

# **Ch10: Centralized Control - Motion Control – Part 1**

# Motion Control

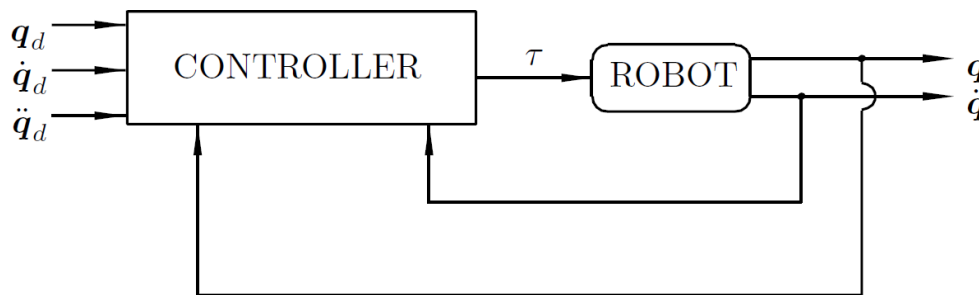
# Motion Control Objective

Given desired joint position  $\mathbf{q}_d(t) \in \mathbb{R}^n$ , velocity  $\dot{\mathbf{q}}_d(t) \in \mathbb{R}^n$ , and acceleration  $\ddot{\mathbf{q}}_d(t) \in \mathbb{R}^n$ , we wish to find joint torques/forces  $\boldsymbol{\tau} \in \mathbb{R}^n$  such that the joint position  $\mathbf{q}(t) \in \mathbb{R}^n$  follow (asymptotically)  $\mathbf{q}_d(t)$  accurately:

$$\lim_{t \rightarrow \infty} \mathbf{q}(t) = \mathbf{q}_d(t) \quad \Rightarrow \quad \lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0}$$

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t) \in \mathbb{R}^n$$

position error



$$\dot{\mathbf{e}}(t) = \dot{\mathbf{q}}_d(t) - \dot{\mathbf{q}}(t) \in \mathbb{R}^n$$

velocity error

The most common motion controllers:

- PD/PID Control
- PD with Feedforward Control
- PD Control with Gravity Compensation
- PD+ Control
- Inverse Dynamics Control

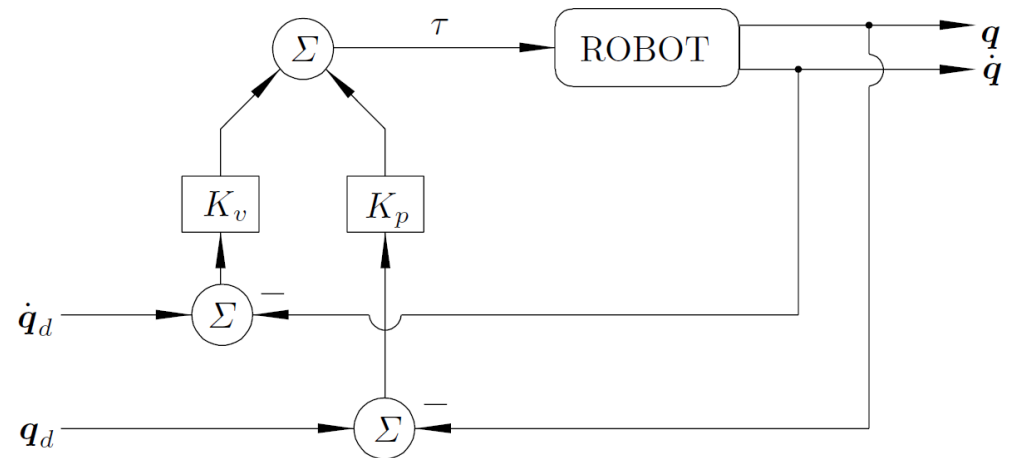
# PD/PID Control

# PD Control

The PD control law is given by  $\tau = K_p e + K_v \dot{e}$   $e = q_d - q$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices. If  $K_p = \text{diag}\{K_{p,i}\}$ ,  $K_v = \text{diag}\{K_{v,i}\}$ , the controller is called PD Independent Joint Control.

This is the simplest (linear) controller that may be used to control robot manipulators.



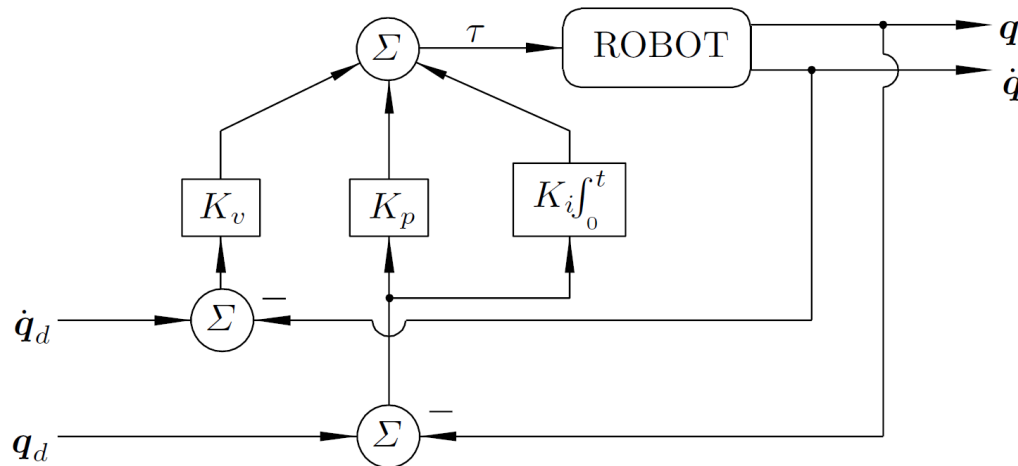
- For any  $K_p = K_p^T > 0$ ,  $K_v = K_v^T > 0$ , it is guaranteed that  $e(t)$  and  $\dot{e}(t)$  are bounded for all initial conditions.
- The error bound decreases, as  $K_{v,i}$  become larger (in case  $K_v = \text{diag}\{K_{v,i}\}$ ), however, large  $K_{v,i}$  can saturate the robot actuators.

# PID Control

The residual error at steady state due to gravity with PD control can be removed to some extent using the PID control law which is given by

$$\tau = K_p e + K_v \dot{e} + K_i \int_0^t e(\tau) d\tau \quad e = q_d - q$$

$K_p, K_v, K_i \in \mathbb{R}^{n \times n}$  (position, velocity, and integral gains) are symmetric positive definite matrices. If  $K_p = \text{diag}\{K_{p,i}\}$ ,  $K_v = \text{diag}\{K_{v,i}\}$ ,  $K_i = \text{diag}\{K_{i,i}\}$ , the controller is called PID Independent Joint Control.



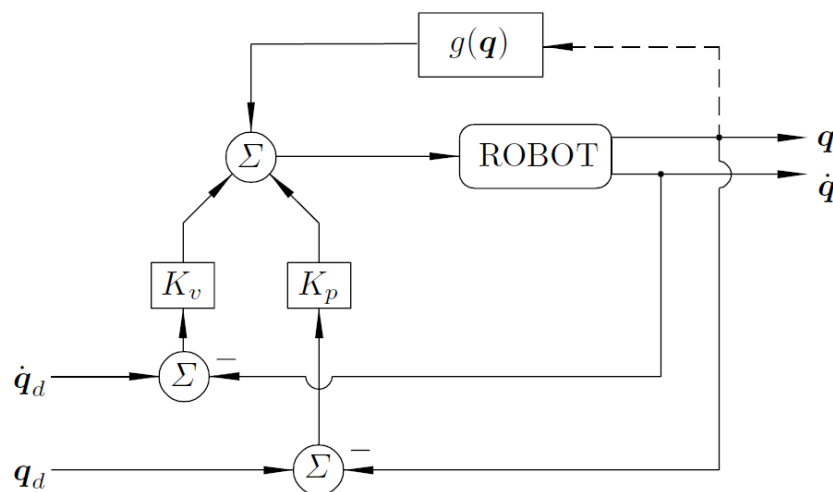
# PD Control with Gravity Compensation

# PD Control with Gravity Compensation

The PD control law with gravity compensation is given by  $\tau = K_p e + K_v \dot{e} + g(q)$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.

$$e = q_d - q$$



- In general, if  $q_d$  is not constant (i.e., motion control), then the controller guarantees bounded tracking errors  $e(t)$  about zero, but the error never goes exactly to zero.
- The error bound decreases, as the PD gains become larger (in case  $K_p = \text{diag}\{K_{p,i}\}$ ,  $K_v = \text{diag}\{K_{v,i}\}$ ).
- PD Controller with Gravity Compensation may behave better than a PID controller for the same PD gains. However, the integral term can reject other terms besides gravity (e.g., friction of some sorts).



# PD with Feedforward Control (PD+ Control)

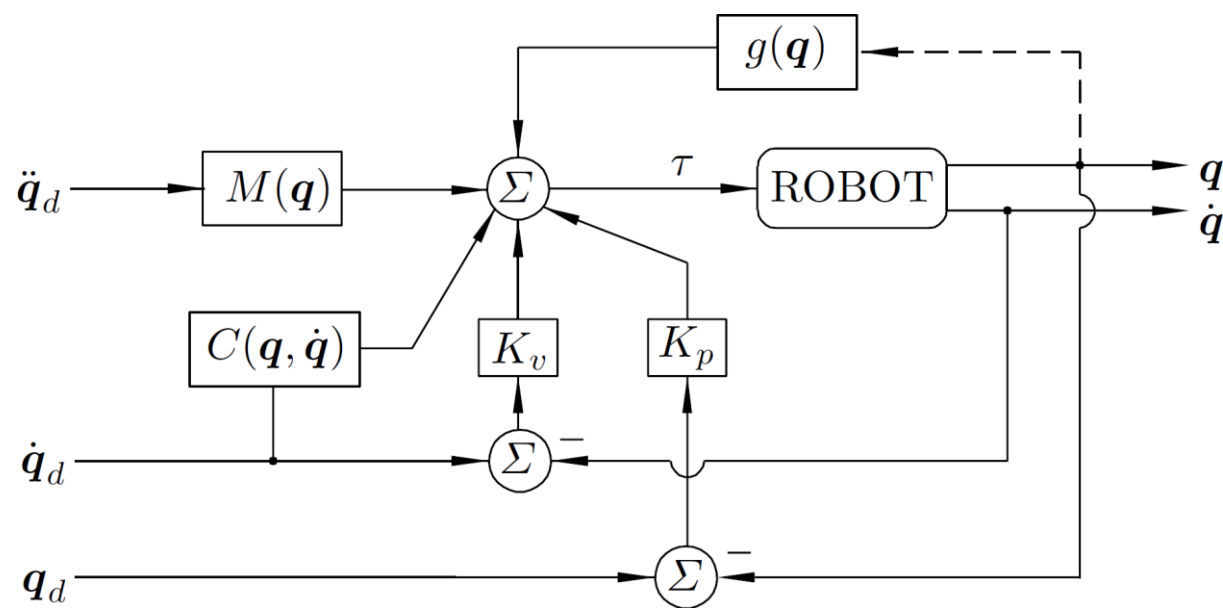
# PD with Feedforward Control (PD+ Control)

The PD with feedforward control law is given by

$$\tau = K_p e + K_v \dot{e} + M(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + g(q)$$

$$e = q_d - q$$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.



# PD with Feedforward Control (PD+ Control)

(cont.)

The closed-loop dynamic equation is derived as  $M(q)\ddot{e} + C(q, \dot{q})\dot{e} = -K_p e - K_v \dot{e}$

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ M(q)^{-1} [-K_p e - K_v \dot{e} - C(q, \dot{q})\dot{e}] \end{bmatrix} \quad q = q_d - e$$

The system is **nonautonomous**, and origin  $(e, \dot{e}) = \mathbf{0} \in \mathbb{R}^{2n}$  is the only equilibrium point.

Consider a Lyapunov function candidate as  $V(t, e, \dot{e}) = \frac{1}{2} \dot{e}^T M(q) \dot{e} + \frac{1}{2} e^T K_p e > 0$ .

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

Using closed-loop dynamic equation, and

$$\dot{e}^T \left[ \frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{e} = 0$$



$$\dot{V}(t, e, \dot{e}) = -\dot{e}^T K_v \dot{e} \leq 0$$

Thus, the origin  $(e, \dot{e}) = \mathbf{0}$  is stable.

- Using more advance theorems (e.g., Matrosov's theorem) or a different Lyapunov function, we can show that the origin is globally uniformly asymptotically stable.

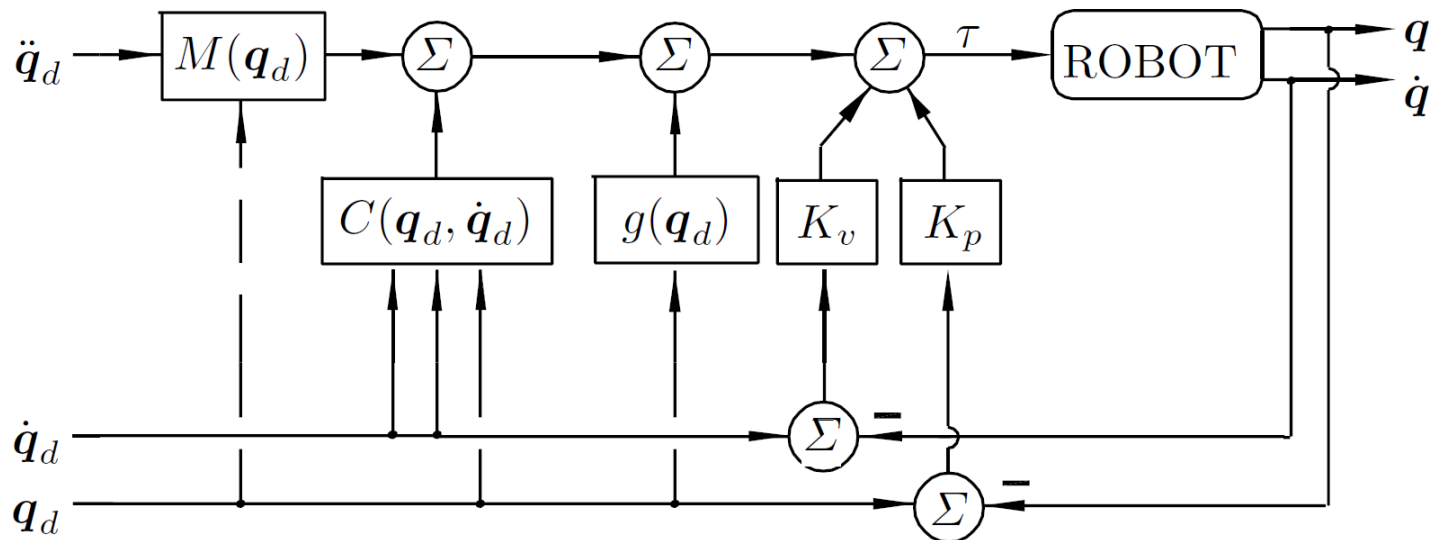
# PD with Desired Feedforward Control

# PD with Desired Feedforward Control

The PD with desired feedforward control law is given by

$$\tau = K_p e + K_v \dot{e} + \underbrace{M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d)}_{\text{For reducing the number of computations in real time implementation.}} \quad e = q_d - q$$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.



# PD with Desired Feedforward Control (cont.)

The closed-loop dynamic equation is derived as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{K}_p \mathbf{e} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}_d)$$

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}} \\ \mathbf{M}(\mathbf{q})^{-1} [-\mathbf{K}_p \mathbf{e} - \mathbf{K}_v \dot{\mathbf{e}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}} - \mathbf{h}(t, \mathbf{e}, \dot{\mathbf{e}})] \end{bmatrix} \quad \mathbf{q} = \mathbf{q}_d - \mathbf{e}$$

$$\mathbf{h}(t, \mathbf{e}, \dot{\mathbf{e}}) = [\mathbf{M}(\mathbf{q}_d) - \mathbf{M}(\mathbf{q})]\ddot{\mathbf{q}}_d + [\mathbf{C}(\mathbf{q}_d, \dot{\mathbf{q}}_d) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]\dot{\mathbf{q}}_d + [\mathbf{g}(\mathbf{q}_d) - \mathbf{g}(\mathbf{q})]$$

We can show that

- Origin  $(\mathbf{e}, \dot{\mathbf{e}}) = \mathbf{0} \in \mathbb{R}^{2n}$  is an equilibrium, independently of the gain matrices  $\mathbf{K}_p, \mathbf{K}_v$ .
- The number of equilibria of the system depends on the proportional gain  $\mathbf{K}_p$ .
- By choosing  $\mathbf{K}_p$  sufficiently large, then the system has a unique equilibrium at origin.
- By choosing  $\mathbf{K}_p, \mathbf{K}_v$  sufficiently large, the origin is globally uniformly asymptotically stable.

# Inverse Dynamics Control

# Inverse Dynamics Control (or Computed Torque Control)

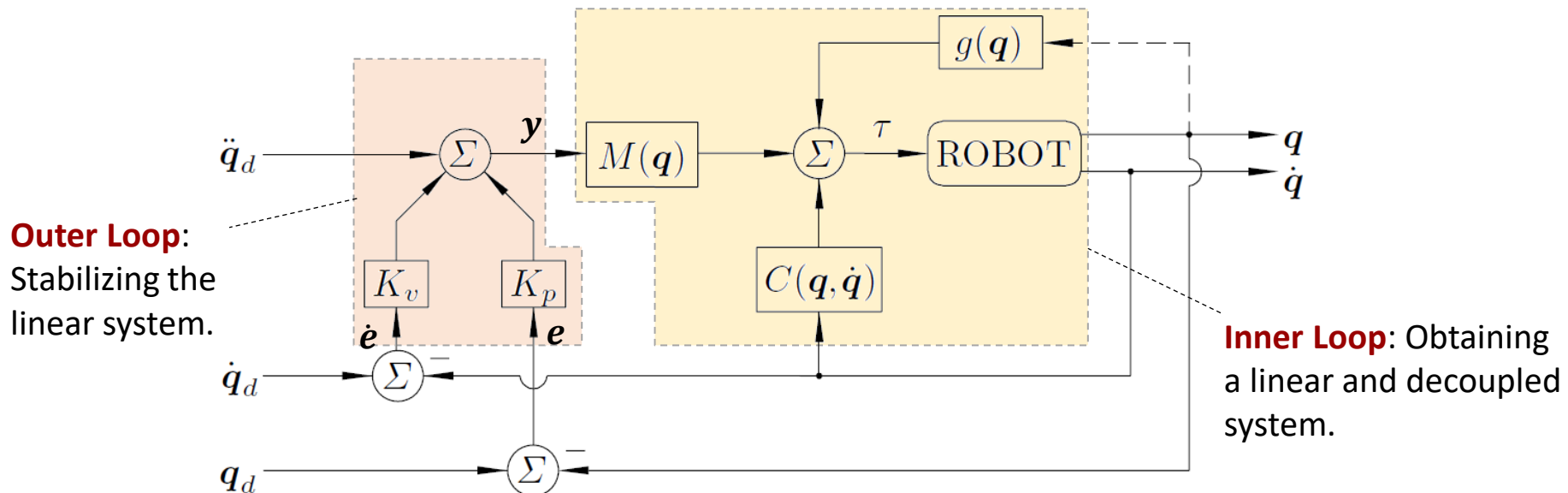
The idea of Inverse Dynamics control is to seek a nonlinear feedback control law which results in a linear closed-loop system. The control law is given by

$$\tau = M(q)y + C(q, \dot{q})\dot{q} + g(q) \qquad e = q_d - q$$

$$y = \ddot{q}_d + K_v \dot{e} + K_p e \quad (\rightarrow \text{PD control with feedforward acceleration})$$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.

This is a model-based motion control approach.





# Inverse Dynamics Control

The closed-loop dynamic equation is derived as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{M}(\mathbf{q})\mathbf{y} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \longrightarrow \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})\mathbf{y}$$

( $n$  decoupled linear double integrators)  $\ddot{\mathbf{q}} = \mathbf{y} \longrightarrow \ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}$

$$\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \mathbf{0} \longrightarrow \frac{d}{dt} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}} \\ -\mathbf{K}_p\mathbf{e} - \mathbf{K}_v\dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_n \\ -\mathbf{K}_p & -\mathbf{K}_v \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix}$$

The system is **linear** and **autonomous**, and the origin  $(\mathbf{e}, \dot{\mathbf{e}}) = \mathbf{0} \in \mathbb{R}^{2n}$  is the unique equilibrium point.

Let's introduce the constant  $\varepsilon \in \mathbb{R}_{++}$  satisfying  $\lambda_{\min}(\mathbf{K}_v) > \varepsilon > 0$ , thus,  $\lambda_{\min}(\mathbf{K}_v)\mathbf{x}^T\mathbf{x} > \varepsilon\mathbf{x}^T\mathbf{x}$ , and since  $\lambda_{\min}(\mathbf{K}_v)\mathbf{x}^T\mathbf{x} \leq \mathbf{x}^T\mathbf{K}_v\mathbf{x}$ , then  $\mathbf{x}^T(\mathbf{K}_v - \varepsilon\mathbf{I}_n)\mathbf{x} > 0 \forall \mathbf{x} \neq \mathbf{0} \in \mathbb{R}^n$ . This means  $\mathbf{K}_v - \varepsilon\mathbf{I}_n > 0$  and  $\mathbf{K}_p + \varepsilon\mathbf{K}_v - \varepsilon^2\mathbf{I}_n > 0$ . Now, a Lyapunov function candidate is

$$V(\mathbf{e}, \dot{\mathbf{e}}) = \frac{1}{2} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix}^T \begin{bmatrix} \mathbf{K}_p + \varepsilon\mathbf{K}_v & \varepsilon\mathbf{I}_n \\ \varepsilon\mathbf{I}_n & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix} = \frac{1}{2} [\dot{\mathbf{e}} + \varepsilon\mathbf{e}]^T [\dot{\mathbf{e}} + \varepsilon\mathbf{e}] + \frac{1}{2} \mathbf{e}^T [\mathbf{K}_p + \varepsilon\mathbf{K}_v - \varepsilon^2\mathbf{I}_n] \mathbf{e} > 0$$

$$\Rightarrow V(\mathbf{e}, \dot{\mathbf{e}}) = \frac{1}{2} \dot{\mathbf{e}}^T \dot{\mathbf{e}} + \frac{1}{2} \mathbf{e}^T [\mathbf{K}_p + \varepsilon\mathbf{K}_v] \mathbf{e} + \varepsilon \mathbf{e}^T \dot{\mathbf{e}} > 0$$

# Inverse Dynamics Control

$$\dot{V}(\mathbf{e}, \dot{\mathbf{e}}) = \ddot{\mathbf{e}}^T \dot{\mathbf{e}} + \mathbf{e}^T [\mathbf{K}_p + \varepsilon \mathbf{K}_v] \dot{\mathbf{e}} + \varepsilon \dot{\mathbf{e}}^T \dot{\mathbf{e}} + \varepsilon \mathbf{e}^T \ddot{\mathbf{e}} \quad \xrightarrow{\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0}}$$

$$\Rightarrow \dot{V}(\mathbf{e}, \dot{\mathbf{e}}) = -\dot{\mathbf{e}}^T [\mathbf{K}_v - \varepsilon \mathbf{I}_n] \dot{\mathbf{e}} - \varepsilon \mathbf{e}^T \mathbf{K}_p \mathbf{e} < 0$$

Thus, the origin  $(\mathbf{e}, \dot{\mathbf{e}}) = \mathbf{0}$  is globally asymptotically stable for any initial condition  $\mathbf{q}(0), \dot{\mathbf{q}}(0) \in \mathbb{R}^n$ :

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0} \quad \lim_{t \rightarrow \infty} \dot{\mathbf{e}}(t) = \mathbf{0}$$

$\Rightarrow$  Thus, the motion control objective is achieved.

**Note:** Since the closed-loop equation is linear and autonomous, the origin is globally exponentially stable.

**Note:** Friction at the joints may also affect the position error. Moreover, in the presence of bounded disturbance  $\boldsymbol{\tau}_{\text{dist}}(t)$ , error  $\mathbf{e}(t)$  remains bounded.

**Note:** This controller is a special case of the method of feedback linearization for nonlinear systems.

# Inverse Dynamics Control: Parameter Selection

$K_p$  and  $K_v$  may be chosen diagonal as:

$$K_p = \text{diag}\{K_{p,i}\} = \text{diag}\{\omega_{n,i}^2\}$$

$$K_v = \text{diag}\{K_{v,i}\} = \text{diag}\{2\zeta_i\omega_{n,i}\}$$

With this choice, the closed-loop equation is  $n$  **decoupled** 2nd-order linear ODEs. The natural frequency  $\omega_{n,i} \in \mathbb{R}$  determines the speed of response (the larger, the faster) and the damping ratio  $\zeta_i \in \mathbb{R}$  determines the existence of overshoot in joint error  $e(t)$ .

**Note 1:** It may be useful to select the desired responses at the end of the arm faster than near the base, where the masses that must be moved are heavier.

**Note 2:** It is undesirable for the robot to exhibit overshoot (e.g., since this could cause impact for paths near the workpiece surface). Therefore, the damping ratios are usually selected  $\zeta_i = 1$  to have a critically damped responses.

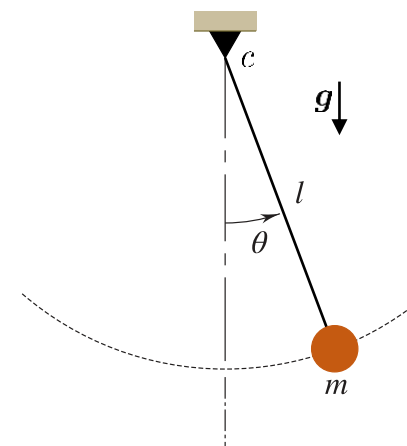
**Note 3:** If the gains  $K_{p,i}, K_{v,i}$  are too large, the control torque may reach its upper limits and saturate some or all of the actuators.

**Note 4:** The implementation of this control scheme requires the real-time computation of  $M(q)$  and  $C(q, \dot{q})$ , and  $g(q)$ , which may be computationally expensive.

# Inverse Dynamics Control: Example

Consider the equation of a pendulum of length  $l$  and mass  $m$  concentrated at its tip, subject to the action of gravity  $g$  and to which is applied a torque  $\tau$  at the axis of rotation. Drive the inverse dynamics control law.

$$ml^2\ddot{\theta} + mgl\sin\theta = \tau$$



# Approximate Inverse Dynamics Control

In some cases,  $\mathbf{M}(\mathbf{q})$  is not known exactly (e.g., unknown payload mass), or  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$  is not known exactly (e.g., unknown friction terms). Then  $\hat{\mathbf{M}}(\mathbf{q})$  and  $\hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$  could be the best estimate we have for these terms. On the other hand, we might simply wish to avoid computing  $\mathbf{M}(\mathbf{q})$  and  $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$  at each short sample time and instead compute more simpler  $\hat{\mathbf{M}}(\mathbf{q})$  and  $\hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$ . The approximate inverse dynamics control law is given by

$$\boldsymbol{\tau} = \hat{\mathbf{M}}(\mathbf{q})\mathbf{y} + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$$

$$\mathbf{y} = \ddot{\mathbf{q}}_d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}$$

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}$$

It can be shown that even if  $\hat{\mathbf{M}} \neq \mathbf{M}$  and  $\hat{\mathbf{h}} \neq \mathbf{h}$ , the performance of the controller can be acceptable if the symmetric positive definite matrices  $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{n \times n}$  are selected large enough.

# PID Inverse Dynamics Control

In the presence of unknown constant disturbances ( $\tau_{\text{dist}}$ ), PD control gives a nonzero steady-state error. Thus, by including an integrator (I) in the outer loop, we can achieve a PID inverse dynamics controller as

$$\tau = M(q)y + C(q, \dot{q})\dot{q} + g(q)$$

$$e = q_d - q$$

$$y = \ddot{q}_d + K_v \dot{e} + K_p e + K_i \int_0^t e(\tau) d\tau$$

$K_p, K_v, K_i \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.

The closed-loop dynamic equation is derived as  $\ddot{e} + K_v \dot{e} + K_p e + K_i e = 0$

**Note:** If control gains are diagonal ( $K_p = \text{diag}\{K_{p,i}\}$ ,  $K_v = \text{diag}\{K_{v,i}\}$ ,  $K_i = \text{diag}\{K_{i,i}\}$ ), for closed-loop stability, based on Routh-Hurwitz criterion, we require that

$$K_{i,i} < K_{p,i} K_{v,i}$$

# Task Space Control

# Task Space Control

Since the robot interacts with the external environment and objects in it, it may be more convenient to express the motion as a trajectory of the end-effector in task space. If the end-effector trajectory be specified by  $\mathbf{x}_d(t) \in \mathbb{R}^r$  or  $\mathbf{T}_d(t) \in SE(3)$ ,  $\mathbf{v}_d(t) \in \mathbb{R}^6$ :

**Method 1:** Converting a desired trajectory in task space to joint-space and proceed with joint-space control.

$$\left\{ \begin{array}{l} \mathbf{q}_d(t) = \mathbf{f}^{-1}(\mathbf{x}_d(t)) \\ \dot{\mathbf{q}}_d(t) = \bar{\mathbf{f}}^{-1}(\dot{\mathbf{x}}_d(t)) \\ \ddot{\mathbf{q}}_d(t) = \bar{\bar{\mathbf{f}}}^{-1}(\ddot{\mathbf{x}}_d(t)) \end{array} \right. \quad \text{or} \quad \left\{ \begin{array}{l} \mathbf{q}_d(t) = \mathbf{T}^{-1}(\mathbf{T}_d(t)) \\ \dot{\mathbf{q}}_d(t) = \mathbf{J}^\dagger(\mathbf{q}_d(t))\mathbf{v}_d(t) \\ \ddot{\mathbf{q}}_d(t) = \mathbf{J}^\dagger(\mathbf{q}_d(t))(\dot{\mathbf{v}}_d(t) - \dot{\mathbf{J}}(\mathbf{q}_d(t))\dot{\mathbf{q}}_d(t)) \end{array} \right.$$

**Drawback:** This requires significant computing power. To reduce the computational load, we can first compute  $\mathbf{q}_d(t)$ , then perform a numerical differentiation to compute  $\dot{\mathbf{q}}_d(t)$  and  $\ddot{\mathbf{q}}_d(t)$ .



# Task Space Control (cont.)

**Method 2:** Developing a control law in the task space using the robot dynamics expressed either in joint space or task space.

$$\begin{array}{l}
 \downarrow \\
 \tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})
 \end{array}
 \quad
 \begin{array}{l}
 \nearrow \\
 \begin{array}{l}
 \mathcal{F} = \mathbf{M}_c(\mathbf{q})\dot{\mathcal{V}} + \mathbf{C}_c(\boldsymbol{\theta}, \mathcal{V})\mathcal{V} + \mathbf{g}_c(\mathbf{q}) \\
 \tau = \mathbf{J}^T(\mathbf{q})\mathcal{F}
 \end{array}
 \end{array}
 \quad
 \begin{array}{l}
 \text{(or)} \\
 \begin{array}{l}
 \mathbf{F} = \mathbf{M}_c(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{C}_c(\mathbf{q}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{g}_c(\mathbf{q}) \\
 \tau = \mathbf{J}_a^T(\mathbf{q})\mathbf{F}
 \end{array}
 \end{array}$$

## Difficulties:

- Task space controllers always require computation of manipulator Jacobian. Thus, the presence of **singularities** and/or **redundancy** influences the Jacobian, and the induced effects are somewhat difficult to handle with a task space controller (e.g., we must use Jacobian pseudoinverse or other redundancy handling techniques).
- Expressing the **joint limits** is easier in joint space than task space.

- Here, let's consider a nonredundant manipulator avoiding singularities to develop the control laws.

# Position Control: PD Control with Gravity Compensation

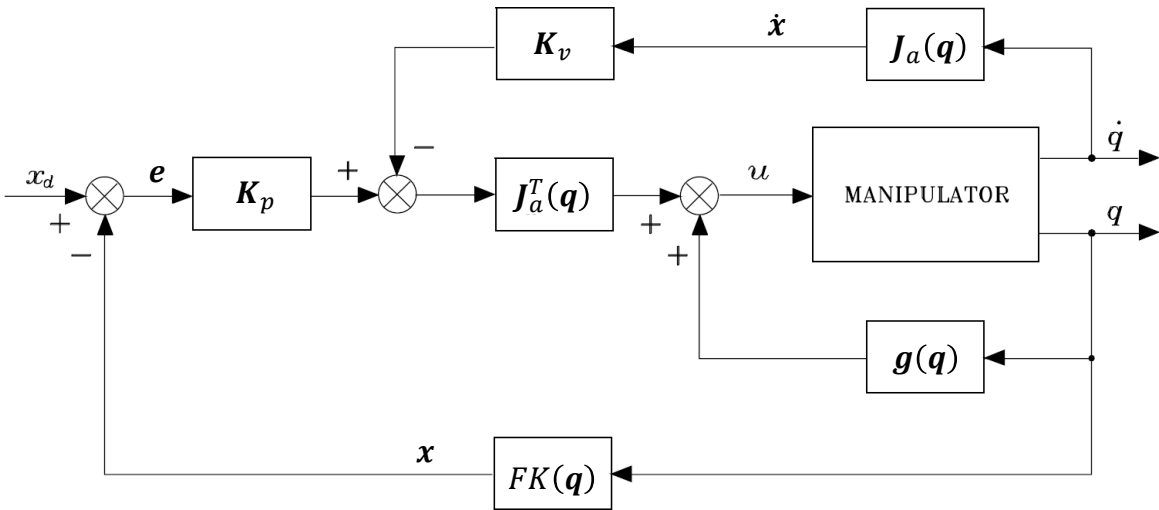
(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame)

Given a constant end-effector pose  $x_d$ , the PD control law with gravity compensation is given by

$$\tau = J_a^T(q)(K_p e - K_v \dot{x}) + g(q)$$

$$e = x_d - x$$
$$\dot{x} = J_a(q)\dot{q}$$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.



**Note:** If measurements of  $x$  and  $\dot{x}$  are made directly in the task space,  $FK(q)$  and  $J_a(q)$  are not required; however, it is necessary to measure  $q$  to update both  $J_a^T(q)$  and  $g(q)$  on-line.

# Position Control: PD Control with Gravity Compensation

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame) (cont.)

The closed-loop dynamic equation is derived as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{J}_a^T(\mathbf{q})\mathbf{K}_p\mathbf{e} - \mathbf{J}_a^T(\mathbf{q})\mathbf{K}_v\mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

$$\frac{d}{dt} \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} -\dot{\mathbf{x}} \\ \mathbf{M}(\mathbf{q})^{-1}(\mathbf{J}_a^T(\mathbf{q})\mathbf{K}_p\mathbf{e} - \mathbf{J}_a^T(\mathbf{q})\mathbf{K}_v\mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}) \end{bmatrix} \quad \mathbf{x} = \mathbf{x}_d - \mathbf{e}$$

The system is **autonomous** (since  $\mathbf{x}_d$  is constant), and it has a unique equilibrium point at origin  $(\mathbf{e}, \dot{\mathbf{q}}) = \mathbf{0} \in \mathbb{R}^{2n}$ .

Consider a Lyapunov function candidate as  $V(\dot{\mathbf{q}}, \mathbf{e}) = \underbrace{\frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}}_{\text{Kinetic energy of the arm}} + \frac{1}{2} \mathbf{e}^T \mathbf{K}_p \mathbf{e} > 0$  (PD)

$$\dot{V}(\dot{\mathbf{q}}, \mathbf{e}) = \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{\mathbf{M}}(\mathbf{q}) \dot{\mathbf{q}} + \dot{\mathbf{e}}^T \mathbf{K}_p \mathbf{e} \quad \forall \mathbf{e}, \dot{\mathbf{q}} \neq \mathbf{0}$$

$$\downarrow \quad \dot{\mathbf{q}}^T \left[ \frac{1}{2} \dot{\mathbf{M}} - \mathbf{C} \right] \dot{\mathbf{q}} = 0$$

$$\dot{V}(\dot{\mathbf{q}}, \mathbf{e}) = -\dot{\mathbf{x}}^T \mathbf{K}_v \dot{\mathbf{x}} \leq 0 \quad \Rightarrow$$

Using LaSalle (invariant set) theorem, the origin  $(\mathbf{e}, \dot{\mathbf{q}}) = \mathbf{0}$  is globally asymptotically stable.

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{0}$$

# Motion Control: Inverse Dynamics Control

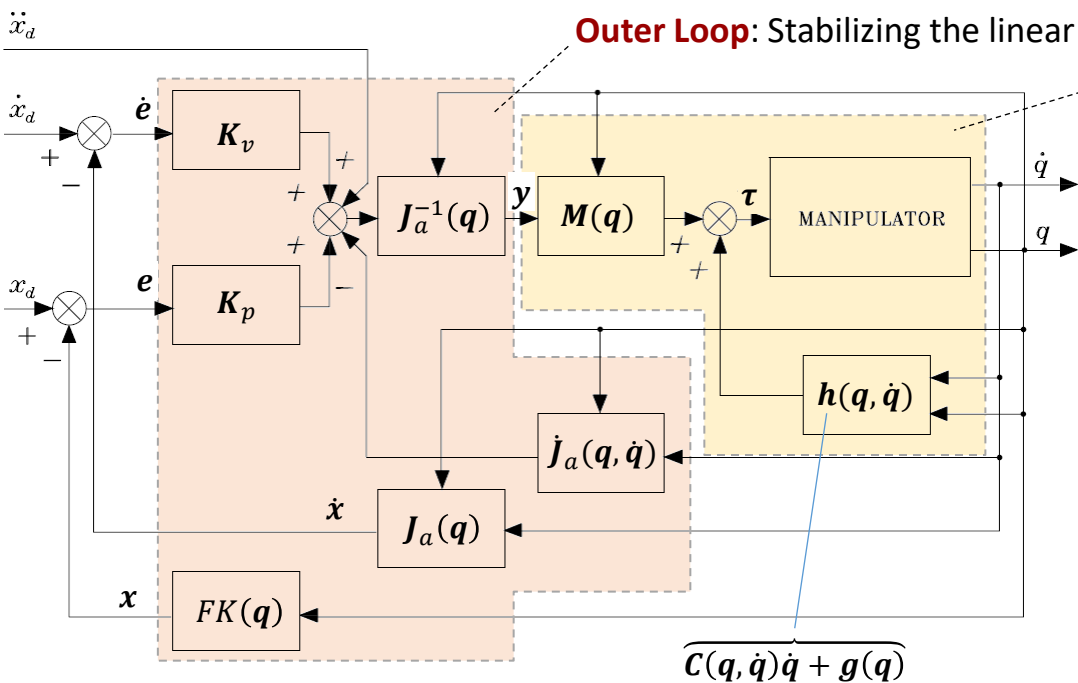
(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame)

The inverse dynamics control law is given by

$$\tau = M(q)y + C(q,\dot{q})\dot{q} + g(q)$$
$$y = J_a^{-1}(q)(\ddot{x}_d + K_v\dot{e} + K_p e - \dot{J}_a(q,\dot{q})\dot{q})$$

$$e = x_d - x$$

$K_p, K_v \in \mathbb{R}^{n \times n}$  are symmetric positive definite matrices.



# Motion Control: Inverse Dynamics Control

(Based on Robot Dynamics in J-Space & Min. Coord. Rep. of EE Frame) (cont.)

The closed-loop dynamic equation is derived as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = M(q)[J_a^{-1}(q)(\ddot{x}_d + K_v\dot{e} + K_pe - \dot{J}_a(q, \dot{q})\dot{q})] + C(q, \dot{q})\dot{q} + g(q)$$



$$\ddot{q} = J_a^{-1}(q)(\ddot{x}_d + K_v\dot{e} + K_pe - \dot{J}_a(q, \dot{q})\dot{q}) = y$$



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

$$\ddot{e} + K_v\dot{e} + K_pe = 0 \quad \longrightarrow \quad \frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ -K_pe - K_v\dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

The system is **decoupled**, **linear** and **autonomous**, and the origin  $(e, \dot{e}) = \mathbf{0} \in \mathbb{R}^{2n}$  is the unique equilibrium point.

Similar to Inverse Dynamics Control in joint space, the origin  $(e, \dot{e}) = \mathbf{0}$  is globally asymptotically (exponentially) stable for any initial condition  $q(0), \dot{q}(0) \in \mathbb{R}^n$ :

$$\lim_{t \rightarrow \infty} e(t) = \mathbf{0}$$

# A Remark on Computation of Error

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_R \\ \mathbf{e}_p \end{bmatrix} = \begin{bmatrix} \mathbf{e}_R \\ \mathbf{p}_d - \mathbf{p} \end{bmatrix}, \quad \dot{\mathbf{e}} = \begin{bmatrix} \dot{\mathbf{e}}_R \\ \dot{\mathbf{e}}_p \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{e}}_R \\ \dot{\mathbf{p}}_d - \dot{\mathbf{p}} \end{bmatrix}$$

Computation of  $\mathbf{e}_R$ ,  $\dot{\mathbf{e}}_R$  depends on the orientation representation of end-effector frame:

**(1) Euler Angles:**

Method 1:  $\mathbf{e}_R = \boldsymbol{\phi}_d - \boldsymbol{\phi}$

Method 2:  $\mathbf{e}_R = \text{EulerAngles}(\mathbf{R}^T \mathbf{R}_d)$

$\dot{\mathbf{e}}_R = \dot{\boldsymbol{\phi}}_d - \dot{\boldsymbol{\phi}}$ 

$\boldsymbol{\phi} \in \mathbb{R}^3$   
 $\mathbf{R} \in SO(3)$

Assumption: There is no kinematic or representation singularities.

## (2) Angle and Axis (Exponential Coordinates):

Method 1:  $\mathbf{R}_e = \mathbf{R}^T \mathbf{R}_d$ ,  $\log(\mathbf{R}_e) = [\hat{\boldsymbol{\omega}}]\theta$ ,

(in EE frame)

$\mathbf{e}_R := \hat{\boldsymbol{\omega}}\theta$  (in EE frame)  
 $\mathbf{e}_R := \mathbf{R}\hat{\boldsymbol{\omega}}\theta$  (in base frame)

# A Remark on Computation of Error

Method 2:  $\mathbf{R}_e = \mathbf{R}^T \mathbf{R}_d$ ,  $\text{UnitQuat}(\mathbf{R}_e) = \begin{bmatrix} \cos \theta/2 \\ \sin \theta/2 \hat{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$

$\downarrow$   
(in EE frame)

$\mathbf{e}_R := \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$  (in EE frame)

$\mathbf{e}_R := \mathbf{R} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$  (in base frame)

- If  $\boldsymbol{\omega}$  and  $\boldsymbol{\omega}_d$  are measured/given in the base frame:  $\dot{\mathbf{e}}_R := \boldsymbol{\omega}_d - \boldsymbol{\omega}$

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{\mathbf{e}}_R \\ \dot{\mathbf{e}}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d - \boldsymbol{\omega} \\ \dot{\mathbf{p}}_d - \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d \\ \dot{\mathbf{p}}_d \end{bmatrix} - \begin{bmatrix} \boldsymbol{\omega} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_d \\ \dot{\mathbf{p}}_d \end{bmatrix} - \mathbf{J}_g \dot{\mathbf{q}}$$

# Motion Control: Inverse Dynamics Control

(Based on Robot Dynamics in T-Space)

Inverse dynamics control law when end-effector trajectory is specified by  $T_d(t) \in SE(3)$ ,  $\mathbf{v}_d(t) \in \mathbb{R}^6$ :

$$\boldsymbol{\tau} = J^T(\mathbf{q})(\mathbf{M}_c(\mathbf{q})\mathbf{y} + \mathbf{C}_c(\boldsymbol{\theta}, \mathbf{v})\mathbf{v} + \mathbf{g}_c(\mathbf{q}))$$

PD:

$$\mathbf{y} = \underbrace{\frac{d}{dt}([\text{Ad}_{T^{-1}T_d}]\mathbf{v}_d)}_{\text{Feedforward acceleration Expressed in the actual EE frame } T.} + \underbrace{K_p \mathbf{S}_e \theta_e}_{\text{Configuration Error Expressed in the actual EE frame } T.} + \underbrace{K_d \mathbf{v}_e}_{\text{Twist Error Expressed in the actual EE frame } T.}$$

$$[\mathbf{S}_e]\theta_e = \log(T^{-1}T_d)$$

$$\mathbf{v}_e = [\text{Ad}_{T^{-1}T_d}]\mathbf{v}_d - \mathbf{v}$$

PID:

$$\mathbf{y} = \frac{d}{dt}([\text{Ad}_{T^{-1}T_d}]\mathbf{v}_d) + K_p \mathbf{S}_e \theta_e + K_d \mathbf{v}_e + K_i \int \mathbf{S}_e \theta_e(t) dt$$



# Velocity Control or Kinematic Control

# Velocity Control in Joint Space

Due to the presence of a low-level control loop, which ‘ideally’ imposes any desired input velocity, the velocity control is feasible.

By having  $\theta_d(k\Delta t)$  at each discrete timestep  $k$ , we can control the joint velocities  $\dot{\theta}$  during each time interval  $[(k-1)\Delta t, k\Delta t]$  as follows

$$\dot{\theta} = (\theta_d(k\Delta t) - \theta((k-1)\Delta t))/\Delta t$$

This amounts to a feedback controller since the desired new joint angles  $\theta_d(k\Delta t)$  are being compared with the most recently measured actual joint angles  $\theta((k-1)\Delta t)$  to calculate the required joint velocities.

# Velocity Control in Task Space

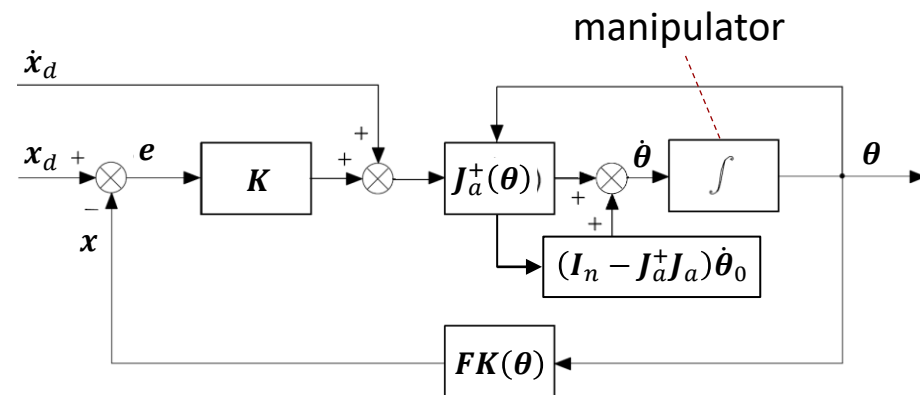
The structure of the velocity-level inverse kinematics algorithm based on Jacobian (pseudo-)inverse method can be conceptually adopted for a simple robot control technique provided that the integrator is regarded as a simplified model of the robot, thanks to the presence of single joint local servos, which ensure a more or less accurate reproduction of the velocity commands.

$$\dot{\theta} = J_a^+(\dot{x}_d + Ke) + \underbrace{(I_n - J_a^+ J_a) \dot{\theta}_0}_{\text{For Exploiting Redundant DOFs}}$$

$$e = x_d - x$$

$$\forall \dot{\theta}_0 \in \mathbb{R}^n$$

For Exploiting  
Redundant DOFs



**Note:** This technique yields satisfactory performance only when one does not require too fast motions or rapid accelerations.