



# Chap.5

## Control of Robot Manipulators.

### ■ 5.1 Introduction

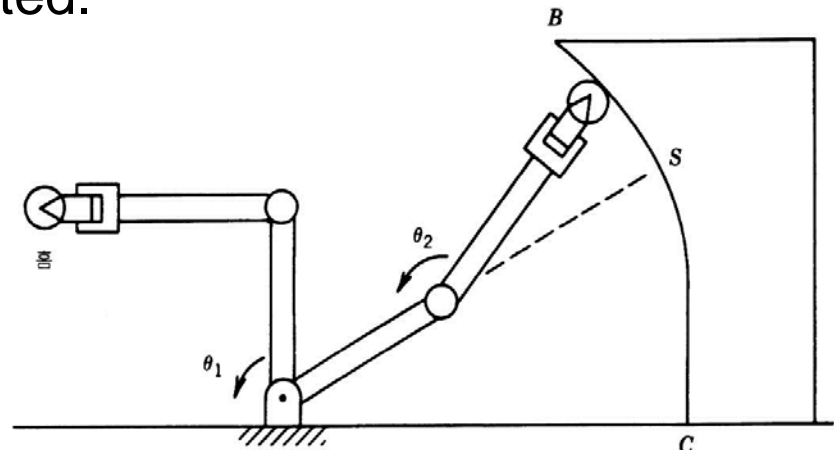
How can we control the robot manipulators?

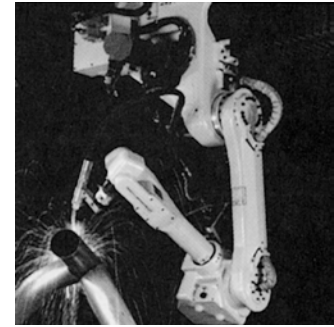
The control problem consists of

- (1) obtaining dynamic models of the manipulator
- (2) using these models to determine control laws or strategies to achieve the desired system response and performance

If the manipulator dynamics is obtained completely, the control input torque of manipulator is calculated.

=> Computed Torque Technique





- The movement of a robot arm

- 1) Gross motion control

A robot arm moves from an initial position/orientation to the vicinity of the desired target position/orientation along a planned trajectory.

- 2) Fine motion control

The end-effector of the arm dynamically interacts with the object using sensory feedback information to complete the task.

The traditional industrial approaches to robot arm control system design treated each joint of the robot arm as a simple joint servo mechanism.

=> The result reduced servo response speed and damping, limiting the precision and speed of the end-effector and making it appropriate only for limited-precision task.



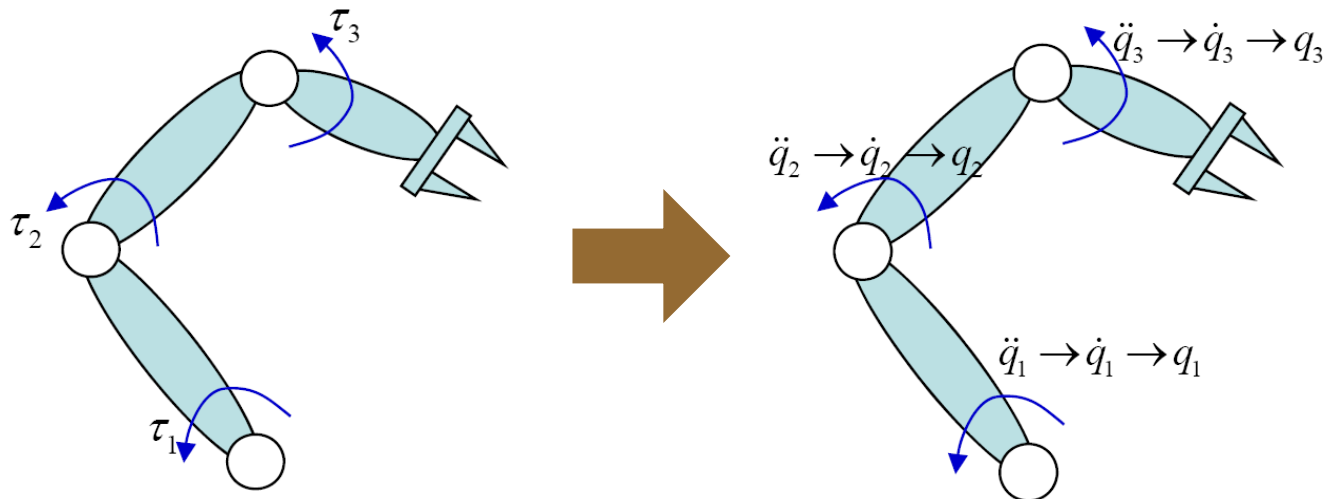
## ROBOT DYNAMICS AND CONTROL SIMULATION

**Robot dynamics Simulation:** Position / attitude / velocity changes of joints

- Robot dynamic equation of motion:

$$H\ddot{q} + C(q, \dot{q}) + G(q) = \tau(t)$$

$$\left( \Leftrightarrow \sum_{j=1}^n H_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i = \tau_i, \quad i = 1 \sim n \right)$$





## ***ROBOT DYNAMICS***

### ***AND CONTROL SIMULATION***

- Calculation of the trajectory of the robot for a given joint torque  $\tau(t)$  profile (Forward dynamics)

$$H\ddot{q} = \tau - C(q, \dot{q}) - G(q)$$

- $\ddot{q} = H^{-1}(\tau - C(q, \dot{q}) - G(q))$

- Perform a numerical integration for differential equations :  
(Ex.) RK4 Algorithm
- Obtain position trajectory  $q(t)$  and velocity trajectory  $\dot{q}(t)$

- How generated torque of each joint  $\tau(t)$  in the simulation?
  - Method that suggest random function as Torque profile
  - Torque profile used to input to control to tracking a specific trajectory



## AND CONTROL SIMULATION

**Robot dynamics Simulation:** Position / attitude / velocity changes of end-effector

Robot dynamic equation

$$H\ddot{q} + C(q, \dot{q}) + G(q) = \tau(t)$$

The diagram shows a large yellow arrow pointing from the robot dynamic equation to a set of joint acceleration vectors. A smaller yellow arrow labeled 'RK4 integration' points from the acceleration vectors to a set of joint velocity and position vectors. To the right of these vectors is a large curly brace containing the text 'Trajectory of Joints (position & velocity)'.

$$\begin{Bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{Bmatrix} \xrightarrow{\text{RK4 integration}} \begin{Bmatrix} \dot{q}_1, q_1 \\ \dot{q}_2, q_2 \\ \vdots \\ \dot{q}_n, q_n \end{Bmatrix} \cdot \left( \begin{array}{l} \text{Trajectory of Joints} \\ \text{(position \& velocity)} \end{array} \right)$$

(1) Position/attitude of End-effector

- Position:  $(p_x, p_y, p_z)$
- Orientation:  $(\phi, \theta, \psi)$

The diagram shows a large yellow arrow pointing from the joint trajectories to a large curly brace. Inside the brace is the text 'Position kinematics' followed by the Denavit-Hartenberg transformation matrix  ${}^0T_n$  and its decomposition into individual joint transformations  $A_1(q_1)A_2(q_2)A_3(q_3)\dots A_n(q_n)$ . A smaller yellow arrow points from this brace to the final end-effector position and orientation.

$$\left( \begin{array}{l} \bullet \text{ Position kinematics} \\ {}^0T_n = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = A_1(q_1)A_2(q_2)A_3(q_3)\dots A_n(q_n) \end{array} \right)$$



# ROBOT DYNAMICS AND CONTROL SIMULATION

Robot dynamic equation  
 $H\ddot{q} + C(q, \dot{q}) + G(q) = \tau(t)$

$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} \xrightarrow[\text{integration}]{\text{RK4}} \begin{bmatrix} \dot{q}_1, q_1 \\ \dot{q}_2, q_2 \\ \vdots \\ \dot{q}_n, q_n \end{bmatrix} : \begin{pmatrix} \text{Trajectory of Joints} \\ \text{(position \& velocity)} \end{pmatrix}$

(2) Velocity of End-effector  
• Velocity:  $(v_x, v_y, v_z)$   
• RPY rate:  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$

• Velocity kinematics  
 $\dot{\mathbf{p}} = [\mathbf{J}] \dot{\mathbf{q}} = [\mathbf{J}_P \quad \mathbf{J}_O]^T \dot{\mathbf{q}}$   
 $\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathbf{J}_P \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}, \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \mathbf{J}_O \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}$



# ROBOT DYNAMICS

## AND CONTROL SIMULATION

### Robot Trajectory (Task Space)

#### 1) Position trajectory

[*position*] end-effector :  $x_e(t), y_e(t), z_e(t)$

[*Orientation*] :  $\phi(t), \theta(t), \psi(t)$

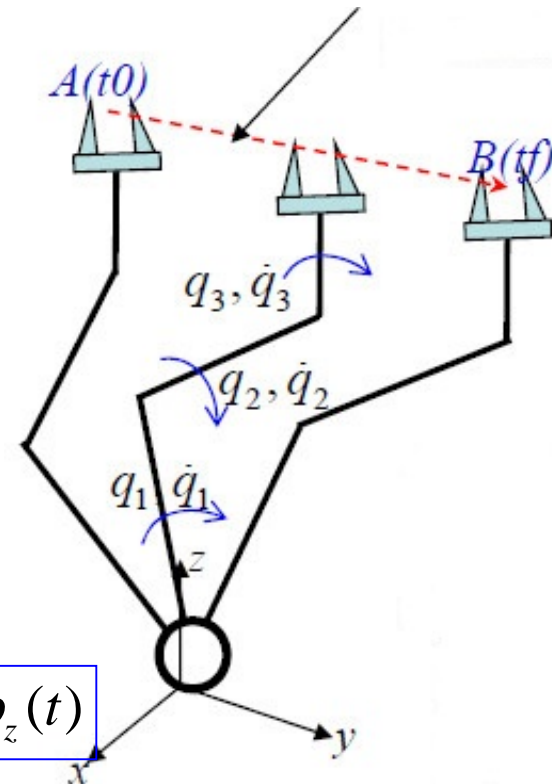
#### 2) Velocity trajectory

[*Linear velocity*] end-effector :  $\dot{x}_e(t), \dot{y}_e(t), \dot{z}_e(t)$

[*Angular velocity*] (Orientation rate):

or (x,y,z)-axis:  $\omega_x(t), \omega_y(t), \omega_z(t)$

Robot trajectory (A  $\rightarrow$  B)  
: Position profile & velocity  
profile





## ROBOT DYNAMICS

## AND CONTROL SIMULATION

### Task space trajectory $\rightarrow$ Joint space trajectory

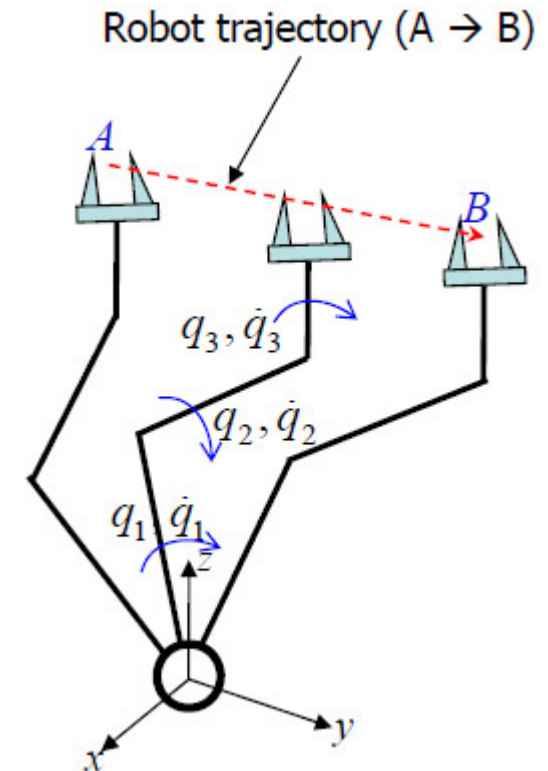
- How to find Joint trajectory

#### 1) Inverse kinematics:

Figure out the joint angle each sampling time for end-effector's position/appearance trajectory  
 $\rightarrow$  Input for position command of robot joint position control loop

#### 2) Inverse differential kinematics:

Figure out the angle for end-effector's linear velocity/Orientation rate trajectory  
 $\rightarrow$  Find out joint angle by an integrating







## ROBOT DYNAMICS

## AND CONTROL SIMULATION

$$\dot{p} = [J]\dot{q} \rightarrow \dot{q} = [J]^{-1} \dot{p} \Leftrightarrow \begin{Bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \Rightarrow \begin{Bmatrix} \dot{q}_{1d} \\ \dot{q}_{2d} \\ \dots \\ \dot{q}_{nd} \end{Bmatrix}$$

- Relationship of the angular velocity  $(\omega_x, \omega_y, \omega_z)$  and RPY rate  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$  on the reference coordinate system  $(x, y, z)$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 & -s\phi & c\theta c\phi \\ 0 & c\phi & c\theta s\phi \\ 1 & 0 & s\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$



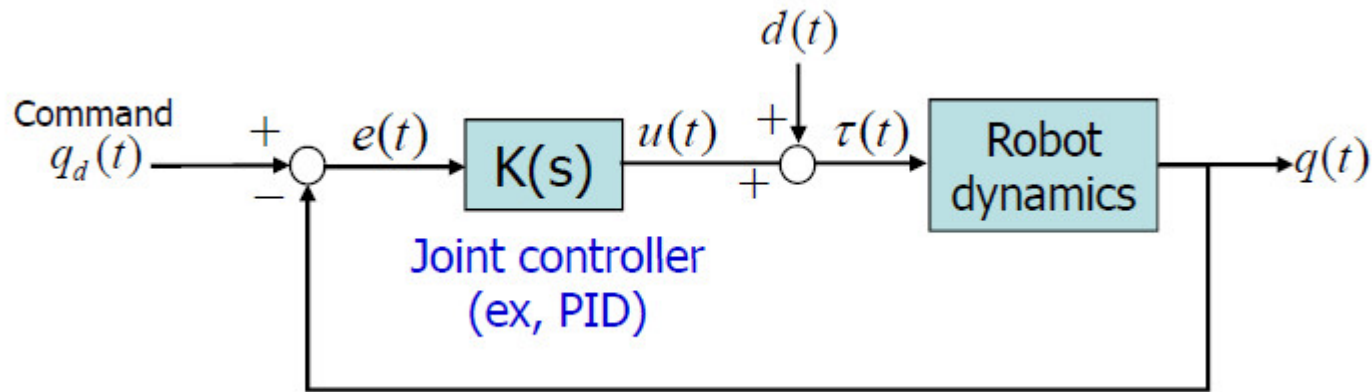
# ***ROBOT DYNAMICS AND CONTROL SIMULATION***

## **Joint Position Control**

- Loop configure of position control for each joint
  - Task space trajectory  $\rightarrow$  Joint space trajectory  $\rightarrow$  desired command of Joint Position control loop
  - Feedback robot dynamics output(joint angle)
  - Calculation joint angle error (e)
  - Calculation input to control by joint control algorithm (ex, PID control)
  - Input to control(u) + External disturbance(d)  $\rightarrow$  Torque of joint input



# ***ROBOT DYNAMICS AND CONTROL SIMULATION***



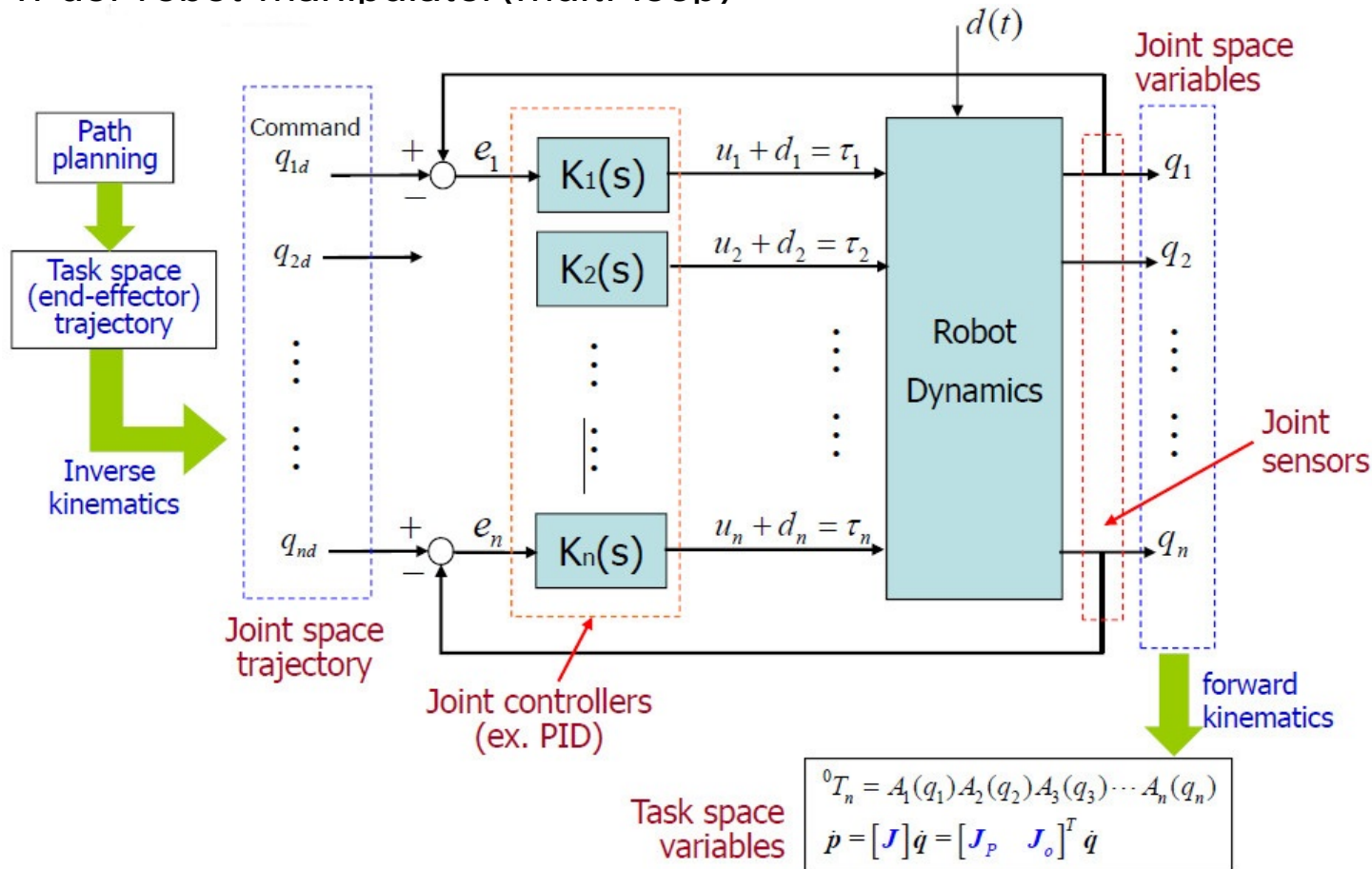
- Joint angle error  $e(t) = q_d(t) - q(t)$
- Joint velocity error :  $\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$
- PID control :  $K(s) = \left( k_p + k_D s + k_I \frac{1}{s} \right) e(s) \rightarrow u(t) = k_p e(t) + k_D \dot{e}(t) + k_I \int e(t) dt$
- Input torque :  $\tau(t) = u(t) + d(t)$



## AND CONTROL SIMULATION

### Independent Joint Control (IJC)

- $n$ -dof robot manipulator(multi-loop)





## ● Robot arm control

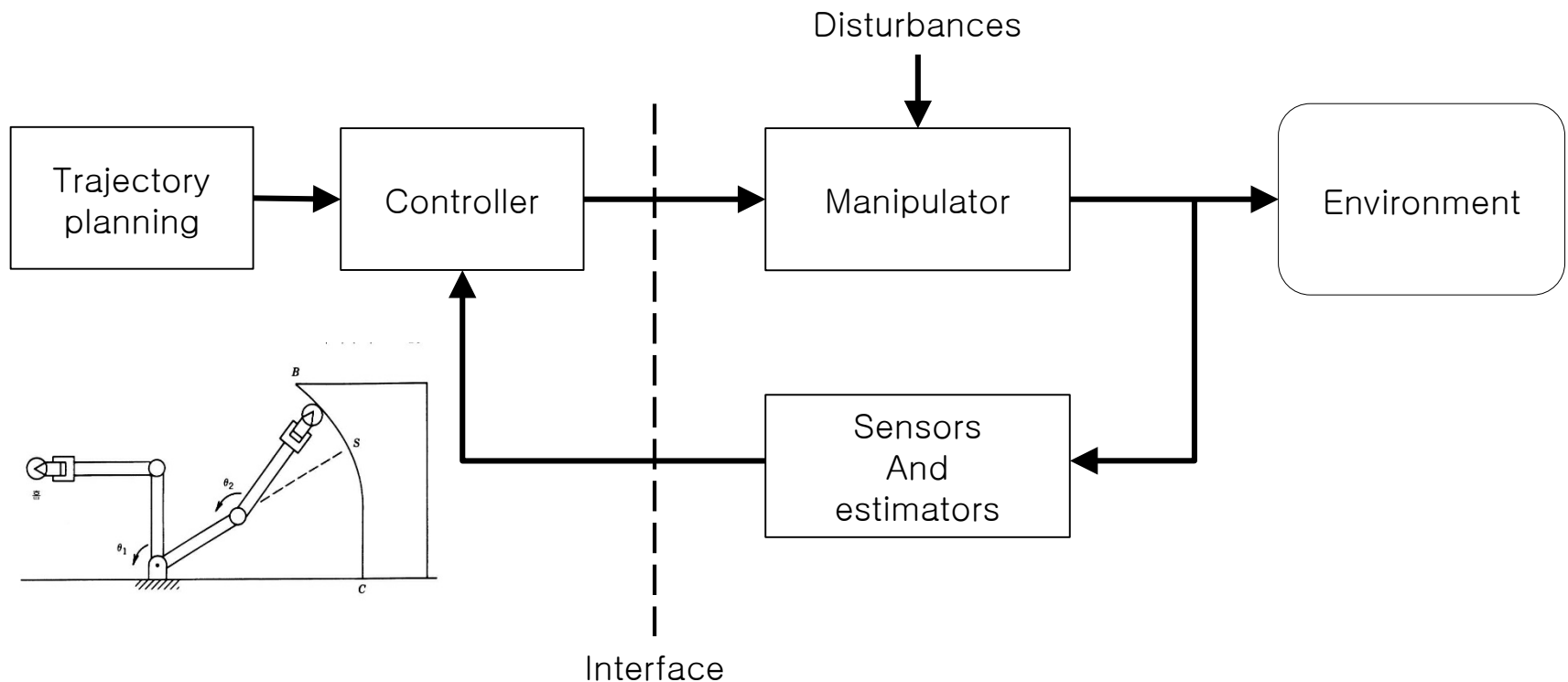


Figure 5.1 Basic control blocs diagram for robot manipulators



Motion Control is classified into the following three major categories

1. Joint motion Controls

- Joint servomechanism
- Computed torque technique (계산 토크법)
- Minimum time control
- Variable structure control

2. Resolved motion control ( cartesian space control ) (분해 동작제어)

- Resolved motion rate control.
- Resolved motion acceleration control.
- Resolved motion force control.

3. Adaptive control.

- Model-referenced adaptive control.
- Self-tuning adaptive control.
- Adaptive perturbation control with feedforward compensation.
- Resolved motion adaptive control.



## ■ 5.2 (PUMA) Control of the PUMA robot arm.

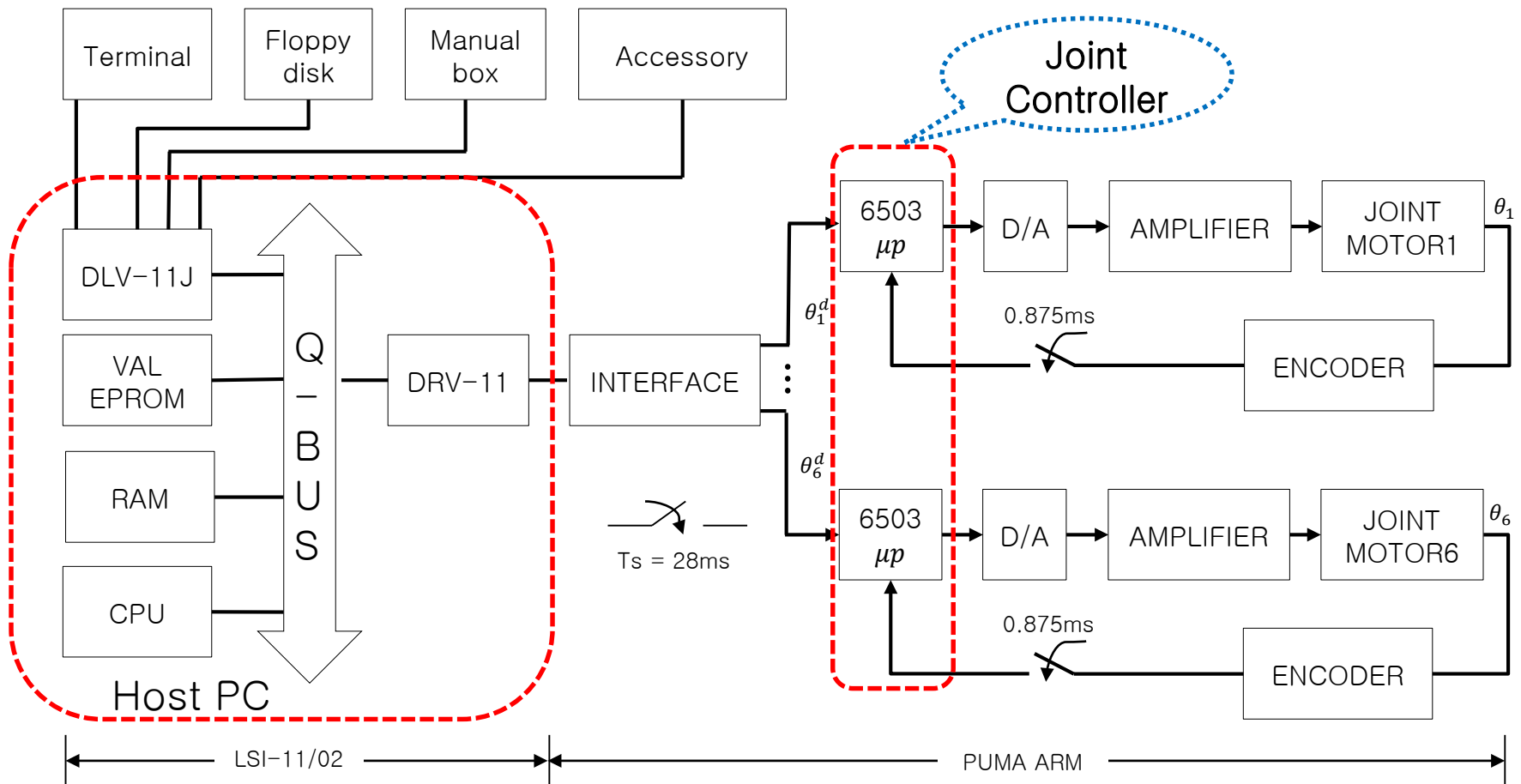


Figure 5.2 PUMA robot arm servo control architecture



- The function of host microprocessor of PUMA robot arm
  1. Coordinate system transformations( e.g., from world to joint coordinates).
  2. Joint-interpolated trajectory planning ; this involves sending incremental location updates corresponding to each set point to each joint every 28 ms
  3. Acknowledging from the micro processors that each axis of motion has completed its required incremental motion.
  4. Looking ahead two instructions to perform continuous path interpolation if the robot is in a continuous path mode.





## ● The main function of joint controllers (6503 microprocessor)

1. Every 28ms, receive and acknowledge trajectory set points from the host computer (LSI-11/02) and perform interpolation between the current joint value and the desired joint value.
  2. Every 0.875 ms, read the register value which stores the incremental values from the encoder mounted at each axis of rotation.
  3. Update the error actuating signals derived from the joint-interpolated set points and the values from the axis encoders
  4. Convert the error actuating signal to current using the DACs, and send the current to the analog servo board which moves the joint.
- PUMA robot control schemes is based on PID control method.  
=> Disadvantage is that the feedback gains are constant and pre-specified. It doesn't have the capability updating the feedback gains under varying payloads.



## ■ 5.3 Computed Torque Technique

- ▶ The control problem is to find appropriate torque/forces to servo all the joints of the manipulator in real time in order to track a desired time-based trajectory as closely as possible.

The drive motor torque required to servo the manipulator is based on a dynamic model of the manipulator which is derived by Lagrange-Euler equation

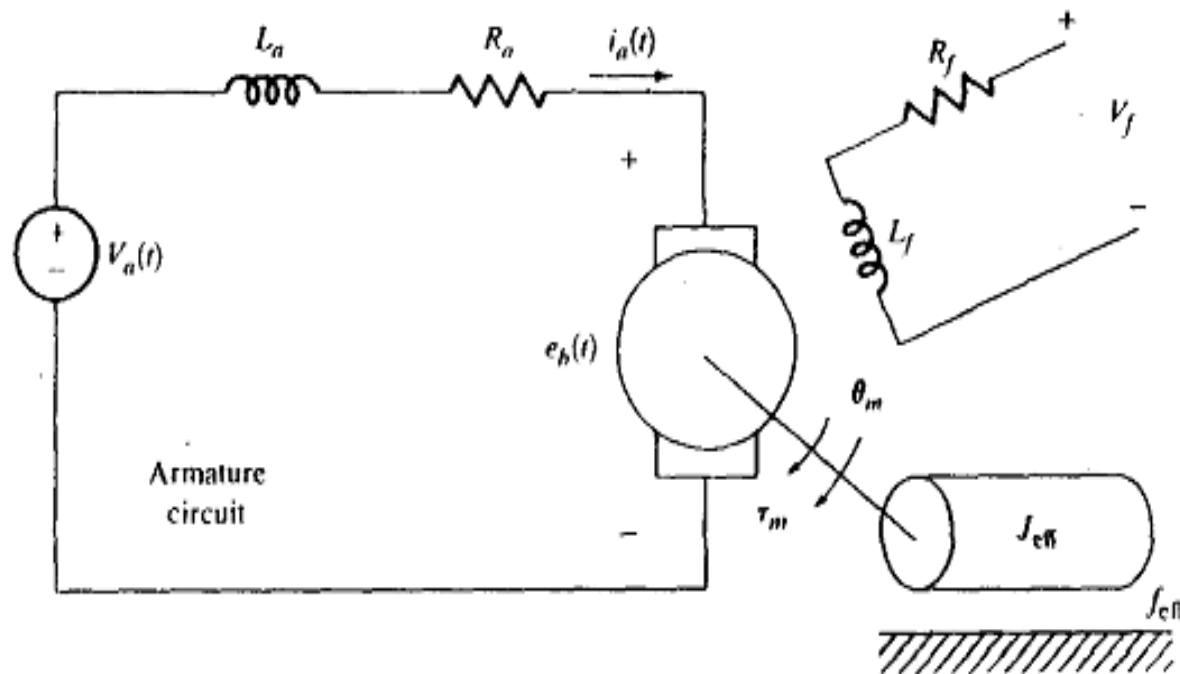
=> The computed torque is converted to the applied motor voltage and inputted to joint motor to control it.

- Modeling errors and parameter variations in the model are compensated by PD control and the torques computed based on the manipulator model.

=>

But it has limitation in robustness against dynamic model error and parameter uncertainty.

## ■ 5.3.1 Transfer function of a single joint



$V_a$ : armature voltage

$V_f$ : field voltage

$L_a$ : armature inductance

$L_f$ : field inductance

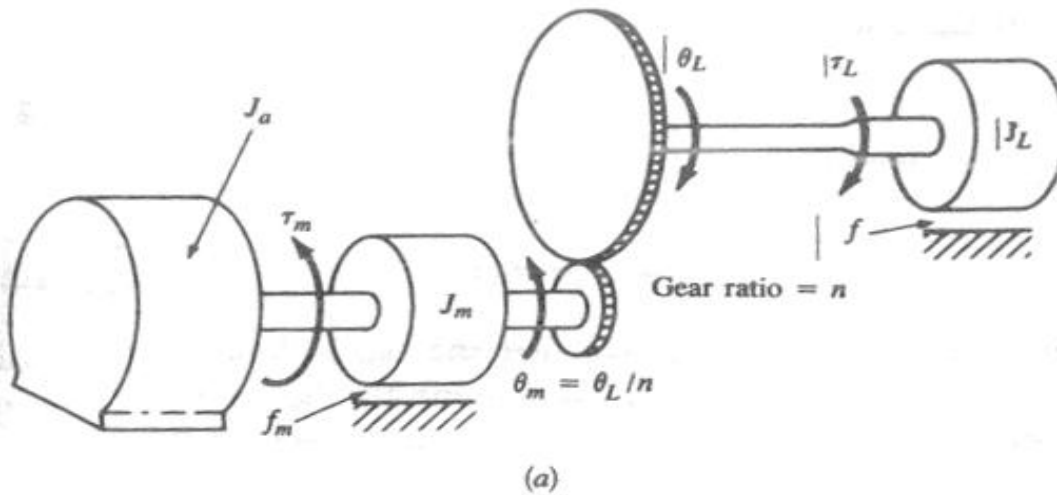
$R_a$ : armature resistance

$R_f$ : Field resistance

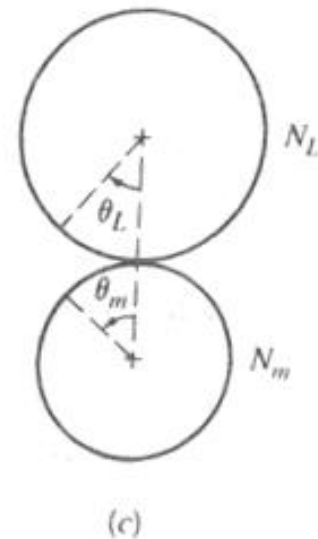
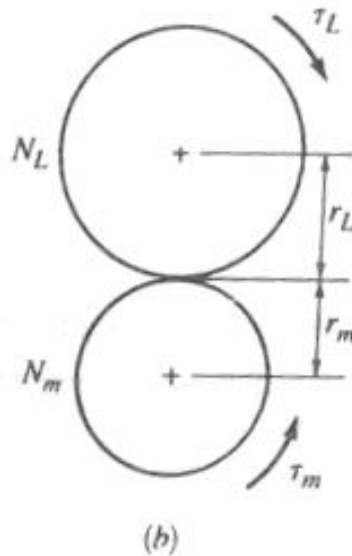
$i_a$ : armature current

$N_m$ : Number of teeth of the  
input gear(motor gear)

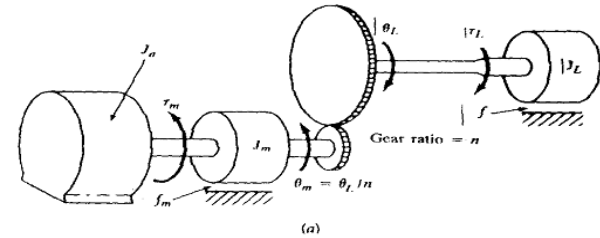
$N_L$ : Number of teeth of the  
input gear(load gear)



The total linear distance traveled on each gear is same



## ● Modeling of servo system

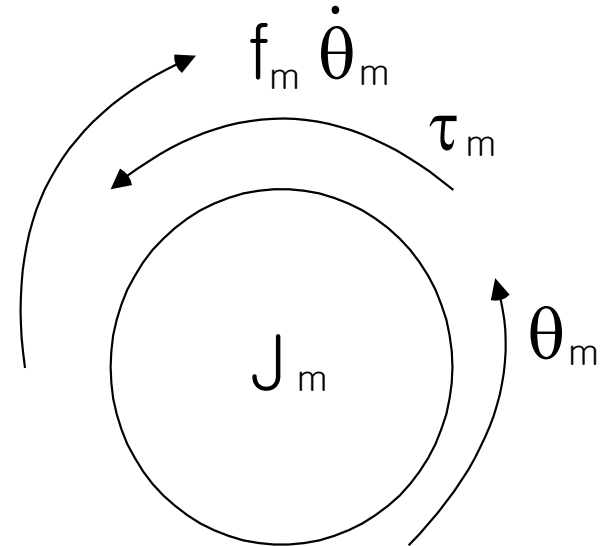
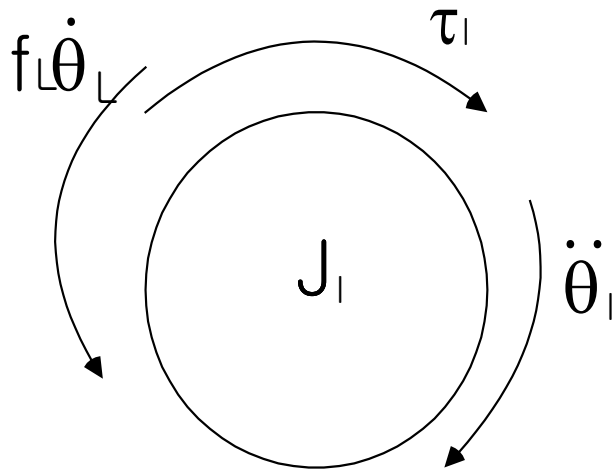


$$r_m \theta_m = r_L \theta_L \Rightarrow \frac{N_m}{N_L} = \frac{\theta_L}{\theta_m} = n < 1 : \text{gear ratio}$$

$$\therefore \theta_L = n\theta_m, \dot{\theta} = n\dot{\theta}_m, \ddot{\theta}_L = n\ddot{\theta}_L = n\ddot{\theta}_m$$

$$\left[ \begin{array}{c} \text{Torque from} \\ \text{motor shaft} \end{array} \right] = \left[ \begin{array}{c} \text{torque} \\ \text{on} \\ \text{motor} \end{array} \right] + \left[ \begin{array}{c} \text{torque on load} \\ \text{referred to} \\ \text{the motor shaft} \end{array} \right]$$

$$\tau(t) = \tau_m(t) + \tau_L^*(t)$$



- The load torque referred to the load shaft

$$\tau_L(t) = J_L \ddot{\theta}_L(t) + f_L \dot{\theta}_L(t) \quad (5.3-9)$$

- The motor torque referred to the motor shaft

$$\tau_m(t) = J_m \ddot{\theta}_m(t) + f_m \dot{\theta}_m(t) \quad (5.3-10)$$



$\tau_L(\theta)$  is converted to the torque to the motor referred to the motor shaft

From conservation of work, the work done by the load referred to the load shaft is equal to the work done by the load referred to the motor shaft,

$$\tau_L \theta_L = \tau_L^* \theta_m \quad \Rightarrow \quad \therefore \tau_L^*(t) = \frac{\tau_L \theta_L}{\theta_m} = n \tau_L(t)$$

From (5.3-9)

$$\therefore \tau_L(t) = \frac{\tau_L^*(t)}{n} = n J_L \ddot{\theta}_m + n f_L \dot{\theta}_m$$

$$\therefore \tau_L^*(t) = n^2 [J_L \ddot{\theta}_m + f_L \dot{\theta}_m(t)] \quad (5.3.12)$$

$$\begin{aligned} \therefore \tau(t) &= \tau_m(t) + \tau_L^*(t) = \tau_m(t) + n^2 (J_L \ddot{\theta}_m + f_L \dot{\theta}_m) \\ &= (J_m + n^2 J_L) \ddot{\theta}_m(t) + (f_m + n^2 f_L) \dot{\theta}_m(t) \end{aligned}$$

$$\therefore \tau(t) = J_{eff} \ddot{\theta}_m(t) + f_{eff} \dot{\theta}_m(t) \quad (5.3.13)$$

$$\text{where, } J_{eff} = J_m(t) + n^2 J_L, \quad f_{eff} = f_m + n^2 f_L$$



Input torque to the motor

$$\tau(t) = K_a i_a(t)$$

$K_a$  is known as the motor-torque proportional constant

By Kirchhoff's law

$$V_a(t) = R_a i_a(t) + L_a \frac{d i_a(t)}{dt} + e_b$$

$$e_b(t) = K_b \dot{\theta}_m(t) \quad K_b [V \bullet s / rad]$$

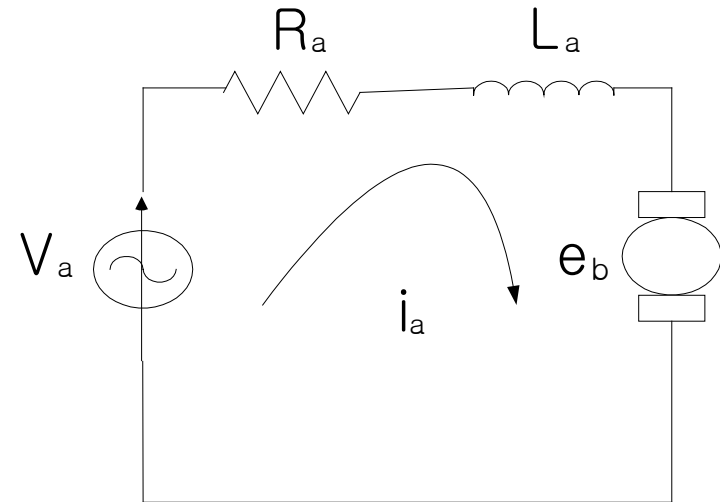
where  $e_b$  is the back electromotive force(emf).

Taking the Laplace transform,  $I_a(s)$  is derived as

$$I_a(s) = \frac{V_a(s) - sK_b \theta_m(s)}{R_a + sL_a}$$

From torque equation

$$T(s) = s^2 J_{eff} \theta_m(s) + s f_{eff} \theta_m(s) = (s^2 J_{eff} + s f_{eff}) \theta_m(s) = K_a I_a(s) \quad (5.3.18)$$







$$T(s) = K_a I_a(s) = K_a \left[ \frac{V_a(s) - sK_b \theta_m(s)}{R_a + sL_a} \right] = (s^2 J_{eff} + s f_{eff}) \theta_m(s)$$

$$\therefore \frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{s[s^2 J_{eff} L_a + (L_a f_{eff} + R_a J_{eff})s + R_a f_{eff} + K_a K_b]} \quad (5.3-20)$$

if  $L_a \approx 0$

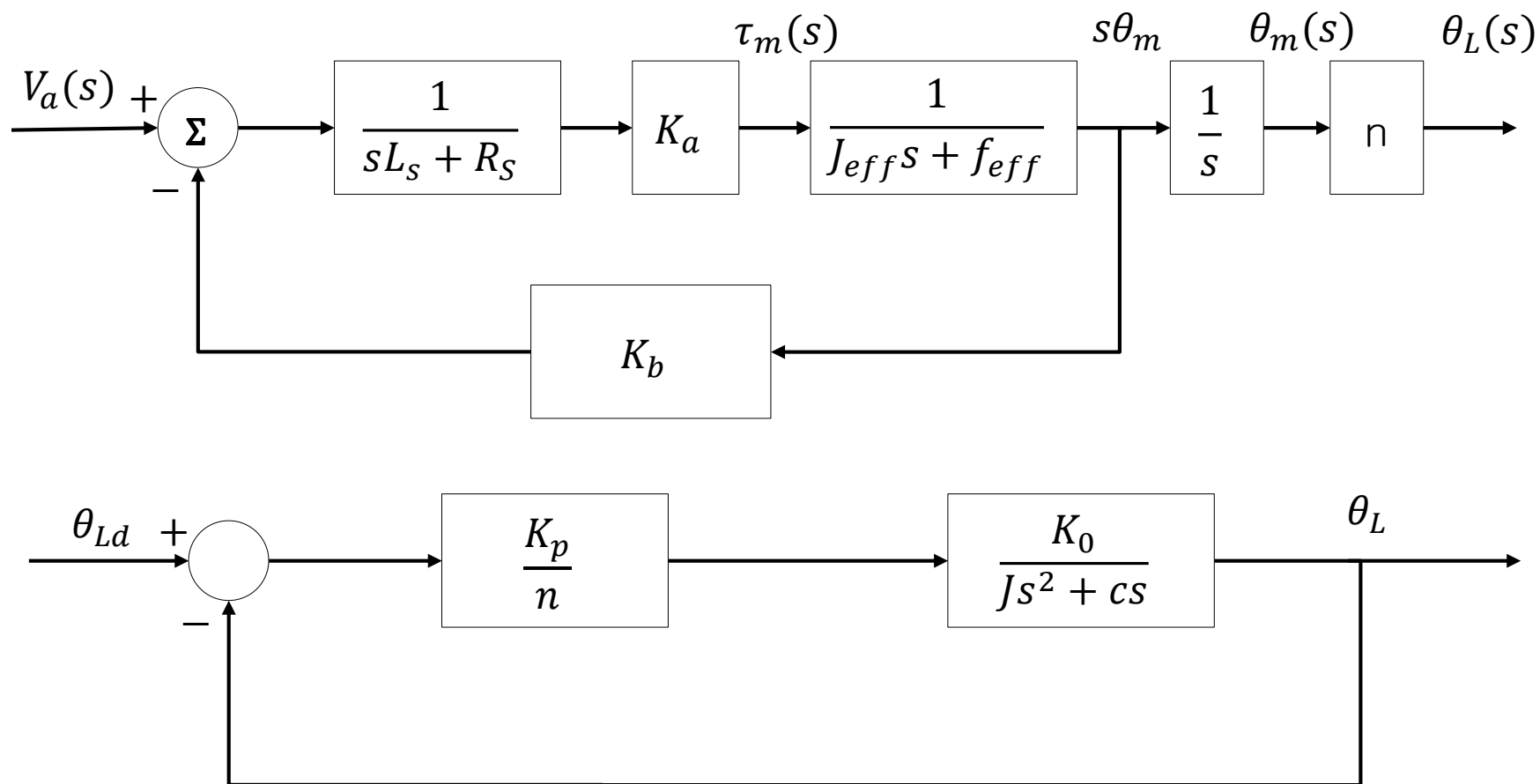
$$\therefore \frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{s[sJ_{eff} R_a + R_a f_{eff} + K_a K_b]} = \frac{K}{s(T_m s + 1)}$$

where,  $K = \frac{K_a}{R_a f_{eff} + K_a K_b}$  : motor gain constant

$$T_m = \frac{R_a J_{eff}}{R_a f_{eff} + K_a K_b} : \text{motor time constant}$$

Transfer function from input  $V_a(s)$  to output  $\theta_L$ , where  $\theta_m(s) = n\theta_2(s)$

$$\frac{\theta_L(s)}{V_a(s)} = \frac{n\theta_m(s)}{V_a(s)} = \frac{nK}{s(T_m s + 1)} = \frac{nK_a}{s[sR_a J_{eff} + R_a f_{eff} + K_a K_b]} \quad (5.3-22)$$





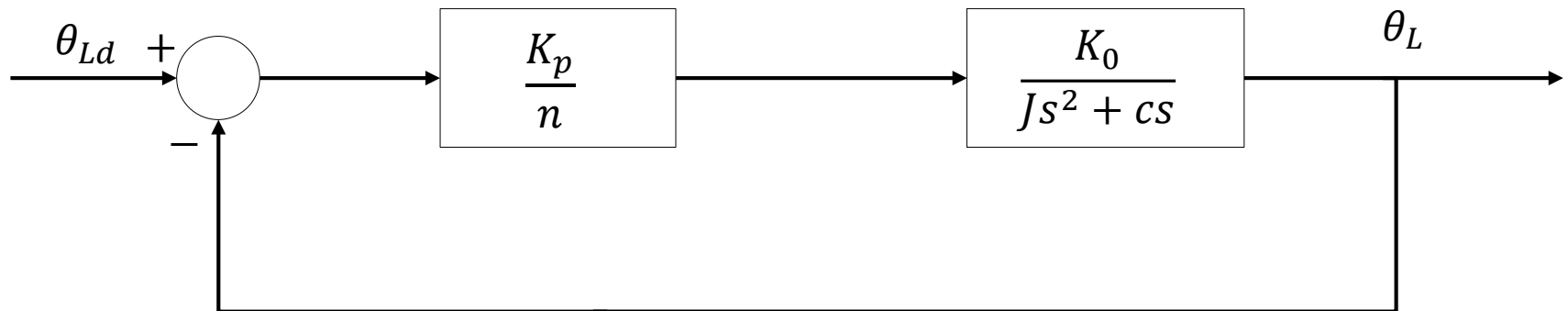
### ■ 5.3.2 Positional Controller for a Single Joint

#### ● Position feedback control

$$V_a(t) = \frac{K_p e(t)}{n},$$

$$e(t) = \theta_L^d(t) - \theta_L(t)$$

$$= \frac{K_p (\theta_L^d(t) - \theta_L(t))}{n}$$





$$L_a \approx 0$$

$$\frac{\theta_L(s)}{E(s)} = G(s) = \frac{K_a K_p}{s[sR_a J_{eff} + R_a f_{eff} + K_a K_b]}$$

$$\frac{\theta_L(s)}{\theta_L^d(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_a K_p / R_a J_{eff}}{s^2 + [(R_a J_{eff} + K_a K_b) / R_a J_{eff}]s + K_a K_p / R_a J_{eff}}$$

$$E(s) = \theta_L^d(s) - \theta_L(s) = \theta_L^d(s) \left( 1 - \frac{\theta_L(s)}{\theta_L^d(s)} \right) = \frac{1}{1 + G(s)} \theta_L^d(s)$$

- In order to increase the system response and reduce the steady-state error, the positional feedback gain  $K_p$  should be increased.

=>Overshoot and chattering or vibration problem effects to a control performance badly

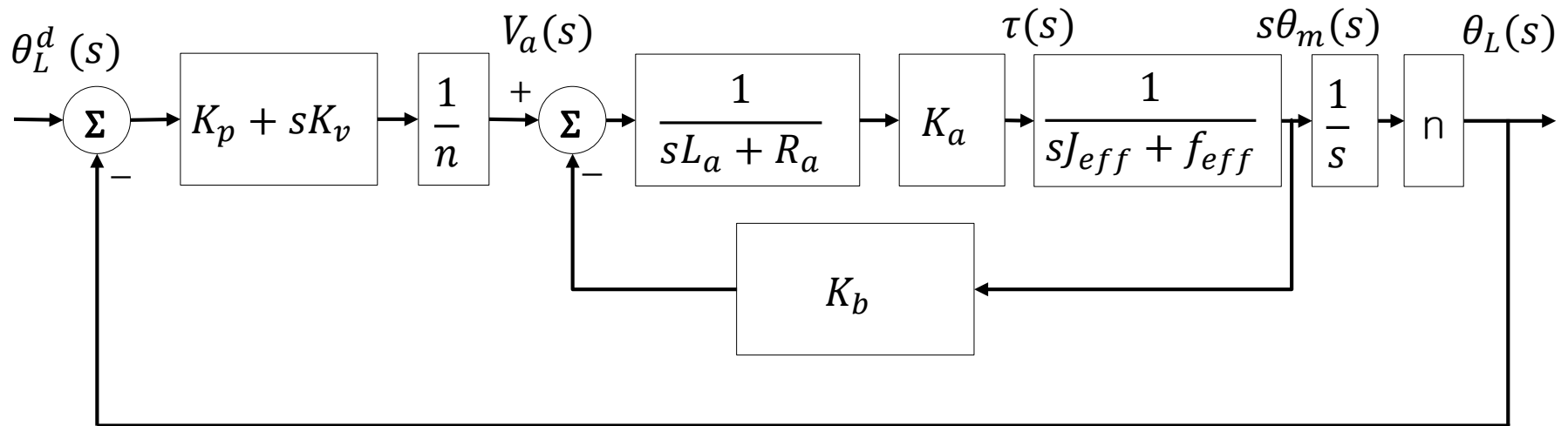


We can modulate some damping into the system by adding a derivative of the position error.  $\longrightarrow$  decreasing overshoot

The angular velocity of the joint can be measured by a tachometer or approximated from position data between two consecutive sampling periods.

The applied control voltage is given by

$$V_a(t) = \frac{K_p[\theta_L^d(t) - \theta_L(t)] + K_v[\dot{\theta}_L^d(t) - \dot{\theta}_L(t)]}{n} = \frac{K_p e(t) + K_v \dot{e}(t)}{n}$$





$$V_a(s) = \frac{(K_p + K_v s) E(s)}{n}$$

$$\frac{\theta_L(s)}{E(s)} = G_{PD}(s) = \frac{K_a K_v s + K_a K_p}{s(sR_a J_{eff} + R_a f_{eff} + K_a K_b)}$$

$$\frac{\theta_L(s)}{\theta_L^d(s)} = \frac{G_{PD}(s)}{1 + G_{PD}(s)} = \frac{K_a K_v s + K_a K_p}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \quad (5.3-29)$$

$$\text{Finite zero: } s = -\frac{K_v}{K_p}$$

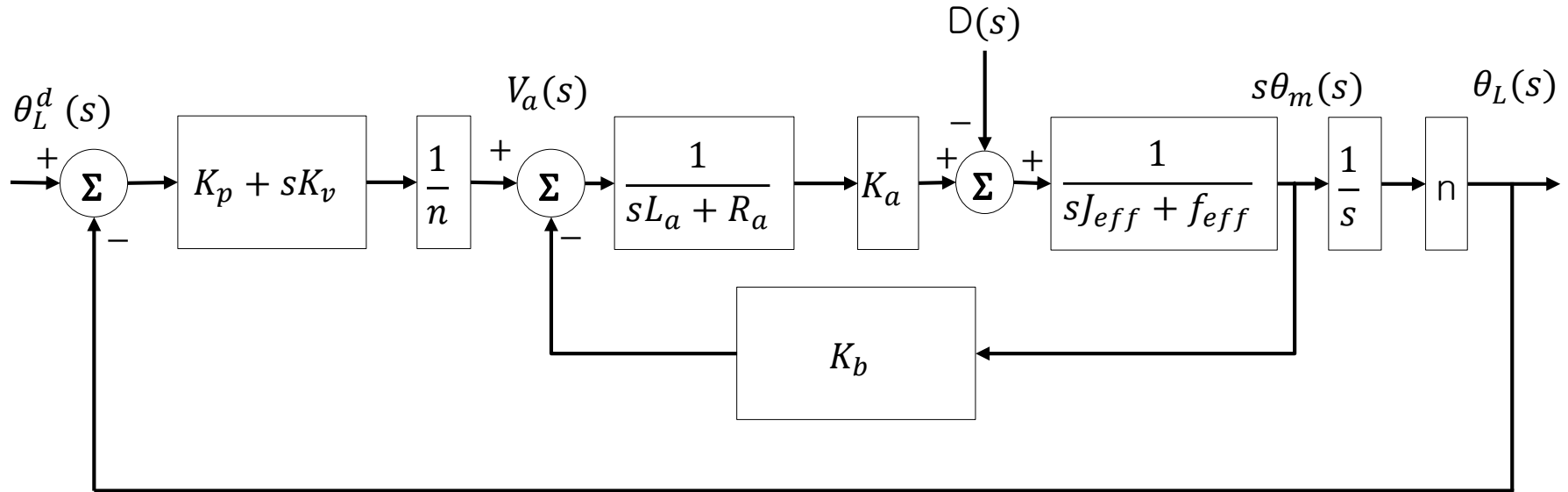
Depending on the location of this zero, the system could have large overshoot and a long settling time.



Because of disturbance, the torque generated at the motor shaft has to compensate for the torques dissipated by the motor, the load, and also the disturbances.

In case of adding disturbance  $D(s)$ , from Eq. (5.3.18) the torque equation is derived as

$$T(s) = (s^2 J_{eff} + s f_{eff}) \theta_m(s) + D(s)$$





$$\left. \frac{\theta_L(s)}{D(s)} \right|_{\theta_L^d(s)=0} = \frac{-nR_a}{s^2 R_a J_{eff} + s(R_a f_{eff} + K_a K_b + K_a K_v) + K_a K_p}$$

Using the superposition principle

$$\theta_L(s) = \frac{K_a (K_p + sK_v) \theta_L^d(s) - nR_a D(s)}{s^2 R_a J_{eff} + s(R_a f_{eff} + K_a K_b + K_a K_v) + K_a K_p}$$





## ● Steady State error

$$\theta_L(s) = \frac{K_a(K_p + sK_v)]\theta_L^d(s) - nR_a D(s)}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \quad (5.3-32)$$

$$\begin{aligned} E(s) &= \theta_L^d(s) - \theta_L(s) \\ &= \frac{[s^2 J_{eff} R_a + s(R_a J_{eff} + K_a K_b)]\theta_L^d(s) + nR_a D(s)}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \end{aligned} \quad (5.3-47)$$

*i) In case of step input  $\theta_L^d(t) = A$*

$$\begin{aligned} e_{ss}(stop) &= \lim_{s \rightarrow 0} sE(s) \\ &= \lim_{s \rightarrow 0} s \frac{[s^2 J_{eff} R_a + s(R_a J_{eff} + K_a K_b)]A/s + nR_a D(s)}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \\ &= \lim_{s \rightarrow 0} s \left[ \frac{nR_a D(s)}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \right] \end{aligned} \quad (5.3-48)$$

which is a function of the disturbance  $D(s)$



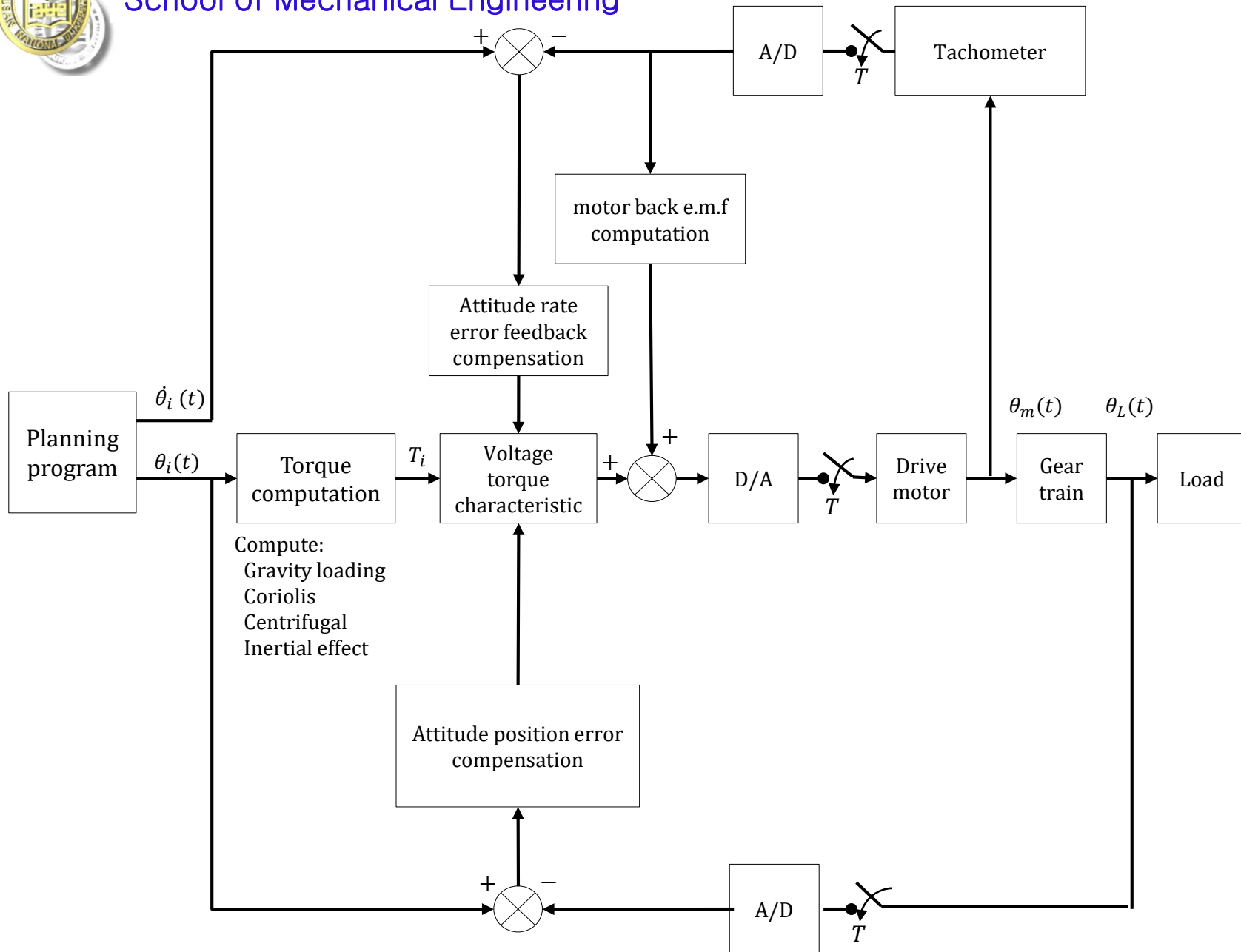
If we can identify each of these torques separately the disturbance is written by

$$\tau_D(t) = \tau_G(t) + \tau_C(t) + \tau_e \quad (5.3-49)$$

where  $\tau_G(t)$  and  $\tau_C(t)$  are torques due to gravity centrifugal effects of the link, and  $\tau_e$  is disturbances which can be assumed to be a very small constant value.

The Lapalce transform of Eq.(5.3-49),

$$D(s) = T_G(s) + T_C(s) + \frac{T_e}{s} \quad (5.3-50)$$





Let us denote the computed torques as  $\tau_{comp}(t)$  whose Laplace transform is  $T_{comp}(s)$ .

With this computed torque and using Eq. (5.3–50), the error equation of (5.3–47) is modified to

$$E(s) = \frac{[s^2 J_{eff} R_a + s(R_a J_{eff} + K_a K_b)]\theta_L^d(s) + nR_a [T_g(s) + T_C(s) + T_e / s - T_{comp}(s)]}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \quad (5.3-51)$$

For a step input  $\theta_L^d(s) = \frac{a}{s}$ , the steady state position error is given by

$$e_{ssp} = \lim_{s \rightarrow 0} s \frac{nR_a [T_G(s) + T_C(s) + T_e / s - T_{comp}(s)]}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p} \quad (5.3-52)$$



If the disturbance due to the centrifugal effect is zero as time approaches infinity,  $\dot{\theta}_L(\infty) \approx 0$  because the centrifugal force is a function of  $\dot{\theta}_L^2$

If the computed torque  $\tau_{comp}(t)$  is equivalent to the gravity loading of the link, then the steady-state position error reduces to

$$e_{ssp} = \frac{nR_a T_e}{K_a K_p} = \frac{nR_a T_e 4K_a}{K_a \omega_0^2 J_0 R_a} = \frac{4nT_e}{\omega_0^2 J_0}$$

$$\because 0 < K_p \leq \frac{\omega_r^2 J_{eff} R_a}{4K_a} \approx \frac{\omega_0^2 J_0 R_a}{4K_a}$$

where,  $\tau_{comp}(t) = \tau_G(t)$



ii) Ramp input  $\theta_L^d = \frac{A}{s^2}$

$$e_{ss}(\text{ramp}) = \lim_{s \rightarrow 0} sE(s)$$

$$= \frac{(R_a f_{eff} + K_a K_b)A}{K_a K_p} +$$

$$\lim_{s \rightarrow 0} s \frac{nR_a [T_G + T_C + T_e / s - T_{comp}]}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p}$$

$$= \frac{(R_a f_{eff} + K_a K_b)A}{K_a K_p} + e_{ssp}$$



The computation of  $\tau_{comp}$  depends on the dynamic model of the manipulator

$$\begin{aligned}\tau_i(t) = & \sum_{k=i}^6 \sum_{j=1}^k T_r \left[ \frac{\partial^0 T_k}{\partial q_j} J_k \left[ \frac{\partial^0 T_k}{\partial q_i} \right]^T \right] \ddot{q}_j(t) \\ & + \sum_{r=i}^6 \sum_{j=0}^r \sum_{k=1}^r T_r \left[ \frac{\partial^{20} T_r}{\partial q_j \partial q_k} J_r \left[ \frac{\partial^0 T_r}{\partial q_i} \right]^T \right] \dot{q}_j(t) \dot{q}_k(t) \\ & - \sum_{j=i}^6 m_j g \left[ \frac{\partial^0 T_j}{\partial q_j} \right] \bar{r}_j, \text{ for } i=1, 2, \dots, 6\end{aligned}$$



### ■ 5.3.3 Performance and Stability Criteria

$$\frac{K_a K_v s + K_a K_p}{s^2 R_a J_{eff} + s(R_a J_{eff} + K_a K_b + K_a K_v) + K_a K_p}$$

#### ● The characteristic equation of a 2nd order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

where  $\zeta$  and  $\omega_n$  are the damping ratio and the undamped natural frequency of the system, respectively. Relating the closed loop poles of Eq. (5.3-29) to Eq. (5.3-33)

$$\omega_n^2 = \frac{K_a K_p}{J_{eff} R_a} \quad (5.3-34)$$

$$2\zeta\omega_n = \frac{R_a J_{eff} + K_a K_b + K_a K_v}{J_{eff} R_a} \quad (5.4-35)$$

In order to have good performance, we may select a critically damped or an overdamped system. That is,  $\zeta \geq 1$

From Eq. (5.3-34), the position feedback gain is derived as

$$K_p = \frac{J_{eff} R_a \omega_n^2}{K_a} > 0 \quad (5.3-36)$$





Substituting  $\omega_n$  from Eq. (5.3-34) into Eq. (5.3-35)

$$\zeta = \frac{R_a f_{eff} + K_a K_b + K_a K_v}{2\sqrt{K_a K_p J_{eff} R_a}} \geq 1 \quad (5.3-37)$$

From (5.3-37), the velocity feedback gain  $K_v$  can be found to be

$$K_v \geq \frac{2\sqrt{K_a K_p J_{eff} R_a} - R_a f_{eff} - K_a K_b}{K_a} \quad (5.3-38)$$

In order not to excite the structural oscillation and resonance of the joint, the undamped natural frequency  $\omega_n$  may be set to no more than one-half of the structural resonance frequency  $\omega_r$  of the joint [R.P.Paul(1981)]

$$\omega_n \leq 0.5\omega_r \quad (5.3-39)$$



If the effective stiffness of the joint is  $K_{stiff}$ , then the restoring torque  $K_{stiff}\theta_m(t)$  opposes the inertial torque of the motor,

$$J_{eff}\ddot{\theta}_m(t) + K_{stiff}\theta_m(t) = 0 \quad (5.3-40)$$

$$LT \Rightarrow J_{eff}s^2 + K_{stiff} = 0 \quad (5.3-41)$$

Solving the above characteristic equation gives the structural resonant frequency of the system

$$\omega_r = \left[ \frac{K_{stiff}}{J_{eff}} \right]^{\frac{1}{2}} \quad (5.3-42)$$



If a load is added to the manipulator's end-effector, the effective moment of inertia will increase which, in effect, reduces the structural resonant frequency  $\omega_r$ .

If a structural resonant frequency  $\omega_0$  is measured at a known moment of inertia  $J_0$ , then the structural resonant frequency at the other moment of inertia  $J_{eff}$  is given by

$$\omega_0 = \left[ \frac{K_{stiff}}{J_0} \right]^{\frac{1}{2}} \Rightarrow K_{stiff} = J_0 \omega_0^2$$
$$\therefore \omega_r = \left[ \frac{K_{stiff}}{J_{eff}} \right]^{\frac{1}{2}} = \left[ \frac{J_0 \omega_0^2}{J_{eff}} \right]^{\frac{1}{2}} = \omega_0 \left[ \frac{J_0}{J_{eff}} \right]^{\frac{1}{2}} \quad (5.3-39)$$

$$\omega_n^2 = \frac{K_a K_p}{J_{eff} R_a} \leq \frac{1}{4} \omega_r^2 = \frac{1}{4} \omega_0^2 \left[ \frac{J_0}{J_{eff}} \right]$$



Using the condition of Eq.(5.3-39) and Eq.(5.3-43), from Eq.(5.3-36) is bounded by

$$\therefore 0 < K_p \leq \frac{\omega_r^2 J_{eff} R_a}{4K_a} \approx \frac{\omega_0^2 J_0 R_a}{4K_a} \quad (5.3-45)$$

After finding  $k_p$ , the velocity feedback gain can be found from Eq.(5.3-38)

$$\begin{aligned} K_v &\geq \frac{2\sqrt{K_a \frac{\omega_0^2 J_0 R_a}{4K_a} J_{eff} R_a - R_a J_{eff} - K_a K_b}}{K_a} \\ &= \frac{R_a \omega_0 \sqrt{J_0 J_{eff}} - R_a J_{eff} - K_a K_b}{K_a} \end{aligned} \quad (5.3-46)$$



### 5.3.4 Controller for Multi-joint Robots

- ▶ For a manipulator with multiple joints, one of the basic schemes is the computed torque based on L-E equations which is a feedforward control and has feedforward and feedback components.
- Feed forward control components compensate for the interaction forces among all the various joints.
- Feedback control computes the necessary correction torques to compensate for any deviation from the desired trajectory.
- ▶ The structure of the control law to supply to the motor

$$\tau(t) = D_a(q) \left\{ \ddot{q}^d(t) + K_v[\dot{q}^d(t) - \dot{q}(t)] + K_p[q^d(t) - q(t)] \right\} \\ + h_a(q, \dot{q}) + C_a(q)$$

(5.3–65)



Substituting  $\tau(t)$  from Eq.(5.3-65) into  $\tau(t) = D(q(t))\ddot{q}(t) + h(q(t), \dot{q}(t)) + C(q(t))$

$$D(q)\ddot{q} + h(q, \dot{q}) + C(q) =$$

$$D_a(q)\{\ddot{q}^d + K_v[\dot{q}^d - \dot{q}] + K_p[q^d - q]\} + h_a(q, \dot{q}) + C_a(q) \quad (5.3-66)$$

If  $D_a(q) \approx D(q)$ ,  $C_a(q) \approx C(q)$ ,  $h_a(q, \dot{q}) \approx h(q, \dot{q})$

$$D(q)[\ddot{e} + K_v\dot{e} + K_p e] = 0 \quad (5.3-67)$$

where,  $e = q^d - q$ ,  $\dot{e} = \dot{q}^d - \dot{q}$

Since  $D(q)$  is always nonsingular,  $K_p$  and  $K_v$  can be chosen appropriately so the characteristic roots of Eq. (5.3-67) have negative real parts, then the position **error vector  $e(t)$  approaches zero asymptotically.**



# ***ROBOT DYNAMICS AND CONTROL SIMULATION***

## **Task Space Control**

- Joint space control → Tracking the trajectory of end-effector by control the trajectory of joint indirectly
- Task space control → Tracking directly the trajectory of end-effector by feedback control error of task space (end-effector)



# ***COLLISION-FREE PATH PLANNING***

It would be extremely convenient if we could simply tell the robot system what the desired goal point of the manipulator motion is, and let the system determine where and how many via points are required so that the goal is reached without the manipulator's hitting any obstacles. In order to do this, the system must have models of the manipulator, the work area, and all potential obstacles in the area.

Systems that plan collision-free paths are not available commercially.





# ***COLLISION-FREE PATH PLANNING***

Research in this area has led to two competing techniques, and several variations and combinations thereof.

One approach solves the problem by forming a connected-graph representation of the free space and then searching the graph for a collision-free path.

A second approach is based on creating artificial potential fields around obstacles, which cause the manipulator(s) to avoid the obstacles while they are drawn toward an artificial attractive pole at the goal point.



## *Video Clips on Trajectory Planning*



< Intelligent Robot Manipulator >



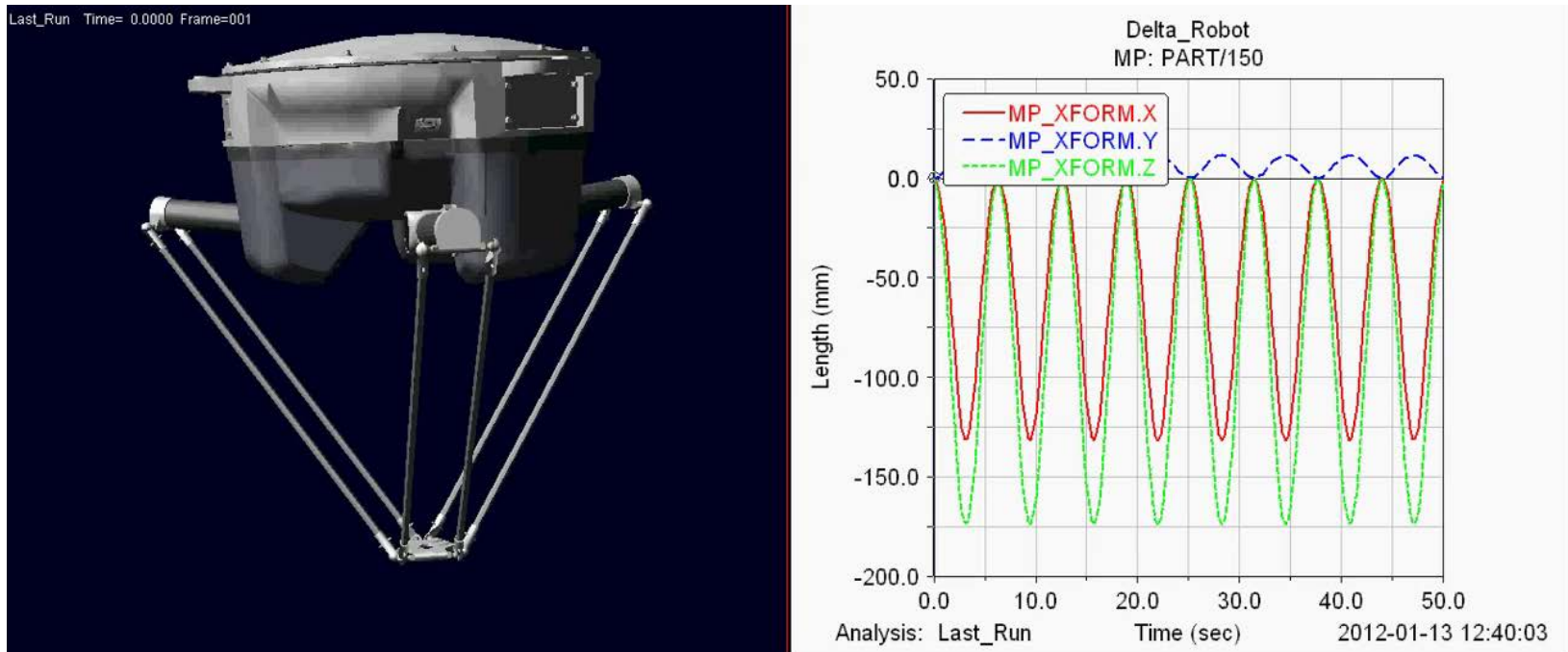
## *Video Clips on Relative Impedance Control*



<Relative Impedance Control>



## *Video Clips on Trajectory Planning*



<Robot dynamics and control with MSC ADMAS and Matlab>



## *Parallel kinematic industrial robot*



<Parallel kinematic industrial robot, Delta Robot>