta} = J\,[\ddot{\theta}_d + k_D(\dot{\theta}_d - \dot{\theta}) + k_P\,(\theta_d - \theta)] + \cancel{b\dot{\theta}} + \cancel{k\theta}$$

- And we get $\Rightarrow$ $\ddot{\theta} = \ddot{\theta}_d + k_D(\dot{\theta}_d - \dot{\theta}) + k_P\,(\theta_d - \theta)$

i) $\theta_d, \dot{\theta}_d \;\&\; \ddot{\theta}_d = 0 \Rightarrow \ddot{\theta} = -k_D\dot{\theta} - k_P\theta$

$$\boxed{\ddot{\theta} + k_D\dot{\theta} + k_P\theta = 0}$$

ii) Trajectory / Reference Tracking

$\theta_d \neq 0$, $\dot{\theta}_d \neq 0$ & $\ddot{\theta}_d \neq 0$

Content

- Introduction
- Control of Second Order Linear Systems
- Control-Law Partitioning
- Trajectory Following
- Nonlinear and Time-Varying Systems
- Nonlinear Control of Multi-Input-Multi-Output System
- Matlab Simulink Simulation

Trajectory Following

- Suppose now we not only want to regulate to a constant position, but to track a desired trajectory.
- Assume trajectory $\theta_d(t)$ is smooth, and thus $\dot{\theta}_d(t)$ and $\ddot{\theta}_d(t)$ are available.
- Define: $e = \theta_d - \theta$ thus we also have $\dot{e}$ and $\ddot{e}$
- We can then design the trajectory following controller as follows:

- For system: $J\ddot{\theta} + b\dot{\theta} + k\theta = \tau$ ←
- Design: $\tau = J\alpha + b\dot{\theta} + k\theta$ 1st step
  - with: $\alpha = \ddot{\theta}_d + k_D(\dot{\theta}_d - \dot{\theta}) + k_P\,(\,\theta_d - \theta\,)$ 2nd step

- Set a desired stiffness k_p
- Then calculate $k_D = 2\sqrt{k_p}$

Trajectory Following

- The closed loop system then becomes:

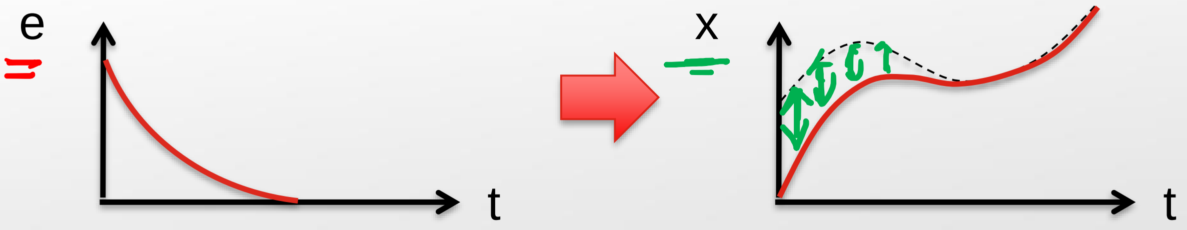
$$\Rightarrow \quad J\ddot{\theta} + b\dot{\theta} + k\theta = J\,[\ddot{\theta}_d + k_D(\dot{\theta}_d - \dot{\theta}) + k_P\,(\theta_d - \theta)] + b\dot{\theta} + k\theta$$

$$\Rightarrow \quad \underset{\ddot{e}}{0} = \underbrace{(\ddot{\theta}_d - \ddot{\theta})} + k_D\underbrace{(\dot{\theta}_d - \dot{\theta})}_{\dot{e}} + k_P\underbrace{(\theta_d - \theta)}_{e}$$

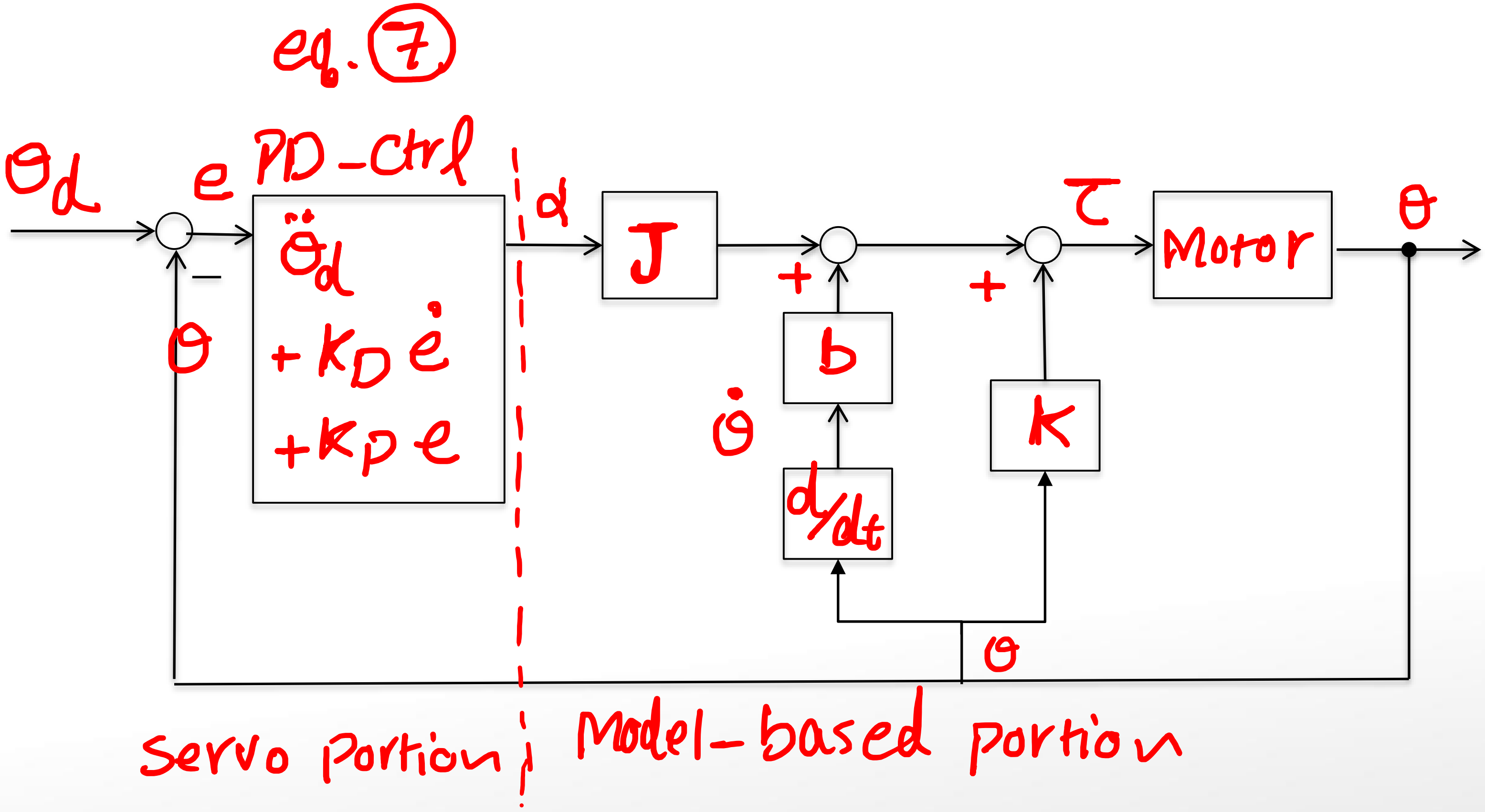
- from which we can obtain the error dynamics:

$$\Rightarrow \quad \ddot{e} + \underline{\underline{k_D}}\,\dot{e} + \underline{\underline{k_p}}\,e = 0$$

- The control parameters have been chosen to achieve critically damped response.
- Therefore, error (x_d – x) decays rapidly and we achieve trajectory following.



Control of only one joint/link



Control Law Partitioning

$$M(q)\ddot{q} + V(q,\dot{q}) + G(q) = \tau$$

- Note:
  - In the linear case, the advantage of such control law partitioning might not be obvious.
  - m, b, k are mostly constants and thus it wouldn’t be too difficult to calculate the controller gains directly from original equation.
  - However, the M, V, G matrices for a robot manipulator are nonlinear, and vary according to the robot configuration and speed.
  - By using the control law partitioning method, we will be able to calculate the controller parameters easily.
  - The model-based compensation for V and G, and the scaling of f by M will allow a constant stiffness and damping for the robot, regardless of the configuration and speed.

Content

- Introduction
- Control of Second Order Linear Systems
- Control-Law Partitioning
- Trajectory Following
- Nonlinear and Time-Varying Systems
- Nonlinear Control of Multi-Input-Multi-Output System
- Matlab Simulink Simulation

Nonlinear and Time-Varying Systems

- Now that you have understood control law partitioning, we will use a very simple nonlinear control method to achieve constant performance (stiffness and damping) throughout the workspace:
  - Just cancel off the nonlinear or time-varying portion of the model!
  - This is called a linearizing control law.
- The control law partitioning method is particularly useful to achieve this.
- Let’s see a few examples to understand the concept.

Nonlinear and Time-Varying Systems

- E.g. 2nd order system with nonlinear spring: $m\ddot{x} + b\dot{x} + qx^3 = F$
  - We shall design the controller F as:

$$\Rightarrow \quad F = m\underbrace{f + b\dot{x} + qx^3}_{} \quad \text{with} \quad f = \ddot{x}_d + k'_D(\dot{x}_d - \dot{x}) + k'_p(x_d - x) = \underbrace{\ddot{x}_d + k'_D\dot{e} + k'_p e}_{\text{servo portion}}$$

$\ddot{x} \leftarrow f$

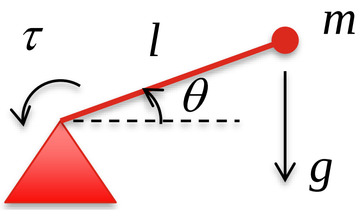
Model-based portion, incorporating the nonlinear term

- The controller leads to the following closed-loop system:

$$\Rightarrow \quad \begin{aligned} m\ddot{x} + b\dot{x} + qx^3 &= F \\ &= mf + b\dot{x} + qx^3 \\ &= m\left(\ddot{x}_d + k'_D\dot{e} + k'_p e\right) \end{aligned} \quad \Rightarrow \quad \begin{aligned} \ddot{x} &= \ddot{x}_d + k'_D\dot{e} + k'_p e \\ 0 &= \ddot{e} + k'_D\dot{e} + k'_p e \end{aligned}$$

- Set the desired stiffness kp', and let $k'_D = 2\sqrt{k'_p}$

Nonlinear and Time-Varying Systems



- E.g. Single-link Manipulator with Coulomb and viscous friction.
  - Its dynamic model is: $ml^2\ddot{\theta} + b\dot{\theta} + \overbrace{c\,\mathrm{sgn}(\dot{\theta})}^{\uparrow} + mgl\cos(\theta) = \tau$
  - We shall design the controller T as:

⇒ $$\underbrace{\tau = ml^2\alpha + b\dot{\theta} + c\,\mathrm{sgn}(\dot{\theta}) + mgl\cos(\theta)}_{}$$

Model-based portion, incorporating the nonlinear term

with $$\alpha = \ddot{\theta}_d + k'_D(\dot{\theta}_d - \dot{\theta}) + k'_p(\theta_d - \theta) = \underbrace{\ddot{\theta}_d + k'_D\dot{e} + k'_p e}_{}$$

servo portion

- The controller leads to the following closed-loop system:

$$ml^2\ddot{\theta} + \cancel{b\dot{\theta}} + \cancel{c\,\mathrm{sgn}(\dot{\theta})} + \cancel{mgl\cos(\theta)} = \tau$$
$$⇒ \quad = ml^2\alpha + \cancel{b\dot{\theta}} + \cancel{c\,\mathrm{sgn}(\dot{\theta})} + \cancel{mgl\cos(\theta)}$$

→ $\ddot{\theta} = \ddot{\theta}_d + k'_D\dot{e} + k'_p e$ → $0 = \ddot{e} + k'_D\dot{e} + k'_p e$

Nonlinear and Time-Varying Systems

- As can be seen from the examples, by using the control law partitioning method, it is not difficult to design a nonlinear controller.
  - Make use of the model to design a model-based control law which “cancels” off the nonlinearities.
  - Then, design a linear servo law for unit mass to achieve desired stiffness and critical damping.

- NOTE: This method is also called the “Computed Torque Control”.
- IMPORTANT ASSUMPTION: The model and the parameters are exactly known.
  - In practice, this can be a problem.

Content

- Introduction
- Control of Second Order Linear Systems
- Control-Law Partitioning
- Trajectory Following
- Nonlinear and Time-Varying Systems
- Nonlinear Control of Multi-Input-Multi-Output System
- Matlab Simulink Simulation

Multi-Input-Multi-Output System

- Apart from being nonlinear and time-varying, the robotic manipulator also has strong coupling amongst its many joints.
- To handle this issue, we will first look at solving a multi-input-multi-output (MIMO) control problem.
  - Instead of one single joint variable (x or θ), we now have a vector of joint positions: $X = [q_1 \quad q_2 \quad \cdots \quad q_n]^T$
  - along with its time derivatives (velocities and accelerations).
- Let the dynamic model of the MIMO system be: $f\ddot{X} + \beta = F$
- Design the control law as: $F = f\alpha + \beta$
- The closed loop system then becomes: $\ddot{X} = \alpha$ or $\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$

Multi-Input-Multi-Output System

- We see that, again, by using the control law partitioning method, we are able to reduce the problem to that of n independent unit mass.
- Therefore, the model based portion of the control law is called "Linearizing and Decoupling" control law. $F = f\alpha + \beta$
- Finally, we will design a servo control law for each of the joints:

$\Rightarrow$

$\ddot{X} = \alpha$

$$\ddot{X} = \ddot{X}_d + K_D \dot{E} + K_p E$$

usually diagonal

usually diagonal

or

$\Rightarrow$

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} = \begin{bmatrix} \ddot{q}_{1d} + k_{D1}\dot{e}_1 + k_{p1}e_1 \\ \ddot{q}_{2d} + k_{D2}\dot{e}_2 + k_{p2}e_2 \\ \vdots \\ \ddot{q}_{nd} + k_{Dn}\dot{e}_n + k_{pn}e_n \end{bmatrix}$$

$\downarrow$

$$K_D = \begin{bmatrix} k_{D1} & 0 & 0 & 0 \\ 0 & k_{D2} & 0 & 0 \\ 0 & 0 & k_{D3} & 0 \\ 0 & 0 & 0 & k_{D4} \end{bmatrix}$$

$\downarrow$

$$K_p = \begin{bmatrix} k_{p1} & 0 & 0 & 0 \\ 0 & k_{p2} & 0 & 0 \\ 0 & 0 & k_{p3} & 0 \\ 0 & 0 & 0 & k_{p4} \end{bmatrix}$$

Manipulator Control

- The same idea of control law partitioning will be used for linearizing, decoupling and servoing of the manipulator.
- The dynamic model of manipulator is:

$$\Rightarrow \qquad M(q)\ddot{q}+V(q,\dot{q})+G(q)=\tau$$

- We could also include non-rigid body effects, e.g. friction into the model:

$$\Rightarrow \qquad M(q)\ddot{q}+V(q,\dot{q})+G(q)+F(q,\dot{q})=\tau$$

- Now, design the model-based control law as:

1st step

$$\tau = M(q)\alpha + V(q,\dot{q}) + G(q) + F(q,\dot{q}) \qquad \ddot{q} \leftarrow \alpha$$

- The servo portion is then designed as:

2nd step

$$\alpha = \ddot{q}_d + K_D\dot{E} + K_pE$$

Manipulator Control

- The control law leads to the following closed loop system:

$$\begin{aligned} M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(q,\dot{q}) &= \tau \\ &= M(q)\alpha + V(q,\dot{q}) + G(q) + F(q,\dot{q}) \\ &= M(q)\left(\ddot{X}_d + K_D\dot{E} + K_pE\right) + V(q,\dot{q}) + G(q) \end{aligned}$$

- Or: $\ddot{q} = \ddot{q}_d + \underline{K_D\dot{E}} + \underline{K_pE}$ $\Longrightarrow$ $\underline{\ddot{E}} + K_D\dot{E} + K_pE = 0$

- Note that the system is decoupled. K_D and K_p are diagonal, thus we can write the closed loop equation for each joint:

$$\ddot{e}_i + k_{Di}\dot{e}_i + k_{pi}e_i = 0$$

  - This is an asymptotically stable system, and the error will decay to zero, meaning that tracking of reference is achieved. $x_i \rightarrow x_{di}$

Manipulator Control

- The control law leads to the following closed loop system:

$$\begin{aligned} M(q)\ddot{q} + V(q,\dot{q}) + G(q) + F(q,\dot{q}) &= \tau \\ &= M(q)\alpha + V(q,\dot{q}) + G(q) + F(q,\dot{q}) \\ &= M(q)\left(\ddot{X}_d + K_D\dot{E} + K_p E\right) + V(q,\dot{q}) + G(q) \end{aligned}$$

- Or: $\ddot{q} = \ddot{q}_d + K_D\dot{E} + K_p E$ $\Rightarrow$ $\ddot{E} + K_D\dot{E} + K_p E = 0$

- Note that the system is decoupled. K_D and K_p are diagonal, thus we can write the closed loop equation for each joint: $\ddot{e}_i + k_{Di}\dot{e}_i + k_{pi}e_i = 0$

  - This is an asymptotically stable system, and the error will decay to zero, meaning that tracking of reference is achieved. $x_i \to x_{di}$

RMIT UNIVERSITY

Content

- Introduction
- Control of Second Order Linear Systems
- Control-Law Partitioning
- Trajectory Following
- Nonlinear and Time-Varying Systems
- Nonlinear Control of Multi-Input-Multi-Output System
- Matlab Simulink Simulation

MATLAB Simulink Simulation

- The Matlab Simulink files are uploaded on Canvas.
- Run the files in the order of A.. B.. C.

Thank you!

Have a good evening.