



*Bài giảng môn học*

*Lý thuyết*  
**ĐIỀU KHIỂN TỰ ĐỘNG**



Liên hệ : [tdkquoc@dng.vnn.vn](mailto:tdkquoc@dng.vnn.vn)

# MỤC LỤC

Phần mở đầu	
1	Khái niệm.....5
2	Các nguyên tắc điều khiển tự động.....6
2.1	Nguyên tắc giữ ổn định.....6
2.2	Nguyên tắc điều khiển theo chương trình.....6
3	Phân loại hệ thống ĐKTD.....6
3.1	Phân loại theo đặc điểm của tín hiệu ra.....6
3.2	Phân loại theo số vòng kín.....6
3.3	Phân loại theo khả năng quan sát tín hiệu.....7
3.4	Phân loại theo mô tả toán học.....7
4	Biểu đồ điều khiển tự động trong một nhà máy.....8
5	Phép biến đổi Laplace.....8
Chương 1: MÔ TẢ TOÁN HỌC CÁC PHẦN TỬ VÀ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG	
1	Khái niệm chung.....10
2	Hàm truyền đạt.....10
2.1	Định nghĩa :.....10
2.2	Phương pháp tìm hàm truyền đạt.....10
2.3	Một số ví dụ về cách tìm hàm truyền đạt.....11
2.4	Hàm truyền đạt của một số thiết bị điển hình.....13
2.5	Đại số sơ đồ khối.....13
3	Phương trình trạng thái.....16
3.1	Phương trình trạng thái tổng quát.....16
3.2	Xây dựng phương trình trạng thái từ hàm truyền đạt.....18
3.3	Chuyển đổi từ phương trình trạng thái sang hàm truyền.....20
Chương 2: ĐẶC TÍNH ĐỘNG HỌC CỦA CÁC KHÂU VÀ CỦA HỆ THỐNG TRONG MIỀN TẦN SỐ	
1	Khái niệm chung.....24
2	Phản ứng của một khâu.....24
2.1	Tín hiệu tác động vào một khâu (các tín hiệu tiền định).....24
2.2	Phản ứng của một khâu.....24
3	Đặc tính tần số của một khâu.....25
3.1	Hàm truyền đạt tần số.....25
3.2	Đặc tính tần số.....26
4	Đặc tính động học của một số khâu cơ bản.....27
4.1	Khâu tỉ lệ.....27
4.2	Khâu quán tính bậc 1.....27
4.3	Khâu dao động bậc 2.....29
4.4	Khâu không ổn định bậc 1.....31
4.5	Khâu vi phân lý tưởng.....32
4.6	Khâu vi phân bậc 1.....32
4.7	Khâu tích phân lý tưởng.....33
4.8	Khâu chậm trễ.....33
Chương 3: TÍNH ỔN ĐỊNH CỦA HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG	
1	Khái niệm chung.....35
2	Tiêu chuẩn ổn định đại số.....36
2.1	Điều kiện cần để hệ thống ổn định.....36
2.2	Tiêu chuẩn Routh.....36
2.3	Tiêu chuẩn ổn định Hurwitz.....37
3	Tiêu chuẩn ổn định tần số.....37
3.1	Tiêu chuẩn Nyquist theo đặc tính tần số biên pha.....37

3.2	Tiêu chuẩn Nyquist theo đặc tính tần số logarit .....	37
3.3	Tiêu chuẩn ổn định Mikhailov .....	38
4	Phương pháp quỹ đạo nghiệm số .....	38
4.1	Phương pháp xây dựng QĐNS .....	38
<b>Chương 4: CHẤT LƯỢNG CỦA QUÁ TRÌNH ĐIỀU KHIỂN</b>		
1	Khái niệm chung .....	41
1.1	Chế độ xác lập .....	41
1.2	Quá trình quá độ .....	41
2	Đánh giá chất lượng ở chế độ xác lập .....	41
2.1	Khi $u(t) = U_0 \cdot 1(t)$ .....	42
2.2	Khi $u(t) = U_0 \cdot t$ .....	42
3	Đánh giá chất lượng ở quá trình quá độ .....	42
3.1	Phân tích thành các biểu thức đơn giản .....	42
3.2	Phương pháp số Tustin .....	42
3.3	Giải phương trình trạng thái .....	44
3.4	Sử dụng các hàm của MATAB .....	44
4	Đánh giá thông qua độ dự trữ ổn định .....	45
4.1	Độ dự trữ biên độ .....	45
4.2	Độ dự trữ về pha .....	45
4.3	Mối liên hệ giữa các độ dự trữ và chất lượng điều khiển .....	45
5	Tính điều khiển được và quan sát được của hệ thống .....	46
5.1	Điều khiển được .....	46
5.2	Tính quan sát được .....	46
<b>Chương 5: NÂNG CAO CHẤT LƯỢNG VÀ TỔNG HỢP HỆ THỐNG</b>		
1	Khái niệm chung .....	48
2	Các bộ điều khiển – Hiệu chỉnh hệ thống .....	48
2.1	Khái niệm .....	48
2.2	Bộ điều khiển tỉ lệ P .....	48
2.3	Bộ bù sớm pha Lead .....	48
2.4	Bộ bù trễ pha Leg .....	49
2.5	Bộ bù trễ-sớm pha Leg -Lead .....	50
2.6	Bộ điều khiển PI (Proportional Integral Controller) .....	51
2.7	Bộ điều khiển PD (Proportional Derivative Controller) .....	51
2.8	Bộ điều khiển PID (Proportional Integral Derivative Controller) .....	52
3	Tổng hợp hệ thống theo các tiêu chuẩn tối ưu .....	53
3.1	Phương pháp tối ưu modun .....	53
3.2	Phương pháp tối ưu đối xứng .....	54
<b>Chương 6: HỆ THỐNG ĐIỀU KHIỂN GIÁN ĐOẠN</b>		
1	Khái niệm chung .....	56
2	Phép biến đổi Z .....	56
2.1	Định nghĩa .....	56
2.2	Một số tính chất của biến đổi Z .....	57
2.3	Biến đổi Z ngược .....	57
3	Lấy mẫu và giữ mẫu .....	58
3.1	Khái niệm .....	58
3.2	Lấy mẫu .....	58
3.3	Giữ mẫu .....	59
4	Hàm truyền đạt hệ gián đoạn .....	60
4.1	Xác định hàm truyền đạt $W(z)$ từ hàm truyền đạt hệ liên tục .....	60
4.2	Xác định hàm truyền đạt từ phương trình sai phân .....	65
5	Tính ổn định của hệ gián đoạn .....	65
5.1	Mối liên hệ giữa mặt phẳng p và mặt phẳng z .....	65
5.2	Phép biến đổi tương đương .....	65

1	Control System Toolbox .....	66
1.1	Định nghĩa một hệ thống tuyến tính .....	66
1.2	Biên đổi sơ đồ tương đương .....	68
1.3	Phân tích hệ thống.....	69
1.4	Ví dụ tổng hợp .....	71
2	SIMULINK .....	73
2.1	Khởi động Simulink.....	73
2.2	Tạo một sơ đồ đơn giản.....	74
2.3	Một số khối thường dùng .....	75
2.4	Ví dụ.....	76
2.5	LTI Viewer .....	77

## Phân mở đầu

Điều khiển học là khoa học nghiên cứu những quá trình điều khiển và thông tin trong các máy móc sinh vật. Trong điều khiển học, đối tượng điều khiển là các thiết bị, các hệ thống kỹ thuật, các cơ thể sinh vật...

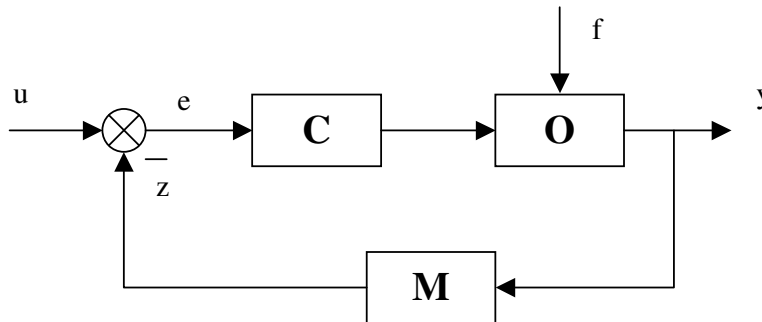
Điều khiển học nghiên cứu quá trình điều khiển các đối tượng kỹ thuật được gọi là điều khiển học kỹ thuật. Trong đó « Điều khiển tự động » là cơ sở lý thuyết của điều khiển học kỹ thuật.

Khi nghiên cứu các qui luật điều khiển của các hệ thống kỹ thuật khác nhau, người ta sử dụng các mô hình toán thay thế cho các đối tượng khảo sát. Cách làm này cho phép chúng ta mở rộng phạm vi nghiên cứu và tổng quát bài toán điều khiển trên nhiều đối tượng có mô tả toán học giống nhau.

Môn học **Điều khiển tự động** cung cấp cho sinh viên các kiến thức cơ bản về xây dựng mô hình toán học của một đối tượng và của cả hệ thống. Trên cơ sở đó, sinh viên có khả năng phân tích, đánh giá chất lượng của hệ thống điều khiển. Ngoài ra, bằng các phương pháp toán học, sinh viên có thể tổng hợp các bộ điều khiển thích hợp để hệ thống đạt được các chỉ tiêu chất lượng đề ra.

### 1 Khái niệm

Một hệ thống ĐKTD được xây dựng từ 3 bộ phận chủ yếu theo sơ đồ sau :



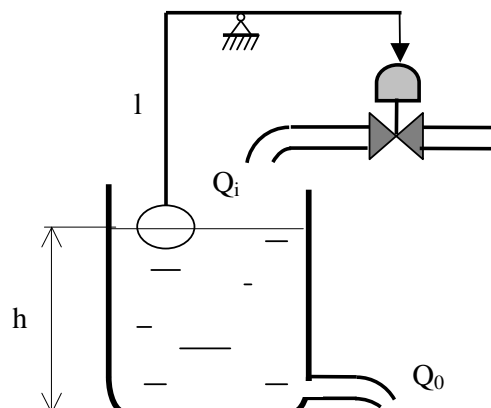
Trong đó :

- O : đối tượng điều khiển
- C : bộ điều khiển, hiệu chỉnh
- M : cơ cấu đo lường

Các loại tín hiệu có trong hệ thống gồm :

- u : tín hiệu chủ đạo (còn gọi là tín hiệu vào, tín hiệu điều khiển)
- y : tín hiệu ra
- f : các tác động từ bên ngoài
- z : tín hiệu phản hồi
- e : sai lệch điều khiển

*Ví dụ về một hệ thống điều khiển đơn giản*

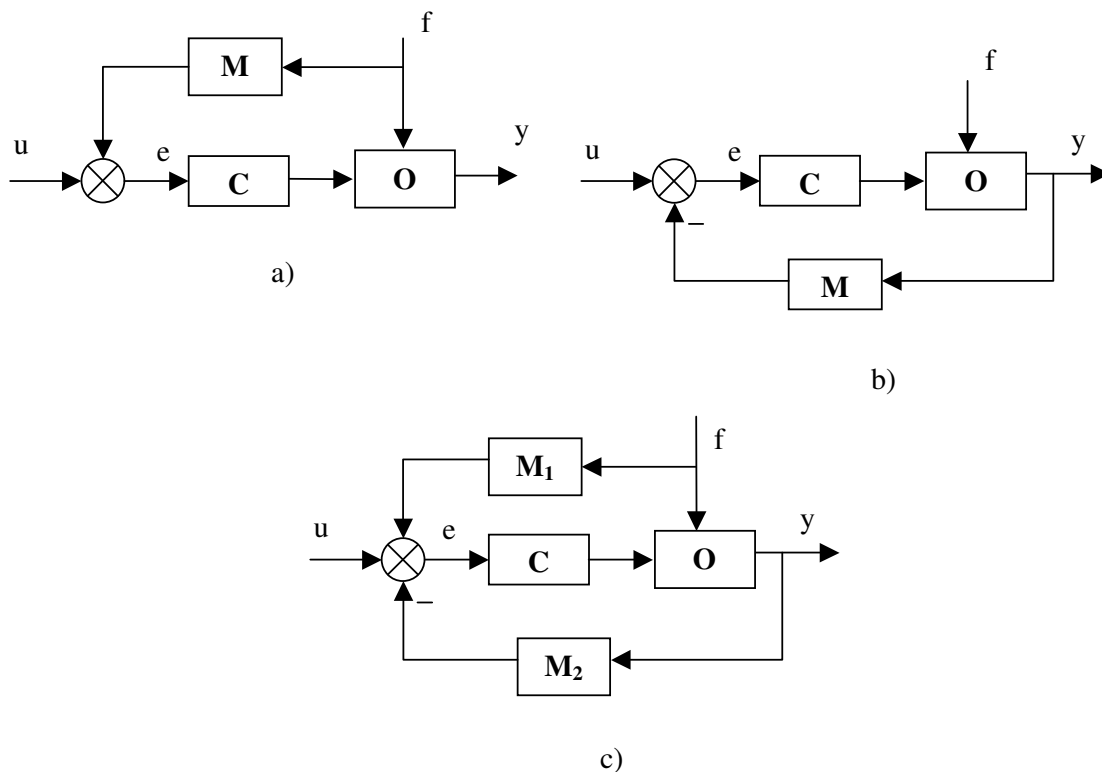


## 2 Các nguyên tắc điều khiển tự động

### 2.1 Nguyên tắc giữ ổn định

Nguyên tắc này giữ tín hiệu ra bằng một hằng số trong quá trình điều khiển,  $y = \text{const}$ . Có 3 phương pháp để thực hiện nguyên tắc giữ ổn định gồm :

- Phương pháp bù tác động bên ngoài (a)
- Phương pháp điều khiển theo sai lệch (b)
- Phương pháp hỗn hợp (c)



### 2.2 Nguyên tắc điều khiển theo chương trình

Nguyên tắc này giữ tín hiệu ra  $y = y(t)$  theo một chương trình đã được định sẵn. Để một tín hiệu ra nào đó thực hiện theo chương trình, cần phải sử dụng máy tính hay các thiết bị có lưu trữ chương trình. 2 thiết bị thông dụng chứa chương trình điều khiển là :

- PLC (Programmable Logic Controller)
- CLC (Computerized Numerical Control)

## 3 Phân loại hệ thống ĐKTD

### 3.1 Phân loại theo đặc điểm của tín hiệu ra

- Tín hiệu ra ổn định
- Tín hiệu ra theo chương trình

### 3.2 Phân loại theo số vòng kín

- Hệ hở: là hệ không có vòng kín nào.
- Hệ kín: có nhiều loại như hệ 1 vòng kín, hệ nhiều vòng kín,...

### 3.3 Phân loại theo khả năng quan sát tín hiệu

#### 3.3.1 Hệ thống liên tục

Quan sát được tất cả các trạng thái của hệ thống theo thời gian.

Mô tả toán học : phương trình đại số, phương trình vi phân, hàm truyền

#### 3.3.2 Hệ thống không liên tục

Quan sát được một phần các trạng thái của hệ thống. Nguyên nhân:

- Do không thể đặt được tất cả các cảm biến.
- Do không cần thiết phải đặt đủ các cảm biến.

Trong hệ thống không liên tục, người ta chia làm 2 loại:

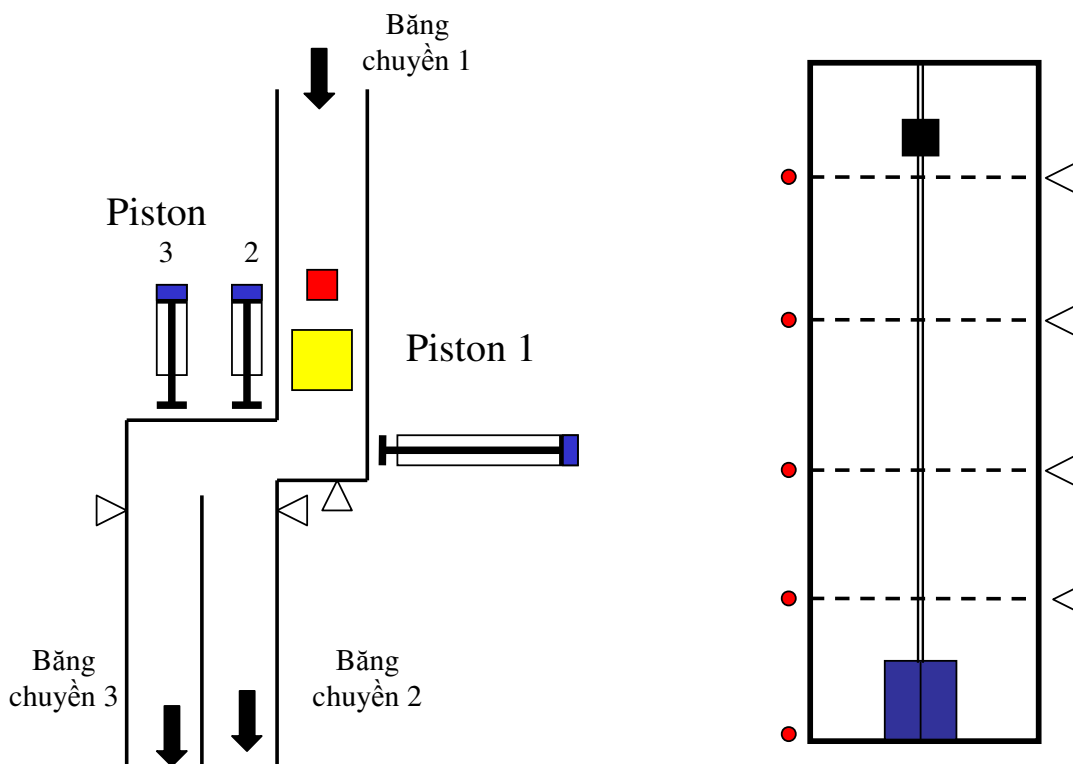
##### a) Hệ thống gián đoạn (*S. discret*)

Là hệ thống mà ta có thể quan sát các trạng thái của hệ thống theo chu kỳ (T). Về bản chất, hệ thống này là một dạng của hệ thống liên tục.

##### b) Hệ thống với các sự kiện gián đoạn (*S à événement discret*)

- Đặc trưng bởi các sự kiện không chu kỳ
- Quan tâm đến các sự kiện/ tác động

**Ví dụ về hệ thống liên tục, gián đoạn, hệ thống với các sự kiện gián đoạn**

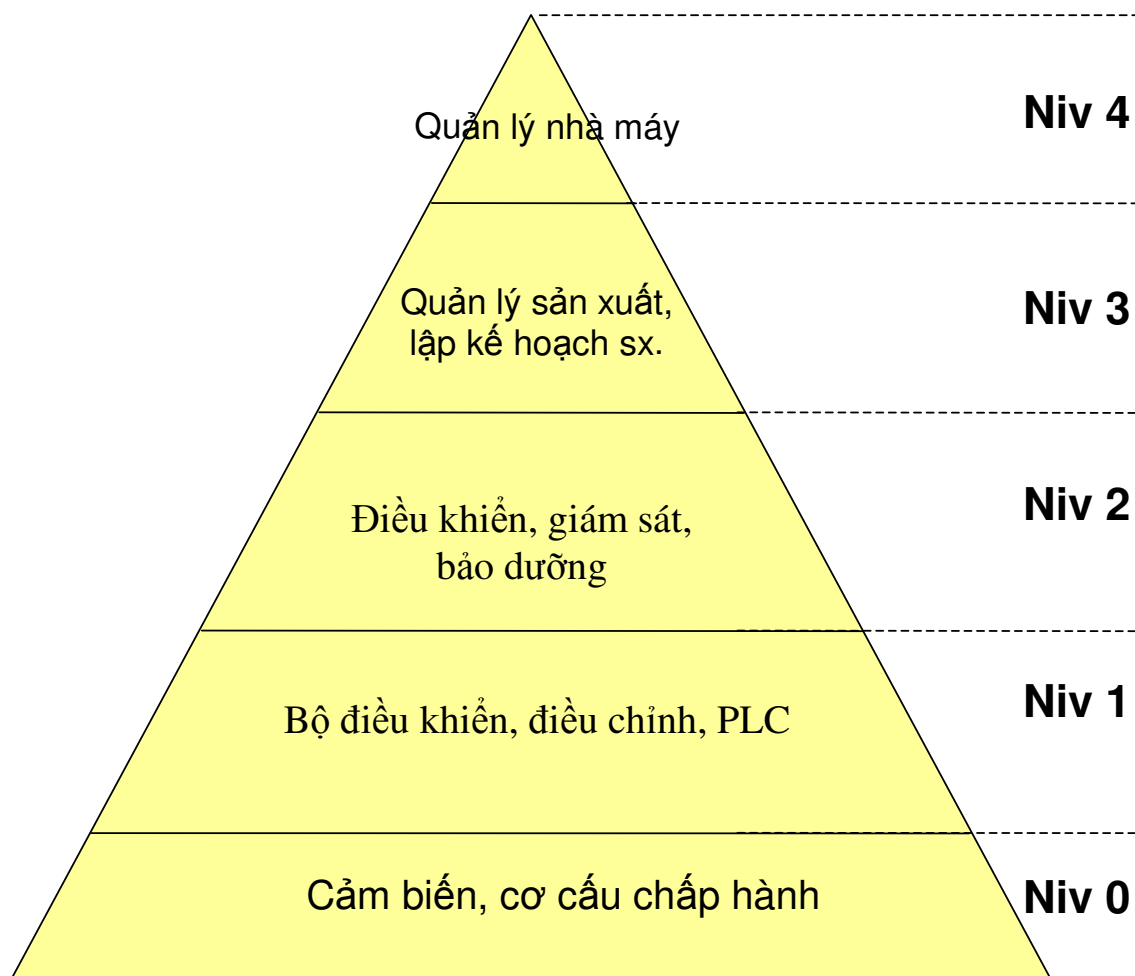


### 3.4 Phân loại theo mô tả toán học

- Hệ tuyến tính: đặc tính tính của tất cả các phân tử có trong hệ thống là tuyến tính. Đặc điểm cơ bản: xếp chồng.
- Hệ phi tuyến: có ít nhất một đặc tính tính của một phân tử là một hàm phi tuyến.
- Hệ thống tuyến tính hóa: tuyến tính hóa từng phần của hệ phi tuyến với một số điều kiện cho trước để được hệ tuyến tính gần đúng.



## 4 Biểu đồ điều khiển tự động trong một nhà máy



## 5 Phép biến đổi Laplace

Giả sử có hàm  $f(t)$  liên tục, khả tích. Ảnh Laplace của  $f(t)$  qua phép biến đổi Laplace, ký hiệu là  $F(p)$  được tính theo định nghĩa:

$$F(p) = \int_0^{\infty} f(t)e^{-pt} dt$$

- $p$ : biến Laplace
- $f(t)$ : hàm gốc
- $F(p)$ : hàm ảnh

### Một số tính chất của phép biến đổi Laplace

#### 1. Tính tuyến tính

$$L\{af_1(t) + bf_2(t)\} = aF_1(p) + bF_2(p)$$

#### 2. Ảnh Laplace của đạo hàm hàm gốc

$$L\{f'(t)\} = pF(p) - f(0)$$

Nếu các điều kiện đầu bằng 0 thì:

$$L\{f^{(n)}(t)\} = p^n F(p)$$

3. Ảnh laplace của tích phân hàm gốc

$$L\left\{\int_0^t f(\tau)d\tau\right\} = \frac{F(p)}{p}$$

4. Ảnh laplace của hàm gốc có trễ

$$L\{f(t-\tau)\} = e^{-p\tau}F(p)$$

5. Hàm ảnh có trễ

$$L\{e^{-at}f(t)\} = F(p+a)$$

6. Giá trị đầu của hàm gốc

$$f(0) = \lim_{p \rightarrow \infty} pF(p)$$

7. Giá trị cuối của hàm gốc

$$f(\infty) = \lim_{p \rightarrow 0} pF(p)$$

### ẢNH LAPLACE VÀ ẢNH Z CỦA MỘT SỐ HÀM THÔNG DỤNG

<b>f(t)</b>	<b>F(p)</b>	<b>F(z)</b>
$\delta(t)$	1	1
1	$\frac{1}{p}$	$\frac{z}{z-1}$
t	$\frac{1}{p^2}$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{2t^2}$	$\frac{1}{p^3}$	$\frac{T^2z(z+1)}{2(z-1)^3}$
$e^{-at}$	$\frac{1}{p+a}$	$\frac{z}{z-e^{-aT}}$
$1-e^{-at}$	$\frac{a}{p(p+a)}$	$\frac{(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$
sinat	$\frac{a}{p^2+a^2}$	$\frac{z \sin aT}{z^2 - 2z \cos aT + 1}$
cosat	$\frac{p}{p^2+a^2}$	$\frac{z^2 - z \cos aT}{z^2 - 2z \cos aT + 1}$

# Chương 1

## MÔ TẢ TOÁN HỌC CÁC PHẦN TỬ VÀ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG

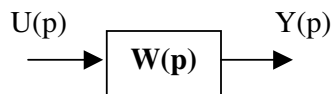
### 1 Khái niệm chung

- Để phân tích một hệ thống, ta phải biết nguyên tắc làm việc của các phần tử trong sơ đồ, bản chất vật lý, các quan hệ vật lý, ...
- Các tính chất của các phần tử/hệ thống được biểu diễn qua các phương trình động học, thường là phương trình vi phân.
- Để thuận lợi hơn trong việc phân tích, giải quyết các bài toán điều khiển, người ta mô tả toán học các phần tử và hệ thống bằng **hàm truyền đạt** (transfer function), **phương trình trạng thái** (state space), v.v

### 2 Hàm truyền đạt

#### 2.1 Định nghĩa :

*Hàm truyền đạt của một khâu (hay hệ thống) là tỉ số giữa tín hiệu ra với tín hiệu vào biểu diễn theo toán tử laplace, ký hiệu là  $W(p)$ , với các điều kiện ban đầu triệt tiêu.*



trong đó  $W(p) = \frac{Y(p)}{U(p)}$

với

$$\begin{aligned} y(0) = y'(0) = \dots = y^{(n-1)}(0) &= 0 \\ u(0) = u'(0) = \dots = u^{(m-1)}(0) &= 0 \end{aligned}$$

#### 2.2 Phương pháp tìm hàm truyền đạt

Từ phương trình vi phân tổng quát của một khâu (hệ thống) có dạng

$$a_n \frac{d^n y(t)}{dt^n} + \dots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_m \frac{d^m u(t)}{dt^m} + \dots + b_1 \frac{du(t)}{dt} + b_0 u(t) \quad (1.1)$$

biến đổi laplace với các điều kiện ban đầu bằng 0 và theo định nghĩa, ta có dạng tổng quát của hàm truyền đạt

$$W(p) = \frac{b_m p^m + \dots + b_1 p + b_0}{a_n p^n + \dots + a_1 p + a_0} = \frac{M(p)}{N(p)} \quad (1.2)$$

$N(p)$  : đa thức đặc tính

#### Ý nghĩa

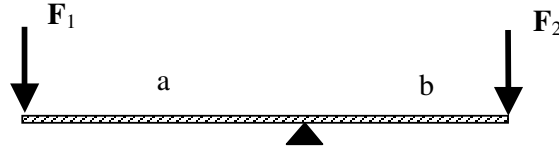
- Quan sát hàm truyền đạt, nhận biết cấu trúc hệ thống
- Xác định tín hiệu ra theo thời gian (biến đổi laplace ngược)
- Xác định các giá trị đầu, giá trị xác lập của hệ thống
- Xác định được hệ số khuếch đại tĩnh của hệ thống
- ...

### 2.3 Một số ví dụ về cách tìm hàm truyền đạt

Nguyên tắc chung :

- Thành lập phương trình vi phân ;
- Sử dụng phép biến đổi laplace để đưa về dạng hàm truyền đạt theo định nghĩa.

**Ví dụ 1 :** Khuếch đại lực bằng cánh tay đòn

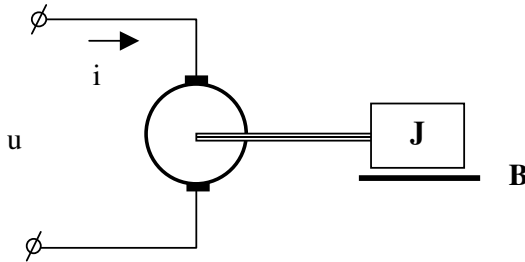


Xét phương trình cân bằng về mômen :

$$F_1(t) \cdot a = F_2(t) \cdot b \quad \Rightarrow \quad F_1(p) \cdot a = F_2(p) \cdot b$$

$$W(p) = \frac{F_2(p)}{F_1(p)} = \frac{a}{b}$$

**Ví dụ 2 :** Động cơ điện một chiều kích từ độc lập



Giả sử từ thông  $\Phi = \text{const}$ ,  $J$  là mômen quán tính quy về trục động cơ,  $B$  là hệ số ma sát ở trục.

Thành lập hàm truyền đạt của động cơ với:

$u$ : tín hiệu vào là điện áp phần ứng

$\omega$ : tín hiệu ra là góc quay của trục động cơ.

*Giải:*

Phương trình quan hệ về điện áp phần ứng:

$$u = Ri + L \frac{di}{dt} + e_u$$

$$e_u = K_e \Phi \omega$$

Suy ra

$$u = Ri + L \frac{di}{dt} + K_e \Phi \omega \tag{1.3}$$

Phương trình quan hệ về momen trên trục động cơ:

$$K_t \Phi i = J \frac{d\omega}{dt} + B\omega \tag{1.4}$$

Thay (1.4) vào (1.3), ta được:

$$u = \frac{R}{K_t \Phi} \left( J \frac{d\omega}{dt} + B\omega \right) + \frac{L}{K_t \Phi} \left( J \frac{d^2\omega}{dt^2} + B \frac{d\omega}{dt} \right) + K_e \Phi \omega$$

$$\Rightarrow u = \frac{LJ}{K_i\Phi} \frac{d^2\omega}{dt^2} + \frac{RJ+LB}{K_i\Phi} \frac{d\omega}{dt} + \left( \frac{RB}{K_i\Phi} + K_e\Phi \right) \omega$$

Vậy

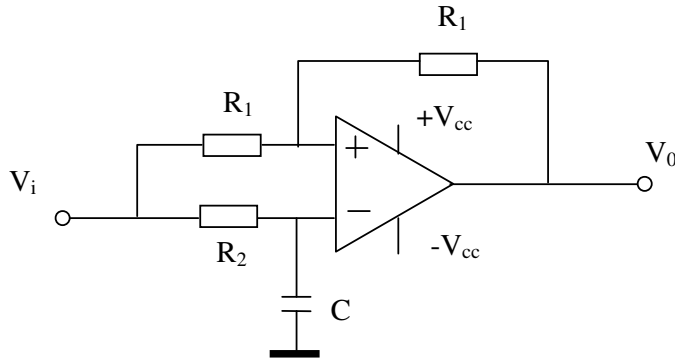
$$U(p) = (a_2p^2 + a_1p + a_0) \omega(p)$$

với  $a_2 = \frac{LJ}{K_i\Phi}; a_1 = \frac{RJ+LB}{K_i\Phi}; a_0 = \left( \frac{RB}{K_i\Phi} + K_e\Phi \right)$

Hàm truyền đạt của động cơ điện một chiều là:

$$W(p) = \frac{\omega(p)}{U(p)} = \frac{1}{a_2p^2 + a_1p + a_0}$$

**Ví dụ 3:** Tìm hàm truyền đạt của mạch điện tử dùng KĐTТ, giả thiết khuếch đại thuật toán là lý tưởng.



Ta có:

$$\frac{V_i - V^-}{R_2} = C \frac{dV^-}{dt} \Rightarrow V_i = V^- + R_2C \frac{dV^-}{dt} \tag{1.5}$$

Xét dòng điện qua V<sup>+</sup>

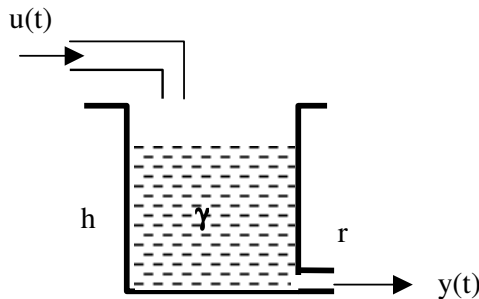
$$\frac{V_i - V^+}{R_1} = \frac{V^+ - V_0}{R_1} \Rightarrow V_i = 2V^+ + V_0 \tag{1.6}$$

Mặt khác, do giả thiết KĐTТ là lý tưởng nên V<sup>-</sup> = V<sup>+</sup>.

Từ (1.5) và (1.6)

$$R_2C \frac{dV_0}{dt} + V_0 = R_2C \frac{dV_i}{dt} - V_i \Rightarrow W(p) = \frac{V_0(p)}{V_i(p)} = \frac{R_2Cp - 1}{R_2Cp + 1}$$

**Ví dụ 4:**



Trong đó:  $u(t)$ : lưu lượng chất lỏng vào;  $y(t)$  là lưu lượng chất lỏng ra;  $A$  là diện tích đáy của bể chất lỏng.

Gọi  $p(t)$  là áp suất của chất lỏng tại đáy bể, biết các quan hệ sau:

$$y(t) = \frac{p(t)}{r} \quad (r \text{ là hệ số})$$

$$p(t) = \gamma h(t)$$

Tìm hàm truyền đạt của bể chất lỏng.

*Giải*

Theo các quan hệ trong giả thiết, ta có:

$$y(t) = \frac{p(t)}{r} = \frac{\gamma}{r} h \tag{1.7}$$

Độ gia tăng chiều cao cột chất lỏng là:

$$\frac{dh}{dt} = \frac{u(t) - y(t)}{A} \tag{1.8}$$

Từ (1.7) và (1.8), suy ra:

$$\frac{dy}{dt} = \frac{\gamma}{r} \frac{u(t) - y(t)}{A} \Rightarrow rA \frac{dy}{dt} + y(t) = \gamma u(t)$$

Hàm truyền đạt của bể chất lỏng trên là:

$$W(p) = \frac{Y(p)}{U(p)} = \frac{\gamma}{rAp + 1} = \frac{K}{Tp + 1}$$

### 2.4 Hàm truyền đạt của một số thiết bị điển hình

- Các thiết bị đo lường và biến đổi tín hiệu:  $W(p) = K$
- Động cơ điện một chiều:  $W(p) = \frac{K}{T_1 T_2 p^2 + T_2 p + 1}$
- Động cơ không đồng bộ 3 pha  $W(p) = \frac{K}{Tp + 1}$
- Lò nhiệt  $W(p) = \frac{K}{Tp + 1}$
- Băng tải  $W(p) = Ke^{-p\tau}$

### 2.5 Đại số sơ đồ khối

Đại số sơ đồ khối là biến đổi một sơ đồ phức tạp về dạng đơn giản hơn để thuận tiện cho việc tính toán.

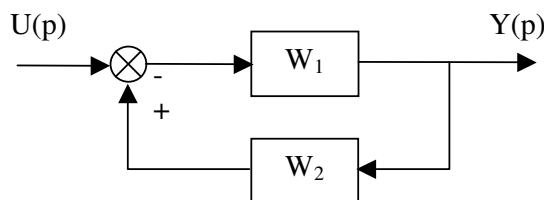
#### 2.5.1 Mắc nối tiếp

$$W(p) = W_1 \cdot W_2 \dots W_n$$

#### 2.5.2 Mắc song song

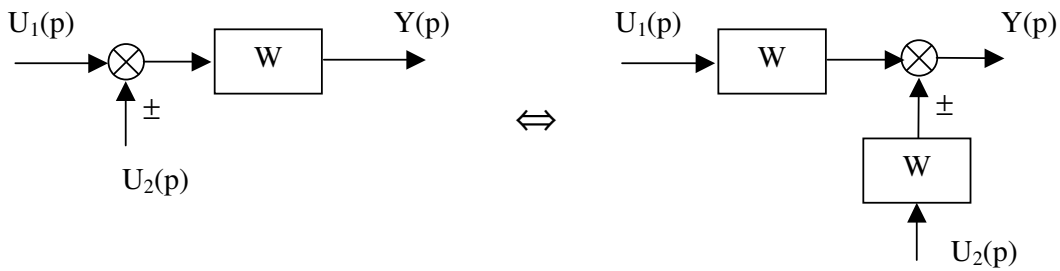
$$W(p) = W_1 \pm W_2 \pm \dots \pm W_n$$

#### 2.5.3 Mắc phản hồi

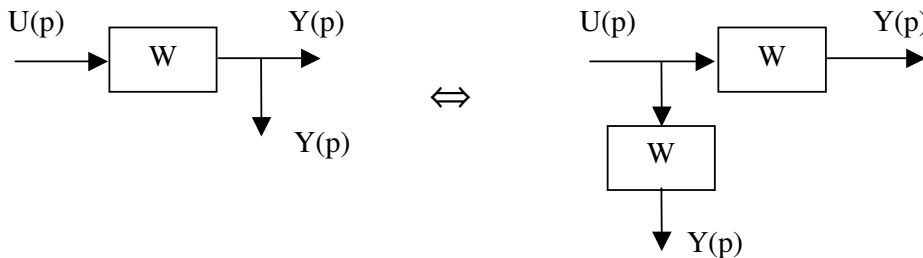


$$W(p) = \frac{W_1}{1 \pm W_1 W_2}$$

2.5.4 Chuyển tín hiệu vào từ trước ra sau một khối

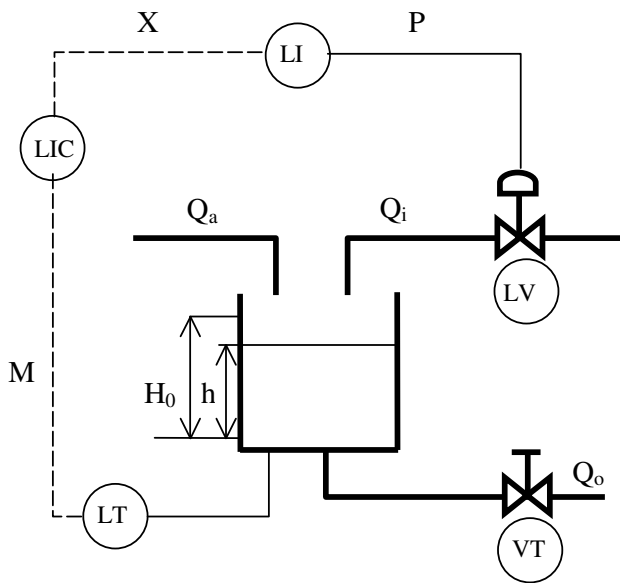


2.5.5 Chuyển tín hiệu ra từ sau ra trước một khối



**Ví dụ 1:** ĐIỀU KHIỂN MỨC CHẤT LỎNG TRONG BỂ CHỨA

Cho một hệ thống điều khiển tự động mức chất lỏng trong bể chứa như hình vẽ, biết rằng:



- LT : chuyển đổi mức chất lỏng
- LIC : Bộ hiệu chỉnh
- LY : chuyển đổi dòng điện/áp suất
- LV : van điều chỉnh tự động
- VT : van điều khiển bằng tay

- Hàm truyền của bộ chuyển đổi mức chất lỏng/dòng điện

$$G_{LR}(p) = \frac{1}{T_c p + 1} \text{ với } T_c = 1$$

- Phương trình vi phân biểu diễn quan hệ giữa lưu lượng và độ cao cột chất lỏng là:

$$\theta \frac{dh(t)}{dt} + h(t) = Q_i(t) + Q_a(t) \text{ với } \theta = 25$$

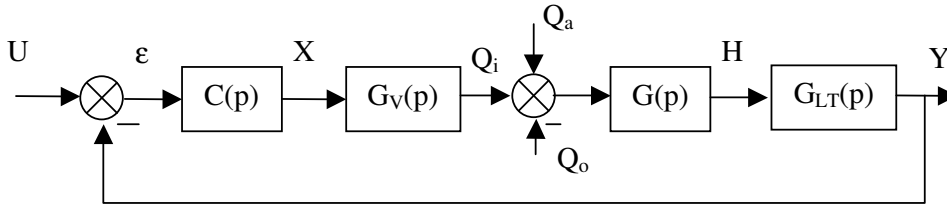
- Hàm truyền của cả bộ chuyển đổi dòng điện sang áp suất và van tự động là:

$$G_v(p) = \frac{Q_e(p)}{N(p)} = \frac{1}{T_v p + 1} = \text{với } T_v = 4$$

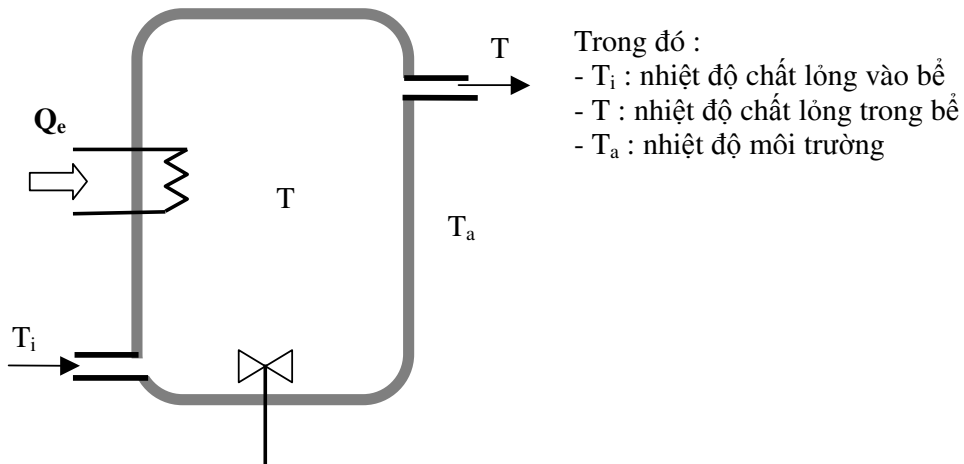
Yêu cầu :

1. Thành lập sơ đồ điều khiển của hệ thống.
2. Tìm các hàm truyền đạt  $W_{HU}(p), W_{HQ_a}(p), W_{HQ_0}(p)$
3. Giả sử chưa có bộ điều khiển  $C(p) = 1$ . Tìm giá trị xác lập của cột nước ở ngõ ra nếu  $u(t) = 5.1(t)$  và  $Q_a = 2.1(t)$ .

**ĐS**



**Ví dụ 2 :** Cho mô hình của một bể điều hòa nhiệt độ chất lỏng như hình vẽ



Trong đó :  
 -  $T_i$  : nhiệt độ chất lỏng vào bể  
 -  $T$  : nhiệt độ chất lỏng trong bể  
 -  $T_a$  : nhiệt độ môi trường

Biết rằng :

- Nhiệt lượng chất lỏng mang vào bể :  $Q_i = VHT_i$   
 với H là hệ số nhiệt ; V là lưu lượng chất lỏng vào bể.
- Nhiệt lượng điện trở cung cấp cho bể  $Q_e(t)$
- Nhiệt lượng chất lỏng mang ra khỏi bể  $Q_0 = VHT$
- Nhiệt lượng tổn thất qua thành bể do chênh lệch với môi trường  $Q_s = \frac{1}{R}(T - T_a)$

Biết nhiệt lượng chất lỏng nhận được sẽ làm tăng nhiệt độ chất lỏng theo biểu thức  $Q_l = C \frac{dT}{dt}$

Hãy thành lập mô hình điều khiển của bể trao đổi nhiệt ở trên.

*Giải*

Phương trình cân bằng nhiệt của bể chất lỏng

$$Q_l = Q_i + Q_e - Q_0 - Q_a$$

Hay



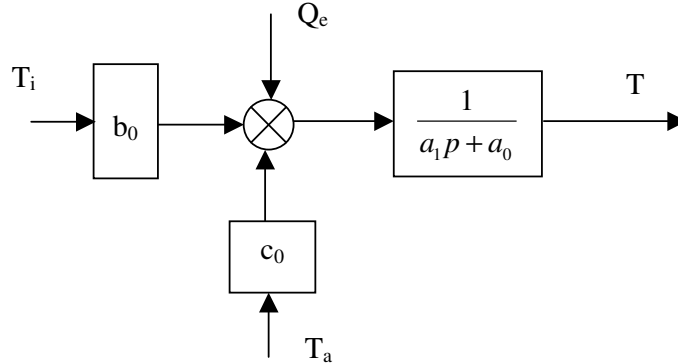
$$C \frac{dT}{dt} = VHT_i + Q_e - VHT - \frac{T - T_a}{R}$$

$$\Leftrightarrow C \frac{dT}{dt} + \left( \frac{1}{R} + VH \right) T = VHT_i + Q_e + \frac{1}{R} T_a$$

$$\Leftrightarrow (a_1 p + a_0) T(p) = b_0 T_i(p) + Q_e(p) + c_0 T_a(p)$$

$$\Leftrightarrow T(p) = \frac{1}{a_1 p + a_0} [b_0 T_i(p) + Q_e(p) + c_0 T_a(p)]$$

Mô hình điều khiển là :



Ngoài phương pháp đại số sơ đồ khối, chúng ta còn có thể dùng phương pháp Graph tín hiệu để tìm hàm truyền đạt tương đương của một hệ thống phức tạp.

### 3 Phương trình trạng thái

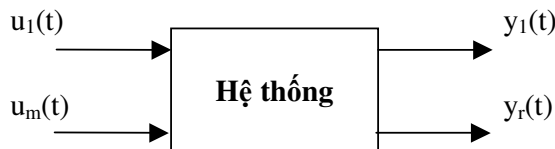
#### 3.1 Phương trình trạng thái tổng quát

##### 3.1.1 Khái niệm

- Đối với một hệ thống, ngoài tín hiệu vào và tín hiệu ra cần phải xác định, đôi khi ta cần quan sát các trạng thái khác. Ví dụ đối với động cơ điện là dòng điện, gia tốc động cơ, tổn hao, v.v...
- Khác với tín hiệu ra phải đo lường được bằng các bộ cảm biến, các biến trạng thái hoặc đo được, hoặc xác định được thông qua các đại lượng khác.
- Từ đó người ta xây dựng một mô hình toán cho phép ta có thể xác định được các biến trạng thái.

##### 3.1.2 Dạng tổng quát của phương trình trạng thái

Xét hệ thống có  $m$  tín hiệu vào và  $r$  tín hiệu ra.



Hệ thống có :

- m tín hiệu vào:  $u_1(t), u_2(t), \dots, u_m(t)$ , viết  $U = \begin{pmatrix} u_1 \\ \dots \\ u_m \end{pmatrix}, U \in \mathbb{R}^m$
- r tín hiệu ra:  $y_1(t), y_2(t), \dots, y_r(t)$ , viết  $Y = \begin{pmatrix} y_1 \\ \dots \\ y_r \end{pmatrix}, Y \in \mathbb{R}^r$
- n biến trạng thái:  $x_1(t), x_2(t), \dots, x_n(t)$ , viết  $X = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix}, X \in \mathbb{R}^n$

Phương trình trạng thái dạng tổng quát của hệ thống được biểu diễn dưới dạng :

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases}$$

Với  $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{r \times n}, D \in \mathbb{R}^{r \times m}$

A, B, C, D gọi là các ma trận trạng thái, nếu không phụ thuộc vào thời gian gọi là hệ thống dừng.

**Nhận xét :**

- Phương trình trạng thái mô tả toán học của hệ thống về mặt thời gian dưới dạng các phương trình vi phân.
- Hệ thống được biểu diễn dưới dạng các phương trình vi phân bậc nhất.

**3.1.3 Ví dụ thành lập phương trình trạng thái**

**Ví dụ 1**

Xây dựng phương trình trạng thái của một hệ thống cho dưới dạng phương trình vi phân như sau :

$$2 \frac{d^2 y}{dt^2} + \frac{dy}{dt} + 5y = u$$

*Giải*

Hệ có một tín hiệu vào và một tín hiệu ra.

$$x_1 = y$$

Đặt  $x_2 = \frac{dy}{dt} = \dot{y}$

Từ phương trình trên, ta có :

$$2\dot{x}_2 + x_2 + 5x_1 = u$$

Như vậy :

$$\begin{cases} \dot{x}_1 = \dot{y} = x_2 \\ \dot{x}_2 = -\frac{5}{2}x_1 - \frac{1}{2}x_2 + \frac{1}{2}u \end{cases}$$

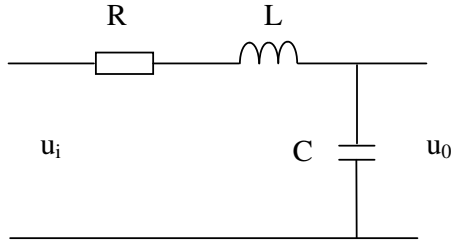
$$\Leftrightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{5}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Đặt A, B, C, D là các ma trận tương ứng, suy ra 
$$\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases}$$

**Ví dụ 2**

Cho mạch điện có sơ đồ như hình vẽ sau, hãy thành lập phương trình trạng thái cho mạch điện này với  $u_1$  là tín hiệu vào,  $u_2$  là tín hiệu ra.



**Giải**

Giả sử mạch hở tải và các điều kiện đầu bằng 0. Gọi  $i$  là dòng điện chạy trong mạch, ta có :

$$\begin{cases} u_i = Ri + L \frac{di}{dt} + \frac{1}{C} \int_0^t i dt \\ u_0 = \frac{1}{C} \int_0^t i dt \end{cases}$$

Đặt các biến trạng thái là :  $x_1 = i, x_2 = u_0$ , ta có :

$$\begin{cases} u_i = Rx_1 + L\dot{x}_1 + x_2 \\ C\dot{x}_2 = x_1 \end{cases} \quad \text{hay} \quad \begin{cases} \dot{x}_1 = -\frac{R}{L}x_1 - \frac{1}{L}x_2 + \frac{1}{L}u_i \\ \dot{x}_2 = \frac{1}{C}x_1 \end{cases} \quad \text{và } x_2 = u_0$$

Vậy :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u_i$$

$$u_0 = [0 \quad 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Hỏi : Trường hợp đặt  $x_1 = u_0, x_2 = i$ , phương trình trạng thái của mạch điện sẽ có dạng như thế nào ?

**Nhận xét**

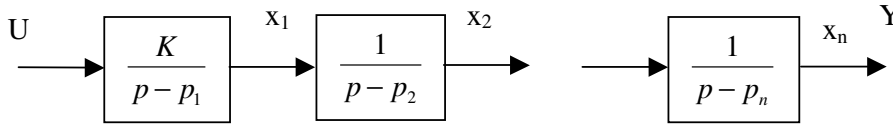
- Với cùng hệ thống sẽ có nhiều phương trình trạng thái khác nhau.
- Hàm truyền đạt của hệ thống là duy nhất.

**3.2 Xây dựng phương trình trạng thái từ hàm truyền đạt**

**3.2.1 Khai triển thành các thừa số đơn giản**

Nếu hàm truyền đạt được biểu diễn dưới dạng tích các thừa số như sau :

$$W(p) = \frac{Y(p)}{U(p)} = K \prod_{i=1}^n \frac{1}{(p - p_i)}$$



Đặt các biến trung gian như hình vẽ, ta có :

$$\begin{cases} \dot{x}_1 = p_1 x_1 + Ku \\ \dot{x}_2 = p_2 x_2 + x_1 \\ \dots \\ \dot{x}_n = p_n x_n + x_{n-1} \end{cases} \quad \text{và } y = x_n$$

Suy ra phương trình trạng thái là :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} p_1 & & & \\ & 1 & & \\ & & p_2 & \\ & & & \ddots \\ & 0 & & & 1 & & \\ & & & & & & p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} K \\ 0 \\ \vdots \\ 0 \end{bmatrix} u$$

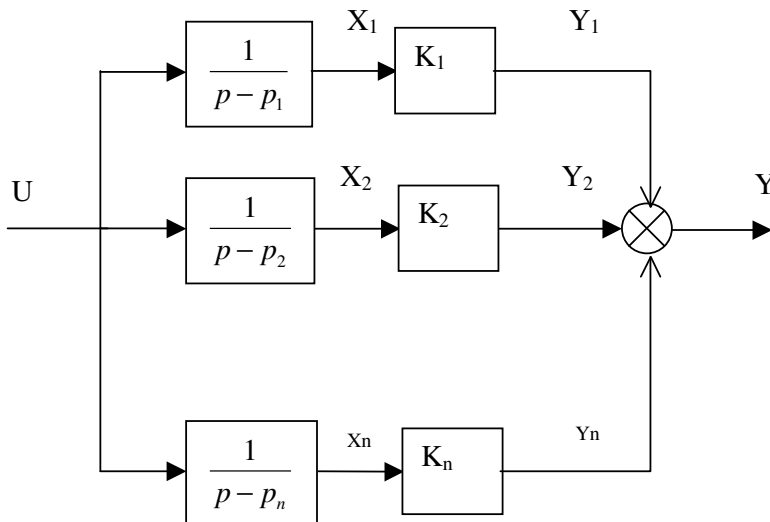
$$y = [0 \quad 0 \quad \dots \quad 1] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^T$$

### 3.2.2 Khai triển thành tổng các phân thức đơn giản

Nếu hàm truyền đạt được khai triển dưới dạng :

$$W(p) = \sum_{i=1}^n \frac{K_i}{p - p_i} = \frac{Y(p)}{U(p)} \quad \Rightarrow \quad Y(p) = \left[ \sum_{i=1}^n \frac{K_i}{p - p_i} \right] U(p)$$

Sơ đồ cấu trúc như sau :



Như vậy :  $pX_i = p_i X_i + U \quad \Rightarrow \quad \dot{x}_i = p_i x_i + u$

Hay 
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} p_1 & & & \\ & p_2 & & \\ & & \ddots & \\ 0 & & & p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u$$

$$y = [K_1 \quad K_2 \quad \dots \quad K_n] [x_1 \quad x_2 \quad \dots \quad x_n]^T$$

3.2.3 Sử dụng mô hình tích phân cơ bản

Trường hợp hàm truyền đạt có dạng

$$W(p) = \frac{Y(p)}{U(p)} = \frac{K}{a_n p^n + \dots + a_1 p + a_0}$$

Đặt  $x_1 = y, x_2 = \dot{x}_1 = \dot{y}, x_3 = \dot{x}_2 = \ddot{y}, \dots, x_n = \dot{y}^{(n-1)}, \dot{x}_n = \dot{y}^{(n)}$

Suy ra :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

...

$$\dot{x}_n = -\frac{a_1}{a_n} x_1 - \dots - \frac{a_{n-1}}{a_n} x_n + \frac{K}{a_n} u$$

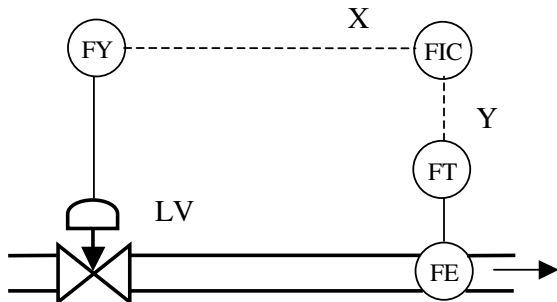
3.3 Chuyển đổi từ phương trình trạng thái sang hàm truyền

$$W(p) = C(pI - A)^{-1} B + D$$

MỘT SỐ BÀI TẬP CHƯƠNG 1

Bài tập 1 ĐIỀU KHIỂN LƯU LƯỢNG CHẤT LỎNG TRONG ỐNG DẪN

Cho sơ đồ điều khiển mực lưu lượng của một đường ống dẫn chất lỏng như hình vẽ



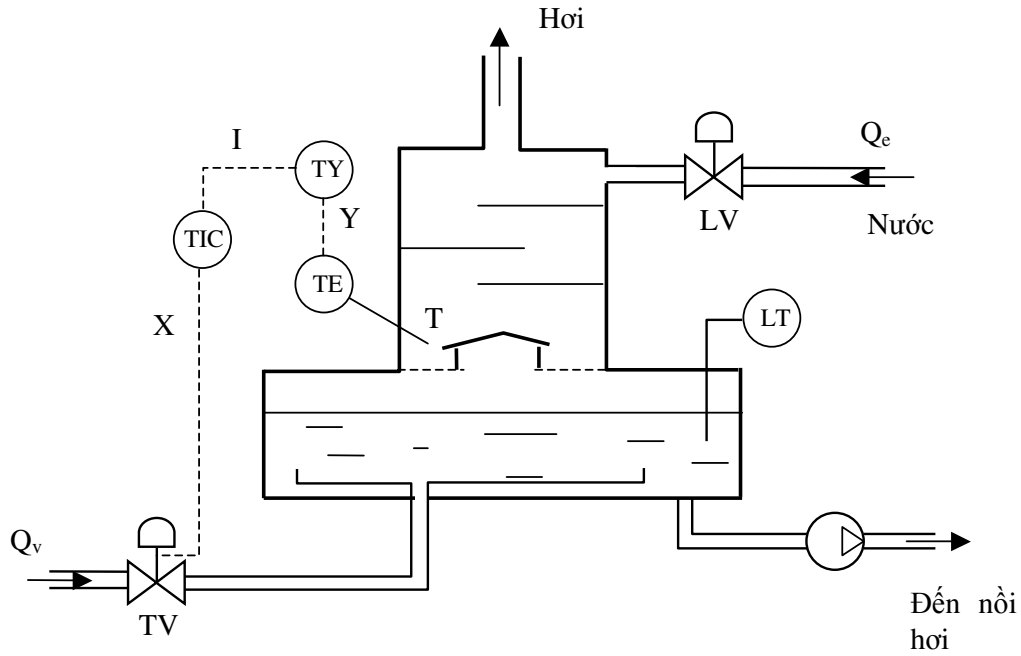
- FE : đo lưu lượng
- FT : chuyển đổi lưu lượng/ dòng điện
- FIC : bộ điều khiển lưu lượng
- FY : chuyển đổi dòng điện/áp suất

Biết hàm truyền của cơ cấu chuyển đổi từ dòng điện sang áp suất + van LV + đường ống + bộ chuyển đổi từ lưu lượng sang dòng điện là  $H(p) = \frac{Y(p)}{X(p)} = \frac{e^{-p}}{2.2p+1}$

Hãy thành lập mô hình điều khiển của hệ thống.

Bài tập 2 ĐIỀU CHỈNH NHIỆT ĐỘ CỦA MÁY LOẠI KHÍ CHO NỒI HƠI

Nước trước khi được đưa vào lò hơi cần phải qua máy loại khí nhằm loại bớt khí CO<sub>2</sub> và O<sub>2</sub> trong nước. Các loại khí này kém tan, chính vì vậy sẽ làm áp suất hơi thấp, nhiệt độ cao. Nước trong máy loại khí này có áp suất thấp và nhiệt độ bão hòa khoảng 104°C. Sơ đồ điều chỉnh nhiệt độ của máy loại khí như sau :



- |                                   |                                      |
|-----------------------------------|--------------------------------------|
| TE : đầu dò nhiệt độ              | TV : van tự động điều chỉnh nhiệt độ |
| TY : chuyển đổi điện áp/dòng điện | LT : bộ chuyển đổi mức               |
| TIC : bộ điều chỉnh nhiệt độ      | LV : van điều chỉnh mức              |

Hàm truyền của van điều chỉnh TV + nồi hơi + bộ đo TE là

$$T(p) = \frac{Y(p)}{X(p)} = \frac{2e^{-4p}}{8p+1}$$

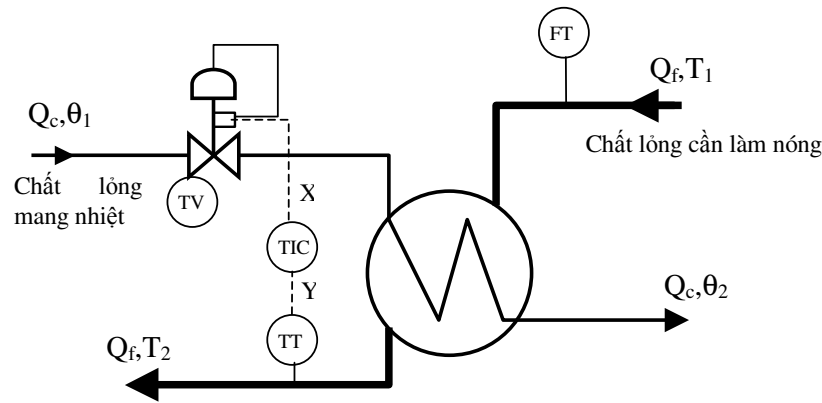
Bộ chuyển đổi điện áp/dòng điện TY có nhiệm vụ chuyển đổi tín hiệu điện áp ( vài micro volt) tỉ lệ với nhiệt độ thành tín hiệu dòng điện I (4-20mA) để đưa đến bộ điều chỉnh TIC. Hàm truyền của bộ chuyển đổi TY là :

$$C(p) = \frac{I(p)}{Y(p)} = \frac{1}{0.3p+1}$$

Hãy thành lập mô hình điều khiển của hệ thống.

**Bài tập 3** ĐIỀU CHỈNH NHIỆT ĐỘ CỦA BỘ TRAO ĐỔI NHIỆT

Sơ đồ của một bộ trao đổi nhiệt như hình vẽ, trong đó  $\theta_1 > T_1$ .



TT : bộ chuyển đổi nhiệt độ  
TIC : bộ điều chỉnh nhiệt độ

TV : van điều chỉnh nhiệt độ  
FT : bộ chuyển đổi lưu lượng

Yêu cầu điều khiển là giữ cho nhiệt độ ra  $T_2$  của chất lỏng cần làm nóng không đổi với mọi lưu lượng  $Q_f$ .

Một tín hiệu điều khiển  $X$  đưa đến van sẽ khống chế nhiệt độ  $T_2$  của chất lỏng, nhiệt độ này được thể hiện qua tín hiệu đo lường  $Y$ . Hàm truyền của van TV + bộ trao đổi nhiệt + bộ đo

TT là  $H(p) = \frac{Y(p)}{X(p)} = \frac{1.4}{(2p+1)^3}$ . Mặt khác, nếu giữ tín hiệu điều khiển  $X$  không đổi nhưng

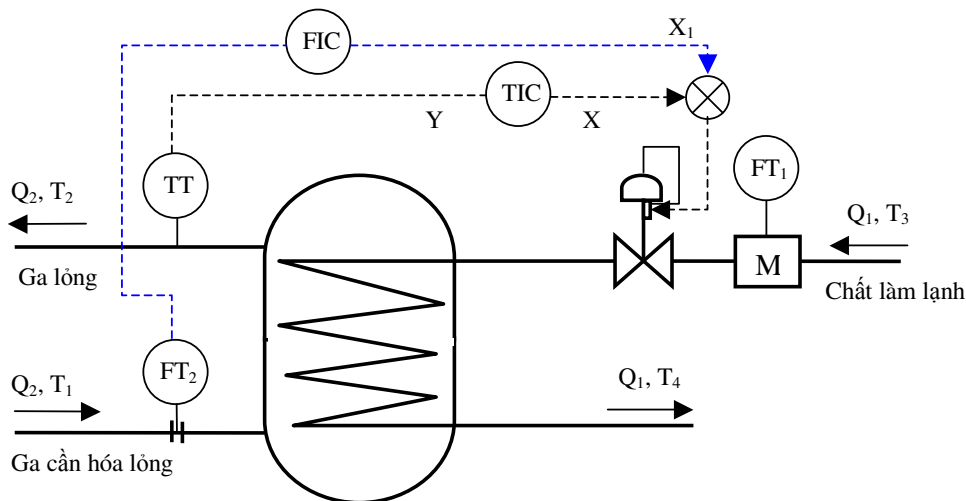
lưu lượng  $Q_f$  của chất lỏng cần làm nóng thay đổi cũng làm ảnh hưởng đến nhiệt độ ra  $T_2$ .

Ảnh hưởng của  $Q_f$  đến  $T_2$  được cho bởi hàm truyền  $D(p) = \frac{Y(p)}{Q_f(p)} = -\frac{2}{(0.5p+1)^2}$

Hãy thành lập mô hình điều khiển của hệ thống.

**Bài tập 4** ĐIỀU KHIỂN NHIỆT ĐỘ CỦA MỘT MÁY HÓA LỎNG GA (liquefacteur)

Sơ đồ khối của một máy hóa lỏng ga được cho trong hình sau :



Trong đó :

TT : bộ chuyển đổi nhiệt độ

TIC : bộ điều chỉnh nhiệt độ

FT<sub>1</sub> : bộ chuyển đổi lưu lượng (điện từ)

FT<sub>2</sub> : bộ chuyển đổi lưu lượng với đo lường tuyến tính

Để điều khiển nhiệt độ của ga đã được hóa lỏng, người ta đổi lưu lượng  $Q_1$  của chất làm lạnh bởi bộ điều khiển TIC. Ga trước khi hóa lỏng có nhiệt độ  $T_1$ , sau khi được hóa lỏng sẽ có nhiệt độ  $T_2$ . Hàm truyền của các khâu trong sơ đồ được định nghĩa như sau :

	$H_1(p) = \frac{T_2(p)}{Q_1(p)} = \frac{K_1 e^{-\tau_1 p}}{1 + \theta_1 p}$	$H_2(p) = \frac{T_2(p)}{Q_2(p)}$	$H_3(p) = \frac{T_2(p)}{T_3(p)}$
$H_4(p) = \frac{T_2(p)}{T_1(p)}$	$H_5(p) = \frac{Y(p)}{T_2(p)} = 1$	$H_6(p) = \frac{Q_1(p)}{X(p)} = 1$	

Với  $K_1=2$ ,  $\tau_1=1$  min,  $\theta_1=4$  min.

Hãy thành lập mô hình điều khiển của hệ thống.



## Chương 2

# ĐẶC TÍNH ĐỘNG HỌC CỦA CÁC KHÂU VÀ CỦA HỆ THỐNG TRONG MIỀN TẦN SỐ

### 1 Khái niệm chung

- Nhiệm vụ của chương : xây dựng đặc tính động học của khâu/hệ thống trong miền tần số. Mục đích :

- + Khảo sát tính ổn định
- + Phân tích tính chất
- + Tổng hợp bộ điều khiển

- Khâu động học : những đối tượng khác nhau có mô tả toán học như nhau được gọi là khâu động học. Có một số khâu động học không có phần tử vật lý nào tương ứng, ví dụ  $W(p) = Tp + 1$  hay  $W(p) = Tp - 1$ .

### 2 Phản ứng của một khâu

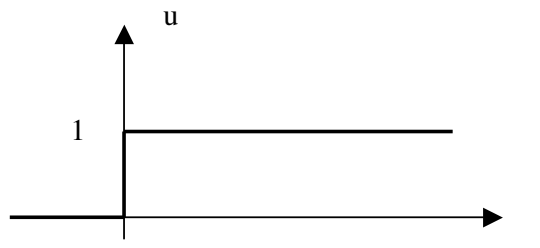
#### 2.1 Tín hiệu tác động vào một khâu (các tín hiệu tiên định)

##### 2.1.1 Tín hiệu bậc thang đơn vị

$$u(t) = 1(t) = \begin{cases} 1 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Dạng tổng quát

$$u(t) = U_0 1(t - t_0) = \begin{cases} U_0 & t \geq t_0 \\ 0 & t < t_0 \end{cases}$$

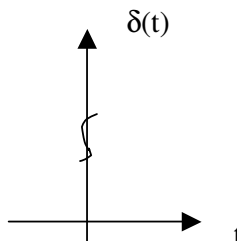


##### 2.1.2 Tín hiệu xung đơn vị

$$u(t) = \delta(t) = \frac{d1(t)}{dt} = \begin{cases} 0 & t \neq 0 \\ \infty & t = 0 \end{cases}$$

Tính chất :

$$\int_0^{\infty} \delta(t) dt = 1$$



##### 2.1.3 Tín hiệu điều hòa

$$u(t) = U_m \sin(\omega t + \varphi)$$

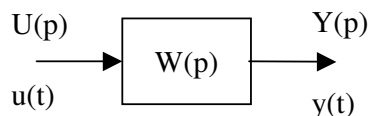
Biểu diễn dưới dạng số phức  $u(t) \rightarrow U_m e^{j(\omega t + \varphi)}$

##### 2.1.4 Tín hiệu bất kỳ

Đối với một tín hiệu vào bất kỳ, ta luôn có thể phân tích thành tổng của các tín hiệu đơn giản ở trên.

### 2.2 Phản ứng của một khâu

Cho một khâu được mô tả toán học như hình vẽ :



**Định nghĩa:** *Phản ứng của một khâu (hệ thống) đối với một tín hiệu vào xác định chính là đặc tính quá độ hay đặc tính thời gian của khâu đó.*

2.2.1 Hàm quá độ của một khâu

*Hàm quá độ của một khâu là phản ứng của khâu đối với tín hiệu vào  $1(t)$ .*

Ký hiệu :  $h(t)$

Biểu thức : 
$$h(t) = L^{-1} \left\{ \frac{W(p)}{p} \right\}$$

2.2.2 Hàm trọng lượng của một khâu

*Hàm trọng lượng của một khâu là phản ứng của khâu đối với tín hiệu vào  $\delta(t)$ .*

Ký hiệu :  $\omega(t)$

Biểu thức : 
$$\omega(t) = L^{-1} \{W(p)\} \quad \text{hay} \quad \omega(t) = \frac{dh(t)}{dt}$$

*Ví dụ :* Cho một khâu có hàm truyền đạt là

$$W(p) = \frac{5}{2p+1}$$

Tìm phản ứng của khâu đối với tín hiệu  $u(t) = 2.1(t-2) - 2.1(t-7)$ .

### 3 Đặc tính tần số của một khâu

#### 3.1 Hàm truyền đạt tần số

3.1.1 Định nghĩa:

*Hàm truyền đạt tần số của một khâu, ký hiệu là  $W(j\omega)$ , là tỉ số giữa tín hiệu ra với tín hiệu vào ở trạng thái xác lập khi tín hiệu vào biến thiên theo qui luật điều hòa  $u(t) = U_m \sin \omega t$ .*

- Ở trạng thái xác lập (nếu hệ thống ổn định):  $y_{xl}(t) = Y_m \sin(\omega t + \varphi)$
- Biểu diễn dưới dạng số phức :

$$u(t) \rightarrow e^{j(\omega t)}$$

$$y_{\infty}(t) \rightarrow Y_m e^{j(\omega t + \varphi)}$$

- Theo định nghĩa : 
$$W(j\omega) = \frac{y_{xl}(t)}{u(t)} = \frac{Y_m e^{j(\omega t + \varphi)}}{U_m e^{j(\omega t)}} = \frac{Y_m}{U_m} e^{j\varphi}$$

**Nhận xét:** Hàm truyền đạt tần số

- Là một số phức
- Phụ thuộc vào tần số tín hiệu.

Do  $W(j\omega)$  là số phức nên có thể biểu diễn nó như sau :

$$W(j\omega) = A(\omega)e^{j\varphi(\omega)}$$

$$W(j\omega) = P(\omega) + jQ(\omega)$$

3.1.2 Cách tìm hàm truyền đạt tần số từ hàm truyền đạt của một khâu

Có thể chứng minh được hàm truyền đạt tần số được tìm được từ hàm truyền đạt của một khâu (hệ thống) theo quan hệ sau :

$$W(j\omega) = W(p) \Big|_{p=j\omega}$$

*Ví dụ :* Tìm hàm truyền đạt tần số của khâu có hàm truyền  $W(p) = \frac{5}{2p+1}$ .

**Ý nghĩa của  $W(j\omega)$**

- Xác định được hệ số khuếch đại / góc lệch pha đối với tín hiệu xoay chiều
- Xác định được phương trình của tín hiệu ra ở trạng thái xác lập.

### 3.2 Đặc tính tần số

#### 3.2.1 Đặc tính tần số biên pha (Nyquist)

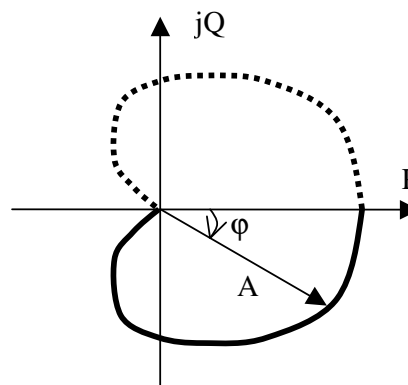
Xuất phát từ cách biểu diễn hàm truyền đạt tần số  $W(j\omega) = P(\omega) + jQ(\omega)$

- Xây dựng hệ trục với trục hoành P, trục tung Q.
- Khi  $\omega$  biến thiên, vẽ nên đặc tính tần số biên pha.

**Định nghĩa:** **Đặc tính tần số biên pha (ĐTBP)** là quỹ đạo của hàm truyền đạt tần số  $W(j\omega)$  trên mặt phẳng phức khi  $\omega$  biến thiên từ  $-\infty$  đến  $\infty$

Đặc điểm :

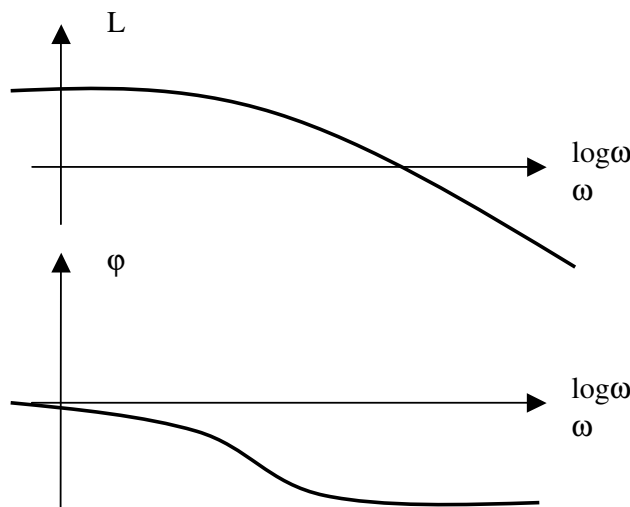
- ĐTBP đối xứng qua trục hoành nên chỉ cần xây dựng  $\frac{1}{2}$  đặc tính khi  $\omega$  biến thiên từ 0 đến  $\infty$  và lấy đối xứng qua trục hoành để được toàn bộ đặc tính.
- Có thể xác định được môđun A, góc pha  $\varphi$  từ ĐTBP



#### 3.2.2 Đặc tính tần số logarit (Bode)

Quan sát sự biến thiên của biên độ và góc pha theo tần số

Xây dựng hệ gồm 2 đặc tính :



#### \* Đặc tính tần số biên độ logarit ĐTBĐ

- Hoành độ là  $\omega$  hay  $\log\omega$  [dec]
- Tung độ L [dB]. Hàm L được xác định  
 $L = 20 \log A(\omega)$

ĐTBĐ biểu diễn biến thiên của hệ số khuếch đại tín hiệu theo tần số tín hiệu vào.

#### \* Đặc tính tần số pha logarit ĐTPL

- Hoành độ là  $\omega$  hay  $\log\omega$  [dec]
- Tung độ  $\varphi$  [rad], được xác định trong  $W(j\omega)$ .

ĐTPL biểu diễn biến thiên của góc pha theo tần số tín hiệu vào.

#### \* Đặc điểm của đặc tính logarit

Khi hệ thống có n khâu nối tiếp :

$$L = L_1 + L_2 + \dots + L_n$$

$$\varphi = \varphi_1 + \varphi_2 + \dots + \varphi_n$$

## 4 Đặc tính động học của một số khâu cơ bản

### 4.1 Khâu tỉ lệ

$$W(p) = K$$

#### 4.1.1 Hàm truyền đạt tần số

#### 4.1.2 Đặc tính Nyquist

$$P = K$$

$$Q = 0$$

#### 4.1.3 Đặc tính Bode

$$L = 20 \lg K$$

$$\varphi = 0$$

#### 4.1.4 Hàm quá độ

$$h(t) = K \cdot 1(t)$$

### 4.2 Khâu quán tính bậc 1

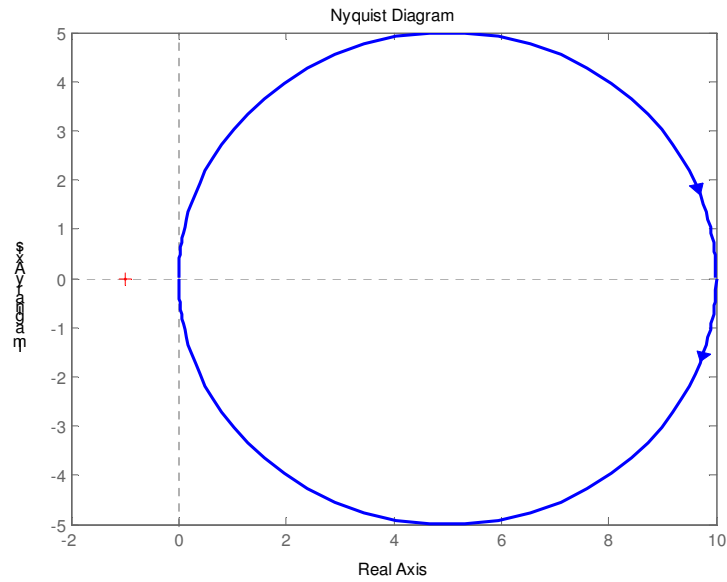
$$W(p) = \frac{K}{Tp+1}$$

#### 4.2.1 Hàm truyền đạt tần số

$$P = \frac{K}{T^2\omega^2 + 1}, \quad Q = -\frac{KT\omega}{T^2\omega^2 + 1}$$

$$A = \frac{K}{\sqrt{T^2\omega^2 + 1}}, \quad \varphi = -\arctg \omega T$$

#### 4.2.2 Đặc tính Nyquist

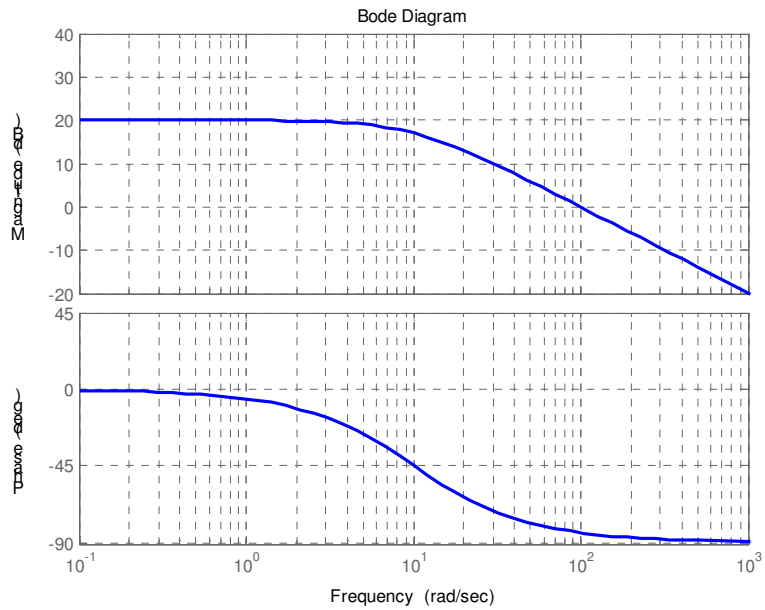


Đặc tính Nyquist của khâu quán tính bậc 1 ( $K = 10, T = 0.1$ )

4.2.3 Đặc tính Bode

$$L = 20 \lg K - 20 \lg \sqrt{T^2 \omega^2 + 1}$$

$$\varphi = -\arctg \omega T$$



Đặc tính Bode của khâu quán tính bậc 1 ( $K = 10, T = 0.1$ )

Trên hệ trục logarit, có thể vẽ đặc tính biên pha gần đúng của khâu quán tính bậc nhất như sau :

\* Đặc tính biên độ logarit

-  $\omega \rightarrow 0 : L \rightarrow L_1 = 20 \lg K;$

-  $\omega \rightarrow \infty : L \rightarrow L_2 = 20 \lg K - 20 \lg \omega;$

-  $\omega = \omega_g = 1/T : L_1(\omega_g) = L_2(\omega_g)$

\* Đặc tính pha logarit

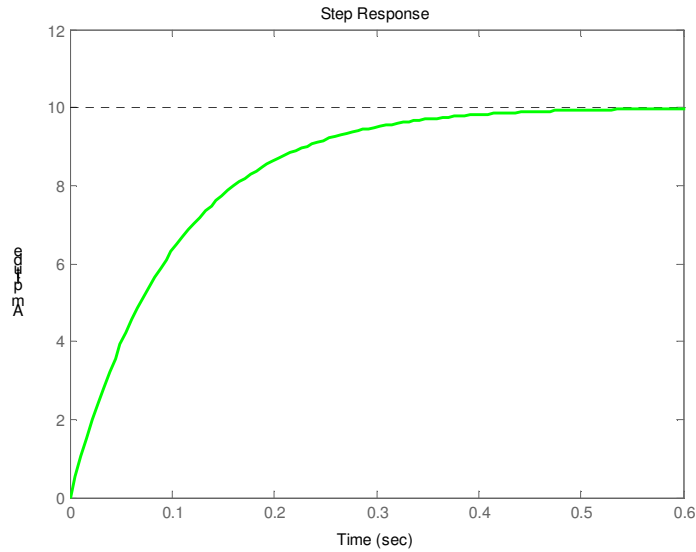
-  $\omega \rightarrow 0 : \varphi \rightarrow 0;$

- $\omega \rightarrow \infty : \varphi \rightarrow -\pi/2;$
- $\omega = \omega_g = 1/T : \varphi(\omega_g) = -\pi/4$

Chú ý: sai lệch giữa đặc tính gần đúng và đặc tính chính xác không được lớn hơn 3dB.

#### 4.2.4 Hàm quá độ

$$h(t) = K(1 - e^{-t/T})$$



Đặc tính quá độ của khâu quán tính bậc 1 (K = 10, T = 0.1)

### 4.3 Khâu dao động bậc 2

$$W(p) = K \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2}$$

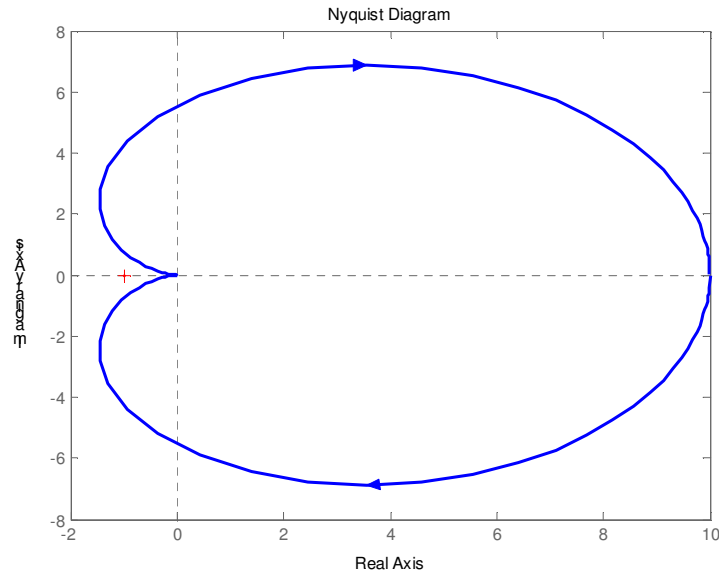
với  $\xi < 1$

#### 4.3.1 Hàm truyền đạt tần số

$$P = \frac{K\omega_0^2(\omega_0^2 - \omega^2)}{(\omega_0^2 - \omega^2)^2 + 4\xi^2\omega_0^2\omega^2}, \quad Q = -\frac{2K\xi\omega_0^3\omega}{(\omega_0^2 - \omega^2)^2 + 4\xi\omega_0^2\omega^2}$$

$$A = \frac{K\omega_0^2}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\xi^2\omega_0^2\omega^2}}, \quad \varphi = -\arctg \frac{2\xi\omega_0\omega}{(\omega_0^2 - \omega^2)}$$

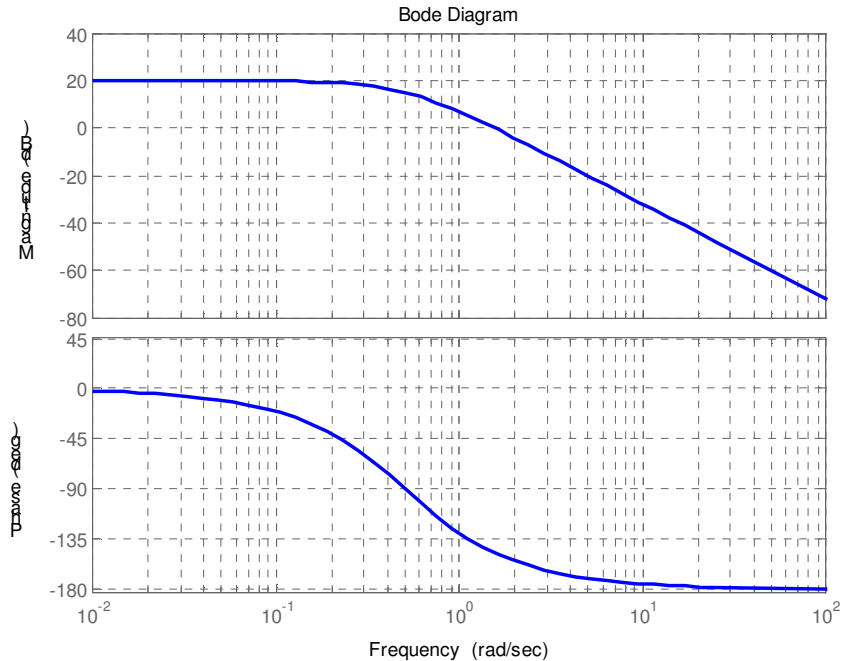
4.3.2 Đặc tính Nyquist



Đặc tính Nyquist của khâu dao động bậc 2 ( $K = 10, \omega_0 = 0.5, \xi = 0.9$ )

4.3.3 Đặc tính Bode

$$L = 20\lg K\omega_0^2 - 20\lg \sqrt{(\omega_0^2 - \omega^2)^2 + 4\xi^2\omega_0^2\omega^2}$$



Đặc tính Bode của khâu dao động bậc 2 ( $K = 10, \omega_0 = 0.5, \xi = 0.9$ )

Cách vẽ đặc tính biên pha gần đúng :

\* Đặc tính biên độ logarit

- $\omega \rightarrow 0 : L \rightarrow L_1 = 20\lg K;$
- $\omega \rightarrow \infty : L \rightarrow L_2 = 20\lg K\omega_0^2 - 40\lg \omega;$
- $\omega = \omega_g = \omega_0: L_1(\omega_g) = L_2(\omega_g).$

$\omega_0$  được gọi là tần số dao động tự nhiên

\* Đặc tính pha logarit

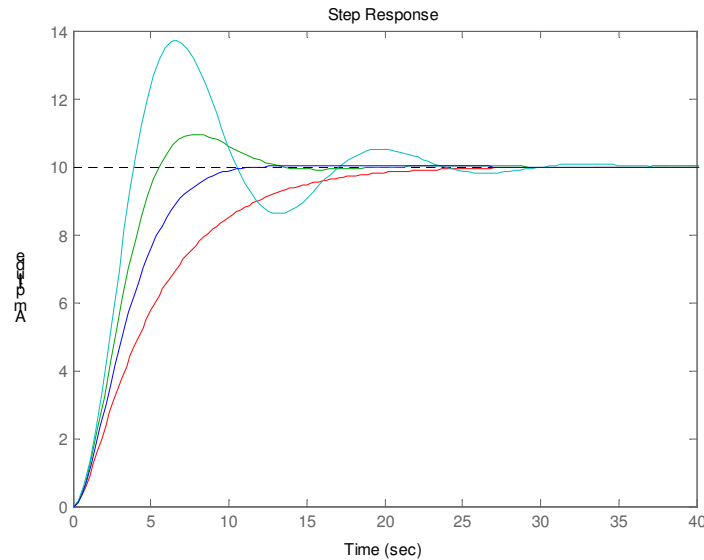
-  $\omega \rightarrow 0 : \varphi \rightarrow 0$ ;

-  $\omega \rightarrow \infty : \varphi \rightarrow -\pi$ ;

-  $\omega = \omega_g = \omega_0 : \varphi(\omega_g) = -\pi/2$

#### 4.3.4 Hàm quá độ

$$h(t) = K \left[ 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_0 t} \sin\left(\omega_0\sqrt{1-\xi^2}t + \arccos \xi\right) \right]$$



Đặc tính quá độ của khâu dao động bậc 2 với các hệ số  $\xi$  khác nhau

#### 4.4 Khâu không ổn định bậc 1

$$W(p) = \frac{K}{Tp-1}$$

##### 4.4.1 Hàm truyền đạt tần số

$$P = -\frac{K}{T^2\omega^2 + 1}, \quad Q = -\frac{KT\omega}{T^2\omega^2 + 1}$$

$$A = \frac{K}{\sqrt{T^2\omega^2 + 1}}, \quad \varphi = \arctg \omega T - \pi$$

##### 4.4.2 Đặc tính Nyquist

##### 4.4.3 Đặc tính Bode

$$L = 20\lg K - 20\lg\sqrt{T^2\omega^2 + 1}$$

$$\varphi = \arctg \omega T - \pi$$

##### 4.4.4 Hàm quá độ

$$h(t) = K(e^{t/T} - 1)$$



**4.5 Khâu vi phân lý tưởng**

$$W(p) = Kp$$

4.5.1 Hàm truyền đạt tần số

$$P = 0, \quad Q = K\omega$$

$$A = K\omega, \quad \varphi = \frac{\pi}{2}$$

4.5.2 Đặc tính Nyquist

4.5.3 Đặc tính Bode

$$L = 20\lg K + 20\lg \omega$$

**4.6 Khâu vi phân bậc 1**

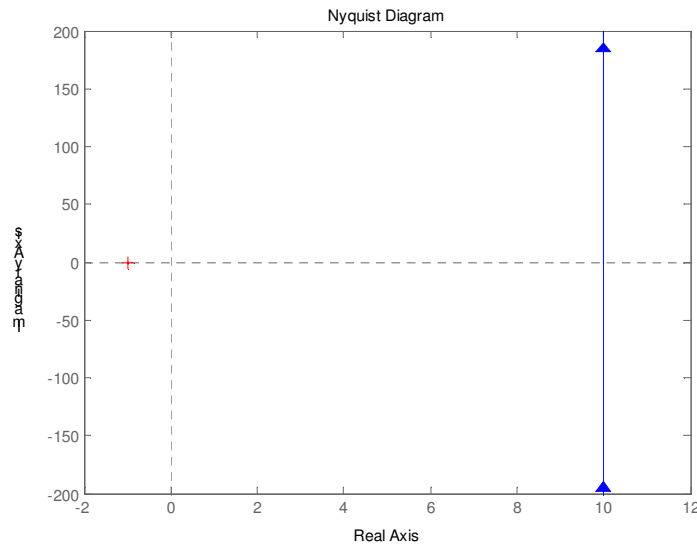
$$W(p) = K(Tp+1)$$

4.6.1 Hàm truyền đạt tần số

$$P = K, \quad Q = KT\omega$$

$$A = K\sqrt{T^2\omega^2 + 1}, \quad \varphi = \arctg T\omega$$

4.6.2 Đặc tính Nyquist

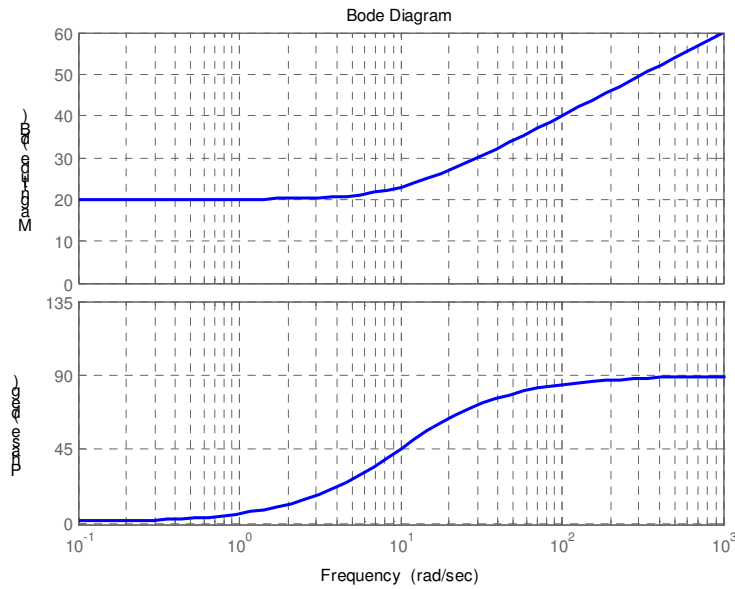


Đặc tính Nyquist của khâu vi phân bậc nhất

4.6.3 Đặc tính Bode

$$L = 20\log K + 20\log \sqrt{T^2\omega^2 + 1}$$

$$\omega_s = \frac{1}{T}$$



Đặc tính Bode của khâu vi phân bậc 1 (K = 10, T = 0.1)

**4.7 Khâu tích phân lý tưởng**

$$W(p) = \frac{K}{p}$$

4.7.1 Hàm truyền đạt tần số

$$P = 0, \quad Q = -\frac{K}{\omega}$$

$$A = \frac{K}{\omega}, \quad \varphi = -\frac{\pi}{2}$$

4.7.2 Đặc tính Nyquist

4.7.3 Đặc tính Bode

$$L = 20 \lg K - 20 \lg \omega$$

**4.8 Khâu chậm trễ**

$$W(p) = e^{-p\tau}$$

4.8.1 Hàm truyền đạt tần số

$$W(j\omega) = e^{-j\omega\tau}$$

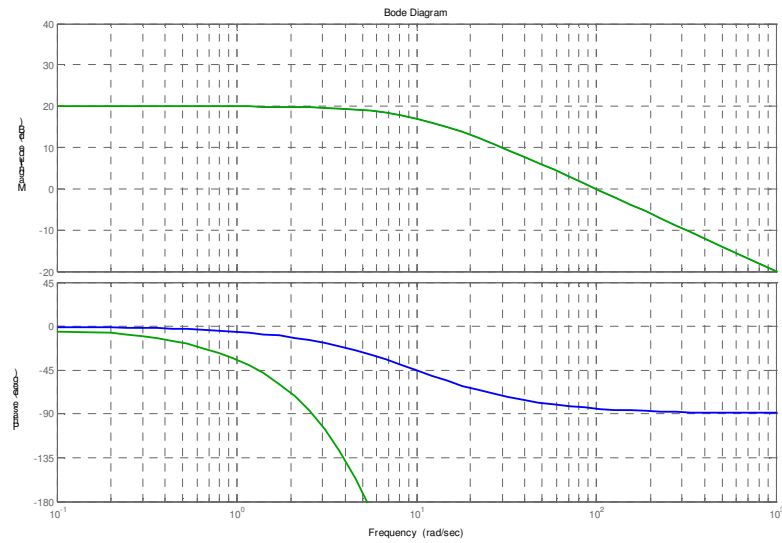
$$A = 1, \quad \varphi = -\omega\tau$$

4.8.2 Đặc tính Nyquist

4.8.3 Đặc tính Bode

$$L = 0$$

$$\varphi = -\omega\tau$$



Đặc tính Bode của khâu quán tính bậc 1 (xanh blue) và khâu quán tính bậc nhất có trễ 0.5s (xanh verte)

Các lệnh thực hiện vẽ đặc tính trên trong MATLAB :

```
num=10
den=[0.1 1]
W1=tf(num,den)
W2=W1;
set(W2,'IODelay,0.5);
W2
bode(W1);
hold on
bode(W2);
```

Chương 3

TÍNH ỔN ĐỊNH CỦA HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG

1 Khái niệm chung

Khảo sát một hệ thống điều khiển tự động được mô tả toán học dưới dạng hàm truyền đạt :

$$W(p) = \frac{b_m p^m + \dots + b_1 p + b_0}{a_n p^n + \dots + a_1 p + a_0} = \frac{Y(p)}{U(p)} \tag{3.1}$$

Phương trình vi phân tương ứng của hệ thống là :

$$a_n \frac{d^n y}{dt^n} + \dots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + \dots + b_1 \frac{du}{dt} + b_0 u \tag{3.2}$$

Nghiệm của phương trình vi phân (3.2) có dạng như sau :

$$y(t) = y_0(t) + y_{qd}(t) \tag{3.3}$$

Trong đó :

- $y_0(t)$  là nghiệm riêng của phương trình (3.2) có vẻ phải, đặc trưng cho quá trình xác lập.
- $y_{qd}(t)$  là nghiệm tổng quát của (3.2), đặc trưng cho quá trình quá độ.

**Tính ổn định của một hệ thống chỉ phụ thuộc vào quá trình quá độ, còn quá trình xác lập là một quá trình ổn định.**

**Định nghĩa :**

a) Một hệ thống ĐKTD ổn định nếu quá trình quá độ tắt dần theo thời gian.

$$\lim_{t \rightarrow \infty} y_{qd}(t) = 0$$

b) Một hệ thống ĐKTD không ổn định nếu quá trình quá độ tăng dần theo thời gian.

$$\lim_{t \rightarrow \infty} y_{qd}(t) = \infty$$

c) Một hệ thống ĐKTD ở biên giới ổn định nếu quá trình quá độ không đổi hay dao động không tắt dần.

Xét nghiệm  $y_{qd}(t)$  trong (3.3), dạng tổng quát của nghiệm quá độ như sau :

$$y_{qd}(t) = \sum_{i=1}^n C_i e^{p_i t} = \sum_{i=1}^n y_{qd,i} \tag{3.4}$$

với  $n$  là bậc và  $p_i$  là nghiệm của phương trình đặc tính

$$N(p) = a_n p^n + \dots + a_1 p + a_0 = 0 \tag{3.5}$$

$C_i$  là các hằng số (tính theo các điều kiện đầu).

**\* Khảo sát các trường hợp nghiệm  $p_i$  :**

i)  $p_i$  là nghiệm thực

$$p_i = \alpha_i \quad \Rightarrow \quad y_{qd,i} = C_i e^{\alpha_i t}$$

$$\lim_{t \rightarrow \infty} y_{qd,i} = \lim_{t \rightarrow \infty} C_i e^{\alpha_i t} = \begin{cases} 0, & \alpha_i < 0 \\ C_i, & \alpha_i = 0 \\ \infty, & \alpha_i > 0 \end{cases}$$

ii)  $p_i$  là cặp nghiệm phức liên hợp:

$$p_{i,i+1} = \alpha_i \pm j\beta_i \quad \Rightarrow \quad y_{qd,i} + y_{qd,i+1} = 2A_i e^{\alpha_i t} \cos(\beta_i t + \varphi_i)$$

$$\lim_{t \rightarrow \infty} (y_{qd,i} + y_{qd,i+1}) = \begin{cases} 0, & \alpha_i < 0 \\ \text{dao dong}, & \alpha_i = 0 \\ \infty, & \alpha_i > 0 \end{cases}$$

**Kết luận :**

- 1) Hệ thống điều khiển tự động ổn định nếu *tất cả* các nghiệm của phương trình đặc tính có *phần thực âm*.
- 2) Hệ thống điều khiển tự động không ổn định nếu có *ít nhất* một nghiệm của phương trình đặc tính có *phần thực dương*.
- 3) Hệ thống điều khiển tự động ở biên giới ổn định nếu có *ít nhất* một nghiệm của phương trình đặc tính có *phần thực bằng 0*, các nghiệm còn lại có *phần thực âm*.

**2 Tiêu chuẩn ổn định đại số**

**2.1 Điều kiện cần để hệ thống ổn định**

Xét một hệ thống điều khiển tự động có phương trình đặc tính tổng quát như sau :

$$N(p) = a_n p^n + \dots + a_1 p + a_0 = 0$$

Phát biểu :

« **Điều kiện cần để một hệ thống ĐKTD tuyến tính ổn định là tất cả các hệ số của phương trình đặc tính dương** »

**2.2 Tiêu chuẩn Routh**

**2.2.1 Cách thành lập bảng Routh**

$p^n$	$a_n$	$a_{n-2}$	$a_{n-4}$	...	$a_0$
$p^{n-1}$	$a_{n-1}$	$a_{n-3}$	$a_{n-5}$	...	$(a_0)$
$p^{n-2}$	$c_{n-2,1}$	$c_{n-2,2}$	...		
...					
$p^2$	$c_{2,1}$	$c_{2,2}$			
$p^1$	$c_{1,1}$	$c_{1,2}$			
$p^0$	$c_{0,1}$				

Với :

$$c_{n-2,1} = -\frac{\begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}}{a_{n-1}}; c_{n-2,2} = -\frac{\begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}}{a_{n-1}}; \dots$$

$$c_{0,1} = -\frac{\begin{vmatrix} c_{2,1} & c_{2,2} \\ c_{1,1} & c_{1,2} \end{vmatrix}}{c_{1,1}}$$

**Quy tắc :**

- Mỗi số hạng trong bảng Routh là một tỉ số, trong đó :
- Tử số là định thức bậc 2, mang dấu âm. Cột thứ nhất của định thức là cột thứ nhất của 2 hàng đứng sát trên hàng có số hạng đang tính ; cột thứ hai của định thức là cột đứng sát bên phải số hạng đang tính cũng của 2 hàng trên.
  - Mẫu số : Tất cả các số hạng trên cùng một hàng có cùng mẫu số là số hạng ở cột thứ nhất của hàng sát trên hàng có số hạng đang tính.

**2.2.2 Phát biểu tiêu chuẩn Routh**

**Điều kiện cần và đủ để hệ thống tuyến tính ổn định là tất cả các số hạng trong cột thứ nhất của bảng Routh phải dương.**

**2.2.3 Các tính chất của bảng Routh**

- Có thể nhân hoặc chia tất cả các số hạng trên cùng một hàng của bảng Routh với một số dương.
- Số lần đổi dấu của các số hạng trong cột thứ nhất của bảng Routh bằng số nghiệm của phương trình đặc tính có phần thực dương.

- Nếu trong cột thứ nhất của bảng Routh có một số hạng bằng 0 thì hệ thống cũng không ổn định. Để xác định số nghiệm âm, có thể thay số 0 bằng số  $\epsilon > 0$  rất bé để tiếp tục xác định các số hạng còn lại.
- Nếu tất cả các số hạng trên cùng 1 hàng của bảng Routh bằng 0 thì hệ thống ở biên giới ổn định.
- Trường hợp hệ thống có khâu chậm trễ, có thể khai triển Fourier hàm mũ như sau :

$$e^{-p\tau} = 1 + \frac{(-p\tau)}{1!} + \frac{(-p\tau)^2}{2!} + \dots$$

### 2.3 Tiêu chuẩn ổn định Hurwitz

#### 2.3.1 Phát biểu

**Điều kiện cần và đủ để hệ thống tuyến tính ổn định là các hệ số  $a_n$  và các định thức Hurwitz dương.**

#### 2.3.2 Cách thành lập định thức Hurwitz

Định thức  $\Delta_n$  có :

- $n$  cột và  $n$  hàng
- Đường chéo chính của  $\Delta_n$  bắt đầu từ  $a_1$  liên tiếp đến  $a_n$ .
- Các số hạng trong cùng một cột có chỉ số tăng dần từ dưới lên trên.
- Các số hạng có chỉ số lớn hơn  $n$  hay nhỏ hơn 0 ghi 0.

## 3 Tiêu chuẩn ổn định tần số

### 3.1 Tiêu chuẩn Nyquist theo đặc tính tần số biên pha

#### 3.1.1 Phát biểu

**Điều kiện cần và đủ để một hệ thống kín phản hồi -1 ổn định là :**

- Khi hệ hở ổn định hoặc ở biên giới ổn định, đặc tính tần số biên pha của hệ hở không bao điểm  $M(-1, j0)$ .
- Khi hệ hở không ổn định, đặc tính tần số biên pha của hệ hở bao điểm  $M(-1, j0)$   $m/2$  vòng kín khi  $\omega$  biến thiên từ 0 đến  $\infty$  với  $m$  là số nghiệm của phương trình đặc tính của hệ hở có phần thực dương.

#### 3.1.2 Áp dụng tiêu chuẩn

- Tiêu chuẩn này chỉ áp dụng cho hệ kín. Trường hợp không phải hệ phản hồi -1 thì chuyển về dạng phản hồi -1 tương đương.

- Có thể xác định số lần bao  $N$  của đặc tính tần số ( $\omega$  biến thiên từ 0 đến  $\infty$ ) với điểm  $M$  như sau :

$$N = \frac{\sum C_{(-\infty, 0)}^+ - \sum C_{(-\infty, 0)}^-}{2}$$

Với :

- +  $C^+$  giao điểm dương : là giao của  $W(j\omega)$  với trục thực, có chiều  $\uparrow$  theo chiều tăng của  $\omega$ .
- +  $C^-$  giao điểm âm : là giao của  $W(j\omega)$  với trục thực, có chiều  $\downarrow$  theo chiều tăng của  $\omega$ .

### 3.2 Tiêu chuẩn Nyquist theo đặc tính tần số logarit

#### 3.2.1 Phát biểu

**Điều kiện cần và đủ để hệ kín phản hồi -1 ổn định khi hệ hở ổn định (hay ở biên giới ổn định) là số giao điểm dương bằng số giao điểm âm trong phạm vi tần số  $\omega$  để  $L(\omega) > 0$ .**

#### 3.2.2 Áp dụng tiêu chuẩn

- Trong đặc tính logarit

- +  $C^+$  giao điểm dương : là giao của  $\varphi(\omega)$  với đường thẳng  $-\pi$ , có chiều  $\downarrow$  theo chiều tăng của  $\omega$ .
- +  $C^-$  giao điểm âm : là giao của  $\varphi(\omega)$  với đường thẳng  $-\pi$ , có chiều  $\uparrow$  theo chiều tăng của  $\omega$ .
- Tiêu chuẩn chỉ áp dụng cho hệ kín phản hồi -1, hệ hở đã ổn định.

### 3.3 Tiêu chuẩn ổn định Mikhailov

#### 3.3.1 Phát biểu

*Điều kiện cần và đủ để hệ thống tuyến tính ổn định là biểu đồ vector đa thức đặc tính  $A(j\omega)$  xuất phát từ trục thực dương quay n góc phần tư ngược chiều kim đồng hồ khi  $\omega$  tăng từ 0 đến  $\infty$ .*

#### 3.3.2 Áp dụng tiêu chuẩn

- Tiêu chuẩn này được áp dụng để xét ổn định cho hệ bất kỳ (hở/kín)
- Đa thức đặc tính là đa thức ở tử số của hàm truyền đạt.

## 4 Phương pháp quỹ đạo nghiệm số

Phương pháp quỹ đạo nghiệm số (QĐNS) thường dùng cho hệ thống có một thông số biến đổi tuyến tính. Với mỗi giá trị của thông số, phương trình đặc tính của hệ thống sẽ có một tập nghiệm, mỗi nghiệm được biểu diễn bằng một điểm trên mặt phẳng phức. Khi thông số biến đổi, nghiệm của phương trình đặc tính cũng biến đổi theo. *Quỹ đạo tạo ra từ các nghiệm của phương trình đặc tính trên mặt phẳng phức khi thông số biến đổi gọi là quỹ đạo nghiệm số.*

### 4.1 Phương pháp xây dựng QĐNS

Xét một hệ thống tuyến tính, trong đó phương trình đặc tính chứa một thông số K biến đổi dưới dạng:

$$N(p) = N_0(p) + KM_0(p) = 0 \tag{3.6}$$

với  $N(p)$ ,  $M(p)$  là hai đa thức bậc n, m tương ứng.

Gọi  $p_i (i = 1, 2, \dots, n)$  là nghiệm của phương trình  $N(p) = 0$

$p_i^* (i = 1, 2, \dots, n)$  là nghiệm của phương trình  $N_0(p) = 0$

$p_j^* (j = 1, 2, \dots, m)$  là nghiệm của phương trình  $M_0(p) = 0$

Có thể viết

$$N_0(p) = \prod_{i=1}^n (p - p_i^*) ; M_0(p) = \prod_{j=1}^m (p - p_j^*)$$

và 
$$N(p) = \prod_{i=1}^n (p - p_i) + K \prod_{j=1}^m (p - p_j^*)$$

#### 4.1.1 Xác định điểm xuất phát của QĐNS

Điểm xuất phát của QĐNS là vị trí nghiệm khi  $K = 0$ . Từ phương trình (3.6), điểm xuất phát của QĐNS chính là n nghiệm  $p_i^*$  của phương trình  $N_0(p) = 0$ .

#### 4.1.2 Xác định điểm kết thúc của QĐNS

Điểm kết thúc của QĐNS là vị trí nghiệm khi  $K \rightarrow \infty$ . Từ phương trình (3.6), có thể viết :

$$N(p) = \frac{1}{K} \prod_{i=1}^n (p - p_i) + \prod_{j=1}^m (p - p_j^*) = 0 \tag{3.7}$$

Rõ ràng, khi  $K \rightarrow \infty$ , nghiệm của  $N(p)$  cũng chính là m nghiệm  $p_j^*$  của phương trình  $M_0(p) = 0$ .

#### 4.1.3 Xác định số lượng quỹ đạo trên mặt phẳng nghiệm

Phương trình  $N(p) = 0$  có n nghiệm xuất phát, do vậy khi K biến thiên sẽ vạch nên n quỹ đạo trên mặt phẳng nghiệm. Do có m điểm kết thúc của quỹ đạo nên nếu  $m < n$  thì :

- $m$  quỹ đạo xuất phát từ  $p_i'$  và kết thúc ở  $p_j''$  ;
- $(n - m)$  quỹ đạo xuất phát từ  $p_i'$  và tiến ra vô cùng.

Khi phương trình  $N_0(p) = 0$  có nghiệm phức liên hợp thì cặp quỹ đạo tương xứng của nó sẽ đối xứng qua trục thực.

#### 4.1.4 Xác định các đường tiệm cận

Có  $(n-m)$  đường thẳng tiệm cận cho các quỹ đạo tiến ra vô cùng.

- Tâm tiệm cận :  $R_0 = \frac{1}{n-m} \sum_{i=1}^n p_i' - \sum_{j=1}^m p_j''$
- Góc tạo bởi các đường tiệm cận và trục hoành :  $\alpha_k = \frac{2k+1}{n-m} \pi, k = 0, 1, \dots, n-m-1$

#### 4.1.5 Xác định điểm tách khỏi trục thực và hướng dịch chuyển của quỹ đạo

- Khảo sát hàm số  $f(p) = \frac{N_0(p)}{M_0(p)}$  để xác định hướng di chuyển của quỹ đạo
- Các nghiệm của phương trình  $\frac{df(p)}{dp} = 0$  chính là các điểm tách khỏi trục thực của QĐNS.

#### 4.1.6 Xác định giao điểm của trục ảo với QĐNS

Gọi  $\pm j\omega_c$  là điểm của QĐNS với trục ảo. Thay  $p = j\omega_c$  vào phương trình đặc tính  $N(p) = 0$ ,  $\omega_c$  được xác định từ hệ phương trình :

$$\begin{aligned} \operatorname{Re}al(N(j\omega_c)) &= 0 \\ \operatorname{Im}(N(j\omega_c)) &= 0 \end{aligned}$$

Ví dụ : Vẽ QĐNS của một hệ thống có phương trình đặc tính có thông số  $K$  biến thiên như sau :

$$N(p) = p^3 + 3p^2 + (K + 2)p + 10K = 0$$

*Giải :*

Trước tiên, ta biến đổi phương trình ở trên về dạng 3.6 như sau :

$$N(p) = (p^3 + 3p^2 + 2p) + K(p + 10) = 0$$

Như vậy :  $N_0(p) = (p^3 + 3p^2 + 2p)$  và  $M_0(p) = (p + 10)$

- Các điểm xuất phát của QĐNS :

$$N_0(p) = 0 \quad p_1' = 0; p_2' = 1; p_3' = 2;$$

- Các điểm kết thúc của QĐNS :

$$M_0(p) = 0 \quad p_1'' = 10$$

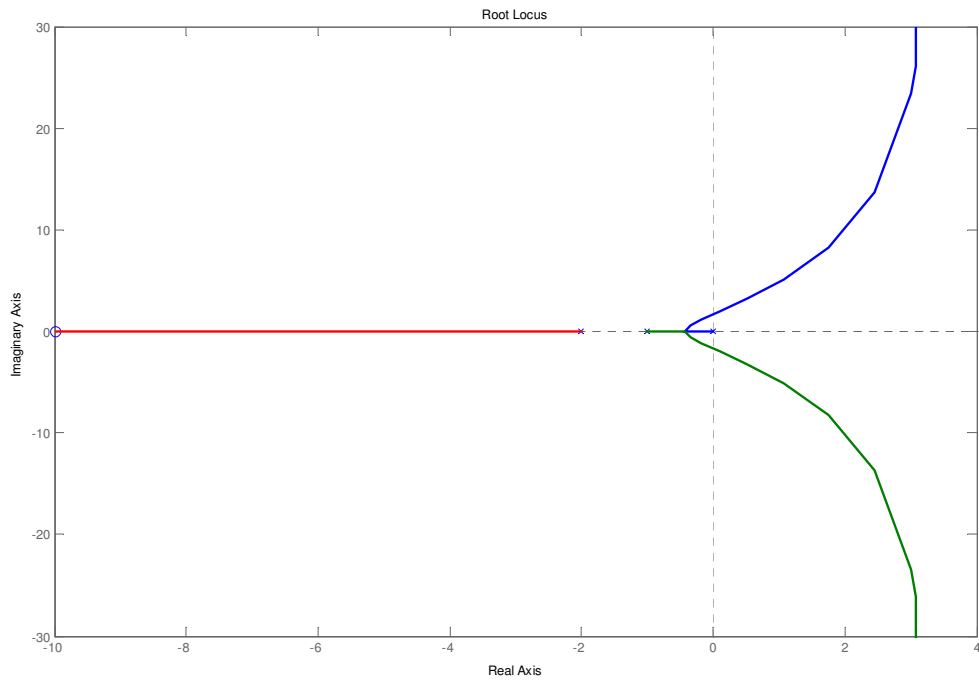
- Vậy có 3 điểm xuất phát, 1 điểm kết thúc nên sẽ có 2 quỹ đạo tiến ra vô cùng (tương ứng với 2 tiệm cận)

- Tâm tiệm cận :  $R_0 = 7$

- Góc các tiệm cận so với trục hoành :  $\alpha_k = (2k + 1) \frac{\pi}{2} = \frac{\pi}{2}; \frac{3\pi}{2}$

- Giao điểm với trục ảo :  $\omega_c = \sqrt{\frac{20}{7}}$  tại  $K = 6/7$ .





Hình vẽ trên biểu diễn Quỹ đạo nghiệm số của hệ thống trong ví dụ trên (được vẽ bằng MATLAB).

## Chương 4 CHẤT LƯỢNG CỦA QUÁ TRÌNH ĐIỀU KHIỂN

### 1 Khái niệm chung

Chất lượng của một hệ thống điều khiển tự động được đánh giá qua 2 chế độ : chế độ xác lập và quá trình quá độ.

#### 1.1 Chế độ xác lập

Chất lượng điều khiển được đánh giá qua sai lệch tĩnh (hay còn gọi là sai số xác lập)  
**Sai lệch tĩnh ( $S_t$ ) là sai lệch không đổi sau khi quá trình quá độ kết thúc.**

#### 1.2 Quá trình quá độ

Chất lượng của hệ thống được đánh giá qua 2 chỉ tiêu chính :

a) **Độ quá điều chỉnh lớn nhất  $\sigma_{max}$**  : là sai lệch cực đại trong quá trình quá độ so với giá trị xác lập, tính theo đơn vị phần trăm.

$$\sigma_{max} = \frac{y_{max} - y_{\infty}}{y_{\infty}} * 100\% \quad (4.1)$$

b) **Thời gian quá độ lớn nhất  $T_{max}$**  :

Về mặt lý thuyết, quá trình quá độ kết thúc khi  $t \rightarrow \infty$ . Trong điều khiển tự động, ta có thể xem quá trình quá độ kết thúc khi sai lệch của tín hiệu được điều khiển với giá trị xác lập của nó không vượt quá 5% (một số tài liệu chọn biên độ là  $\pm 2\%$ ). Khoảng thời gian đó gọi là  **$T_{max}$** .

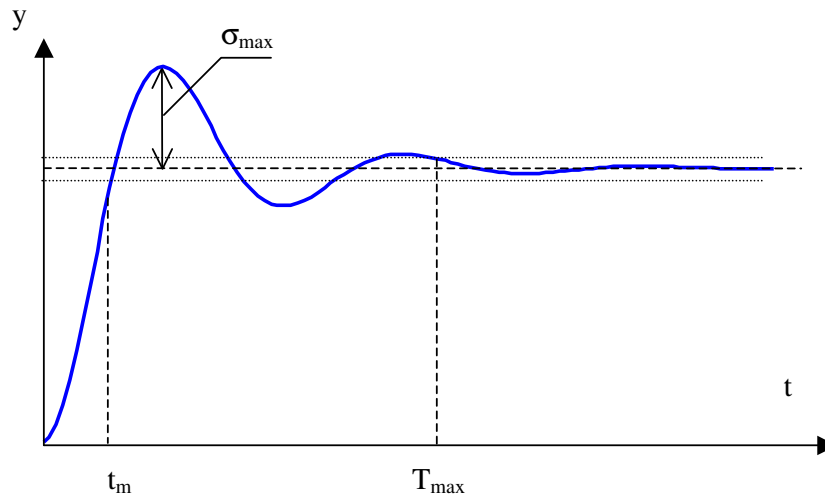
Thực tế điều khiển cho thấy : khi giảm  $\sigma_{max}$  thì  $T_{max}$  tăng và ngược lại.

Thông thường, qui định cho một hệ thống điều khiển :

$$\sigma_{max} = (20 \div 30)\%$$

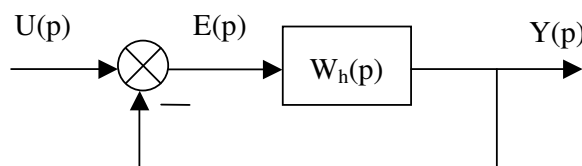
$$T_{max} = 2 \text{ đến } 3 \text{ chu kỳ dao động quanh giá trị xác lập}$$

c) **Thời gian tăng  $t_m$**  : là thời gian từ 0 đến lúc tín hiệu điều khiển đạt được 90% giá trị xác lập lần đầu tiên.



### 2 Đánh giá chất lượng ở chế độ xác lập

Xét một hệ thống kín phản hồi -1.



Theo định nghĩa, ta có :

$$S_t = \lim_{t \rightarrow \infty} e(t) = \lim_{p \rightarrow 0} pE(p)$$

Theo sơ đồ khối ở trên, ta có :  $E(p) = \frac{U(p)}{1+W_h(p)}$

$$\text{Vậy } S_t = \lim_{t \rightarrow \infty} e(t) = \lim_{p \rightarrow 0} p \frac{U(p)}{1+W_h(p)} \quad (4.2)$$

Trường hợp hệ thống kín bất kỳ, ta chuyển về hệ thống kín phản hồi  $-1$  tương đương và áp dụng công thức tính sai lệch tĩnh cho hệ tương đương này.

**Nhận xét :** sai lệch tĩnh  $S_t$  phụ thuộc

- Hàm truyền đạt của hệ hở
- Tín hiệu kích thích.

Hàm truyền đạt của hệ hở có dạng tổng quát như sau :

$$W_h(p) = \frac{K \frac{b_m p^m + \dots + b_1 p + 1}{a_n p^{n-v} + \dots + 1}}{p^v} = \frac{K}{p^v} W_0(p)$$

$v$  là bậc tích phân

### 2.1 Khi $u(t) = U_0 \cdot 1(t)$

$$U(p) = \frac{1}{p} \quad \Rightarrow \quad S_t = \lim_{p \rightarrow 0} \frac{1}{1 + \frac{K}{p^v} W_0(p)}$$

- Với  $v = 0$  :  $S_t = \frac{U_0}{1+K}$
- Với  $v = 1, 2, \dots$   $S_t = 0$

### 2.2 Khi $u(t) = U_0 \cdot t$

$$U(p) = \frac{U_0}{p^2} \quad \Rightarrow \quad S_t = \lim_{p \rightarrow 0} \frac{U_0}{p \left[ 1 + \frac{K}{p^v} W_0(p) \right]}$$

- Với  $v = 0$  :  $S_t = \infty$
- Với  $v = 1$  :  $S_t = \frac{U_0}{K}$
- Với  $v = 2, 3, \dots$   $S_t = 0$

## 3 Đánh giá chất lượng ở quá trình quá độ

Phải vẽ được đáp ứng quá độ  $y(t)$  của hệ thống

### 3.1 Phân tích thành các biểu thức đơn giản

Trong phương pháp này, tín hiệu ra  $Y(p)$  được phân tích thành tổng của các thành phần đơn giản. Sử dụng bảng tra Laplace hay hàm *ilaplace* trong MATLAB để tìm hàm gốc  $y(t)$ .

### 3.2 Phương pháp số Tustin

#### 3.2.1 Nội dung phương pháp

Số hóa tín hiệu liên tục thành tín hiệu gián đoạn để tìm đáp ứng thời gian, nghĩa là : chuyển hàm truyền đạt từ hệ liên tục sang hệ gián đoạn.

- Trong hệ gián đoạn, quan tâm đến  $y(kT)$
- Biến đổi toán học trong hệ gián đoạn là  $Y(z)$

- Đặc điểm :  $y(kT) \rightarrow Y(z)$   
 $y(k+m)T \rightarrow z^m Y(z)$

**Xác định mối liên hệ giữa hệ liên tục và hệ gián đoạn**

Xét một quan hệ giữa  $Y(p)$  và  $U(p)$  dưới dạng hàm truyền đạt :

$$W(p) = \frac{Y(p)}{U(p)} = \frac{1}{p} \tag{4.3}$$

Phương trình vi phân tương ứng là :

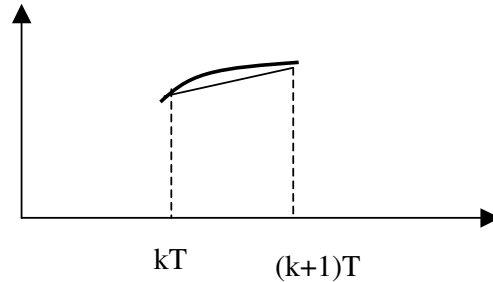
$$y(t) = \int_0^t u(t) dt \quad (\text{giả thiết các điều kiện}$$

đầu bằng 0)

Trên đường cong  $u(t)$ ,  $y(t)$  chính là diện tích xác định bởi đường cong  $u(t)$  với trục hoành.

Ta có :

$$y[(k+1)T] - y(kT) = \frac{T}{2} [u(k+1)T + u(kT)]$$



Chuyển phương trình sai phân ở trên sang toán tử Z, ta có :

$$(z-1)Y(z) = \frac{T}{2}(z+1)U(z)$$

$$\Rightarrow W(z) = \frac{Y(z)}{U(z)} = \frac{T}{2} \frac{z+1}{z-1} \tag{4.4}$$

Từ (4.3) và (4.4), ta có mối liên hệ :

$$\frac{1}{p} \leftrightarrow \frac{T}{2} \frac{z+1}{z-1} \quad \text{hay} \quad p \leftrightarrow \frac{2}{T} \frac{z-1}{z+1} \tag{4.5}$$

**3.2.2 Các bước tiến hành**

- Xác định tín hiệu  $Y(p)$  từ hàm truyền đạt  $W(p)$  và tín hiệu vào  $U(p)$
- Tìm  $Y(z)$  tương đương nhờ thay  $p = \frac{2}{T} \frac{z-1}{z+1}$  vào biểu thức của  $Y(p)$
- Biến đổi Z ngược để tìm  $y(kT)$

**Ví dụ :** Vẽ đặc tính thời gian của hệ thống có hàm truyền đạt :

$$W(p) = \frac{Y(p)}{U(p)} = \frac{10}{p^3 + 2p^2 + p + 1}$$

với  $u(t) = 1t$ .

*Giải :*

Chọn  $T = 1s$ , ta có :

$$p(p^3 + 2p^2 + p + 1)Y(p) = 10$$

$$\Rightarrow \frac{2}{T} \frac{z-1}{z+1} \left[ \left( \frac{2}{T} \frac{z-1}{z+1} \right)^3 + 2 \left( \frac{2}{T} \frac{z-1}{z+1} \right)^2 + \left( \frac{2}{T} \frac{z-1}{z+1} \right) + 1 \right] Y(z) = U(z)$$

Thay  $T = 1$ , ta có :

$$2(z-1) \left[ 8(z-1)^3 + 8(z-1)^2(z+1) + 2(z-1)(z+1)^2 + (z+1)^3 \right] Y(z) = (z+1)^4 U(z)$$

$$\Rightarrow (a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0) Y(z) = (b_4 z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0) U(z)$$

$$\Rightarrow a_4 y(k+4) = -a_3 y(k+3) - a_2 y(k+2) - a_1 y(k+1) - a_0 y(k) + (b_4 + b_3 + b_2 + b_1 + b_0)$$

Các hệ số  $a_i, b_j$  được xác định từ phương trình trên. Giả thiết biết trước các giá trị đầu  $y(0), y(1), y(2), y(3)$ , ta có thể tính lần lượt các giá trị còn lại của tín hiệu ra  $y(kT)$ .

### 3.3 Giải phương trình trạng thái

Nghiệm của phương trình trạng thái :

$$\dot{X} = AX + BU \tag{4.6}$$

$$Y = CX + DU$$

có dạng sau :

$$X(t) = e^{At} X(0) + \int_0^t e^{A(t-\tau)} BU(\tau) d\tau \tag{4.7}$$

$$Y(t) = C e^{At} X(0) + \int_0^t e^{A(t-\tau)} BU(\tau) d\tau + DU \tag{4.8}$$

Trong đó :

$$e^{At} = L^{-1} \{ (pI - A)^{-1} \}$$

**Ghi chú :**

$$A^{-1} = \frac{A_{adj}}{\det(A)} \text{ với } A_{adj} \text{ là ma trận có các phần tử } \tilde{a}_{ij} = (-1)^{i+j} \det(A_{ji}) \text{ trong đó } A_{ji} \text{ là ma}$$

trận có được bằng cách bỏ đi hàng thứ  $j$ , cột thứ  $i$ .

**Ví dụ :** Cho hệ thống được biểu diễn dưới dạng phương trình trạng thái :

$$\dot{X} = \begin{pmatrix} -2 & 1 \\ 0 & -1 \end{pmatrix} X + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$y = x_1$$

Tìm đáp ứng thời gian của hệ thống với  $u(t) = 1(t)$  với trạng thái ban đầu  $X = [0 \ 0]^T$ .

**Giải**

Tính  $e^{At}$

Ta có :

$$(pI - A)^{-1} = \begin{pmatrix} p+2 & -1 \\ 0 & p+1 \end{pmatrix}^{-1} = \frac{1}{(p+1)(p+2)} \begin{pmatrix} p+1 & 1 \\ 0 & p+2 \end{pmatrix} = \begin{pmatrix} \frac{1}{p+2} & \frac{1}{(p+1)(p+2)} \\ 0 & \frac{1}{p+1} \end{pmatrix}$$

$$\Rightarrow e^{At} = L^{-1} \{ (pI - A)^{-1} \} = \begin{pmatrix} e^{-2t} & e^{-t} - e^{-2t} \\ 0 & e^{-t} \end{pmatrix}$$

Theo công thức ở trên, ta có :

$$X(t) = \int_0^t \begin{pmatrix} e^{-2(t-\tau)} & e^{-(t-\tau)} - e^{-2(t-\tau)} \\ 0 & e^{-(t-\tau)} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} 1(\tau) d\tau = \begin{pmatrix} \frac{1}{2} - e^{-t} + \frac{e^{-2t}}{2} \\ 1 - e^{-t} \end{pmatrix}$$

$$y(t) = x_1 = \frac{1}{2} - e^{-t} + \frac{e^{-2t}}{2}$$

### 3.4 Sử dụng các hàm của MATLAB

- Hàm **step**: tìm hàm quá độ của một khâu
- Hàm **impulse**: tìm hàm trọng lượng của một khâu

Hàm **lsim**: phản ứng của khâu đối với tín hiệu vào bất kỳ.

Câu lệnh: **LSIM(sys,u,t)**

Với:

+ sys là tên của hàm truyền đạt đã được định nghĩa trước

+ u là vectơ tín hiệu vào

+ t là vectơ thời gian.

Ví dụ:

$t = 0:0.01:2*pi;$

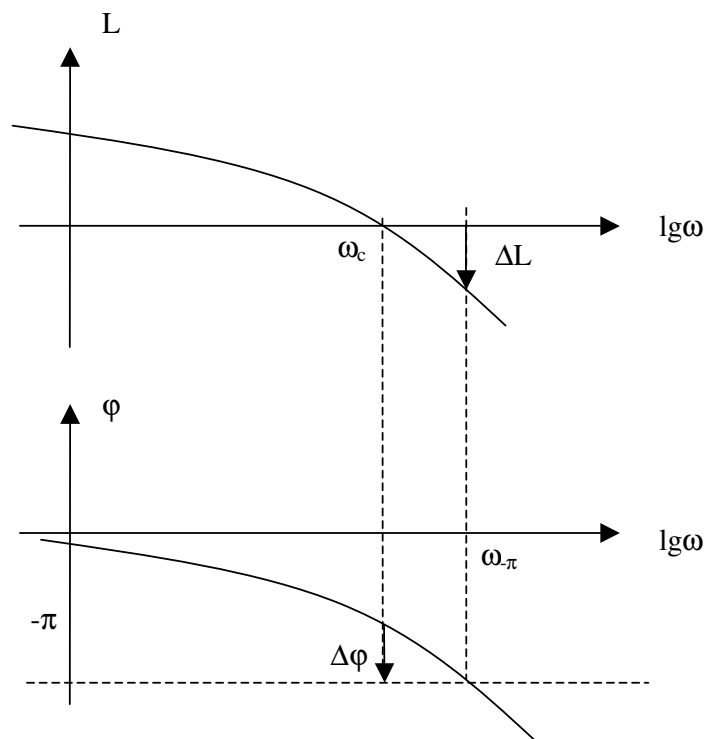
$u = \sin(t);$

$lsim(W1,u,t);$

## 4 Đánh giá thông qua độ dự trữ ổn định

### 4.1 Độ dự trữ biên độ

$$\Delta L = -L(\omega_{-\pi})$$



### 4.2 Độ dự trữ về pha

$$\Delta \varphi = 180 + \varphi(\omega_c)$$

Có thể xác định các độ dự trữ về biên độ, về pha bằng MATLAB

- MARGIN(SYS) : vẽ đặc tính tần số biên pha logarit + ghi các giá trị về độ dự trữ ổn định trên đặc tính
- [Gm,Pm]=MARGIN(SYS) : ghi các giá trị  $G_m = \Delta L$ ;  $P_m = \Delta \varphi$

\* Tính chất : Yêu cầu của quá trình điều khiển (tham khảo)

$$\Delta L = 6 \div 12 \text{ dB}$$

$$\Delta \varphi \approx 45^\circ$$

### 4.3 Mối liên hệ giữa các độ dự trữ và chất lượng điều khiển

- Khi tần số cắt  $\omega_c$  tăng :  $T_{max}$  giảm,  $t_m$  giảm.
- Khi tăng  $\Delta \varphi$  , độ quá điều lớn nhất  $\sigma_{max}$  giảm.

## 5 Tính điều khiển được và quan sát được của hệ thống

### 5.1 Điều khiển được

#### 5.1.1 Định nghĩa

Xét một hệ thống được mô tả toán học dưới dạng phương trình trạng thái :

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases}$$

Với  $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{r \times n}, D \in \mathbb{R}^{r \times m}$

Một hệ thống được gọi là **điều khiển được** nếu từ một vector ban đầu  $X_0$  bất kỳ, ta luôn có thể tìm được vector tín hiệu  $U_d$  để chuyển hệ thống từ trạng thái  $X_0$  đến trạng thái  $X_d$  mong muốn.

#### 5.1.2 Điều kiện

Xây dựng ma trận điều khiển

$$P = [B, AB, A^2B, \dots, A^{n-1}B]$$

**Điều kiện cần và đủ để một hệ thống mô tả toán học dưới dạng phương trình trạng thái điều khiển được là  $\text{rank}(P) = n$ .**

#### Nhận xét :

- Tính điều khiển được chỉ phụ thuộc vào các ma trận trạng thái A, B.
- Liên quan đến việc chọn các biến trạng thái

Ví dụ :

Cho hệ thống có mô tả toán học dưới dạng hàm truyền đạt như sau :

$$W(p) = \frac{20}{2p^2 + p + 4}$$

Giả sử đặt các biến trạng thái là :

$$x_1 = y$$

$$\dot{x}_1 = x_2$$

Xác định tính điều khiển được của hệ thống.

*Giải*

Ta có :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 0.5x_2 + 10u \end{aligned} \quad \text{hay} \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 10 \end{bmatrix} u$$

Ma trận P

$$P = [B, AB] = \begin{bmatrix} 0 & \begin{pmatrix} 0 & 1 \end{pmatrix} \\ 10 & \begin{pmatrix} -2 & -0.5 \end{pmatrix} \end{bmatrix} \begin{pmatrix} 0 \\ 10 \end{pmatrix} = \begin{bmatrix} 0 & 10 \\ 10 & -5 \end{bmatrix}$$

$$\det(P) = -100 \neq 0 \text{ nên } \text{rank}(P) = 2.$$

Vậy hệ thống với cách đặt biến trạng thái như trên là điều khiển được.

### 5.2 Tính quan sát được

#### 5.2.1 Định nghĩa

Một hệ thống được gọi là **quan sát được** nếu từ các vector  $U$  và  $Y$  đã có, ta có thể xác định được các biến trạng thái  $X$  của hệ thống.

#### 5.2.2 Điều kiện

Xây dựng ma trận quan sát

$$L = [C', A'C', (A')^2C', \dots, (A')^{n-1}C']$$

**Điều kiện cần và đủ để một hệ thống mô tả toán học dưới dạng phương trình trạng thái quan sát được là  $\text{rank}(L) = n$ .**

**Nhận xét :**

- Tính điều khiển được chỉ phụ thuộc vào các ma trận trạng thái A, C.

Ví dụ :

Xét trong ví dụ ở trên, ma trận trạng thái C sẽ là :

$$C = [1 \ 0]$$

Ma trận quan sát

$$L = [C' \ A'C'] = \begin{bmatrix} 1 & \begin{pmatrix} 0 & -2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 & \begin{pmatrix} 1 & -0.5 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Do  $\text{rank}(L) = 2$  nên hệ ở trên quan sát được.



## Chương 5

## NÂNG CAO CHẤT LƯỢNG VÀ TỔNG HỢP HỆ THỐNG

## 1 Khái niệm chung

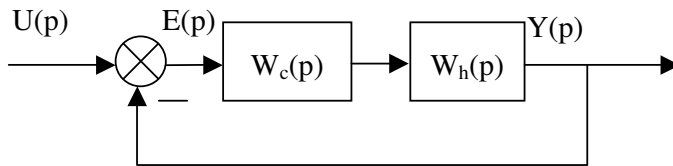
Trong một hệ thống điều khiển tự động, vai trò của bộ điều khiển C là :

- Ổn định hóa hệ thống
- Nâng cao chất lượng điều khiển.

## 2 Các bộ điều khiển – Hiệu chỉnh hệ thống

## 2.1 Khái niệm

- Có nhiều loại bộ điều khiển (khác nhau về cấu tạo, mô tả toán học, tác dụng điều khiển,...)
- Mục đích là nhằm thay đổi các giá trị về  $\Delta L$ ,  $\Delta \varphi$ , tần số cắt  $\rightarrow$  thay đổi chất lượng hệ thống



- Sau khi mắc bộ điều khiển, ta sẽ có :  
 $L' = L_c + L_h$   
 $\varphi' = \varphi_c + \varphi_h$

## 2.2 Bộ điều khiển tỉ lệ P

## 2.2.1 Hàm truyền đạt

$$W(p) = K$$

## 2.2.2 Đặc tính tần số logarit

$$L = 20 \lg K$$

$$\varphi = 0$$

**Nhận xét :**

- Tăng (giảm) biên độ trên toàn đặc tính
- Không làm thay đổi về pha.

## 2.2.3 Tác dụng điều khiển

## 2.3 Bộ bù sớm pha Lead

## 2.3.1 Hàm truyền đạt

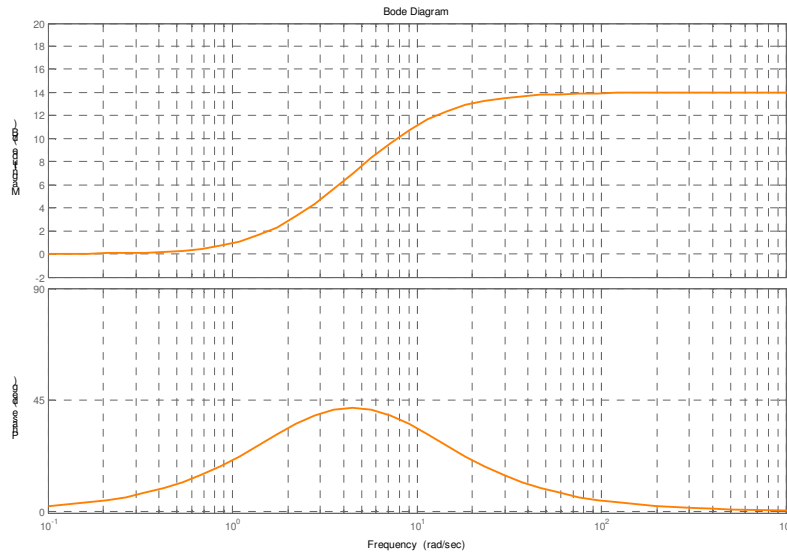
$$W(p) = K \frac{aTp + 1}{Tp + 1}, a > 1$$

## 2.3.2 Đặc tính tần số logarit

$$\varphi = \arctg(aT\omega) - \arctg(T\omega)$$

$$\omega_{max} = \frac{1}{T\sqrt{a}}$$

$$\sin \varphi_{max} = \frac{a-1}{a+1} > 0$$



Đặc tính logarit của bộ bù sớm pha ( $K=1, T=0.1, a = 5$ )

**Nhận xét :**

- Đặc tính biên độ làm tăng hệ số khuếch đại ở vùng tần số cao
- Gây ra sự vượt pha ở vùng tần số trung bình.

**2.3.3 Tác dụng hiệu chỉnh**

Tùy thuộc vào cách chọn hệ số khuếch đại  $K$ , các thông số  $a, T$  mà tác dụng hiệu chỉnh rất khác nhau. Nên tận dụng sự vượt pha ở tần số trung bình để làm tăng độ dự trữ về pha của hệ thống.

**2.4 Bộ bù trễ pha Leg**

**2.4.1 Hàm truyền đạt**

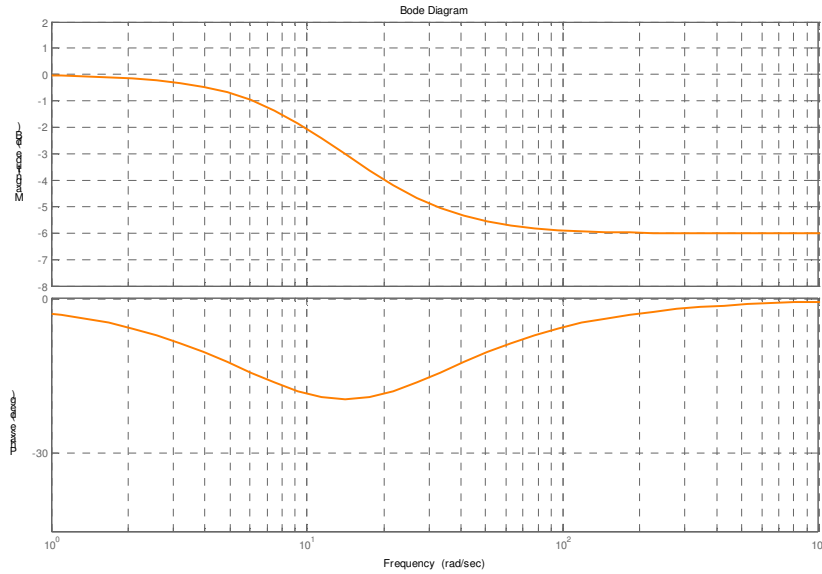
$$W(p) = K \frac{aTp + 1}{Tp + 1}, a < 1$$

**2.4.2 Đặc tính tần số logarit**

$$\varphi = \text{arctg}(aT\omega) - \text{arctg}(T\omega)$$

$$\omega_{max} = \frac{1}{T\sqrt{a}}$$

$$\sin \varphi_{max} = \frac{a-1}{a+1} < 0$$



Đặc tính logarit của bộ bù trễ pha ( $K=1, T=0.1, a = 0.5$ )

**Nhận xét :**

- Đặc tính biên độ làm giảm hệ số khuếch đại ở vùng tần số cao
- Gây ra sự chậm pha ở vùng tần số trung bình.

**2.4.3 Tác dụng hiệu chỉnh**

- Có thể tăng hệ số khuếch đại của hệ thống mà không ảnh hưởng đến tần số cắt.
- Tránh sự chậm pha do bộ điều khiển gây ra làm ảnh hưởng đến độ dự trữ về pha.

**2.5 Bộ bù trễ-sớm pha Leg -Lead**

**2.5.1 Hàm truyền đạt**

$$W(p) = K \left( \frac{a_1 T_1 p + 1}{T_1 p + 1} \right) \left( \frac{a_2 T_2 p + 1}{T_2 p + 1} \right)$$

$$a_1 < 1, a_2 > 1$$

**2.5.2 Đặc tính tần số logarit**

$$\omega_{max1} = \frac{1}{T_1 \sqrt{a_1}}; \sin \varphi_{max1} = \frac{a_1 - 1}{a_1 + 1} < 0$$

$$\omega_{max2} = \frac{1}{T_2 \sqrt{a_2}}; \sin \varphi_{max2} = \frac{a_2 - 1}{a_2 + 1} < 0$$

**Nhận xét :**

- Bộ bù leg-lead gồm 2 bộ bù nối tiếp.
- Để phát huy ưu điểm của bộ bù, phần trễ pha nên ở tần số thấp, phần sớm pha ở tần số trung bình hay tần số cao. Do đó điều kiện các thông số là :

$$\frac{1}{T_1 \sqrt{a_1}} < \frac{1}{T_2 \sqrt{a_2}} \Rightarrow \frac{T_1}{T_2} > \left( \frac{a_2}{a_1} \right)^2$$

### 2.5.3 Tác dụng hiệu chỉnh

- Chọn các thông số thích hợp sẽ làm tăng  $\Delta\varphi$
- Tăng hệ số khuếch đại của hệ thống.

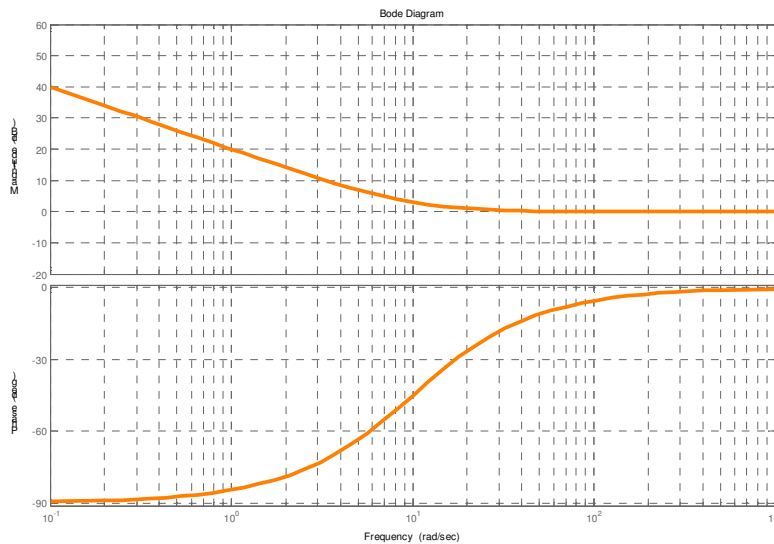
## 2.6 Bộ điều khiển PI (Proportional Integral Controller)

### 2.6.1 Hàm truyền đạt

$$W(p) = K \left( 1 + \frac{1}{T_i p} \right)$$

### 2.6.2 Đặc tính tần số logarit

$$\varphi = \arctg(T_i \omega) - \pi/2$$



Đặc tính logarit của bộ điều khiển PI ( $K=1$ ,  $T_i=0.1$ )

#### Nhận xét :

- Tăng 1 bậc tích phân
- Gây ra sự chậm pha ở vùng tần số thấp.

### 2.6.3 Tác dụng hiệu chỉnh

- Giảm bậc sai lệch tĩnh.
- Tác dụng hiệu chỉnh phụ thuộc rất lớn vào việc chọn thông số bộ điều khiển.

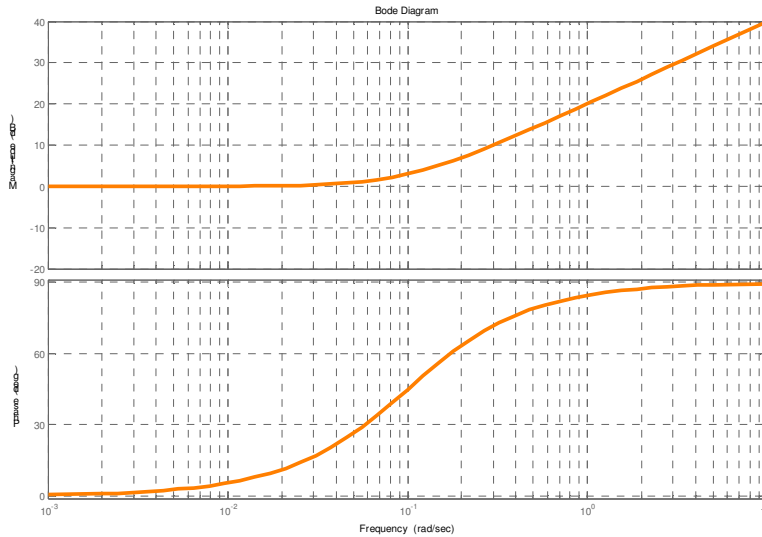
## 2.7 Bộ điều khiển PD (Proportional Derivative Controller)

### 2.7.1 Hàm truyền đạt

$$W(p) = K (1 + T_D p)$$

### 2.7.2 Đặc tính tần số logarit

$$\varphi = \arctg(T_D \omega)$$



Đặc tính logarit của bộ điều khiển PD (K=1, Td=10)

**Nhận xét :**

- Gây ra sự vượt pha ở vùng tần số cao.
- Tăng hệ số khuếch ở tần số cao

**2.7.3 Tác dụng hiệu chỉnh**

- Góp phần cải thiện  $\Delta\phi$ .
- Tăng mạnh hệ số khuếch đại tín hiệu ở tần số cao -> dễ bị ảnh hưởng của nhiễu.

**2.8 Bộ điều khiển PID (Proportional Integral Derivative Controller)**

**2.8.1 Hàm truyền đạt**

$$W(p) = K_p \left( 1 + \frac{1}{T_i p} + T_d p \right) = K_p + \frac{K_I}{p} + K_D p$$

Ta có :

$$W(p) = K_p \left( 1 + \frac{1}{T_i p} + T_d p \right) = \frac{K_p}{T_i p} (1 + T_i p + T_d T_i p^2) = \frac{K_I}{p} (1 + T_1 p)(1 + T_2 p)$$

với  $\begin{cases} T_1 T_2 = T_d T_i \\ T_1 + T_2 = T_i \end{cases} \quad K_I = K/T_i$

Giải hệ phương trình ở trên, ta được

$$\begin{cases} T_1 = \frac{T_i}{2} \left( 1 + \sqrt{1 - 4 \frac{T_d}{T_i}} \right) \\ T_2 = \frac{T_i}{2} \left( 1 - \sqrt{1 - 4 \frac{T_d}{T_i}} \right) \end{cases} \quad \text{nếu } T_i \geq 4T_d \text{ (giả thiết } T_1 > T_2)$$

Hay

$$W(p) = K T_1 \left( 1 + \frac{1}{T_1 p} \right) (1 + T_2 p) = W_{PI}(p) * W_{PD}(p)$$

**2.8.2 Đặc tính tần số logarit**

**Nhận xét :**

- Là sự kết hợp của bộ điều khiển PI và PD

2.8.3 Tác dụng hiệu chỉnh

- PI : giảm bậc sai lệch tĩnh
- PD : tăng  $\Delta\varphi$

**3 Tổng hợp hệ thống theo các tiêu chuẩn tối ưu**

**3.1 Phương pháp tối ưu modun**

- Khảo sát hệ kín phản hồi -1. Hàm truyền hệ kín là  $W_k(p) = \frac{W_c(p) * W_h(p)}{1 + W_c(p) * W_h(p)}$

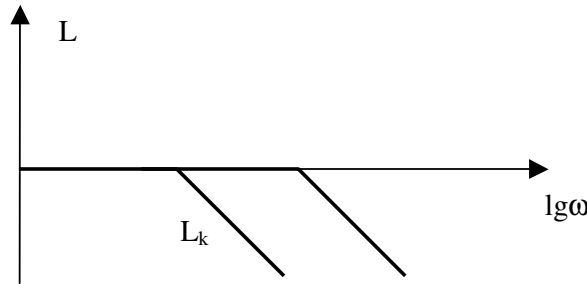
- Một trong những tiêu chuẩn để chọn bộ điều khiển  $W_c(p)$  là tín hiệu ra luôn bám theo tín hiệu vào, nghĩa là  $Y(p) = X(p)$  hay  $|W_k(p)| = 1, \forall \omega$ .

- Thực tế, việc đạt được tiêu chuẩn này là vô cùng khó khăn do : bản thân hệ thống có quán tính, dao động, trễ,... Tuy nhiên những hệ thống thực tế lại có một đặc điểm tự nhiên hợp lý là suy giảm mạnh ở tần số cao, nhờ vậy mà nó tồn tại với nhiều.

- Để thỏa thuận giữa yêu cầu lý tưởng và điều kiện thực tế, yêu cầu là tổng hợp hệ thống sao cho

$$|W_k'(j\omega)| \approx 1 \tag{*}$$

trong một dải tần số càng rộng càng tốt.



hay nói cách khác  $L_k = 20 \lg A_k \approx 0$ . Dải tần số làm  $L_k = 0$  càng lớn thì chất lượng hệ thống kín càng cao.

Phương pháp này hiện nay chỉ mới được áp dụng cho một số hệ hờ đặc biệt dưới đây. Trường hợp các hệ tổng quát, ta đưa về các hệ đặc biệt nhờ phương pháp gần đúng.

3.1.1 Hệ hờ là khâu quán tính bậc nhất

- Hệ hờ :  $W_h(p) = \frac{K}{Tp + 1}$

- Bộ điều khiển  $W_c(p) = \frac{K_p}{T_i p}$

- Hệ hờ với bộ điều khiển :  $W_h'(p) = \frac{K}{T_R(Tp + 1)}$  với  $T_R(p) = \frac{T_i}{K_p}$

- Hàm truyền hệ kín với bộ điều khiển

$$W_k'(p) = \frac{K}{T_R p (Tp + 1) + K}$$

$$\Rightarrow |W_k'(p)| = \frac{K}{\sqrt{(K - T_R T \omega^2)^2 + (\omega T_R)^2}}$$

Do đó  $|W_k'(p)|^2 = \frac{K^2}{K^2 + (T_R^2 - 2KT_R T)\omega^2 + T_R^2 T^2 \omega^4}$

Để điều kiện (\*) thỏa mãn trong dải tần số càng rộng càng tốt, ta có thể chọn  $T_R$  sao cho :

$$T_R^2 - 2KT_R T = 0 \Leftrightarrow T_R = \frac{T_i}{K_p} = 2KT$$

3.1.2 Hệ hở là khâu quán tính bậc 2

- Hệ hở : 
$$W_h(p) = \frac{K}{(1+T_1p)(1+T_2p)}$$

- Bộ điều khiển 
$$W_c(p) = K_p \left( 1 + \frac{1}{T_i p} \right)$$

- Trước tiên chọn  $T_1 = T_1$  để bù mẫu số  $(T_1p + 1)$ . Thực hiện tương tự phần còn lại, ta sẽ được :

$$T_R = \frac{T_i}{K_p} = 2KT_2 \Rightarrow K_p = \frac{T_1}{2KT_2}$$

3.1.3 Hệ hở là khâu quán tính bậc 3

- Hệ hở : 
$$W_h(p) = \frac{K}{(1+T_1p)(1+T_2p)(1+T_3p)}$$

- Bộ điều khiển 
$$W_c(p) = K_p \left( 1 + \frac{1}{T_i p} + T_d p \right) = \frac{(1+T_1'p)(1+T_2'p)}{T_R p}$$
 với  $T_R(p) = \frac{T_i}{K_p}$

trong đó : 
$$T_1' + T_2' = T_i$$

$$T_1' T_2' = T_i T_d$$

- Đầu tiên, ta chọn  $T_1' = T_1; T_2' = T_2$

Sau đó đơn giản các biểu thức và thực hiện như trên, ta được  $K_p = \frac{T_1 + T_2}{2KT_3}$ .

3.2 Phương pháp tối ưu đối xứng

- Nhược điểm của tổng hợp tối ưu modul ở trên là hệ hở phải ổn định, hàm quá độ h(t) có dạng tiếp xúc với trục hoành tại gốc 0.

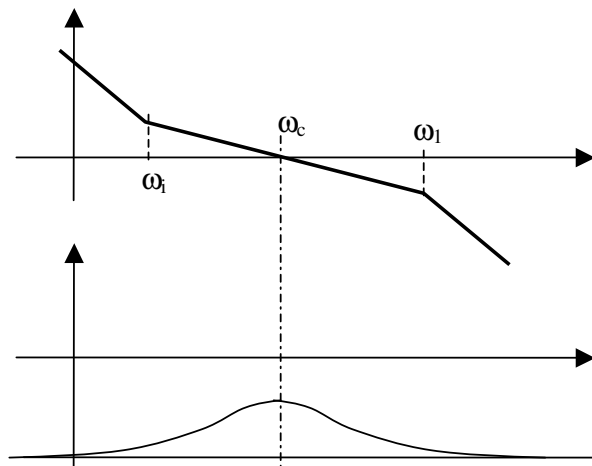
- Xét hệ kín phân hồi -1, ta có :

$$W_k' = \frac{W_h'}{1+W_h'} \Rightarrow W_h' = \frac{W_k'}{1-W_k'}$$

- Từ phương pháp tối ưu modul, thay vì để  $|W_k'(j\omega)| \approx 1$ , ta phải xác định bộ điều khiển sao cho

$$|W_h'(j\omega)| \gg 1 \tag{**}$$

- Đặc tính tần số logarit mong muốn là :



Đặc tính xây dựng có 3 phần

+ Tần số thấp : L cực lớn để sai lệch tĩnh bằng 0

+ Vùng tần số trung bình : liên quan trực tiếp đến chất lượng của hệ kín. Vùng này mang tính chất đối xứng

+ Vùng tần số cao : L cực bé để giảm ảnh hưởng của nhiễu.

- Để có được đặc tính mong muốn như trên, hệ hở với bộ điều khiển có đặc tính là :

$$W_h'(p) = \frac{K_h(1+T_i p)}{p^2(1+T_1 p)}$$

3.2.1 Đối tượng là khâu tích phân - quán tính bậc nhất

$$W_h(p) = \frac{K}{p(1+T_1 p)}$$

$$\Rightarrow W_c(p) = K_p \left( 1 + \frac{1}{T_i p} \right)$$

3.2.2 Đối tượng là khâu tích phân - quán tính bậc hai

$$W_h(p) = \frac{K}{p(1+T_1 p)(1+T_2 p)}$$

$$\Rightarrow W_c(p) = K_p \left( 1 + \frac{1}{T_i p} + T_d p \right)$$



# Chương 6

## HỆ THỐNG ĐIỀU KHIỂN GIÁN ĐOẠN (Hệ xung số)

### 1 Khái niệm chung

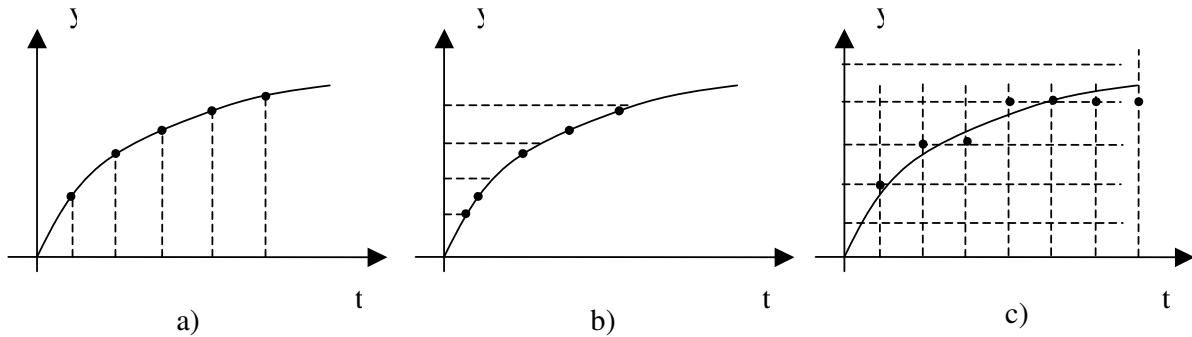
- Trong điều khiển, người ta phân thành 2 loại hệ thống : hệ liên tục và hệ không liên tục. Trong hệ không liên tục lại có 2 loại chính là : hệ gián đoạn (hệ xung số) và hệ thống với các sự kiện gián đoạn. Và đặc điểm của hệ gián đoạn là ta chỉ có thể quan sát các trạng thái của hệ thống một cách gián đoạn nhưng có chu kỳ (T).

- Nguyên nhân hình thành các hệ thống gián đoạn là :

- Sự hình thành của các bộ điều khiển số : linh hoạt, dễ dàng thay đổi và khống chế các thông số.
- Giám sát các tín hiệu bằng các thiết bị điện tử số.

- Quá trình biến đổi tín hiệu liên tục thành gián đoạn gọi là lượng tử hóa (trong kỹ thuật gọi là lấy mẫu). Có 3 hình thức lấy mẫu :

- Theo thời gian (a)
- Theo mức (b)
- Hỗn hợp (c)



### 2 Phép biến đổi Z

Để thuận tiện cho việc giải quyết các bài toán liên quan đến tín hiệu gián đoạn, người ta dùng phép biến đổi Z.

#### 2.1 Định nghĩa

Giả sử  $f(t)$  là hàm liên tục được lượng tử hóa bằng phương pháp thời gian với chu kỳ lấy mẫu T. Trong giải tích, hàm  $f(t)$  được viết như sau :

$$f^*(t) = \sum_{i=0}^{\infty} f(iT)\delta(t-iT) \tag{6.1}$$

Trong đó :

- $f^*(t)$  : là hàm liên tục đã được lấy mẫu (hàm được lượng tử hóa)
- $\delta(t-iT)$  là xung dirac tại thời điểm  $t - iT$

Biến đổi laplace của hàm  $f^*(t)$  như sau :

$$F^*(p) = \int_0^{\infty} f^*(t)e^{-pt} dt = \int_0^{\infty} \left[ \sum_{i=0}^{\infty} f(iT)\delta(t-iT) \right] e^{-pt} dt = \sum_{i=0}^{\infty} \int_0^{\infty} f(iT)\delta(t-iT)e^{-pt} dt$$

$$\Rightarrow F^*(p) = \sum_{i=0}^{\infty} f(iT)e^{-ipT} \tag{6.2}$$

Đặt  $z = e^{pT}$  (6.3)

Từ ( 6.2) và (6.3), ta có :

$$F(z) = \sum_{i=0}^{\infty} f(iT)z^{-i} \tag{6.4}$$

F(z) được gọi là biến đổi Z của hàm gián đoạn f(iT). Ký hiệu là :

$$F(z) = \mathbf{Z}\{f(iT)\}$$

Hay  $f(iT) = \mathbf{Z}^{-1}\{F(z)\}$

**Nhận xét :**

- Biến đổi Z là dạng biến đổi laplace.
- Chỉ có biến đổi Z của hàm gián đoạn chứ không có biến đổi Z của hàm liên tục.

**Ví dụ :** Cho hàm f(t) = e<sup>-at</sup>. Tìm biến đổi Z của hàm f(iT).

*Giải*

Ta có f(t) = e<sup>-at</sup> nên f(iT) = e<sup>-aiT</sup>.

Theo định nghĩa

$$F(z) = \sum_{i=0}^{\infty} f(iT)z^{-i} = 1 + e^{-aT}z^{-1} + e^{-2aT}z^{-2} + \dots$$

$$F(z) = \frac{1}{1 - e^{-aT}z^{-1}} = \frac{z}{z - e^{-aT}}$$

với điều kiện e<sup>-aT</sup>z<sup>-1</sup> < 1.

Một số sách để đơn giản trong cách viết, người ta bỏ thời gian lấy mẫu T, nghĩa là:

$$F(z) = \mathbf{Z}\{f(i)\} = \frac{z}{z - e^{-a}}$$

**2.2 Một số tính chất của biến đổi Z**

- Tính tuyến tính

$$\mathbf{Z}\{af_1(iT) + bf_2(iT)\} = aF_1(z) + bF_2(z)$$

- Tính dịch chuyển hàm gốc

$$\mathbf{Z}\{f(i+1)T\} = zF(z) - zf(0)$$

$$\mathbf{Z}\{f(i+m)T\} = z^m F(z) - \sum_{j=0}^{m-1} f(j)z^{m-j}$$

Nếu tất cả các điều kiện đầu bằng 0 thì

$$\mathbf{Z}\{f(i+m)T\} = z^m F(z)$$

- Giá trị đầu của hàm gốc

$$f(0) = \lim_{z \rightarrow \infty} F(z)$$

- Giá trị cuối của hàm gốc

$$f_{\infty} = \lim_{z \rightarrow 1} (z-1)F(z)$$

**2.3 Biến đổi Z ngược**

**2.3.1 Tra bảng**

Phân tích hàm F(z) thành các thành phần đơn giản và thực hiện tra bảng.

**2.3.2 Phương pháp chuỗi lũy thừa**

Theo định nghĩa, ta có:

$$F(z) = \sum_{i=0}^{\infty} f(iT)z^{-i} = f(0) + f(T)z^{-1} + f(2T)z^{-2} + \dots$$

Do đó nếu có thể phân tích hàm F(z) thành chuỗi lũy thừa có chứa các thành phần z<sup>-i</sup>, ta có thể biết được f(iT).

**Ví dụ :**

$$F(z) = \frac{z}{z^2 - 3z + 2}$$

Phân tích hàm  $F(z)$  ở trên ta được :

$$F(z) = z^{-1} + 3z^{-2} + 7z^{-3} + 15z^{-4} + \dots$$

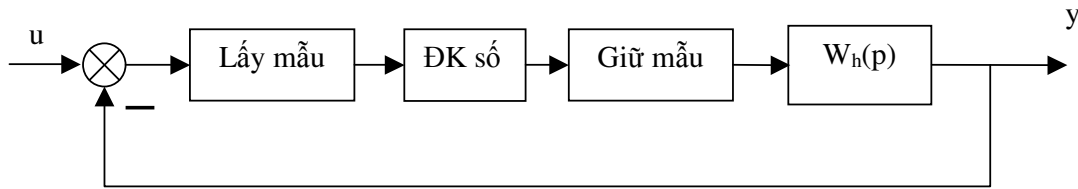
Vậy  $f(iT) = 2^i - 1$ .

### 3 Lấy mẫu và giữ mẫu

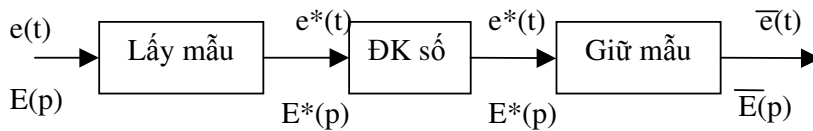
#### 3.1 Khái niệm

Để có thể đưa bộ điều khiển số vào hệ thống, cần có quá trình lấy mẫu và giữ mẫu.

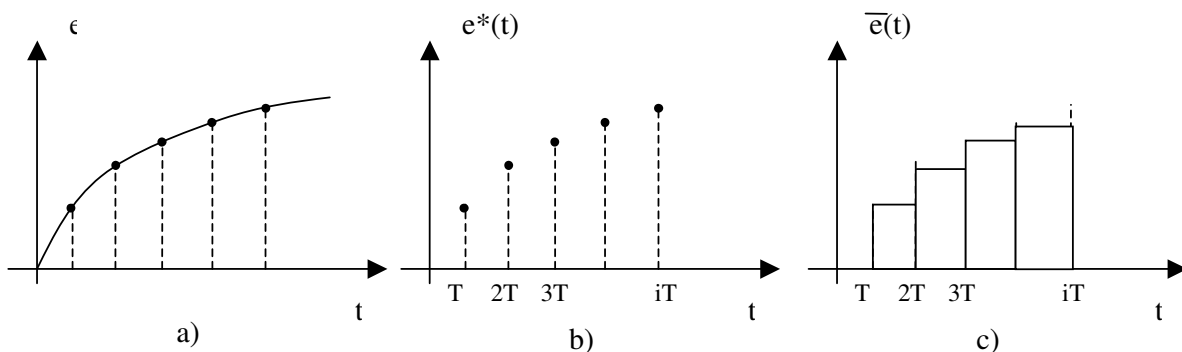
- Lấy mẫu là chuyển tín hiệu liên tục thành tín hiệu gián đoạn.
- Giữ mẫu là quá trình chuyển tín hiệu gián đoạn thành tín hiệu liên tục.



Khảo sát một quá trình lấy mẫu và giữ mẫu đơn giản như hình vẽ sau, trong đó tín hiệu gián đoạn không qua bất kỳ một khâu biến đổi nào.



Đặc điểm thời gian của các tín hiệu trên như sau :



**Nhận xét :**

$\bar{e}(t)$  là tín hiệu liên tục từng đoạn. Sau quá trình biến đổi (lấy mẫu và giữ mẫu),  $\bar{e}(t)$  khác với  $e(t)$  ban đầu. Khi tần số lấy mẫu lớn càng lớn ( $T$  bé) thì  $\bar{e}(t)$  càng gần giống dạng của  $e(t)$ .

#### 3.2 Lấy mẫu

Phương trình của tín hiệu  $e^*(t)$  sau khi được lấy mẫu là :

$$e^*(t) = \sum_{i=0}^{\infty} e(iT)\delta(t-iT) \tag{6.5}$$

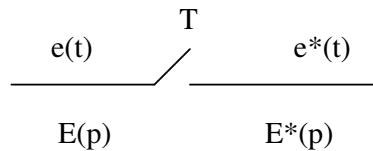
Do đó :

$$E^*(p) = \sum_{i=0}^{\infty} e(iT)e^{-ipT} \tag{6.6}$$

### 3.2.1 Định nghĩa

Một bộ lấy mẫu được gọi là lý tưởng nếu sau khi lấy mẫu, ảnh laplace của tín hiệu lấy mẫu có biểu thức như trong 6.6.

Sơ đồ thay thế của bộ lấy mẫu lý tưởng như sau :



Nếu biết ảnh laplace của tín hiệu cần lấy mẫu E(p), ta có thể tìm được ảnh laplace của tín hiệu đã được lấy mẫu lý tưởng theo biểu thức sau :

$$E^*(p) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E\left(p + jn \frac{2\pi}{T}\right) + \frac{e(0)}{2} \tag{6.7}$$

**Ghi chú :** có khả năng nhiều tín hiệu khác nhau sau khi được lấy mẫu sẽ có phương trình toán học như nhau.

### 3.2.2 Định lý lấy mẫu (định lý Shannon)

Một tín hiệu liên tục theo thời gian e(t) chỉ có thể phục hồi sau quá trình lấy mẫu nếu thỏa mãn điều kiện :

$$f \geq 2f_{max} \tag{6.8}$$

Trong đó :

- f là tần số lấy mẫu (f = 1/T)
- f<sub>max</sub> là tần số cực đại của tín hiệu cần lấy mẫu

### 3.2.3 Tính chất của tín hiệu E\*(p)

#### Tính chất 1

Hàm E\*(p) tuần hoàn trong mặt phẳng p với chu kỳ jω<sub>p</sub> trong đó ω<sub>p</sub> =  $\frac{2\pi}{T}$  (T là chu kỳ lấy mẫu)

#### Tính chất 2

Nếu E(p) có một cực tại p = p<sub>1</sub> thì E\*(p) phải có cực tại p = p<sub>1</sub> + jω<sub>p</sub> với m = 0, ±1, ±2,...

## 3.3 Giữ mẫu

### 3.3.1 Bộ giữ mẫu bậc 0

Đặc điểm của bộ giữ mẫu bậc 0 là tín hiệu được giữ mẫu không đổi giữa 2 lần lấy mẫu và bằng giá trị của lần giữ mẫu trước đó (xem hình vẽ trên)

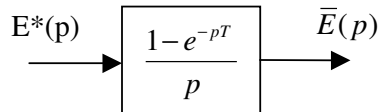
$$\bar{e}(t) = e(0)[1(t) - 1(t-T)] + e(T)[1(t-T) - 1(t-2T)] + \dots$$

$$\begin{aligned} \bar{E}(p) &= e(0) \left[ \frac{1}{p} - \frac{1}{p} e^{-pT} \right] + e(T) \left[ \frac{1}{p} e^{-pT} - \frac{1}{p} e^{-2pT} \right] + \dots \\ \Rightarrow &= \left[ \frac{1 - e^{-pT}}{p} \right] \left[ e(0) + e(T)e^{-pT} + e(2T)e^{-2pT} + \dots \right] \\ &= \left[ \frac{1 - e^{-pT}}{p} \right] \sum_{i=0}^{\infty} e(iT)e^{-ipT} \end{aligned}$$

Kết hợp với 6.6, ta được

$$\bar{E}(p) = \left[ \frac{1 - e^{-pT}}{p} \right] E^*(p) \tag{6.8}$$

Như vậy, mô tả toán học của bộ giữ mẫu bậc 0 (Zero Order Hold) là :



Hàm truyền đạt của bộ giữ mẫu bậc 0 là :

$$W_{ZOH}(p) = \frac{1 - e^{-pT}}{p} \tag{6.9}$$

### 3.3.2 Bộ giữ mẫu bậc 1

Tín hiệu giữ mẫu giữa 2 lần lấy mẫu liên tiếp  $nT$  và  $(n+1)T$  là

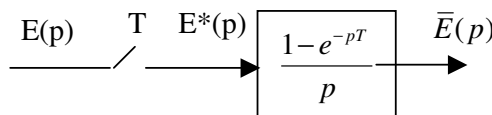
$$e_n(t) = e(nT) + e'(nT)(t - nT), \quad nT \leq t < (n+1)T$$

với 
$$e'(nT) = \frac{e(nT) - e[(n-1)T]}{T}$$

Chứng minh tương tự, ta tìm được hàm truyền đạt của bộ giữ mẫu bậc nhất (First Order Hold) là :

$$W_{FOH}(p) = \left( \frac{1 + pT}{T} \right) \left( \frac{1 - e^{-pT}}{p} \right)^2$$

Như vậy, sơ đồ thay thế của bộ lấy mẫu và giữ mẫu là :



**Chú ý :** Bộ lấy mẫu và giữ mẫu trong sơ đồ trên không thể là mô hình toán học cho một thiết bị cụ thể nào trong thực tế. Tuy nhiên, sự kết hợp giữa bộ lấy mẫu và giữ mẫu lại là mô hình chính xác của bộ chuyển đổi ADC và DAC.

## 4 Hàm truyền đạt hệ gián đoạn

### Định nghĩa

Hàm truyền đạt hệ gián đoạn, ký hiệu là  $W(z)$ , là tỉ số giữa tín hiệu ra với tín hiệu vào dưới dạng toán tử  $z$ .

$$W(z) = \frac{Y(z)}{U(z)} \tag{6.10}$$

### 4.1 Xác định hàm truyền đạt $W(z)$ từ hàm truyền đạt hệ liên tục

#### 4.1.1 Mối liên hệ giữa $E^*(p)$ và $E(z)$

Theo công thức (6.6), ta có ảnh laplace của tín hiệu liên tục  $e(t)$  sau khi được lượng tử hóa là :

$$E^*(p) = \sum_{i=0}^{\infty} e(iT)e^{-ipT}$$

Cũng tín hiệu liên tục e(t), sau khi được lượng tử hóa và thực hiện biến đổi Z, theo công thức (6.4), ta có :

$$E(z) = \sum_{i=0}^{\infty} e(iT)z^{-i}$$

Từ 2 công thức ở trên, có thể thấy rằng :

$$E(z) = E^*(p) \Big|_{e^{pT}=z} \tag{6.11}$$

$$E^*(p) = E(z) \Big|_{z=e^{pT}} \tag{6.12}$$

Ví dụ : Cho một tín hiệu liên tục có ảnh laplace là :

$$E(p) = \frac{1}{(p+1)(p+2)}$$

Tìm ảnh E\*(p) và E(z).

*Giải*

Ta có:

$$E(p) = \frac{1}{(p+1)} - \frac{1}{(p+2)}$$

Tra bảng có sẵn, ta có :

$$E(z) = \frac{z}{(z-e^{-T})} - \frac{z}{(z-e^{-2T})} = \frac{z(e^{-T} - e^{-2T})}{(z-e^{-T})(z-e^{-2T})}$$

$$\Rightarrow E^*(p) = \frac{e^{pT}(e^{-T} - e^{-2T})}{(e^{pT} - e^{-T})(e^{pT} - e^{-2T})}$$

**Chú ý :** chúng ta sẽ dùng ký hiệu sau để biểu diễn ảnh laplace của tín hiệu được lượng tử hóa

$$E^*(p) = \{E(p)\}^* \tag{6.13}$$

**Tính chất của phép biến đổi \*(p)**

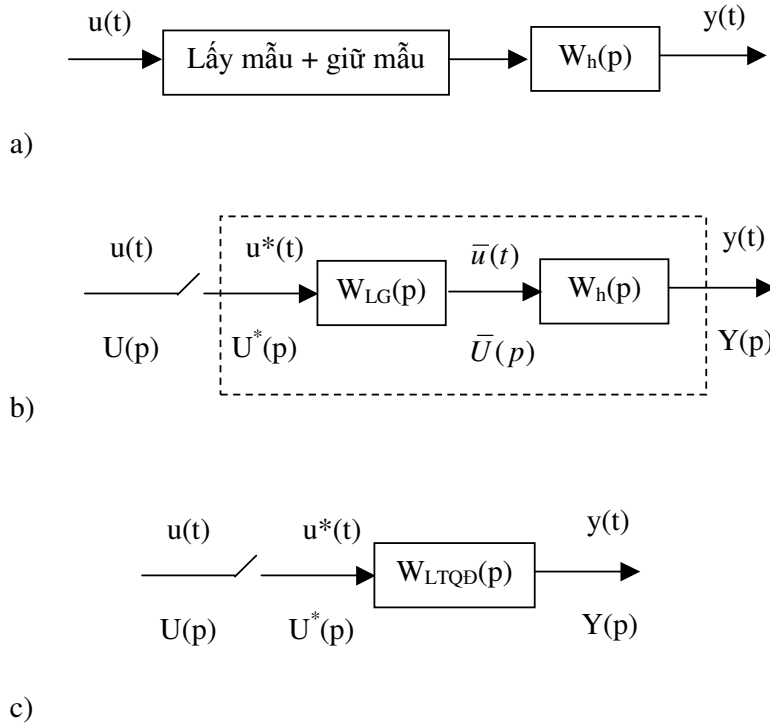
Nếu ta có quan hệ

$$F(p) = H(p).E^*(p) \tag{6.14}$$

thì  $F^*(p) = H^*(p).E^*(p) \tag{6.15}$

**4.1.2 Hàm truyền đạt hệ hở**

Xét một hệ hở gián đoạn có sơ đồ khối như hình vẽ



Hàm truyền đạt phân liên tục quy đổi là :

$$W_{LTQD}(p) = W_{LG}(p)W_h(p)$$

Tín hiệu ra là :

$$Y(p) = W_{LTQD}(p)U^*(p) = W_{LG}(p)W_h(p)U^*(p)$$

Thực hiện biến đổi \*(p) 2 vế phương trình trên, ta được

$$Y^*(p) = \{W_{LG}(p)W_h(p)\}^* U^*(p)$$

Biết rằng biến đổi \*(p) và biến đổi Z là tương đương, do đó :

$$Y(z) = Z\{W_{LG}(p)W_h(p)\}U(z)$$

Hàm truyền đạt hệ gián đoạn hờ vì vậy được tính :

$$W_h(z) = \frac{Y(z)}{U(z)} = Z\{W_{LG}(p)W_h(p)\} \tag{6.16}$$

Trường hợp bộ giữ mẫu là bậc 0,  $W_{LG}(p) = \frac{1-e^{-pT}}{p}$ , ta có :

$$W_h(z) = \frac{Y(z)}{U(z)} = Z\left\{\frac{1-e^{-pT}}{p}W_h(p)\right\} = \frac{z-1}{z}Z\left\{\frac{W_h(p)}{p}\right\} \tag{6.17}$$

Ví dụ : Tìm hàm truyền đạt hệ gián đoạn hờ biết  $W_h(p) = \frac{1}{p+1}$  và bộ giữ mẫu là bậc 0. Giả sử tín

hiệu vào là  $u(t) = 1(t)$ . Tìm phương trình của tín hiệu ra.

*Giải*

Áp dụng công thức trên, ta có :

$$W_h(z) = \frac{z-1}{z}Z\left\{\frac{1}{p(p+1)}\right\} = \frac{1-e^{-T}}{z-e^{-T}}$$

$$u(t) = 1(t) \Rightarrow U(z) = \frac{z}{z-1}$$

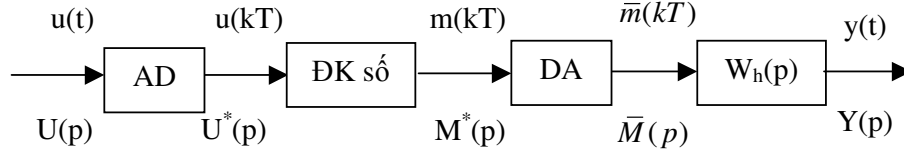
$$Y(z) = W_h(z)U(z) = \frac{z(1-e^{-T})}{(z-1)(z-e^{-T})} = \frac{z}{z-1} - \frac{e^{-T}}{(z-e^{-T})}$$

Biến đổi  $Z^{-1}$ , ta được  
 $y(iT) = 1 - e^{-iT}$

**Chú ý :** Với hệ thống gián đoạn, ta chỉ có thể biết được giá trị của tín hiệu ở ngõ ra tại những thời điểm lấy mẫu. Ở giữa các khoảng lấy mẫu, ta không thể biết được giá trị chính xác của tín hiệu.

### 4.1.3 Hệ hở có bộ điều khiển số

Xét hệ hở có bộ điều khiển số như sau :



Trong đó bộ điều khiển số có hàm truyền là :

$$W_c(z) = \frac{M(z)}{U(z)} \text{ hay } M(z) = W_c(z)U(z)$$

Ta có :

$$Y(p) = W_h(p) \cdot \bar{M}(p) = W_h(p) \cdot W_{LG}(p) M^*(p)$$

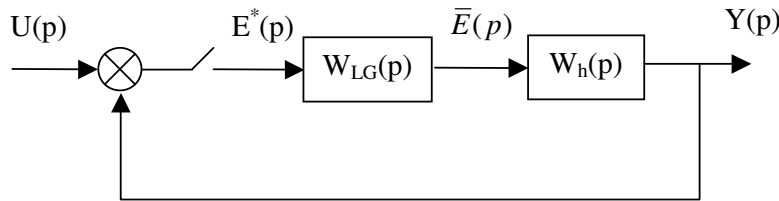
$$\Rightarrow Y^*(p) = \{W_h(p) \cdot W_{LG}(p)\}^* \cdot M^*(p) = \{W_h(p) \cdot W_{LG}(p)\}^* \cdot W_c^*(p) \cdot U^*(p)$$

$$\Rightarrow Y(z) = Z\{W_h(p) \cdot W_{LG}(p)\} \cdot W_c(z) \cdot U(z)$$

$$\Rightarrow W(z) = \frac{Y(z)}{U(z)} = Z\{W_h(p) \cdot W_{LG}(p)\} \cdot W_c(z)$$

### 4.1.4 Hệ kín

Xét hệ kín gián đoạn có sơ đồ khối như sau :



Ta có :

$$Y(p) = W_h(p) \cdot \bar{E}(p) = W_h(p) \cdot W_{LG}(p) \cdot E^*(p) = W_{LTQD}(p) \cdot E^*(p)$$

$$\Rightarrow Y^*(p) = \{W_{LTQD}(p)\}^* \cdot E^*(p)$$

Mặt khác :

$$E(p) = U(p) - Y(p) \Rightarrow E^*(p) = U^*(p) - Y^*(p)$$

$$\Rightarrow Y^*(p) = \{W_{LTQD}(p)\}^* [U^*(p) - Y^*(p)]$$

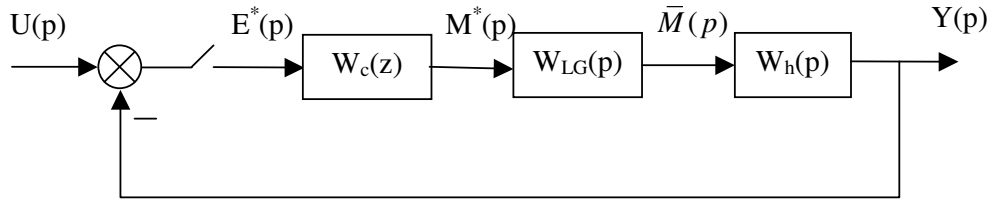
$$\Rightarrow Y^*(p) = \frac{\{W_{LTQD}(p)\}^*}{1 + \{W_{LTQD}(p)\}^*} U^*(p)$$

hay 
$$Y(z) = \frac{Z\{W_{LTQD}(p)\}}{1 + Z\{W_{LTQD}(p)\}} U(z)$$



$$\Rightarrow W_k(z) = \frac{W_h(z)}{1 + W_h(z)}$$

4.1.5 Hệ kín có bộ điều khiển số

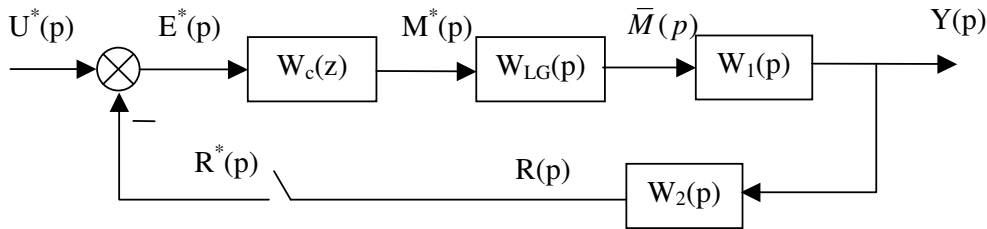
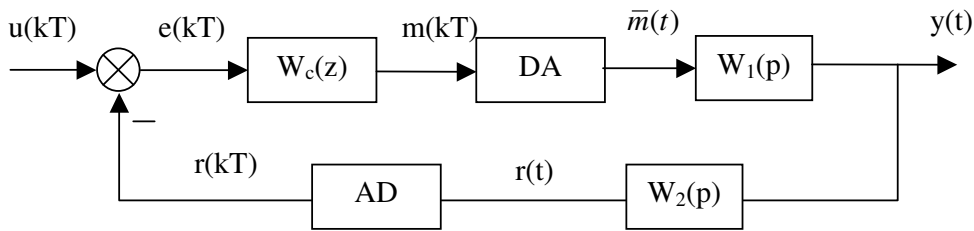


Chứng minh tương tự, ta được :

$$W_k(z) = \frac{W_h(z)W_c(z)}{1 + W_h(z).W_c(z)} \quad \text{với} \quad W_h(z) = \frac{Y(z)}{U(z)} = Z\{W_{LG}(p)W_h(p)\}$$

4.1.6 Hệ gián đoạn điều khiển từ máy tính

Sơ đồ khối của hệ thống như sau :



Ta có :

$$Y(p) = W_1(p)\bar{M}(p) = W_{LG}(p).W_1(p).M^*(p)$$

$$\Rightarrow Y^*(p) = \{W_{LG}(p).W_1(p)\}^* .M^*(p) \text{ hay } Y(z) = Z\{W_{LG}(p).W_1(p)\}.M(z)$$

Theo sơ đồ thì :

$$M^*(p) = W_c^*(p)E^*(p) = W_c^*(p)[U^*(p) - R^*(p)]$$

hay  $M(z) = W_c(z)[U(z) - R(z)]$

Ngoài ra do :

$$R(p) = W_2(p).Y(p) = W_{LG}(p).W_1(p).W_2(p).M^*(p)$$

nên  $R(z) = Z\{W_{LG}(p).W_1(p).W_2(p)\}M(z)$

Suy ra  $M(z) = W_c(z)[U(z) - Z\{W_{LG}(p).W_1(p).W_2(p)\}M(z)]$

Hay  $M(z) = \frac{W_c(z).U(z)}{1 + W_c(z).Z\{W_{LG}(p).W_1(p).W_2(p)\}}$

Thay vào công thức của Y(z), ta được :

$$Y(z) = \frac{W_c(z) \cdot Z\{W_{LG}(p) \cdot W_1(p)\}}{1 + W_c(z) \cdot Z\{W_{LG}(p) \cdot W_1(p) \cdot W_2(p)\}} U(z)$$

Hay  $W(z) = \frac{Y(z)}{U(z)} = \frac{W_c(z) \cdot Z\{W_{LG}(p) \cdot W_1(p)\}}{1 + W_c(z) \cdot Z\{W_{LG}(p) \cdot W_1(p) \cdot W_2(p)\}}$

**Ví dụ :**

Cho hệ điều khiển gián đoạn kín phản hồi -1 trong đó  $W_c(z) = \frac{2z-1}{z}$  và  $W_p(z) = \frac{1}{p+1}$ . Tìm hàm truyền đạt của hệ thống.

**4.2 Xác định hàm truyền đạt từ phương trình sai phân**

Một hệ thống gián đoạn có thể được cho dưới dạng phương trình sai phân tổng quát như sau :

$$a_n y[(i+n)T] + \dots + a_1 y[(i+1)T] + a_0 y(iT) = b_m u[(i+m)T] + \dots + b_1 u[(i+1)T] + b_0 u(iT)$$

Giả sử các điều kiện đầu bằng 0. Thực hiện biến đổi Z cho cả 2 vế phương trình trên, ta được :

$$(a_n z^n + \dots + a_1 z + a_0) Y(z) = (b_m z^m + \dots + b_1 z + b_0) U(z)$$

Vậy hàm truyền đạt là :

$$W(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^m + \dots + b_1 z + b_0}{a_n z^n + \dots + a_1 z + a_0}$$

**5 Tính ổn định của hệ gián đoạn**

**5.1 Mối liên hệ giữa mặt phẳng p và mặt phẳng z**

- Nhắc lại :  $z = e^{pT}$

- Ứng với một điểm  $p = \alpha + j\beta$  trong mặt phẳng p sẽ có điểm  $z = e^{(\alpha + j\beta)T} = e^{\alpha T} (\cos \beta T + j \sin \beta T)$  trong mặt phẳng z.

- Do  $|z| = e^{\alpha T}$  nên khi  $\alpha < 0$  thì  $|z| < 1$

**Kết luận :** Một hệ thống gián đoạn ổn định khi và chỉ khi tất cả các nghiệm của phương trình đặc tính có môđun nhỏ hơn 1.

**5.2 Phép biến đổi tương đương**

Để có thể sử dụng các tiêu chuẩn ổn định trong mặt phẳng p, người ta sử dụng phép biến đổi tương đương như sau :

$$z = \frac{v+1}{v-1}$$

- Với  $|z| < 1 \Rightarrow \text{real}(v) < 0$

- Với  $|z| = 1 \Rightarrow \text{real}(v) = 0$

- Với  $|z| > 1 \Rightarrow \text{real}(v) > 0$

Sau khi chuyển sang mặt phẳng v, ta có thể sử dụng các tiêu chuẩn ổn định của hệ tuyến tính để xét tính ổn định của hệ liên tục tương đương.

## Phụ lục

**CONTROL SYSTEM TOOLBOX & SIMULINK TRONG MATLAB***Ứng dụng để phân tích, thiết kế và mô phỏng các hệ thống tuyến tính***GIỚI THIỆU**

MATLAB, tên viết tắt của từ tiếng Anh **MAT**rix **LAB**oratory, là một môi trường mạnh dành cho các tính toán khoa học. Nó tích hợp các phép tính ma trận và phân tích số dựa trên các hàm cơ bản. Hơn nữa, cấu trúc đồ họa hướng đối tượng của Matlab cho phép tạo ra các hình vẽ chất lượng cao. Ngày nay, Matlab trở thành một ngôn ngữ « chuẩn » được sử dụng rộng rãi trong nhiều ngành và nhiều quốc gia trên thế giới.

Về mặt cấu trúc, Matlab gồm một cửa sổ chính và rất nhiều hàm viết sẵn khác nhau. Các hàm trên cùng lĩnh vực ứng dụng được xếp chung vào một thư viện, điều này giúp người sử dụng dễ dàng tìm được hàm cần quan tâm. Có thể kể ra một số thư viện trong Matlab như sau :

- Control System (dành cho điều khiển tự động)
- Financial Toolbox (lĩnh vực kinh tế)
- Fuzzy Logic (điều khiển mờ)
- Signal Processing (xử lý tín hiệu)
- Statistics (toán học và thống kê)
- Symbolic (tính toán theo biểu thức)
- System Identification (nhận dạng)
- ...

Một tính chất rất mạnh của Matlab là nó có thể liên kết với các ngôn ngữ khác. Matlab có thể gọi các hàm viết bằng ngôn ngữ Fortran, C hay C++, và ngược lại các hàm viết trong Matlab có thể được gọi từ các ngôn ngữ này...

Các bạn có thể xem phần Help trong Matlab để tham khảo cách sử dụng và ví dụ của từng lệnh, hoặc download (miễn phí) các file help dạng \*.pdf tại trang Web của Matlab ở địa chỉ <http://www.mathworks.com>

**1 Control System Toolbox**

Control System Toolbox là một thư viện của Matlab dùng trong lĩnh vực điều khiển tự động. Cùng với các lệnh của Matlab, tập lệnh của Control System Toolbox sẽ giúp ta thiết kế, phân tích và đánh giá các chỉ tiêu chất lượng của một hệ thống tuyến tính.

**1.1 Định nghĩa một hệ thống tuyến tính****1.1.1 Định nghĩa bằng hàm truyền****Hệ thống một tín hiệu vào/ra**

Câu lệnh: `sys=tf(num,den,T)`

- *num*: vectơ chứa các hệ số của đa thức ở tử số, bậc từ cao đến thấp theo toán tử Laplace (hệ liên tục) hoặc theo toán tử *z* (hệ gián đoạn)
- *den*: vectơ chứa các hệ số của đa thức ở mẫu số, bậc từ cao đến thấp
- *T*: chu kỳ lấy mẫu, chỉ dùng cho hệ gián đoạn (tính bằng *s*)

Ví dụ:

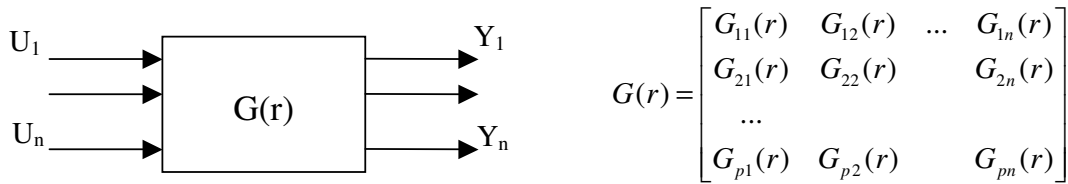
Định nghĩa một hàm truyền trong Matlab

$$F(p) = 3 \frac{p+2}{p^2+2p+4} \quad \text{num}=3*[1 \ 2]; \text{den}=[1 \ 2 \ 4]; \text{sys1}=\text{tf}(\text{num}, \text{den});$$

$$F(z) = 2,1 * \frac{z-0,6}{z^2-0,56z+0,4} \quad \text{num}=2.1*[1 \ -0.6]; \text{den}=[1 \ -0.56];$$

$$T=0.5; \text{sys2}=\text{tf}(\text{num}, \text{den}, T)$$

### Hệ thống nhiều tín hiệu vào/ra



Câu lệnh :

$$G11=\text{tf}(\text{num11}, \text{den11}, T); G12=\text{tf}(\text{num12}, \text{den12}, T); \dots; G1n=\text{tf}(\text{num1n}, \text{den1n}, T); \\ G21=\text{tf}(\text{num21}, \text{den21}, T); G22=\text{tf}(\text{num22}, \text{den22}, T); \dots; G2n=\text{tf}(\text{num2n}, \text{den2n}, T);$$

$$Gp1=\text{tf}(\text{nump1}, \text{denp1}, T); G12=\text{tf}(\text{nump2}, \text{denp2}, T); \dots; Gpn=\text{tf}(\text{numpn}, \text{denpn}, T); \\ \text{sys}=[G11, G12, \dots, G1n; G21; G22; \dots; G2n; \dots; Gp1, Gp2, \dots, Gpn];$$

#### 1.1.2 Định nghĩa bằng zero và cực

##### Hệ thống một tín hiệu vào/ra

Câu lệnh:  $\text{sys}=\text{zpk}(Z, P, K, T)$

- $Z, P$  là các vectơ hàng chứa danh sách các điểm zerô và cực của hệ thống.
- $K$  là hệ số khuếch đại

Chú ý: nếu hệ thống không có điểm zerô (cực) thì ta đặt là []

Ví dụ:

$$F(p) = \frac{p+2}{p(p+5)} \quad Z=-2; P=[0 \ -5]; K=1; \text{sys}=\text{zpk}(Z, P, K);$$

##### Hệ thống nhiều tín hiệu vào/ra

Câu lệnh :

$$G11=\text{zpk}(Z11, P11, T); G12=\text{zpk}(Z12, P12, T); \dots; G1n=\text{zpk}(Z1n, P1n, T); \\ G21=\text{zpk}(Z21, P21, T); G22=\text{zpk}(Z22, P22, T); \dots; G2n=\text{zpk}(Z2n, P2n, T);$$

$$Gp1=\text{zpk}(Zp1, Pp1, T); G12=\text{zpk}(Zp2, Pp2, T); \dots; Gpn=\text{zpk}(Zpn, Ppn, T); \\ \text{sys}=[G11, G12, \dots, G1n; G21; G22; \dots; G2n; \dots; Gp1, Gp2, \dots, Gpn];$$

#### 1.1.3 Phương trình trạng thái

Câu lệnh:  $\text{sys}=\text{ss}(A, B, C, D, T)$

- $A, B, C, D$  là các ma trận trạng thái định nghĩa hệ thống
- $T$  là chu kỳ lấy mẫu.

##### Chuyển đổi giữa các dạng biểu diễn

- Chuyển từ phương trình trạng thái sang hàm truyền  
 $[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)$
- Chuyển từ dạng zero/cực sang hàm truyền  
 $[\text{num}, \text{den}] = \text{zp2tf}(Z, P, K)$
- Chuyển từ hàm truyền sang phương trình trạng thái  
 $[A, B, C, D] = \text{tf2ss}(\text{num}, \text{den})$

### 1.1.4 Chuyển đổi giữa hệ liên tục và gián đoạn

#### Số hóa một hệ thống liên tục

Câu lệnh: `sys_dis=c2d(sys,T,method)`

- `sys`, `sys_dis` hệ thống liên tục và hệ thống gián đoạn tương ứng
- `Ts` thời gian lấy mẫu
- `method` phương pháp lấy mẫu: 'zoh' lấy mẫu bậc 0, 'foh' lấy mẫu bậc 1, 'tustin' phương pháp Tustin...

Ví dụ: chuyển một khâu liên tục có hàm truyền  $G(p) = \frac{2}{0.5p+1}$  sang khâu gián đoạn bằng phương

pháp giữ mẫu bậc 0, chu kỳ lấy mẫu  $T=0.01s$

`num=2`

`den=[0.5 1]`

`sysc=tf(num,den)`

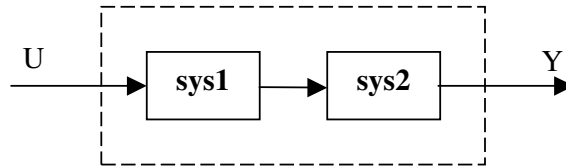
`sysd=c2d(sysc,0.01,'zoh')`

#### Hệ liên tục tương đương của một hệ thống gián đoạn

Câu lệnh: `sys=d2c(sys_dis,method)`

## 1.2 Biến đổi sơ đồ tương đương

### 1.2.1 Mắc nối tiếp



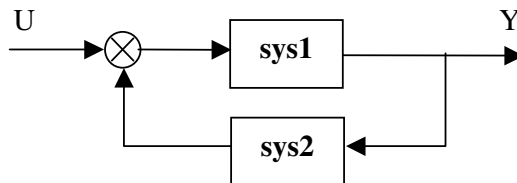
Câu lệnh: `sys=series(sys1,sys2)`

### 1.2.2 Mắc song song

Câu lệnh: `sys=parallel(sys1,sys2)`

### 1.2.3 Mắc phản hồi

Câu lệnh: `sys=feedback(sys1,sys2,sign)`



`sign = +1` nếu phản hồi dương và `sign=-1` (hoặc không có `sign`) nếu phản hồi âm.

### 1.3 Phân tích hệ thống

#### 1.3.1 Trong miền thời gian

##### Hàm quá độ $h(t)$

Câu lệnh: `step(sys)`

Vẽ hàm quá độ của hệ thống tuyến tính  $sys$ . Khoảng thời gian vẽ và bước thời gian do Matlab tự chọn.

Một số trường hợp khác

- `step(sys,t_end)`: vẽ hàm quá độ từ thời điểm  $t=0$  đến thời điểm  $t_{end}$ .
- `step(sys,T)`: vẽ hàm quá độ trong khoảng thời gian  $T$ .  $T$  được định nghĩa như sau  $T=Ti:dt:Tf$ . Đối với hệ liên tục,  $dt$  là bước vẽ, đối với hệ gián đoạn,  $dt=Ts$  là chu kỳ lấy mẫu.
- `step(sys1,sys2,sys3,...)`: vẽ hàm  $h(t)$  cho nhiều hệ thống đồng thời.
- `[y,t]=step(sys)`: tính đáp ứng  $h(t)$  và lưu vào các biến  $y$  và  $t$  tương ứng

##### Hàm trọng lượng $\omega(t)$

Câu lệnh: `impulse(sys)`

#### 1.3.2 Trong miền tần số

##### Đặc tính bode

Câu lệnh: `bode(sys)`

Vẽ đặc tính tần số Bode của hệ thống tuyến tính  $sys$ . Dải tần số vẽ do Matlab tự chọn.

Một số trường hợp khác

- `bode(sys,{w_start,w_end})`: vẽ đặc tính bode từ tần số  $w_{start}$  đến tần số  $w_{end}$ .
- `bode(sys,w)` vẽ đặc tính bode theo vectơ tần số  $w$ . Vectơ tần số  $w$  được định nghĩa bằng hàm `logspace`. Ví dụ: `w=logspace(-2,2,100)` định nghĩa vectơ  $w$  gồm 100 điểm, từ tần số  $10^{-2}$  đến  $10^2$ .
- `bode(sys1,sys2,sys3,...)` vẽ đặc tính bode của nhiều hệ thống đồng thời.
- `[mag,phi,w]=bode(sys,...)` lưu tất cả các điểm tính toán của đặc tính bode vào vectơ  $mag$ ,  $phi$  ứng với tần số  $w$  tương ứng.

Chú ý: Đối với hệ thống gián đoạn, dải tần số để vẽ phải thỏa mãn định lý Shannon.

##### Đặc tính Nyquist

Câu lệnh: `nyquist(sys)`  
`nyquist(sys,{w_start,w_end})`  
`nyquist(sys,w)`  
`nyquist(sys1,sys2,sys3,...,w)`  
`[real,ima,w]=nyquist(sys,...)`

##### Đặc tính Nichols

Câu lệnh: `nichols(sys)`  
`nichols(sys,{w_start,w_end})`  
`nichols(sys,w)`  
`nichols(sys1, sys2, sys3,...,w)`  
`[mag,phi,w]=nichols(sys,...)`

Tính toán  $|G(\omega)|$ ,  $\arg[G(\omega)]$  và vẽ trong mặt phẳng Black.

Ví dụ: Vẽ các đặc tính tần số của hệ thống sau

$$G(p) = \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2} \quad \text{với } \omega_0=1\text{rad/s và } \xi=0,5$$

```
w0=1 ;xi=0.5 ;num=w0^2 ;den=[1 2*xi*w0^2 w0^2] ;G=tf(num,den);
w=logspace(-2,2,100) ;
bode(G,w) ; % vẽ đặc tính bode trong dải tần số w
nichols(G) ; % vẽ đặc tính nichols trong dải tần số tự chọn của Matlab
nyquist(G) ; % vẽ đặc tính nyquist
```

### 1.3.3 Một số hàm để phân tích

#### Hàm *margin*

- *margin(sys)* vẽ đặc tính *Bode* của hệ thống SISO và chỉ ra độ dự trữ biên độ, độ dự trữ pha tại các tần số tương ứng.
- *[delta\_L,delta\_phi,w\_L,w\_phi]=margin(sys)* tính và lưu độ dự trữ biên độ vào biến *delta\_L* tại tần số *w\_L*, lưu độ dự trữ về pha vào biến *delta\_phi* tại tần số *w\_phi*.

#### Hàm *pole*

*vec\_pol=pole(sys)* tính các điểm cực của hệ thống và lưu vào biến *vec\_pol*.

#### Hàm *tzero*

*vec\_zer=tzero(sys)* tính các điểm zero của hệ thống và lưu vào biến *vec\_zer*.

#### Hàm *pzmap*

- *[vec\_pol,vec\_zer]=pzmap(sys)* tính các điểm cực và zero của hệ thống và lưu vào các biến tương ứng.
- *pzmap(sys)* tính các điểm cực, zero và biểu diễn trên mặt phẳng phức.

#### Hàm *dcgain*

*G0=dcgain(sys)* tính hệ số khuếch đại tĩnh của hệ thống và lưu vào biến *G0*.

### 1.3.4 Một số hàm đặc biệt trong không gian trạng thái

#### Hàm *ctrl*

Câu lệnh: *C\_com=ctrl(A,B)*  
*C\_com=ctrl(sys)*

Tính ma trận “điều khiển được” **C** của một hệ thống. Ma trận **C** được định nghĩa như sau:

$$\mathbf{C}=[\mathbf{B} \ \mathbf{A}\mathbf{B} \ \mathbf{A}^2\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad \text{với } \mathbf{A} \in \mathcal{R}^{n \times n}$$

#### Hàm *obsv*

Câu lệnh: *O\_obs=obsv(A,C)*  
*O\_obs=obsv(sys)*

Tính ma trận “quan sát được” **O** của một hệ thống. Ma trận **O** được định nghĩa như sau:  $\mathbf{O}=[\mathbf{C} \ \mathbf{C}\mathbf{A} \ \mathbf{C}\mathbf{A}^2 \ \dots \ \mathbf{C}\mathbf{A}^{n-1}]$

#### Hàm *ctrbf*

Câu lệnh: *[Ab,Bb,Cb,T,k]=ctrbf(A,B,C)*

Chuyển về dạng chuẩn (canonique) “điều khiển được” của một hệ thống biểu diễn dưới dạng phương trình trạng thái.

Trong đó:  $\mathbf{A}_b=\mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ ,  $\mathbf{B}_b=\mathbf{T}\mathbf{B}$ ,  $\mathbf{C}_b=\mathbf{C}\mathbf{T}^{-1}$ , **T** là ma trận chuyển đổi.

#### Hàm *obsvf*

Câu lệnh: *[Ab,Bb,Cb,T,k]=obsvf(A,B,C)*

Chuyển về dạng chuẩn “quan sát được“ của một hệ thống biểu diễn dưới dạng phương trình trạng thái.

Trong đó:  $A_b = TAT^{-1}$ ,  $B_b = TB$ ,  $C_b = CT^{-1}$ ,  $T$  là ma trận chuyển đổi.

#### 1.4 Ví dụ tổng hợp

Cho một hệ thống kín phản hồi -1, trong đó hàm truyền của hệ hở là

$$G(p) = \frac{K}{p(1+\tau p)} * \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2} \quad \text{với } K=1, \tau=10s, \omega_0=1\text{rad/s và } \xi=0.5$$

1. Vẽ đặc tính tần số Nyquist. Chứng tỏ rằng hệ kín không ổn định.
2. Vẽ đáp ứng quá độ của hệ kín.
3. Để hệ thống ổn định, người ta hiệu chỉnh hệ số khuếch đại  $K=0.111$ . Xác định tần số cắt, độ dự trữ biên độ và độ dự trữ về pha của hệ thống trong trường hợp này.
4. Xác định các thông số quá độ (thời gian quá độ lớn nhất  $T_{\max}$ , độ quá điều chỉnh lớn nhất  $\sigma_{\max}$ ) của hệ thống đã hiệu chỉnh.

Giải

##### Câu 1

```
>>K=1;to=10;w0=1;xi=0.5;
>>num1=K;den1=[to 1 0];
>>num2=w0^2;den2=[1 2*xi*w0 w0^2];
>>G=tf(num1,den1)*tf(num2,den2)
```

Transfer function:

1

-----  
10 s^4 + 11 s^3 + 11 s^2 + s

```
>>w=logspace(-3,2,100); % tạo vector tần số để vẽ các đặc tính tần số
```

```
>>nyquist(G,w);
```

Đặc tính được biểu diễn trên hình 6.1

Để xét tính ổn định của hệ kín dùng tiêu chuẩn Nyquist, trước tiên ta xét tính ổn định của hệ hở. Nghiệm của phương trình đặc tính của hệ hở được xác định :

```
>>pole(G)
```

ans =

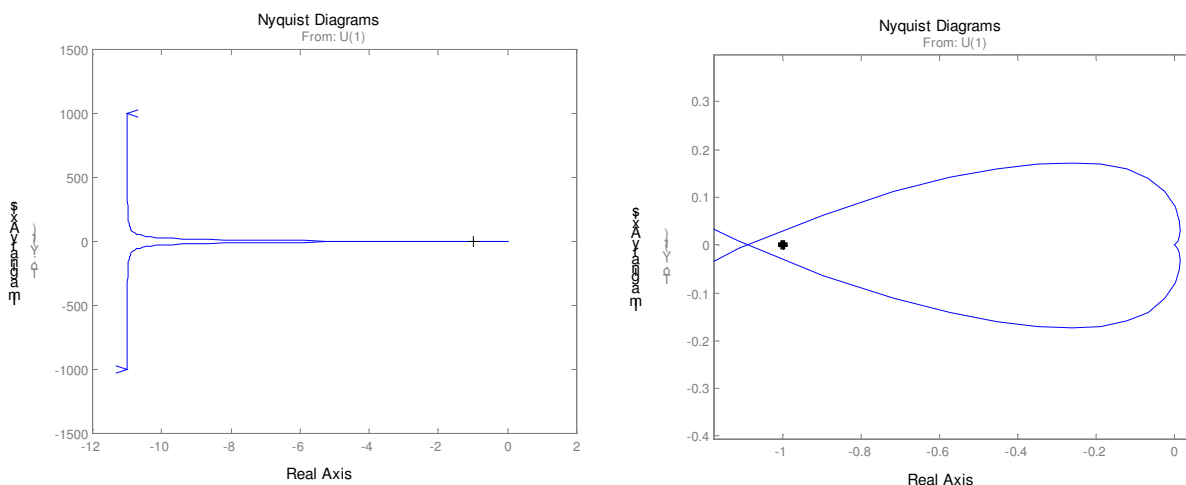
0

-0.5000 + 0.8660i

-0.5000 - 0.8660i

-0.1000

Hệ hở có 1 nghiệm bằng 0 nên ở biên giới ổn định.



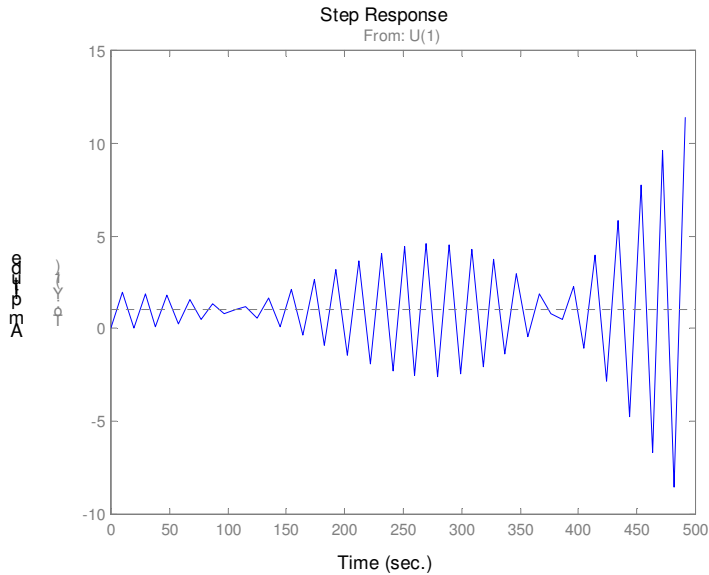
Hình 6.1 : Đặc tính tần số Nyquist của hệ hở



Quan sát đặc tính tần số Nyquist của hệ hở trên hình 6.1 (phần zoom bên phải), ta thấy đặc tính Nyquist bao điểm  $(-1, j0)$ , và do hệ hở ở biên giới ổn định nên theo tiêu chuẩn Nyquist, **hệ thống kín sẽ không ổn định.**

### Câu 2

```
>>G_loop=feedback(G,1,-1);           % hàm truyền hệ kín
>>step(G_loop);
```



Hình 6.2 :  
Đáp ứng quá độ hệ kín

### Câu 3

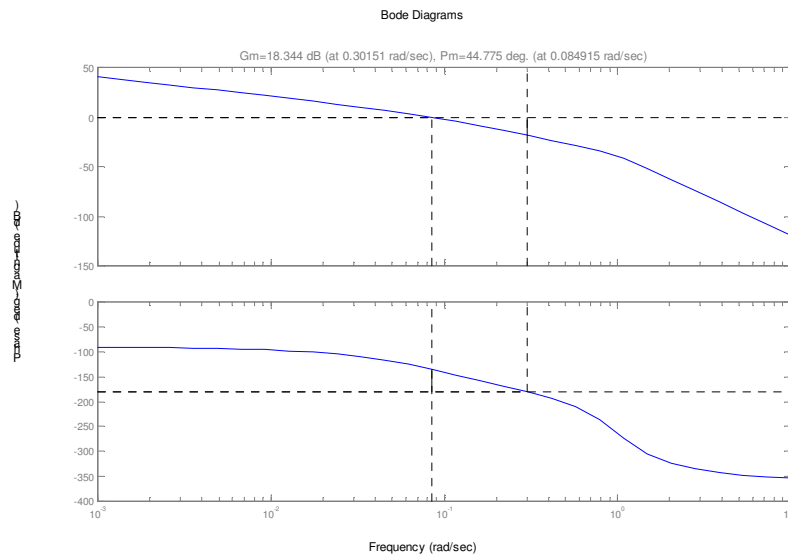
```
>>K=0.111 ; num1=K ;                 % thay đổi hệ số khuếch đại K
>>GK=tf(num1,den1)*tf(num2,den2)
Transfer function:
0.111
```

-----  
 $10 s^4 + 11 s^3 + 11 s^2 + s$

```
>>margin(GK)
```

Đặc tính tần số Bode của hệ hở đã hiệu chỉnh được biểu diễn trên hình 6.3. Từ đặc tính này, ta có thể xác định được

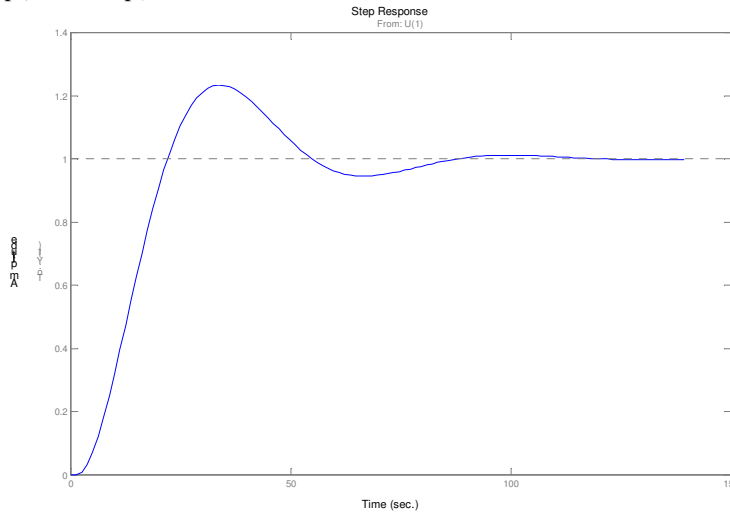
$$\Delta L=18.34\text{dB} ; \Delta\varphi = 44.78^\circ ; \omega_c=0.085\text{rad/s}$$



Hình 6.3 : Đặc tính tần số Bode của hệ hở đã hiệu chỉnh

**Câu 4**

```
>>GK_loop=feedback(GK,1,-1);
>>step(GK_loop);
```



Hình 6.4  
áp ứng quá độ hệ  
tín đã hiệu chỉnh

Sử dụng con trỏ chuột và kích vào các điểm cần tìm trên đặc tính, ta xác định được

$$\sigma_{\max}=23\%; T_{\max}= 70.7s$$

## 2 SIMULINK

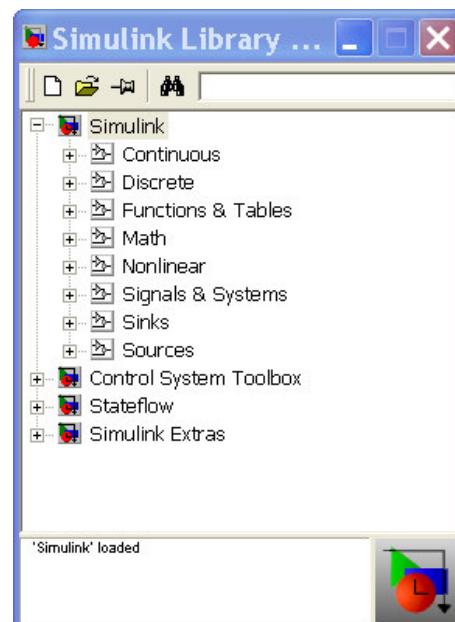
Simulink được tích hợp vào Matlab (vào khoảng đầu những năm 1990) như một công cụ để mô phỏng hệ thống, giúp người sử dụng phân tích và tổng hợp hệ thống một cách trực quan. Trong Simulink, hệ thống không được mô tả dưới dạng dòng lệnh theo kiểu truyền thống mà ở dưới dạng sơ đồ khối. Với dạng sơ đồ khối này, ta có thể quan sát các đáp ứng thời gian của hệ thống với nhiều tín hiệu vào khác nhau như : tín hiệu bậc thang, tín hiệu sinus, xung chữ nhật, tín hiệu ngẫu nhiên... bằng cách thực hiện mô phỏng. Kết quả mô phỏng có thể được xem theo thời gian thực trên các Oscilloscope trong môi trường Simulink, hay trong môi trường Matlab.

Simulink hoàn toàn tương thích với Matlab, nhưng nó là một giao diện đồ họa. Vì vậy tất cả các hàm trong Matlab đều có thể truy cập được từ Simulink, ngay cả các hàm do người sử dụng tạo ra. Ngược lại, các kết quả tìm được trong Simulink đều có thể được sử dụng và khai thác trong môi trường Matlab.

Cuối cùng, Simulink cho phép người sử dụng khả năng tạo ra một thư viện khối riêng. Ví dụ, nếu bạn muốn làm việc trong lĩnh vực điều khiển các máy điện, bạn có thể tạo ra một thư viện riêng chứa các mô hình máy điện... Như vậy, với công cụ Simulink, ta có thể tự tiến hành mô phỏng thí nghiệm, quan sát kết quả, kiểm chứng với lý thuyết trước khi tiến hành thí nghiệm trên mô hình thật.

### 2.1 Khởi động Simulink

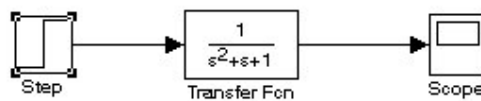
Để khởi động Simulink từ môi trường Matlab, ta gõ dòng lệnh *simulink*. Lúc này một cửa sổ như trên hình 6.5 sẽ xuất hiện, trên đó có các thư mục chính và các thư viện con của Simulink. Để bắt đầu làm việc, ta tạo cửa sổ mới bằng cách kích vào biểu tượng « New ». Có 8 thư viện chính của Simulink được phân loại như sau :



- *Continuous* : hệ thống tuyến tính và liên tục
- *Discrete* : hệ thống tuyến tính gián đoạn
- *Nonlinear* : mô hình hóa những phần tử phi tuyến như role, phần tử bão hòa...
- *Source* : các khối nguồn tín hiệu
- *Sinks* : các khối thu nhận tín hiệu
- *Function & Table* : các hàm bậc cao của Matlab
- *Math* : các khối của simulink với các hàm toán học tương ứng của Matlab
- *Signals & System* : các khối liên hệ tín hiệu, hệ thống con...

## 2.2 Tạo một sơ đồ đơn giản

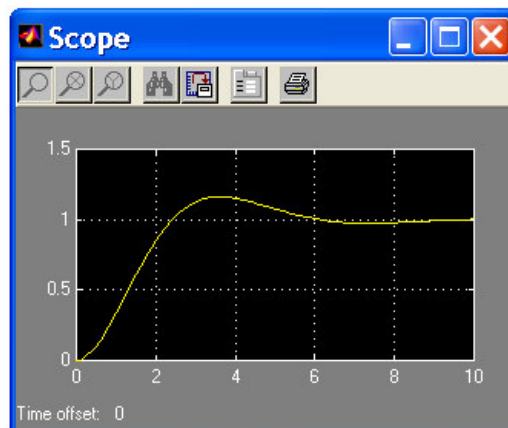
Để làm quen với Simulink, ta bắt đầu bằng một ví dụ đơn giản : phân tích hàm quá độ của một khâu bậc hai có hàm truyền  $G(p) = \frac{\omega_0^2}{p^2 + 2\xi\omega_0 p + \omega_0^2}$  với  $\omega_0=1\text{rad/s}$  và  $\xi=0,5$ . Các bước thực hiện để được sơ đồ mô phỏng như hình 6.6 như sau :



Hình 6.6 : Một sơ đồ Simulink đơn giản

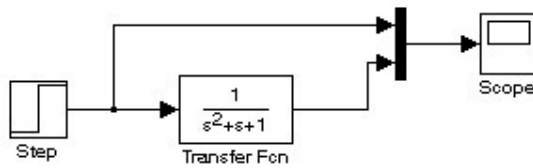
- Khởi động Simulink từ Matlab bằng dòng lệnh *simulink*
- Trong cửa sổ chính của Simulink, chọn biểu tượng « New » để tạo cửa sổ ứng dụng.
- Muốn tạo một khối trong cửa sổ ứng dụng, ta tìm khối đó trong các thư viện của Simulink, kích chọn và kéo nó vào cửa sổ ứng dụng. Ví dụ, để tạo khối *Step*, ta vào thư viện *Simulink* -> *Continuous* -> *Sources* -> *Step*, khối *Transfer Fcn* trong *Simulink* -> *Continuous* -> *Transfer Fcn*...
- Để đặt thông số cho từng khối, ta mở khối đó ra bằng cách double-click chuột vào nó. Lúc này đặt các thông số theo hướng dẫn trên màn hình.
- Đường nối giữa các khối được thực hiện bằng cách dùng chuột kéo các mũi tên ở đầu (cuối) mỗi khối đến vị trí cần nối.

Sau khi tạo được sơ đồ khối như hình 6.6, ta có thể bắt đầu tiến hành mô phỏng (với các tham số mặc định) bằng cách chọn **Simulation -> Start**. Xem kết quả mô phỏng bằng cách mở khối *Scope* như hình 6.7.

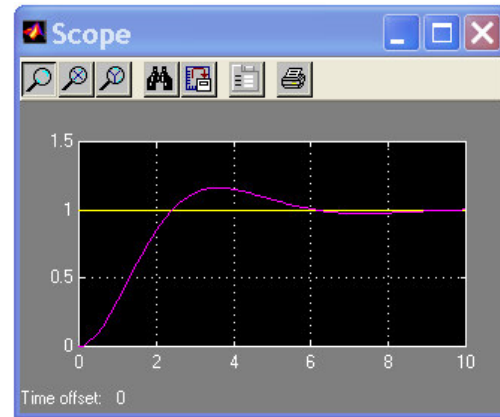


Hình 6.7 : Kết quả mô phỏng

Để xem đồng thời tín hiệu vào và ra trên cùng một *Scope*, ta tạo sơ đồ mô phỏng như hình 6.8. Kết quả mô phỏng biểu diễn trên hình 6.9.



Hình 6.8



Hình 6.9

### 2.3 Một số khối thường dùng

Thư viện « Sources »

- Step** Tạo ra tín hiệu bậc thang liên tục hay gián đoạn.  
**Ramp** Tạo tín hiệu dốc tuyến tính (rampe) liên tục.  
**Sine Wave** Tạo tín hiệu sinus liên tục hay gián đoạn.  
**Constant** Tạo tín hiệu không đổi theo thời gian.  
**Clock** Cung cấp đồng hồ chỉ thời gian mô phỏng. Có thể xem được « đồng hồ » này khi đang thực hiện mô phỏng.

**Chú ý :** Muốn khối clock chỉ đúng thời điểm đang mô phỏng, tham số **Sample time** được đặt như sau

→ 0 : hệ liên tục

→ >0 : hệ gián đoạn, *clock* lúc này sẽ chỉ số chu kỳ lấy mẫu đặt trong **Sample time**.

Thư viện « Sinks »

- Scope** Hiển thị các tín hiệu được tạo ra trong mô phỏng.  
**XY Graph** Vẽ quan hệ giữa 2 tín hiệu theo dạng XY. Khối này cần phải có 2 tín hiệu vào, tín hiệu thứ nhất tương ứng với trục X, tín hiệu vào thứ hai tương ứng với trục Y.  
**To Workspace** Tất cả các tín hiệu nối vào khối này sẽ được chuyển sang không gian tham số của Matlab khi thực hiện mô phỏng. Tên của biến chuyển vào Matlab do người sử dụng chọn.

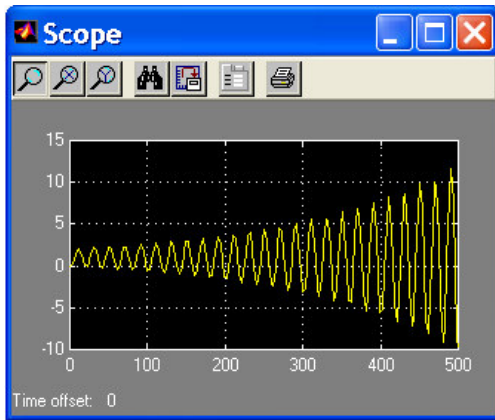
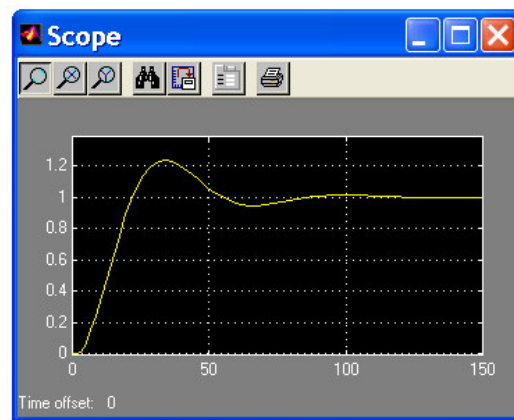
#### 2.3.1 Thư viện « Continuous »

**Transfer Fcn** Mô tả hàm truyền của một hệ thống liên tục dưới dạng *Đa thức tử số/Đa thức mẫu số*. Các hệ số của đa thức tử số và mẫu số do người sử dụng nhập vào, theo bậc giảm dần của toán tử *Laplace*. Ví dụ để nhập vào hàm truyền có dạng  $\frac{2s+1}{s^2+s+1}$ , ta nhập vào như sau : *Numerator* [2 1], *Denominator* [1 1 1].

**State Space** Mô tả hàm truyền của một hệ thống liên tục dưới dạng *phương trình trạng thái*. Các ma trận trạng thái A, B, C, D được nhập vào theo qui ước ma trận của Matlab.

- Integrator** Khâu tích phân.  
**sDerivative** Khâu đạo hàm  
**Transport Delay** Khâu tạo trễ



Hình 6.11 : Đáp ứng quá độ ( $K=1$ )Hình 6.12 : Đáp ứng quá độ ( $K=0.111$ )

## 2.5 LTI Viewer

Như ta đã biết, khi thực hiện mô phỏng trên Simulink, ta chỉ có thể quan sát được các đặc tính thời gian của hệ thống. Để có thể phân tích toàn diện một hệ thống, ta cần các đặc tính tần số như đặc tính Bode, đặc tính Nyquist, quỹ đạo nghiệm số v.v...

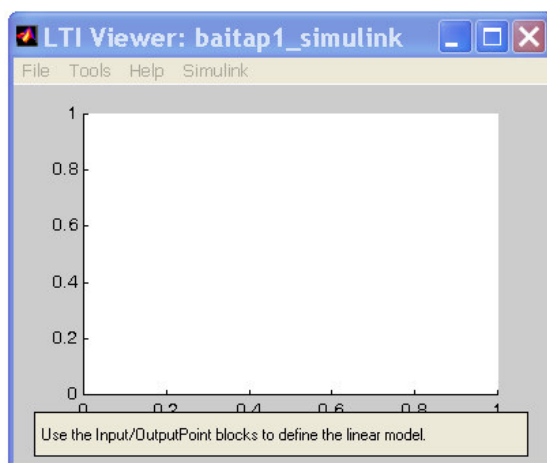
« **LTI Viewer** » là một giao diện đồ họa cho phép quan sát đáp ứng của một hệ thống tuyến tính, trong lĩnh vực tần số cũng như thời gian, mà không cần gõ lại lệnh hay lập trình theo từng dòng lệnh như trong Control System Toolbox. Nó sử dụng trực tiếp sơ đồ khối trong Simulink.

### 2.5.1 Khởi động LTI Viewer

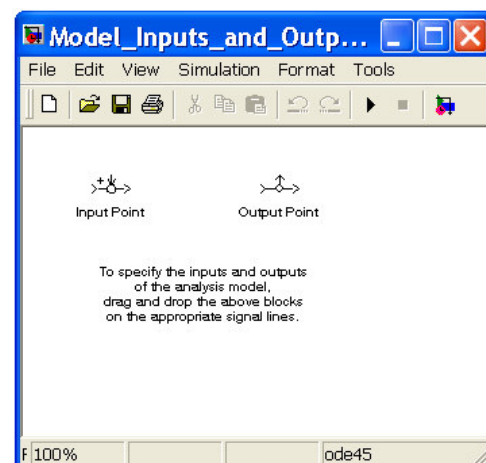
Để khởi động LTI Viewer từ Simulink, ta chọn menu **Tool -> Linear Analysis**.

Lúc này, Matlab sẽ mở 2 cửa sổ mới:

- Cửa sổ LTI Viewer (hình 6.13) có 2 phần chính:
  - o Phần cửa sổ đồ họa dùng để biểu diễn các đường đặc tính.
  - o Thanh công cụ phía dưới chỉ dẫn cách sử dụng LTI Viewer
- Cửa sổ chứa các điểm input và output (hình 6.14). Các điểm này được dùng để xác định điểm vào/ra trên sơ đồ Simulink cần phân tích.



Hình 6.13



Hình 6.14

### 2.5.2 Thiết lập các điểm vào/ra cho LTI Viewer

Dùng chuột kéo rê các điểm “input point”, “output point” trên cửa sổ hình 6.14 và đặt lên các vị trí tương ứng trên sơ đồ Simulink.

**Chú ý:** Việc chọn các điểm đặt “input”, “output” phải phù hợp yêu cầu phân tích. LTI Viewer tính hàm truyền bằng cách tuyến tính hóa hệ thống với 2 điểm input/output đã được định nghĩa. Khi vẽ các đặc tính tần số cũng như thời gian, LTI sử dụng các hệ thống đã được tuyến tính hóa này.

### 2.5.3 Tuyến tính hóa một mô hình

Để tìm mô hình giữa 2 điểm input/output đã định nghĩa, ta thực hiện như sau:

Chọn cửa sổ LTI Viewer (hình 6.13) → Chọn menu **Simulink → Get linearized model**

Lúc này, trong phần đồ họa của cửa sổ LTI Viewer sẽ xuất hiện đặc tính quá độ của mô hình tuyến tính hóa tìm được.

Để xem các đặc tính khác trên LTI Viewer, ta chỉ việc kích chuột phải vào phần đồ họa, chọn menu **Plot Type** → chọn loại đặc tính cần quan sát.

#### Ghi chú:

- Cứ mỗi lần thực hiện tuyến tính hóa một mô hình (**Simulink → Get linearized model**) thì LTI Viewer sẽ nạp mô hình hiện hành tại cửa sổ Simulink vào không gian của nó. Nếu giữa 2 lần thực hiện tuyến tính hóa, mô hình không có sự thay đổi (cấu trúc hay thông số) thì 2 mô hình tìm được tương ứng sẽ giống nhau.
- Có thể bật/tắt đặc tính của một hay nhiều mô hình đã tìm được trong LTI Viewer bằng cách: kích chuột phải vào cửa sổ đồ họa → chọn *Systems* → chọn mô hình cần bật/tắt. Tiện ích này rất cần thiết khi ta muốn so sánh tác động do sự biến đổi một thông số nào đó đến hệ thống.

### 2.5.4 Lưu và sử dụng các thông số của mô hình tuyến tính hóa

- Để lưu mô hình tuyến tính hóa vừa tìm được, chọn menu **File → Export...**
- Để sử dụng các thông số của mô hình :
  - o Dạng hàm truyền  $[num,den]=tfdata(\ll\text{bien file}\gg, 'v')$
  - o Dạng phương trình trạng thái  $[A,B,C,D]=ssdata(\ll\text{bien file}\gg)$

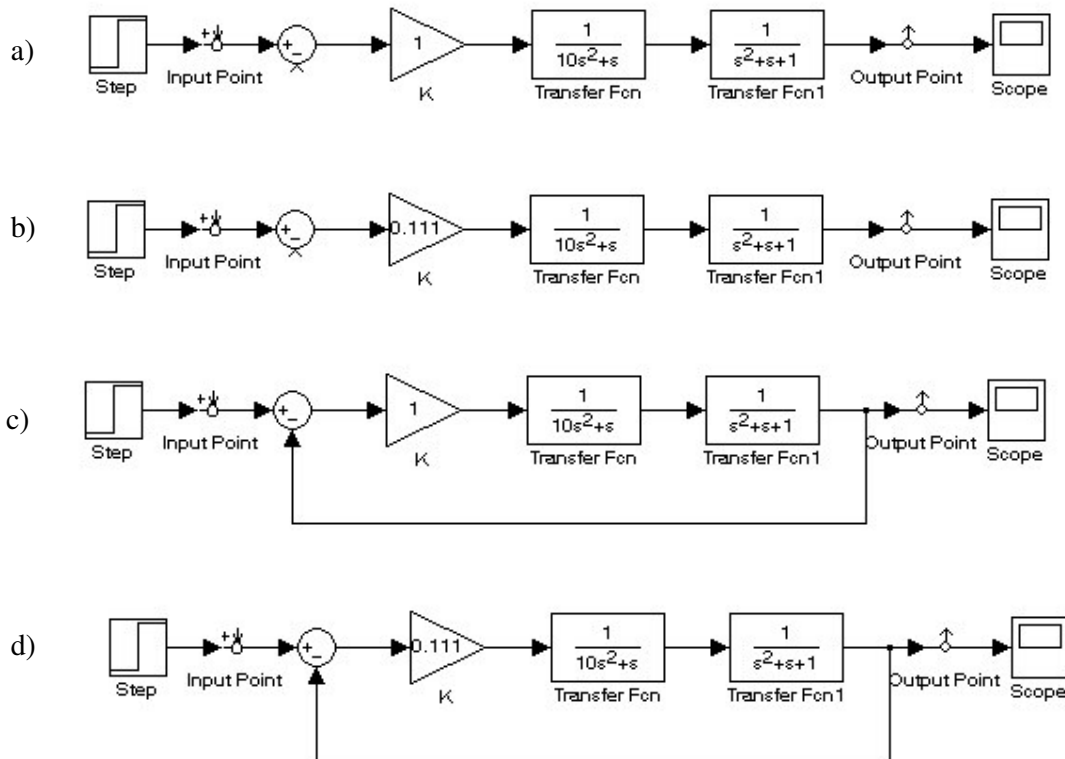
### 2.5.5 Ví dụ sử dụng LTI Viewer

Giả sử đã có hàm mô hình mô phỏng trên cửa sổ Simulink như hình 2.6. Sử dụng LTI Viewer để quan sát các đặc tính sau:

- Đặc tính tần số Nyquist của hệ hở khi chưa hiệu chỉnh ( $K=1$ ) và đã hiệu chỉnh ( $K=0.111$ ).
- Đặc tính tần số Bode của hệ hở đã hiệu chỉnh .
- Đặc tính quá độ của hệ kín chưa hiệu chỉnh và đã hiệu chỉnh.

#### THỰC HIỆN

Theo yêu cầu đặt ra, ta cần phải có 4 hệ thống có thông số và cấu trúc khác nhau: hệ hở với  $K=1$ , hệ hở với  $K=0.111$ , hệ kín  $K=1$  và hệ kín  $K=0.111$ . Do vậy, ta cần thực hiện 4 lần tuyến tính hóa để có được 4 mô hình khác nhau trong LTI Viewer. Các bước thực hiện tuần tự như trong hình 6.15.



Hình 6.15 : Sơ đồ và cấu trúc để tuyến tính hóa

Sau 4 lần tuyến tính hóa trong LTI Viewer, ta được 4 hệ thống lần lượt là *baitap1\_simulink\_1* đến *baitap1\_simulink\_4* (sơ đồ trong Simulink có tên là *baitap1\_simulink*).

Trên cửa sổ đồ họa lúc này sẽ hiển thị đồng thời đặc tính quá độ của cả 4 mô hình ở trên.

- Để xem đặc tính Nyquist của hệ hở trước và sau hiệu chỉnh:
  - o Kích chuột phải vào phần đồ họa, chọn *Systems*, chọn 2 mô hình 1 và 2.
  - o Tiếp tục kích chuột phải vào phần đồ họa, chọn *Plot Type* → *Nyquist*.
 Trên cửa sổ đồ họa sẽ xuất hiện 2 đặc tính Nyquist với 2 màu phân biệt.
- Để xem đặc tính quá độ của hệ kín trước và sau hiệu chỉnh:
  - o Kích chuột phải vào phần đồ họa, chọn *Systems*, chọn 2 mô hình 3 và 4.
  - o Tiếp tục kích chuột phải vào phần đồ họa, chọn *Plot Type* → *Step*.

Các đặc tính khác được tiến hành một cách tương tự.



# GIÁO TRÌNH ROBOCON

Robocon đã được tổ chức từ năm 2002. Cuộc thi này đã cuốn hút rất nhiều bạn sinh viên có niềm đam mê khoa học kỹ thuật. Cuộc thi này đã đem lại cho

Chúng ta rất nhiều những kiến thức về kỹ thuật ,rèn luyện đức tính kiên trì ,bền bỉ ,dám đối mặt với gian khổ .Hơn nữa ,cuộc thi đã để lại những kỉ niệm đẹp về một thời sinh viên.

Giáo trình này được biên soạn nhằm giúp các bạn có được một cái nhìn tổng quan về robot ,sự định hướng cụ thể và những kinh nghiệm quý báu khi tham gia robocon

Giáo trình gồm 8 bài :

**Bài 1: Giới thiệu về Robot Các Modul của robot.**

**Bài 2: Mạch điều khiển robot bằng tay.**

**Bài 3: Mạch cảm biến dò đường.**

**Bài 4: Mạch công suất điều khiển động cơ .**

**Bài 5: Mạch vi điều khiển.**

**Bài 6: Lập trình và kỹ thuật dò đường .**

**Bài 7: Lập trình các thao tác khác và chiến thuật.**

**Bài 8: Chương trình hoàn thiện cho một robot tự động**

Toàn bộ các bài học đều được thực hành trên mạch và robot thật.Sử dụng mạch điện và robot của đội BK-FIRE.

## BÀI 1.GIỚI THIỆU VỀ ROBOT VÀ CÁC MODUL CỦA ROBOT

**Bài này sẽ giới thiệu cho các bạn một cách tổng quan về robot và các modul của robot cũng như một số kinh nghiệm khi làm robocon.**

### **1.1.Giới thiệu về robot**

#### 1.1.1.Những hình ảnh về robot trong các cuộc thi robocon

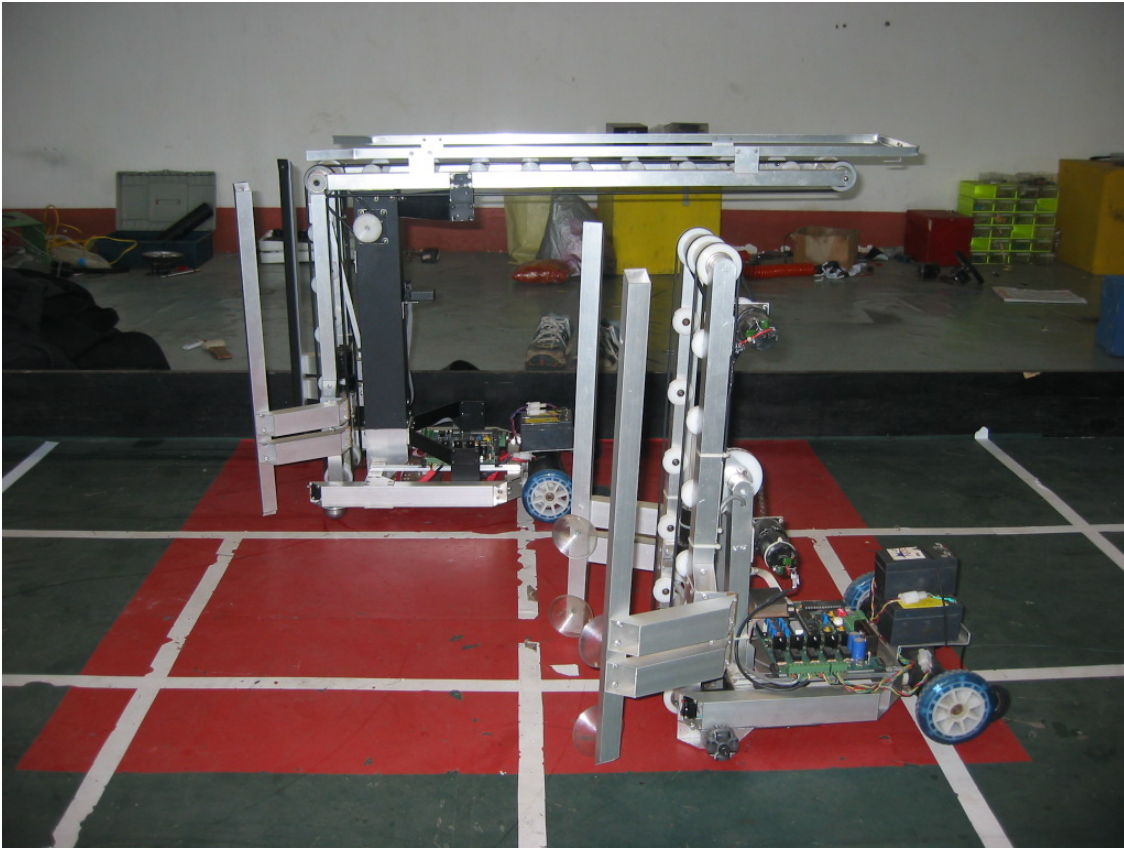
Hình 1.1.BK-FIRE 2005 (Robot tự động)



Hình 1.2.BK-FIRE 2005 (Robot bằng tay)

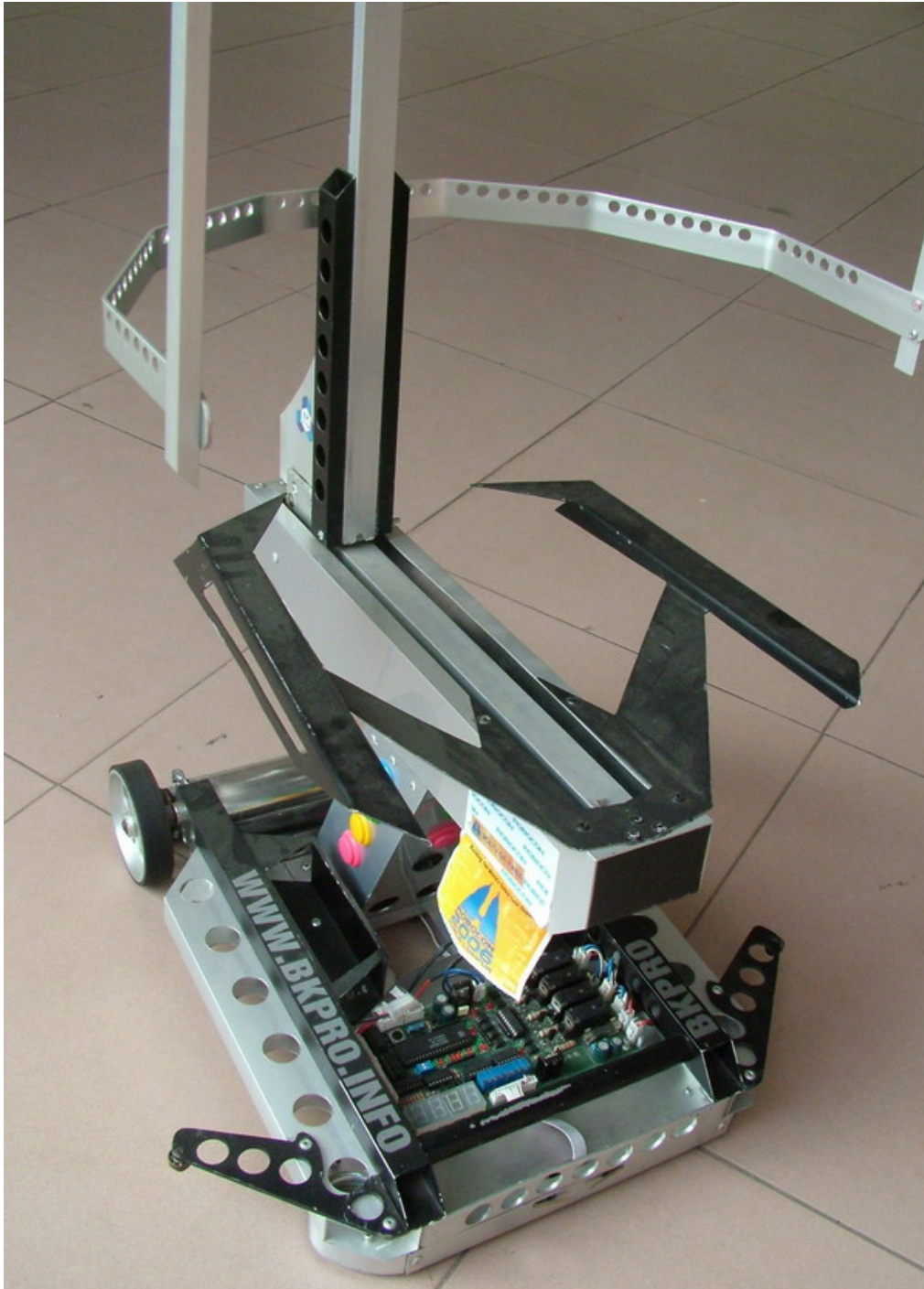


Hình 1.3. FXR 2004 (robot tự động)



Hình 1.4.BKPRO 2006





### 1.1.2. Các modul trong robot

Trong robocon ,có 2 loại robot là robot tự động và robot điều khiển bằng tay.

Robot bao gồm 3 phần chính :Cơ khí ,mạch điện tử và lập trình

+ **Cơ khí :**

Phần này bao gồm các kết cấu cơ khí của robot .Phần này mỗi năm đều thay đổi tùy thuộc vào đề thi.Đây chính là phần thể hiện ý tưởng của bạn.Khi có một ý tưởng độc đáo ,một kết cấu cơ khí tốt ,bạn sẽ có nhiều cơ hội dành chiến thắng.Trong cuộc thi robocon ,ý tưởng đóng vai trò rất quan trọng .Do đó ,khi thiết kế cơ khí ,toàn đội cần phải họp bàn kỹ lưỡng để đưa ra một giải pháp thiết kế cơ khí tối ưu nhất.Để thiết kế cơ khí ,các bạn có thể dung các phần mềm thiết kế cơ khí chuyên dụng như Solid work ,Autocad.

Hình 1.3.Cơ cấu bánh xích



Hình 1.4.Cơ cấu khí nén



Cơ cấu khí nén được BK-FIRE sử dụng đầu tiên trong cuộc thi robocon 2005 .Từ đó ,cơ cấu này đã được rất nhiều đội sử dụng và đã rất thành công như FAS-01 ,BK-FIRE (2006).

#### + Mạch điện tử

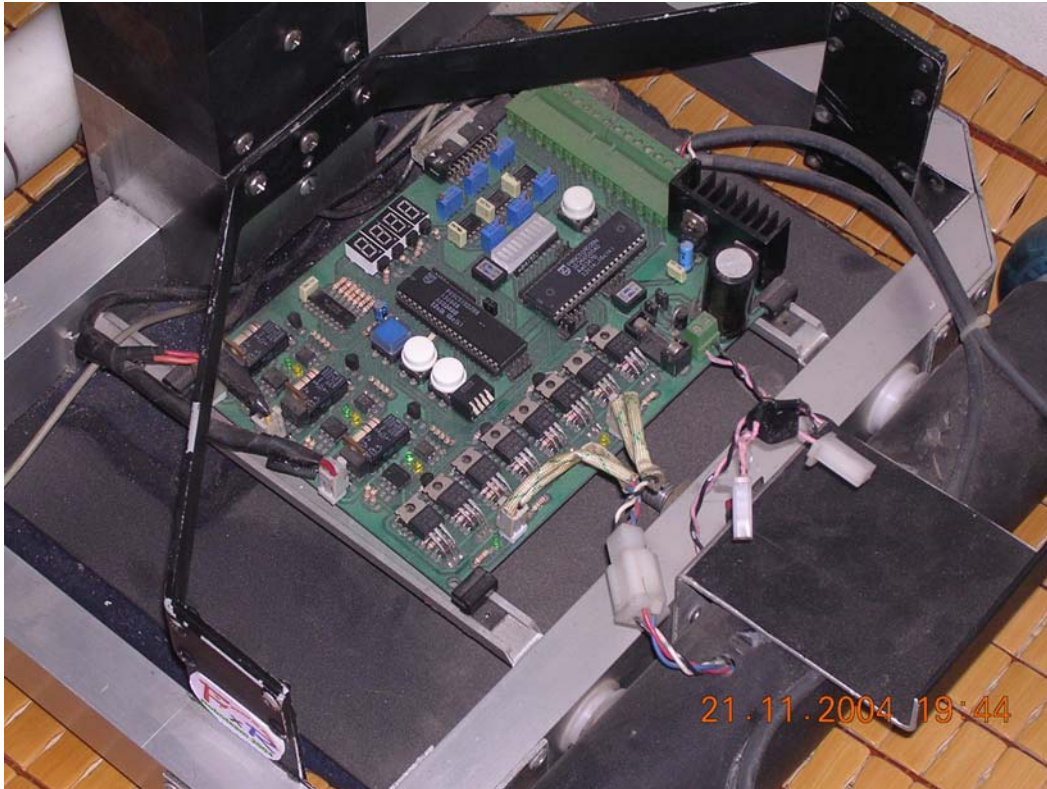
Mạch điện tử trong robot đóng vai trò như những mạch máu trong một cơ thể người .Nó đóng vai trò kết nối toàn bộ những bộ phận khác trong robot thành một thể thống nhất.

Mạch điện tử trong robot bao gồm các phần chính :

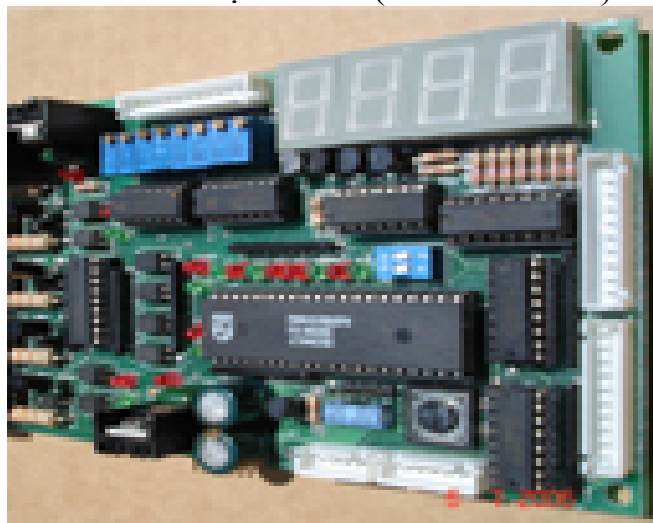
- Mạch vi điều khiển (8051 ,PIC ,AVR) đóng vai trò như bộ não của robot điều khiển toàn bộ hoạt động của robot theo chương trình lập sẵn.
- Mạch sensor đóng vai trò như các giác quan của robot để giúp robot nhận dạng môi trường xung quanh để gửi tín hiệu về vi điều khiển.
- Mạch công suất điều khiển động cơ ,dùng để điều khiển các cơ cấu của robot
- Mạch chiến thuật thi đấu :cho phép người sử dụng có thể tùy ý lựa chọn chiến thuật khi ở trên sân thi đấu.

Hình 1.5.Mạch điện tử trong robocon (FXR)





Hình 1.6.Mạch robot (BK-PRO 2006)



### **+Lập trình**

Chương trình trong robot giống như bộ não của một cơ thể người ,robot chỉ hoạt động được khi có chương trình cài đặt sẵn cho nó.Các chiến thuật thi đấu ,đường chạy của robot là do người lập trình quyết định .Hai ngôn ngữ được sử dụng nhiều trong robocon đó là C và ASM .Tuy nhiên ,ngôn ngữ C được sử dụng nhiều nhất vì C có cấu trúc dễ debug lỗi và đặc biệt thuận tiện



khi làm những chương trình lớn (Các chương trình trong robocon rất phức tạp).

Để một robot có thể hoạt động tốt ,cần phải có sự phối hợp chặt chẽ giữa các phần cơ khí ,mạch ,và lập trình .Nếu một bộ phận không tốt ,thì robot sẽ không hoạt động được.

## **1.2.Một số kinh nghiệm khi tham gia robocon.**

### **1.2.1.Tổ chức đội hình.**

Một đội robot bao gồm 7 đến 10 người ,và thường được chia thành 3 nhóm

-Nhóm cơ khí :Thiết kế ,gia công cơ khí (3-4 người).

-Nhóm mạch điện tử :Thiết kế ,làm mạch cho robot (2-3 người).

-Nhóm lập trình: Lập trình ,test robot (3 người)

Người đội trưởng phải có trách nhiệm phân công rõ ràng công việc cho từng người ,các thành viên phải nghiêm túc chấp hành tránh hiện tượng chây ,ý lại.

### **1.2.2.Mua sắm thiết bị linh kiện**

Đội robot phải có định hướng rõ ràng ,làm robot cần có những phương tiện gì ,địa điểm mua hàng cụ thể.

Công cụ cơ khí :Khoan máy ,cưa máy ,cưa tay ,đinh rút : Địa điểm :Chợ giờ (Phố Huế)

Công cụ điện tử :Mỏ hàn ,hút thiếc ,đồng hồ điện tử.v.v.v Địa điểm :Trần Cao Vân ,17 Hàn Thuyên ,70 Hàng Trống ,269 Đội Cấn.

### **1.2.3.Tìm kiếm tài trợ**

Để làm robot ,bạn phải tốn rất nhiều tiền (trung bình từ 20 đến 30 triệu và có thể nhiều hơn nữa) .Với điều kiện sinh viên ,các bạn rất khó có thể có được số tiền như thế ,Do đó ,bạn cần phải tìm kiếm các nhà tài trợ từ các doanh nghiệp.Tuy nhiên để xin được tài trợ ,bạn cần phải có phương án kế hoạch cụ thể ,hay những mối quan hệ khác.

-Địa chỉ một số doanh nghiệp các bạn có thể xin tài trợ

a) Công ty Elcom 18 Nguyễn Chí Thanh

b) Công ty cổ phần kỹ thuật SEEN (Từ Liêm)

c) Công ty Ameco (Tầng 1 toà nhà sông đà Phương Liệt)

## **BÀI 2. MẠCH ĐIỀU KHIỂN ROBOT BẰNG TAY**

Robot điều khiển bằng tay là một phần không thể thiếu được trong robocon .Trong một số trận đấu ,robot bằng tay có vai trò hỗ trợ cho robot tự

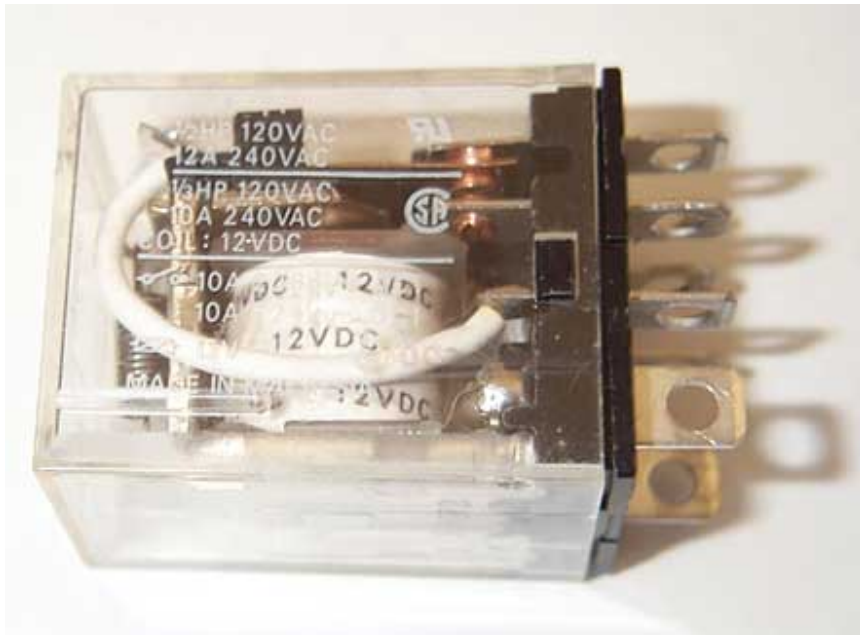
động .Đôi khi ,robot bằng tay đã giúp thay đổi cục diện trên sân.Mạch điều khiển robot bằng tay phần lớn là mạch relay điều khiển động cơ.

### 2.1.Chức năng của mạch điều khiển bằng tay

Mạch điều khiển bằng tay dùng để điều khiển các động cơ của robot bằng tay.Việc điều khiển động cơ bao gồm các chức năng sau

- Điều khiển động cơ quay ngược (robot lùi )
- Điều khiển động cơ quay xuôi (robot tiến )
- Dừng động cơ (dừng robot)

Relay được dùng trong robot bằng tay là loại relay OMRON ,dòng lớn



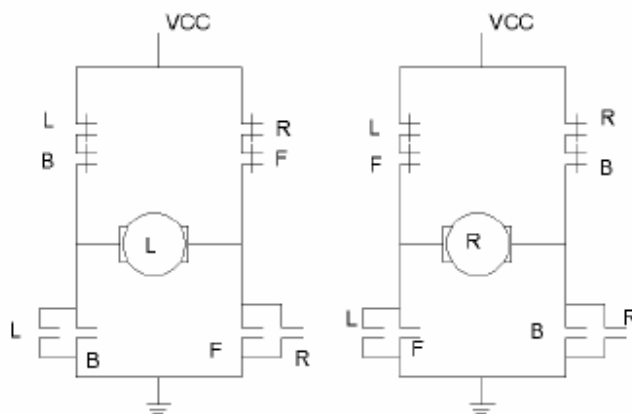
Hình 2.1.Role omron



### 2.2.Nguyên tắc hoạt động mạch điều khiển robot bằng tay

Mạch này sử dụng 2 relay điều khiển 1 động cơ ,cấp nguồn cho relay thứ 1 thì động cơ quay phải ,cấp nguồn cho relay thứ 2 thì động cơ đảo chiều.

Sơ đồ mạch điều khiển bằng tay



Bạn thiết kế một mạch điều khiển sử dụng các nút bấm để đóng mở các relay ,mạch rất đơn giản và hiệu quả.Tuy nhiên ,làm như vậy sẽ không thẩm mỹ và bất tiện .Do đó bạn có thể thiết kế bộ điều khiển bằng gamepad PS 2 để điều khiển .

(Phần này có tham khảo tài liệu trên mạng)

Gamepad cho Robot điều khiển bằng tay chia làm 2 loại : điều khiển từ xa bằng hồng ngoại và điều khiển qua dây cáp. Nếu dùng các mạch thu phát hồng ngoại thông thường thì tính ổn định khi thi đấu không cao (!), tốt nhất nên dùng Vi xử lí (VXL) để thu phát theo tần số riêng của mình (như POL chẳng hạn). Điều khiển qua dây cáp mang tính ổn định rất cao, ở đây chúng tôi đề cập tới cách sử dụng gamepad qua dây cáp.

### **1.Sử dụng gamepad playstation của SONY.**

Kết nối trực tiếp vào Vi xử lí. Rất gọn khi thiết kế phần cứng, tuy nhiên chúng ta nên kèm theo nút Reset VXL trên gamepad để tăng tính ổn định. Đây là phương pháp sử dụng của đội FXR.

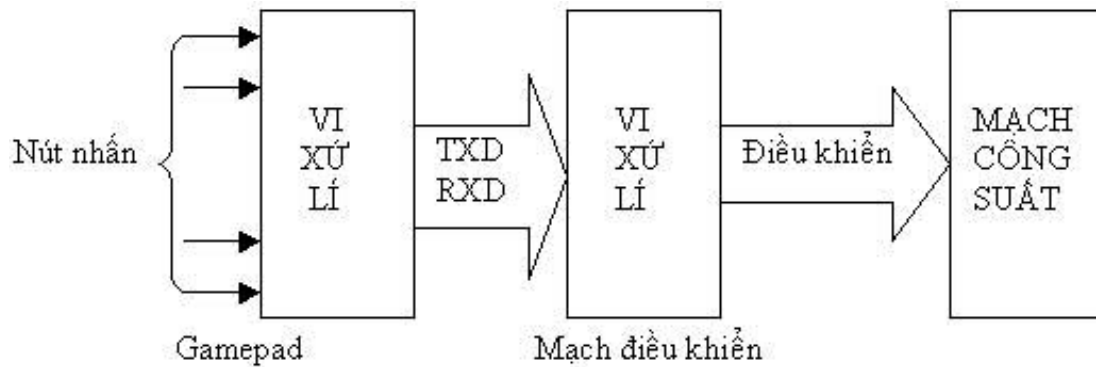
### **2.Cải tạo phần mạch của 1 gamepad bất kì:**

Trong mạch của các gamepad bất kì, các nút nhấn được thiết kế như là các công tắc hành trình (tức là có 1 điểm chung nối mass).

+ Cắt bỏ đường mạch in nối các nút nhấn với phần điều khiển của gamepad.

+ Sử dụng các nút nhấn như là các công tắc hành trình độc lập.

#### **a. Truyền nối tiếp :**

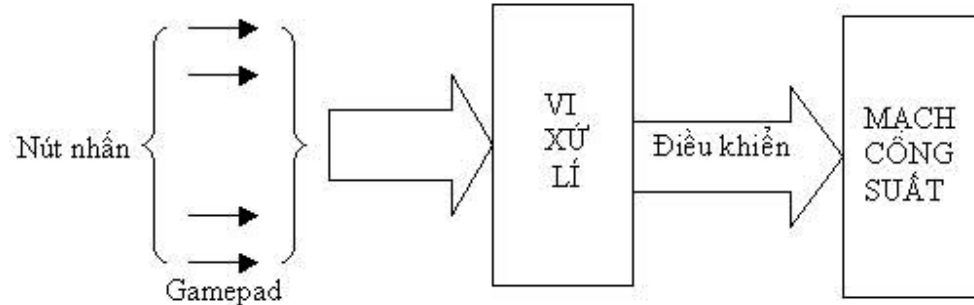


Theo cách xử lý này chúng ta chỉ sử dụng cáp 4 sợi : +5v, mass, TXD, RXD.

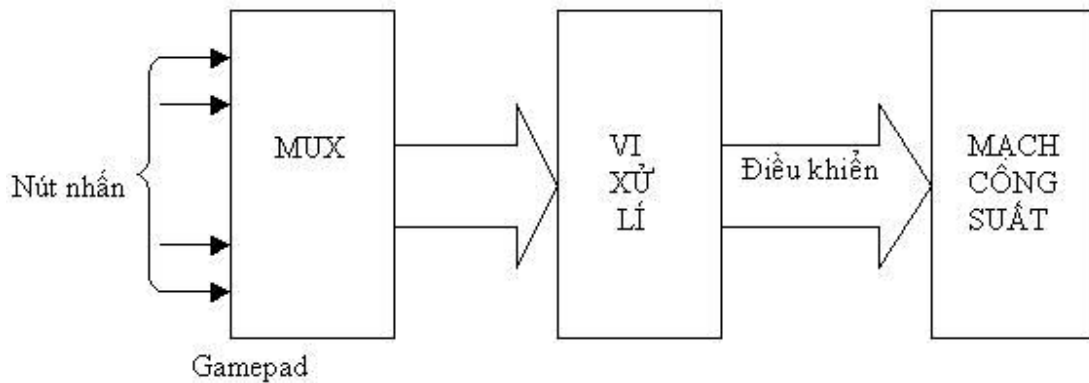
Cách xử lý này vẫn chưa ổn định tuyệt đối nhưng sử dụng ít Port VXL. Nên kèm theo nút Reset VXL trên gamepad để tăng tính ổn định. Đây là phương pháp Basic Bot sử dụng năm 2003 và 2004.

b. Truyền song song :

Thực chất ta nối tất cả các nút nhấn xuống thẳng mạch điều khiển, khi đó các nút nhấn đóng vai trò như là những công tắc hành trình. Phương pháp này rất ổn định, tuy nhiên nhược điểm là rất tốn Port VXL (mỗi nút nhấn ứng với 1 Port).



Cách khắc phục nhược điểm trên :  
Sử dụng mạch dồn kênh 16 sang 1.



Phương pháp này giảm bớt được đáng kể số lượng Port VXL, tính ổn định cao.

Thiết kế mạch robot bằng tay kiểu bán tự động đã được FXR sử dụng trong cuộc thi robocon 2004 rất thành công (đoạt chức vô địch)

## BÀI 3: MẠCH CẢM BIẾN DÒ ĐƯỜNG

Bài học này giới thiệu về mạch cảm biến trong robocon. Mạch cảm biến đóng vai trò như “mắt” của robot giúp cho robot có khả năng nhận biết được môi trường xung quanh (vạch trắng, chướng ngại vật) để từ đó có biện pháp xử lý.

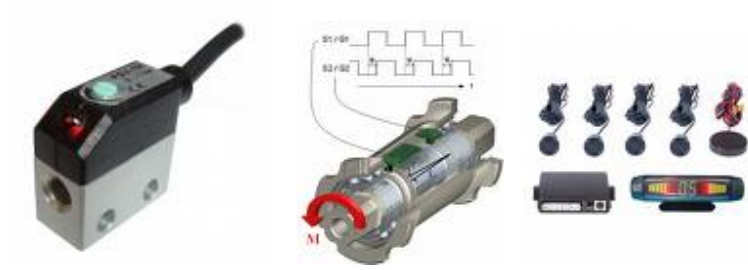
Trong mạch cảm biến, các bạn có thể sử dụng các loại led thu phát hồng ngoại hoặc quang trở. Led thu phát hồng ngoại có giá thành rẻ (2500 /cặp).

Hình 2.1. Sensor hồng ngoại.



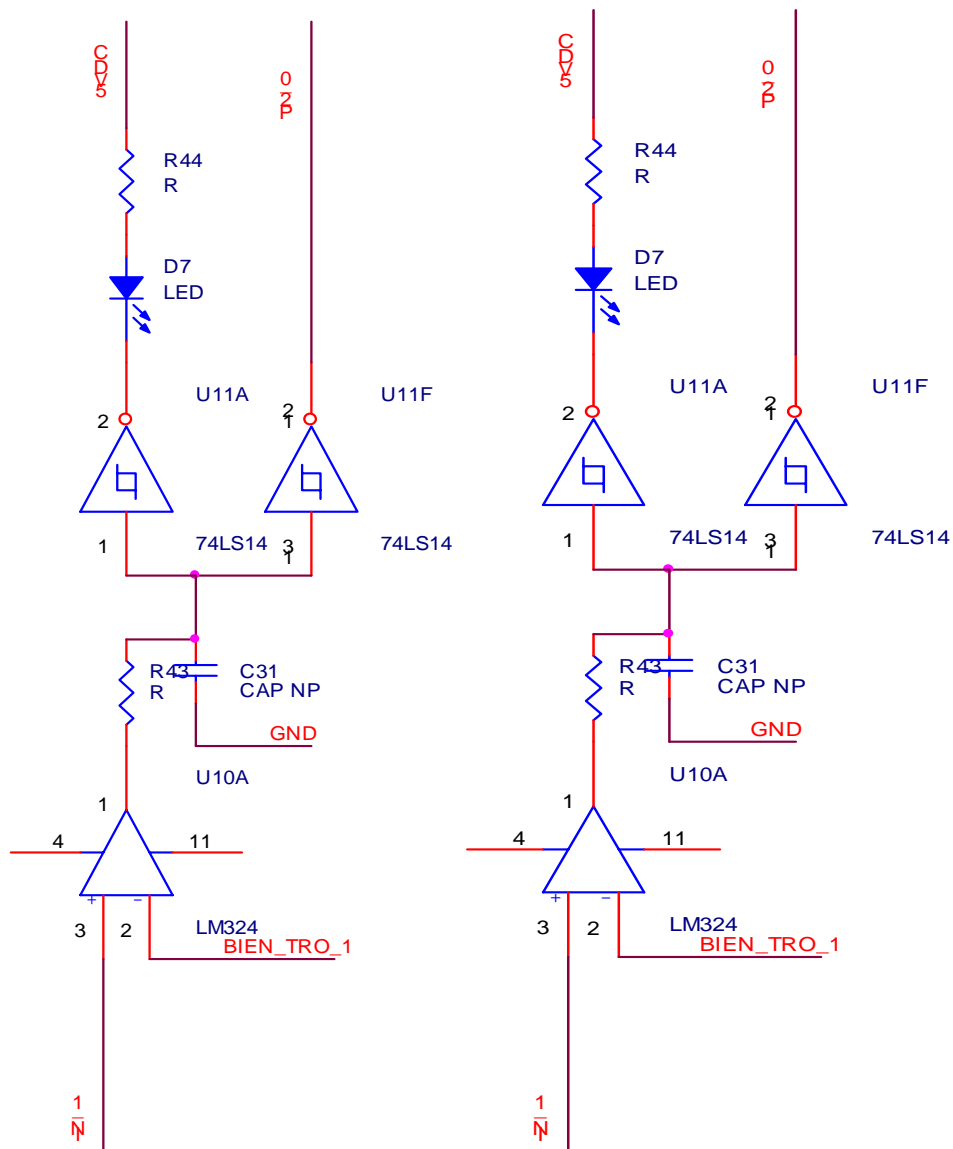
Đối với robot công nghiệp, bạn có thể sử dụng những loại cảm biến công nghiệp của các hãng Omron, hay Siemens. Các loại cảm biến này có độ nhạy cao và khả năng chống nhiễu tốt. Tuy nhiên giá thành rất đắt (hàng trăm đô), cho nên sử dụng chúng trong robocon thì không phù hợp. Bạn hoàn toàn có thể tự chế tạo mạch sensor cho mình.

Hình 2.2. Một số sensor dùng trong công nghiệp

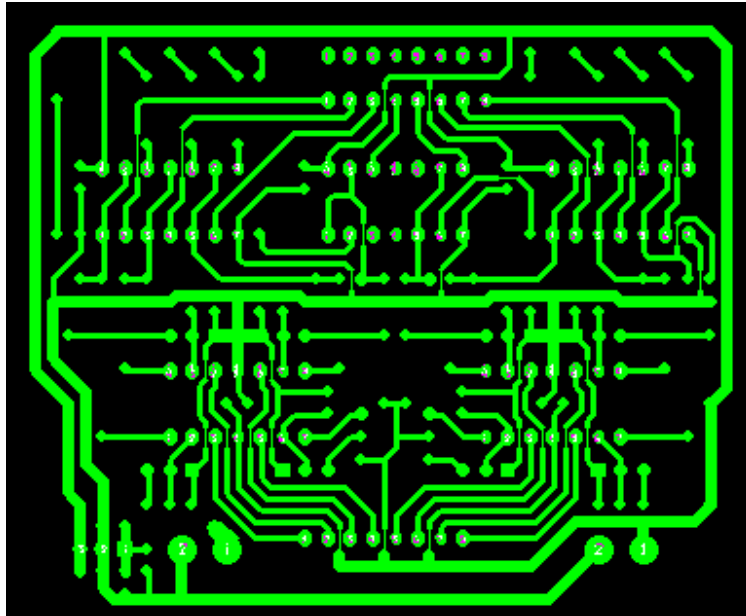


### 3.1. Mạch sensor phát thẳng.

### 3.1.1. Sơ đồ nguyên lý



### 3.1.2. Sơ đồ mạch in



### 3.1.3. Nguyên lý hoạt động.

Trong sơ đồ nguyên lý trên ,cặp led thu phát được đặt sát nhau ,một chân của led thu được đưa và LM324 .LM324 là IC khuếch đại so sánh ,mục đích khuếch đại tín hiệu từ sensor .Tín hiệu ra từ LM324 được đưa và IC 7414 (Trigger inverter) để đảo mức tín hiệu ,tín hiệu ra được đưa vào chân của vi điều khiển.Chiết áp đóng vai trò chỉnh độ nhạy của sensor ,đèn led có tác dụng báo hiệu khi sensor gặp vạch trắng.

Khi hoạt động ,led phát chiếu tia hồng ngoại xuống sân thi đấu ,khi chùm tia hồng ngoại chiếu xuống nền đen , led phát không nhận được chùm phản xạ nên không dẫn ,điện trở bằng vô cùng ,tín hiệu ra LM324 là mức 1 ,qua 7414 bị đảo mức tín hiệu về 0 (led báo sang ,tín hiệu vào vi điều khiển là mức 0).Tương tự ,khi robot gặp nền trắng ,mức tín hiệu ngược lại ,led báo tắt ,tín hiệu vào vi điều khiển mức 1.

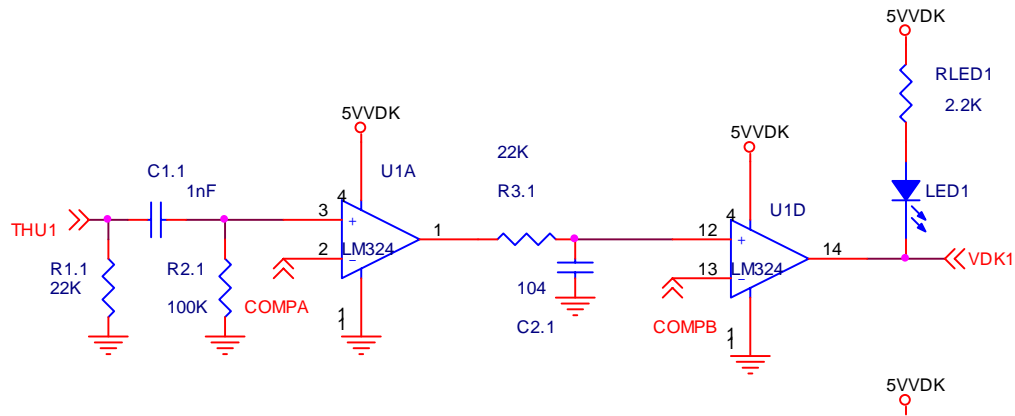
### 3.1.4. Ưu nhược điểm

Mạch sensor phát thẳng có ưu điểm là đơn giản ,dễ chế tạo ,tiết kiệm chi phí .Tuy nhiên ,nhược điểm của mạch sensor phát thẳng là khả năng chống nhiễu kém .Khi trên sân thi đấu ,ánh sáng đèn cao áp sẽ ảnh hưởng đến khả năng hoạt động của sensor .Do đó ,sensor phải được che chắn kỹ lưỡng để khỏi ảnh hưởng đến khả năng hoạt động của sensor (đặt trong hộp kín ,hoặc bọc bằng dính đen các sensor).

## 3.2. Mạch sensor phát xung



Đặc điểm của mạch sensor phát xung đó là khả năng chống nhiễu tốt, không bị ảnh hưởng bởi các điều kiện bên ngoài như ánh sáng. Tuy nhiên, mạch phát xung có nhược điểm là phức tạp, không tiện lợi



Trên thực tế, trong các cuộc thi robocon đa số các đội sử dụng mạch sensor phát xung. Bởi vì mạch sensor phát xung tương đối đơn giản, dễ sử dụng

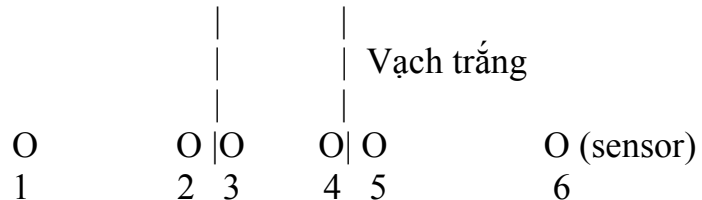
### 3.3. Bố trí sensor

Trên robot, về nguyên tắc, sử dụng nhiều sensor thì khả năng bám đường càng tốt. Tuy nhiên, trên thực tế bạn chỉ cần sử dụng 6 đến 8 sensor là

đủ. Cách bố trí sensor sẽ tùy thuộc vào ý tưởng và chất lượng các loại động cơ bạn sử dụng trong robot.

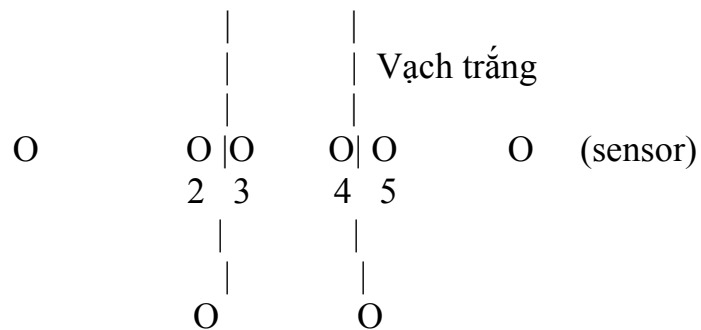
Xin giới thiệu 2 phương pháp bố trí sensor trên robot

a) Sử dụng 6 sensor xếp thành hàng ngang.



Cách này thích hợp đối với việc sử dụng các loại động cơ có độ hãm tốt, quay ngã tư quay bằng 1 bánh (bánh quay, bánh dừng). Trong cách bố trí này, sensor 2, 3, 4, 5 được sử dụng để bám đường, hai sensor ngoài cùng 1 và 6 được dùng để đếm ngã tư và quay phải, quay trái. Phương pháp bố trí sensor này đã được FXR sử dụng năm 2004 và đã đoạt chức vô địch.

a) Sử dụng 8 sensor

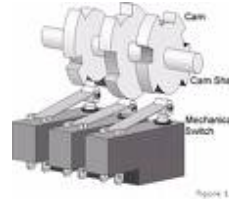


Cách này thường được áp dụng với các loại động cơ có độ hãm kém, quay ngã tư bằng 2 bánh. (2 bánh quay ngược chiều nhau). Ở sơ đồ trên, 2 sensor được lắp ở phía dưới để của robot để bắt ngã tư, còn 6 sensor phía trên dùng để bám đường. Đây là cách mà BK-FIRE sử dụng trong cuộc thi robocon năm 2005.

### 3.4. Công tắc hành trình

Công tắc hành trình được sử dụng khá nhiều trong robocon. Công tắc hành trình thường được lắp đặt ở các cơ cấu phía trên, như cơ cấu truyền động, gắp quà, bỏ bóng.

Hình 3.3. Công tắc hành trình



Công tắc hành trình có 3 chân : Ở dạng thường đóng ,thường mở. Một chân tín hiệu được nối đất hoặc VCC , một chân được nối với vi điều khiển.

Trong robocon ,việc bố trí công tắc hành trình cần phải linh hoạt ,khéo léo để phát huy tối đa khả năng nhận biết của robot.

### 3.5.Encoder

Một công nghệ mới đã được một số đội robot áp dụng trong những năm gần đây là sử dụng các bộ encoder .

Cơ sở của phương pháp này là sử dụng đặc tính độ dài quãng đường đi được và sai số chuyển động giữa hai bánh của động cơ nhờ khai thác tính chất của bộ mã hoá xung vòng quay (Rotary encoder).



Bộ encoder thực chất là một đĩa có đục lỗ có gắn cặp sensor thu phát ở 2 bên .Bộ encoder được gắn trên bánh xe hoặc trên động cơ của robot

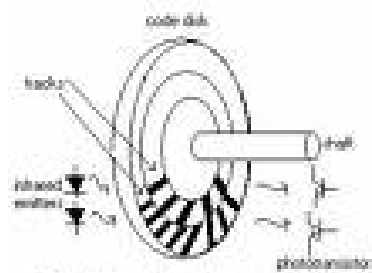
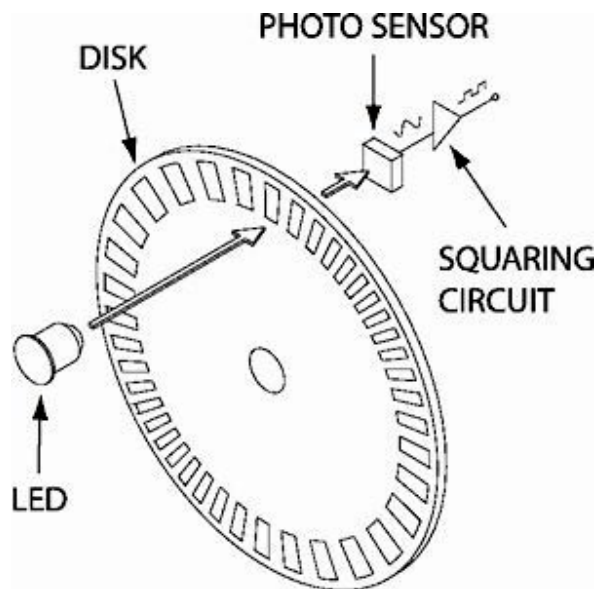


Fig 1. là sơ đồ optical encoder

### Nguyên tắc hoạt động của encoder

Gồm một bộ thu phát hồng ngoại và một đĩa cho chia lỗ được đặt giữa hệ thống thu phát này. Đĩa được gắn trên trục của động cơ hoặc trục chuyển

động. Quá trình đĩa chuyển động làm cho phần photo sensor thay đổi trạng thái và tạo ra một chuỗi các xung vuông trên đầu ra. Đây là thông số kỹ thuật quan trọng của một encoder. Tùy theo số lỗ trên đĩa mà số xung tạo ra trong một vòng quay của đĩa khác nhau. Số lượng xung càng lớn nghĩa là số lỗ càng nhiều trên một vòng tròn 360. Nghĩa là ta càng có thể điều khiển chính xác. Và dĩ nhiên bộ encoder càng đắt tiền. Chúng ta có thể thấy có nhiều loại encoder dùng từ trường hoặc trên đĩa có nhiều vòng lỗ nhưng tôi giới thiệu loại phổ dụng và đơn giản nhất là sử dụng ánh sáng như trên. Trong thực tế chúng ta có thể thấy các bộ encoder trên các động cơ DC chứ không chỉ là các thành phần độc lập. Việc lựa chọn và sử dụng hai loại encoder này đều có những ưu và nhược điểm riêng mà tôi không đề cập ở đây. Hình dạng các encoder :



Khi robot muốn dịch chuyển theo một quỹ đạo xác định cần căn cứ vào hai trạng thái cơ bản đó là quãng đường đã đi được của hai bánh (tham chiếu với chương trình, và hướng chuyển động), Sai số quãng đường giữa hai động cơ.

Phương pháp này thể hiện đặc tính rất mềm dẻo khi phải chuyển hướng vì có thể xoay robot một góc bất kì ra khỏi quỹ đạo chính.

Encoder thường có 6 dây (hoặc 4 dây tùy loại ) bao gồm 2 dây nguồn ,2 dây tín hiệu A và B và dây pha Z.2 dây tín hiệu A và B cho phép bạn xác định số vòng quay của động cơ , vận tốc và chiều quay của động cơ.Để lập trình xử l tín hiệu encoder ,bạn có thể nối 2 dây tín hiệu A và B vào 2 chân timer hoặc ngắt ngoài của vi điều khiển,thiết lập vi điều khiển ở chế độ counter ,vi điều khiển sẽ đếm xung từ vi điều khiển.

Bạn có thể tham khảo code dưới đây

```
void ngat0(void) interrupt 1 // ngat bo dinh thoi 0
{
    if (kt0==0) // bit nay set de dung trong luc quay va di lap ma
    {
        P1_2=1;
        P1_3=1;
        P1_0=1;
        P1_1=1;
    }

    if (kt0==1) // bit nay xoa de bam xung dinh thoi new
    {
        TR0=0;
        if (dem_trai==tocdo_trai)
        { P1_2=~P1_2;tocdo_trai=10-tocdo_trai;dem_trai=0;}
        dem_trai++;
        TR0=1;
    }

    if (kt0==2) // dung cho bamxung quay
    {
        TR0=0;
        TH0=0xFF;
        TLO=0x47;
        d2++;
        TR0=1;
    }
}
```

```
    }  
        // quay dung bo lap ma  
void quayphai(unsigned char bytecao,unsigned char bytethap)  
    {  
        TMOD=0x55;// dem su kien  
        P1_2=1;  
        P1_3=0;  
        P1_0=0;  
        P1_1=1;  
        delay(500);  
        P1_0=1;  
        kt0=0;// bit nay set de chuong trinh ngat kiem tra lua chon  
        TH0=bytecao;  
        TL0=bytethap; // 230 XUNG  
        TR0=1;  
        while(1)  
            {  
                if(P1_3==1) break;  
            }  
    }  
  
void ngat1(void) interrupt 3 // ngat bo dinh thoi 1  
    {  
        if(kt1==0) // dung de di lap ma  
            {  
                P1_0=1;  
                P1_1=1;  
                P1_2=1;  
                P1_3=1;  
            }  
        if(kt1==1) // bam xung dinh thoi kieu moi  
            {  
                TR1=0;  
                if(dem_phai==tocdo_phai)  
                    { P1_0=~P1_0;tocdo_phai=10-tocdo_phai;dem_phai=0;}  
                dem_phai++;  
                TR1=1;  
            }  
    }
```

```
    }  
    }  
    // quay dung bo lap ma
```

```
void quaytrai(unsigned char bytecao,unsigned char bytethap)  
{  
    TMOD=0x55; //khai dong bo dinh thoi dem su kien  
    kt1=0;  
    P1_0=1;  
    P1_1=0;  
    P1_2=0;  
    P1_3=1;  
    delay(500);  
    P1_2=1;  
    TH1=bytecao;  
    TL1=bytethap;  
    TR1=1;  
    while(1)  
    {  
        if (P1_1==1) break;  
    }  
}
```

```
void quay(unsigned char status ,unsigned char bytecao ,unsigned char  
bytethap)  
{  
  
    if (status==turn_right)  
        quayphai(bytecao,bytethap);  
    if (status==turn_left)  
        quaytrai(bytecao,bytethap);  
}
```

```
void delay (unsigned long time)  
  
{  
    unsigned long i;  
    for (i=0;i<time ;i++)
```

```
    {}  
}
```

*// bam xung bang bo dinh thoi*

```
void quayphaingatu()  
{  
    motor(backward);  
    delay(3500);  
    motor(stop);  
    delay(10000);  
    motor(left_go);  
    delay(4500);  
    bamxung_quay(5,motor_left);  
    motor(stop);  
}
```

```
void quaytraingatu()  
{  
    motor(backward);  
    delay(3500);  
    motor(stop);  
    delay(10000);  
    motor(right_go);  
    delay(4500);  
    bamxung_quay(5,motor_right);  
    motor(stop);  
}
```

## BÀI 4.MẠCH CÔNG SUẤT ĐIỀU KHIỂN ĐỘNG CƠ

### 4.1. Động cơ.

Động cơ được sử dụng trong robocon là loại động cơ 1 chiều (DC motor).

Hình 4.1.Động cơ 1 chiều





Việc lựa chọn động cơ căn cứ vào các tiêu chí sau:

- Tốc độ
- Khả năng chịu tải
- Độ hãm
- Dòng ,áp

-Đối với động cơ dùng cho cơ cấu chuyển động (phần đế của robot) yêu cầu đặt ra là phải có tốc độ nhanh ,và có độ hãm tốt .Động cơ thường được sử dụng ở phần này là loại động cơ pitman.

Ngoài động cơ pitman ,bạn có thể sử dụng các loại động cơ khác ,miễn là đạt được các tiêu chí nêu trên.Bạn có thể sử dụng động cơ vuông tháo bánh răng .Đây là các mà BK-FIRE đã sử dụng trong cuộc thi robocon2005 ,bằng cách này ,robot của BK-FIRE đạt được tốc độ khá cao.



Hình 4.3. Động cơ pitman.

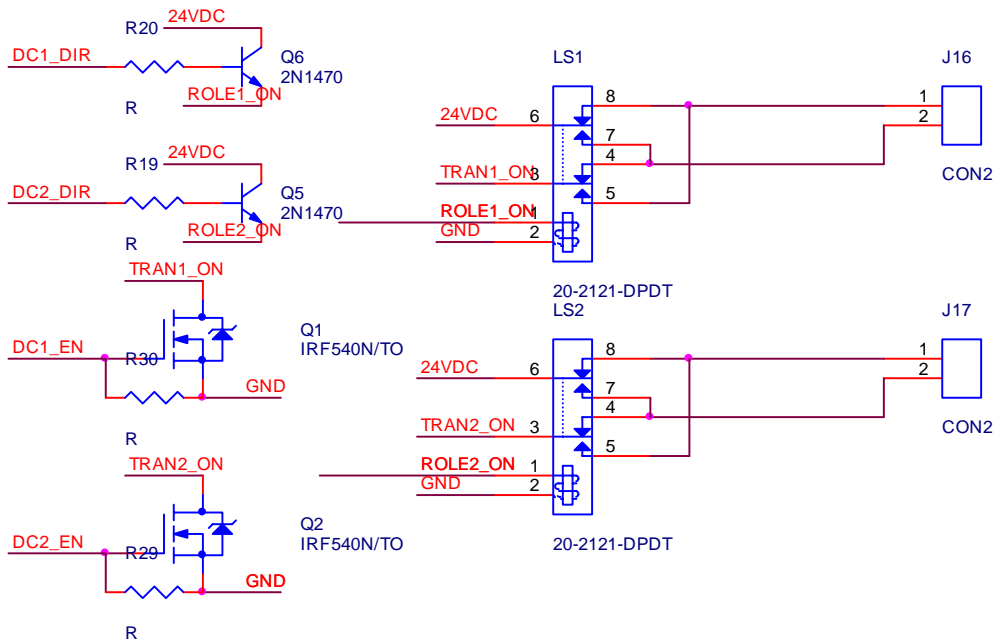
-Đối với động cơ dùng cho cơ cấu nâng hạ ,gấp quà (phần trên ) yêu cầu phải có khả năng chịu tải ,khoẻ .Loại động cơ thường được sử dụng là loại động cơ gạt nước hoặc động cơ có hộp điều tốc .Đặc điểm của loại động cơ này là tốc độ không cao nhưng lực quay rất khoẻ có thể chịu tải lớn.Trong năm 2005 ,các đội robocon BK-Đà Nẵng đã sử dụng động cơ này cho robot trung tâm và đạt hiệu quả rất cao.

-Trong một số trường hợp ,robot cần phải có tốc độ cực nhanh .VD năm 2005 và 2006 ,các robot trung tâm phải có tốc độ đặc biệt nhanh (trong vòng 1,5s-2s phải chiếm lĩnh được đuốc trung tâm).Để giải quyết vấn đề này,một số đội đã sử dụng động cơ đề xe máy cho robot trung tâm.Động cơ đề xe máy có dòng lớn ,và tốc độ rất nhanh.Tuy nhiên với loại động cơ này thì các mạch công suất không thể điều khiển được.Để gia tăng tốc độ ,bạn có thể lắp đặt cơ cấu xích cho robot của mình.

## **4.2.Mạch điều khiển động cơ**

### **4.2.1.Mạch relay**

#### **4.2.1.1.Sơ đồ nguyên lý**



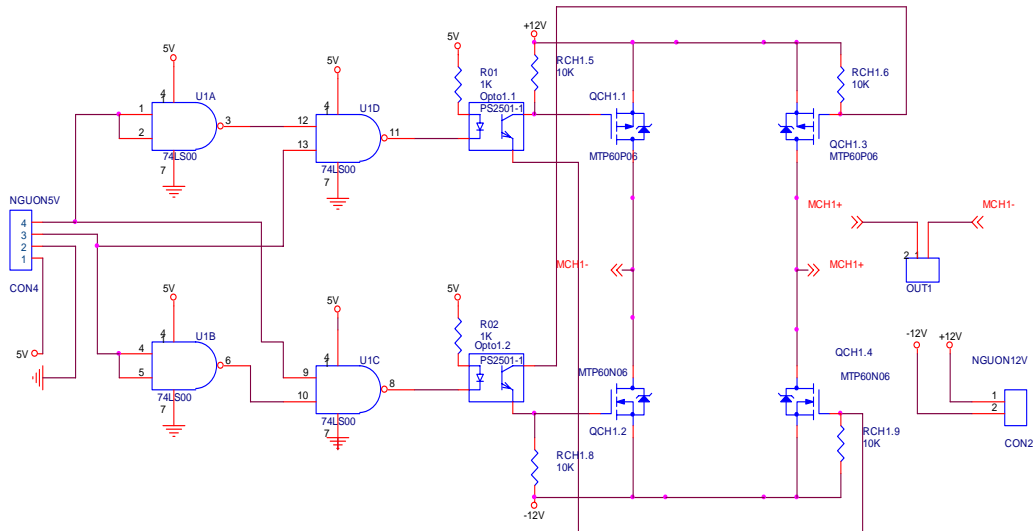
#### 4.2.1.2. Nguyên lý hoạt động.

Loại relay được sử dụng trong mạch robot tự động là loại relay 8 chân (2 tiếp điểm)

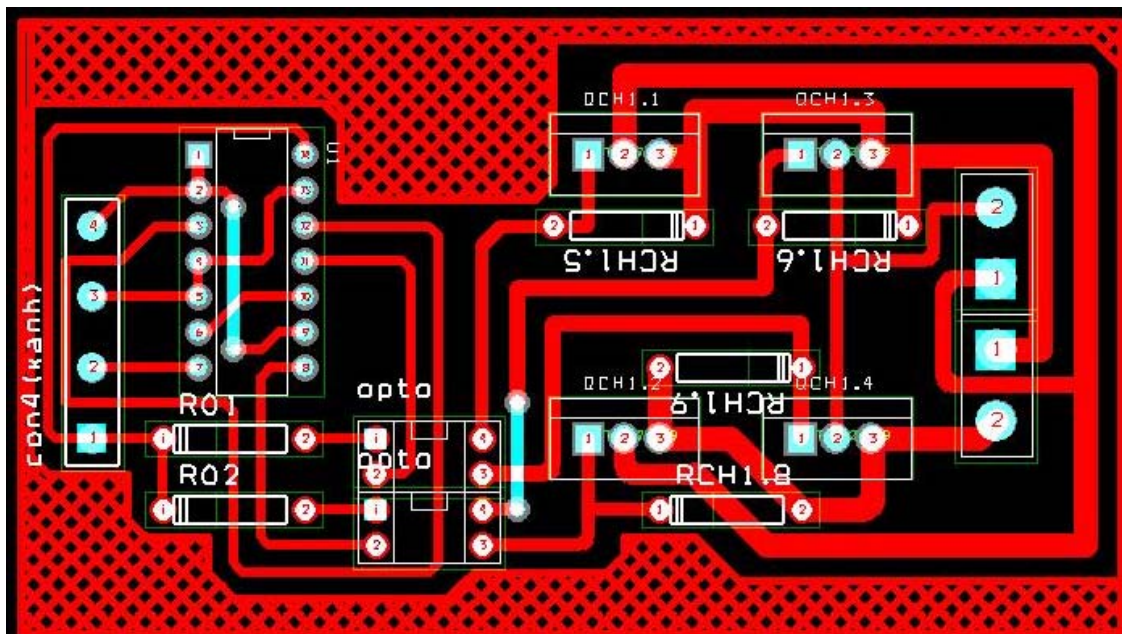
Mạch relay điều khiển động cơ sử dụng 2 tín hiệu điều khiển. Chân DC\_EN và chân DC\_DIR, chân DC\_EN là chân kích FET (đóng mở FET) chân DC\_DIR là chân đảo chiều. Tuy nhiên, trước khi đảo chiều động cơ bạn phải tắt FET trước, nếu không sẽ gây chết FET.

### 4.3. Mạch cầu H điều khiển động cơ

#### 4.3.1. Sơ đồ nguyên lý



#### 4.3.2. Sơ đồ mạch in



#### 4.3.3. Nguyên lý hoạt động

Mạch cầu H điều khiển động cơ sử dụng 4 MOSFET IRF540 để khuếch đại công suất và đóng mở.

Hình 4.4. IRF 540



Mosfet là loại transistor hiệu ứng trường (MOSFET = metal-oxide semiconductor field-effect transistor) MOSFET là mô hình nguồn dòng phụ thuộc áp. Như vậy để kích dẫn MOSFET cần đưa điện áp vào hai cực Vgs. Một điểm khác của MOSFET nữa là MOSFET có Vds không ổn định như Vce (BJT) mà nó phụ thuộc vào giá trị ổn định khác là Rds-on (Ở chế độ đóng ngắt). Như vậy  $Vds = Id * Rds$ . Như vậy công suất tiêu tán trên MOSFET chủ yếu phụ thuộc vào Rds.

Có hai loại MOSFET kênh P và kênh N.

Loại kênh N thường có Rds nhỏ hơn kênh P.

Loại kênh N cần  $Vgs > 0$  để dẫn.

Loại kênh P cần  $Vgs < 0$  để dẫn.

Đa số các MOSFET bắt đầu dẫn khi  $abs(Vgs)$  xấp xỉ 5. Thường giá trị khuyến cáo là 10V. Khi điện áp lớn hơn 15 bắt đầu có nguy cơ đánh thủng Vgs. Khá nguy hiểm là khi bị thủng MOSFET thường dẫn luôn dẫn đến dễ nổ. Trong mô hình của MOSFET có một cái tụ ký sinh từ gọi là Cgs cần quan tâm đến tụ này khi thiết kế mạch lái sử dụng PWM. Khi làm Robot thường dùng con IRF540 để lái động cơ DC hoặc dùng mạch cầu IRF540 để điều khiển hướng (Mặc dù cách điều khiển này rất khó).

Mạch cầu H điều khiển động cơ có 2 tín hiệu điều khiển .2 chân tín hiệu này được nối với 2 chân vi điều khiển.

Giả sử 2 tín hiệu điều khiển là A,B (tương ứng 2 chân P1.0 ,P1.1)

Điều khiển động cơ DC có các trạng thái điều khiển :Động cơ quay ,đảo chiều ,dừng và điều khiển tốc độ động cơ.

Ta có bảng trạng thái

P1.0	P1.1	Trạng thái
1	0	Động cơ quay xuôi
0	1	Động cơ quay ngược
1	1	Dừng động cơ

1	1	Dừng động cơ
---	---	--------------

#### 4.3.4. Mã nguồn điều khiển động cơ

```
#include <at89x52.h>
#include <robot.h>
unsigned char dutycycle=0 ;
    /*
        P2_0 P2_1 : dieu khien dong co nang ha
        P2_2 P2_3 : dieu khien dong co lay bong
    */
void motor( unsigned char status)
{
    switch (status)
    {

        case nang:

            {
                P2_0=1;
                P2_1=0;
                break;
            }

        case ha:
            {

                P2_0=0;
                P2_1=1;
                break;

            }
        case dung:
            {

                P2_0=1;
                P2_1=1;
                break;
            }
    }
}
```

```
void motorkep(unsigned char state)
{

    switch (state)
    {

        case kep:
        {
            P2_2=0;
            P2_3=1;
            break;
        }

        case nha:
        {
            P2_2=1;
            P2_3=0;
            break;
        }

        case phanh :
        {

            P2_2=1;
            P2_3=1;
            break;
        }

    }
}

void delay ( unsigned char time)
{

    unsigned char i;
    for ( i=0; i<time ;i++)
    { }
}

void nangnhanhdan(unsigned char time1,unsigned time2)
{
```

```
unsigned char x,y;

for (x=0;x <time1 ;x++)
{
  for ( y=0;y<time2; y++)
  {
    P2_0=1;
    P2_1=0;
    delay(dutycycle);
    P2_1=1;
    delay(Time-dutycycle);
  }
  dutycycle++;

  if (dutycycle==255) break;;

}

}

void nangchamdan(unsigned char Time1,unsigned Time2)
{

  unsigned char x,y;

  for (x=0;x <Time1 ;x++)
  {
    for ( y=0;y<Time2; y++)
    {
      P2_0=1;
      P2_1=0;
      delay(dutycycle);
      P2_1=1;
      delay(Time-dutycycle);
    }
    dutycycle--;
    if (dutycycle==0) break;

  }

}
```



```
}  
  
void nhabong(unsigned char timedelay)  
{  
    motorkep(nha);  
    delay(timedelay);  
    motorkep(phanh);  
}  
  
void main()  
{  
    // trang thai gap bong  
  
    motorkep(kep);  
    delay(255);  
    motorkep(phanh);  
  
    // co cau nang bong hoat dong  
  
    nanghanhdan(100,100); // chang 1 nang nhanh dan  
  
    // chang 2 : nang het toc luc  
    motor(nang);  
  
    delay(100);  
  
    // chang 3 : nang cham dan  
  
    nangchamdan(100,100);  
  
    motor(dung);  
  
    delay(100);  
    motorkep(phanh);  
    /* trang thai nha bong  
    khi P0 cua slaver nhan tin hieu dieu khien tu master  
    thi trang thai nha bong hoat dong  
    */
```

```
nhabong(100);  
  
while(1)  
{  
    motor(dung);  
    motorkep(phanh);  
}  
// P0=0xFB;  
}  
  
/* bo qua thu 1 */
```

#### 4.3.5. Kỹ thuật điều chế độ rộng xung điều khiển tốc độ động cơ (PWM)

Điều khiển tốc độ động cơ là một kỹ thuật hết sức quan trọng trong việc lập trình robot. Kỹ thuật này giúp bạn có thể điều khiển robot của mình chạy nhanh chậm tùy ý, giúp cho hoạt động của robot luôn linh hoạt, thích ứng với mọi tính huống xảy ra trên sân.

a) Cách tạo xung có độ rộng thay đổi bằng VĐK.

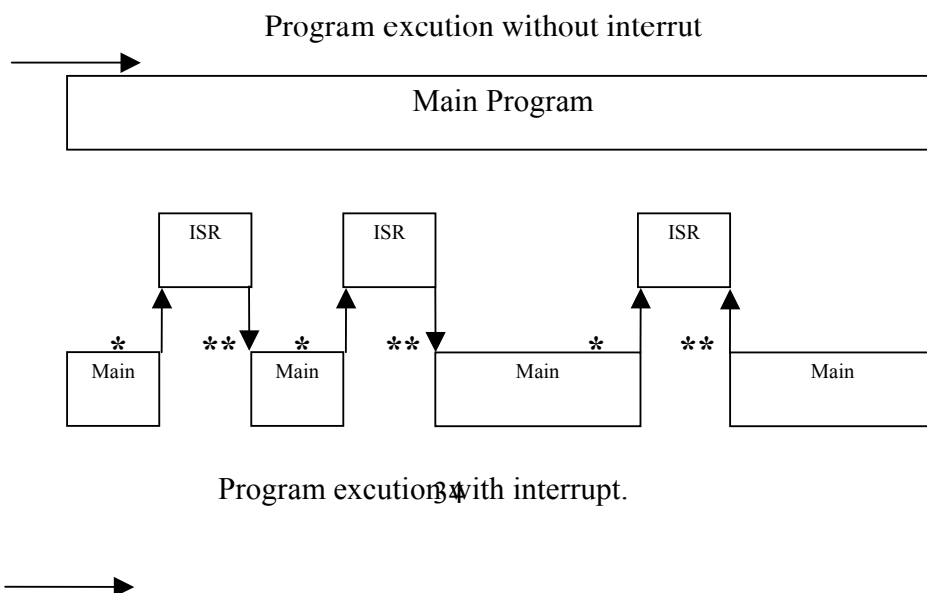
Cách 1: Như các bạn điều khiển nhấp nháy 1 con led, đó là tạo ra 1 xung ở 1 chân của vi điều khiển, nhưng xung đó có độ rộng cố định, tần số lớn, cách bạn có thể điều chỉnh lại hàm delay để tần số của nó đúng 1 Khz. Tuy nhiên vì là dùng hàm delay nên trong thời gian có xung lên 1(5V) và thời gian không có xung(0V) vi điều khiển không làm gì cả, hơn nữa tạo xung bằng việc delay mà các bạn có nhu cầu cần 2 bộ phát xung ở 2 kênh, có cùng tần số mà khác độ rộng xung thì trở nên rất khó khăn. Cho nên chúng ta dùng bộ định thời Timer của vi điều khiển trong trường hợp này rất tiện.

+ Cách 2: Dùng ngắt Timer của bộ vi điều khiển.

Trước hết nhắc lại về ngắt của vi điều khiển:

+ Ngắt là gì? để trả lời câu hỏi này tôi xin trích đoạn về ngắt trong bài 2 ví dụ cho ngắt timer:

#### Timer



Một chương trình chính không có ngắt thì chạy liên tục, còn chương trình có ngắt thì cứ khi nào điều kiện ngắt được đảm bảo thì con trở sẽ nhảy sang hàm ngắt thực hiện xong hàm ngắt lại quay về đúng chỗ cũ thực hiện tiếp chương trình chính. Tôi có 1 ví dụ như sau: Bạn đang ăn cơm, có tiếng điện thoại, bạn đặt bát cơm ra nghe điện thoại, nghe xong lại quay về bưng bát cơm lên ăn tiếp. Thì quá trình ăn cơm của bạn là chương trình chính, có điện thoại gọi đến là điều kiện ngắt, bạn ra nghe điện thoại là thực hiện chương trình ngắt(Interrupt Service Routine), quay về ăn cơm tiếp là tiếp tục thực hiện chương trình chính.

Ngắt đối với người mới học vi điều khiển là rất khó hiểu, vì đa số các tài liệu đều không giải thích ngắt để làm gì. Có nhiều loại ngắt khác nhau nhưng tất cả đều có chung 1 đặc điểm, ngắt dùng cho mục đích đa nhiệm. Đa tức là nhiều, nhiệm tức là nhiệm vụ. Thực hiện nhiều nhiệm vụ.

Các bạn nhìn vào tiến trình của hàm main với chương trình có ngắt :

Chương trình chính đang chạy, ngắt xảy ra, thực hiện hàm ngắt rồi quay lại chương trình chính. Chương trình trong vi điều khiển khác với ví dụ ăn cơm nghe điện thoại của tôi ở chỗ, thời gian thực hiện hàm chính là rất lớn, thời gian thực hiện hàm ngắt là rất nhỏ, cho nên thời gian thực hiện hàm ngắt không ảnh hưởng gì đến thời gian thực hiện hàm chính. Như vậy trong hàm ngắt các bạn làm 1 việc, trong hàm chính của bạn làm 1 việc như vậy coi như các bạn làm được 2 việc(đa nhiệm) trong 1 quãng thời gian tương đối ngắn cỡ mS, chứ thực ra tại 1 thời điểm vi điều khiển chỉ thực thi 1 lệnh.

Ví dụ : Bạn thử nghĩ xem làm thế nào để vừa điều chế xung PWM để điều chỉnh tốc độ động cơ, vừa đọc các cảm biến đầu vào mà tốc độ động cơ phụ thuộc đầu vào cảm biến.

Vậy ngắt là 1 điều kiện nào đó xảy ra ngẫu nhiên mà vi điều khiển có thể biết do phần cứng của vi điều khiển, rồi ta căn cứ vào đó để lập trình.

\* Ví dụ: Với ngắt bộ định thời timer, hay bộ đếm counter là khi tràn bộ đếm thì phần cứng của vi điều khiển sẽ báo có ngắt xảy ra và nhảy đến chương trình phục vụ ngắt( ISR\_ Interrupt Service Routine) 1 cách tự động.

Với ngắt ngoài, chân P3.2 chẳng hạn, nếu ta khai báo trước chân sử dụng chân P3.2 sử dụng cho ngắt ngoài chứ không phải sử dụng cho mục

đích IO thì cứ khi cú 1 xung xuất hiện từ mạch ngoại vi truyền vào chõn P3.2 thì phần cứng của vi điều khiển nhận ra và chuyển tới chương trình phục vụ ngắt.

Với ngắt nối tiếp thì cứ khi có kí tự truyền từ máy tính xuống vi điều khiển thì sẽ có hiện tượng ngắt xảy ra.

- Hàm ngắt:

Cấu trúc:

```
Void Tênhàm(void) interrupt nguồnnắt using bắngthanhghi  
{  
// Chuong trinh phục vụ ngắt o đây  
}
```

Chỳ ý về hàm ngắt:

- + Hàm ngắt không được phép trả lại giá trị hay truyền biến vào hàm.
- + Tên hàm bất kì.
- + interrupt là từ khóa phân biệt hàm ngắt với hàm thường.
- + Nguồn ngắt từ 0 tới 5 theo bảng vector ngắt.
- + Bảng thanh ghi trên ram chọn từ 0 đến 3.

Tựy theo bạn viết hàm ngắt cho nguồn nào bạn chọn nguồn ngắt từ bảng sau:

<b>Ngắt do</b>	<b>Cờ</b>	<b>Địa chỉ vector</b>
<b>Reset hệ thống</b>	<b>RST</b>	<b>0000H</b>
<b>Ngắt ngoài 0</b>	<b>IE0</b>	<b>0003H</b>
<b>Bộ định thời 0</b>	<b>TF0</b>	<b>000BH</b>
<b>Ngắt ngoài 1</b>	<b>IE1</b>	<b>0013H</b>
<b>Bộ định thời 1</b>	<b>TF1</b>	<b>001BH</b>
<b>Port nối tiếp</b>	<b>RI hoặc TI</b>	<b>0023H</b>
<b>Bộ định thời 2</b>	<b>TF2 hoặc EXF2</b>	<b>002BH</b>

Riêng ngắt Reset không tính, bắt đầu đếm từ 0 và từ ngắt ngoài 0. Ví dụ: tôi cần viết hàm ngắt cho bộ định thời timer 1 hàm ngắt sẽ là.

```
void timer1_isr(void) interrupt 3 using 0
```

```
{  
// Lenh can thực hiện.  
}
```

- Về using 0: Có 4 bảng thanh ghi bạn có thể chọn cho chương trình phục vụ ngắt, cái này cũng không quan trọng. Trong hàm ngắt các bạn có thể bỏ đi từ using 0, khi đó vi điều khiển sẽ tự sắp xếp là dùng bảng thanh ghi nào.

- Hàm ngắt khác hàm bình thường chỗ nào. Hàm bình thường ví dụ hàm delay, cứ khi bạn gọi nó thì nó sẽ được thực hiện, có nghĩa là nó có vị trí cố định trong tiến trình hàm main, có nghĩa là bạn biết nó xảy ra khi nào. Còn hàm ngắt thì không có tiến trình cố định, điều kiện ngắt có thể xảy ra bất kì lúc nào trong tiến trình hàm main và cứ khi nào có điều kiện ngắt thì hàm ngắt sẽ được gọi tự động.

- Để sử dụng ngắt ta phải làm các công việc sau:

1) Khởi tạo ngắt: dùng ngắt nào thì cho phép ngắt đó hoạt động bằng cách gán giá trị tương ứng cho thanh ghi cho phép ngắt IE( Interrupt Enable):

EA	ET2	ES	ET1	EX1	EX0	ET0
Điều khiển các nguồn ngắt						
IE		(0: không cho phép; 1: cho phép)				
IE.7	EA	Cho phép/ không cho phép toàn cục				
IE.6	---	Không sử dụng				
IE.5	ET2	Cho phép ngắt do bộ định thời 2				
IE.4	ES	Cho phép ngắt do port nối tiếp				
IE.3	ET1	Cho phép ngắt cho bộ định thời 1				
IE.2	EX1	Cho phép ngắt từ bên ngoài (ngắt ngoài 1)				
IE.1	EX0	Cho phép ngắt từ bên ngoài (ngắt ngoài 0)				
IE.0	ET0	Cho phép ngắt do bộ định thời 0				

IE là thanh ghi có thể xử lý từng bit. Ví dụ : bạn muốn cho phép ngắt timer 1 bạn dùng lệnh: ET1=1; Không cho phép nữa bạn dùng lệnh : ET1=0; Hoặc bạn có thể dùng lệnh IE= 0x08; thì bit 3 của thanh ghi IE tức(IE) sẽ lên 1. Nhưng cách thứ nhất tiện hơn.

2) Cấu hình cho ngắt: Trong 1 ngắt nó lại có nhiều chế độ ví dụ: với ngắt timer. Bạn phải cấu hình cho nó chạy ở chế độ nào, chế độ timer hay counter, chế độ 16 bit, hay 8 bit,... bằng cách gán các giá trị tương ứng cho thanh ghi TMOD( Timer MODE).

TMOD		Chọn model cho bộ định thời 1			
7	GATE	Chọn model cho bộ định thời 1			
6	C/T	Bit chọn chức năng đếm hoặc định thời:			
5	M1	Bit chọn chế độ thứ nhất			
4	M0	Bit chọn chế độ thứ 2			
		M1	M0	Chế độ	Chức năng
		0	0	0	Chế độ định thời 13 bit
		0	1	1	Chế độ định thời 16 bit

		<b>1</b>	<b>0</b>	<b>2</b>	<b>Chế độ tự động nạp lại 8 bit</b>
		<b>1</b>	<b>1</b>	<b>3</b>	<b>Chế độ định thời chia xẻ</b>
<b>3</b>	<b>GATE</b>	<b>Bit điều khiển công cho bộ định thời 0</b>			
<b>2</b>	<b>C/T</b>	<b>Bit chọn chức năng đếm / định thời cho bộ định thời 0</b>			
<b>1</b>	<b>M1</b>	<b>Bit chọn chế độ thứ nhất cho bộ định thời 0</b>			
<b>0</b>	<b>M0</b>	<b>Bit chọn chế độ thứ 2 cho bộ định thời 0</b>			

Ví dụ tôi cấu hình cho bộ định thời 1 chế độ timer, với bộ đếm 8 bit tự động nạp lại(auto reload) dùng lệnh sau: TMOD=0x20.

Các bạn đừng lo vì việc phải nhớ bảng thanh ghi này, các bạn không phải nhớ nói trắng ra như vậy, chuyển sang phần lập trình các bạn sẽ được hướng dẫn làm thế nào để không phải nhớ, nhưng chỉ lập trình với C mới làm được còn lập trình Asem thì bắt buộc phải nhớ .

### 3) Bắt đầu chương trình có ngắt:

-Trước khi bắt đầu cho chạy chương trình ta phải cho phép ngắt toàn cục được xảy ra bằng cách gán EA(Enable All interrupt) bằng 1, thì ngắt mới xảy ra.

-Thường thì ngay vào đầu chương trình(hàm main) trước vòng while(1) chúng ta đặt công việc khởi tạo, cấu hình và cho phép kiểm tra ngắt. Ví dụ với bộ định thời timer ta gán các giá trị phù hợp cho thanh ghi TCON( Timer CONTROL).

TCON		Điều khiển bộ định thời
TCON.7	TF1	Cờ tràn của bộ định thời 1. Cờ này được set bởi phần cứng khi có tràn, được xoá bởi phần mềm, hoặc bởi phần cứng khi bộ vi xử lý trở đến trình phục vụ ngắt
TCON.6	TR1	Bit điều khiển hoạt động của bộ định thời 1. Bit này được set hoặc xoá bởi phần mềm để điều khiển bộ định thời hoạt động hay ngưng
TCON.5	TF0	Cờ tràn của bộ định thời 0
TCON.4	TR0	Bit điều khiển hoạt động của bộ định thời 0
TCON.3	IE1	Cờ ngắt bên ngoài 1 (kích khởi cạnh). Cờ này được set bởi phần cứng khi có cạnh âm (cuông) xuất hiện trên chân INT1, được xoá

		bởi phần mềm, hoặc phần cứng khi CPU trở đến trình phục vụ ngắt
TCON.2	IT1	Cờ ngắt bên ngoài 1 (kích khởi cạnh hoặc mức). Cờ này được set hoặc xoá bởi phần mềm khi xảy ra cạnh âm hoặc mức thấp tại chân ngắt ngoài
TCON.1	IE0	Cờ ngắt bên ngoài 0 (kích khởi cạnh)
TCON.0	IT0	Cờ ngắt bên ngoài 0 (kích khởi cạnh hoặc mức)

Ví dụ để chạy bộ định thời timer 1 ta dùng câu lệnh: TR1=0;

TR1(Timer Run 1). Còn bạn nào thích khó thì: TCON=0xxx;

Còn các loại ngắt khác quá trình tương tự, đây là khóa học cơ bản chỉ làm việc với ngắt timer, trong khóa nâng cao sẽ có các ngắt còn lại, tuy nhiên làm việc được với ngắt timer thì các ngắt khác các bạn cũng có thể làm tương tự, các bạn làm đến ngắt nào thì dùng tài liệu tra bảng thanh ghi của ngắt đó. Tài liệu tôi sẽ gửi cùng bài này.

- Quay trở lại bài học:

Sau khi khởi tạo xong và cho ngắt timer 1 chạy thì điều gì xảy ra?

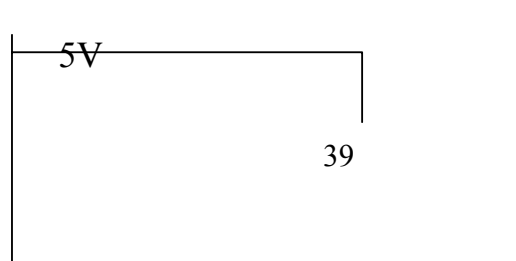
Khi bắt đầu cho timer 1 chạy thì bộ đếm của timer sẽ đếm dao động của thạch anh, cứ 12 dao động của thạch anh(1 chu kì máy), bộ đếm của timer 1 TL1(Timer Low1) sẽ tăng 1, có thể nói timer 1 đếm số chu kì máy. Đối với chế độ 8 bit.

TL1 là 1 thanh ghi 8 bit, là bộ đếm của bộ định thời rồi. Nó đếm được từ 0, đến 255. Nếu nó đếm đến 256 thì bộ đếm tràn, TL1 quay vòng lại bằng 0, và cờ ngắt TF1(Timer Flag 1) tự động được gán lên 1(bảng phần cứng của vi điều khiển) như 1 công tắc tự động bật, và ngắt xảy ra.

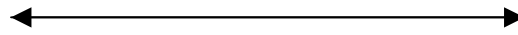
Còn với chế độ 16 bit, bộ đếm của bộ định thời còn 1 thanh ghi 8 bit nữa là TH1(Timer high 1), nếu cấu hình cho timer 1 hoạt động ở chế độ 16 bit thì khi TL1 tràn nó sẽ đếm sang TH1(TH1 sẽ tăng 1). Như vậy ta có thể đếm:  $2^{16}$  chu kì máy( 2 thanh ghi 8+8=16 bit).

Chú ý là khi bộ đếm tràn ngắt sẽ xảy ra. Nếu ta cần đếm 256 chu kì máy thì khi khởi tạo ta cho TL1=0; , còn nếu không muốn đếm 256 chu kì máy mà ta chỉ cần đếm 100 thôi ngắt đã xảy ra rồi thì ta phải làm như sau:  $256-100 = 156$ ; và khi khởi tạo ta gán : TL1=155; vì đếm từ 155 đến 255 là đủ 100 lần thì ngắt xảy ra.

Với yêu cầu của bài. Tạo xung tần số 1Khz  $\rightarrow$  Chu kì =  $1/10^3 = 0,001$  giây= 1 mili giây=1000 uS= 1000 chu kì máy. Với 10 cấp tốc độ, tức là bạn phải tạo ra được xung 10%, 20%, 30%, 40%, ..., 90%, 100%. 1 xung như sau:



0V



T : Chu kì  
1000 micro giây.

Khoảng thời gian xung kéo dài 5V là T1. Xung 10% tức là  $T1/T = 10\% = 1/10$ . Xung 20%  $T2/T = 2/10$ ... PWM (Thay đổi độ rộng xung)  
Bây giờ tôi mới xin nói về phần 2.

#### 4.3.5. Nguyên lý của PWM

- Xung PWM: Đưa ra mở transistor, xung với độ rộng lớn hơn transistor sẽ mở lâu hơn động cơ sẽ quay nhanh hơn, dĩ nhiên không tuyến tính. Không có xung động cơ sẽ không quay, có xung 100% động cơ sẽ quay max. Tuy nhiên xung phải lớn hơn 1 mức nào đó thì mới đủ khởi động cho động cơ. Các đặc tính này các bạn tham khảo trong giáo trình về máy điện, khí cụ điện, nếu các bạn cần thông số chính xác.

Để có thể thay đổi 10 cấp tốc độ với chu kì 1000uS, ta khởi tạo cho ngắt timer: 100 uS ngắt 1 lần. Trong hàm ngắt kiểm tra xem ta cần cấp xung bao nhiêu % thì ta sẽ gán giá trị cho nó. Cụ thể như sau:

\* Hàm khởi tạo ngắt.

Dùng ngắt timer 0, 100 uS ngắt 1 lần, dùng chế độ 2 8 bit tự động nạp lại của timer (vì mình chỉ cần đếm đến 100). TL0 nạp bằng 156. Đối với chế độ 2 khi tràn bộ đếm TL0 sẽ quay vòng giá trị bằng 0, nhưng sau đó nó lại được nạp giá trị lưu trong TH0 (giá trị nạp lại), do đó ta chỉ cần gán giá trị cho TL0 và TH0 trong hàm khởi tạo, còn ở các chế độ khác 16 bit, 2 timer counter 8 bit, khi tràn bộ đếm TL0 không được nạp lại mà ta phải tự gán lại giá trị cho nó trong hàm ngắt.

```
void khoitaotimer0(void) // Hàm khởi tạo
{
EA=0; // Cấm ngắt toàn cục
TMOD=0x02; // Timer 0 chế độ 2 8 bit auto reload
TH0=0x9B; // Giá trị nạp lại 155 đổi ra số hex
TL0=0x9B; // Giá trị khởi tạo 155 đổi ra số hex
ET0=1; // Cho phép ngắt timer 0
EA=1; // Cho phép ngắt toàn cục
TR0=1; // Chạy timer 0 bắt đầu đếm số chu kỳ máy
}
```



## BÀI 5. MẠCH VI ĐIỀU KHIỂN

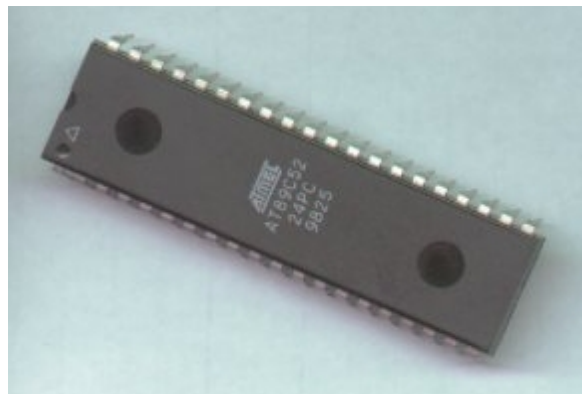
Mạch vi điều khiển đóng vai trò là phần trung tâm của robot. Vi điều khiển có thể ví như là bộ não của một cơ thể người, điều khiển mọi hoạt động của robot.

5.1. Giới thiệu một số loại vi điều khiển được sử dụng trong robocon

5.1.1. Vi điều khiển 8051

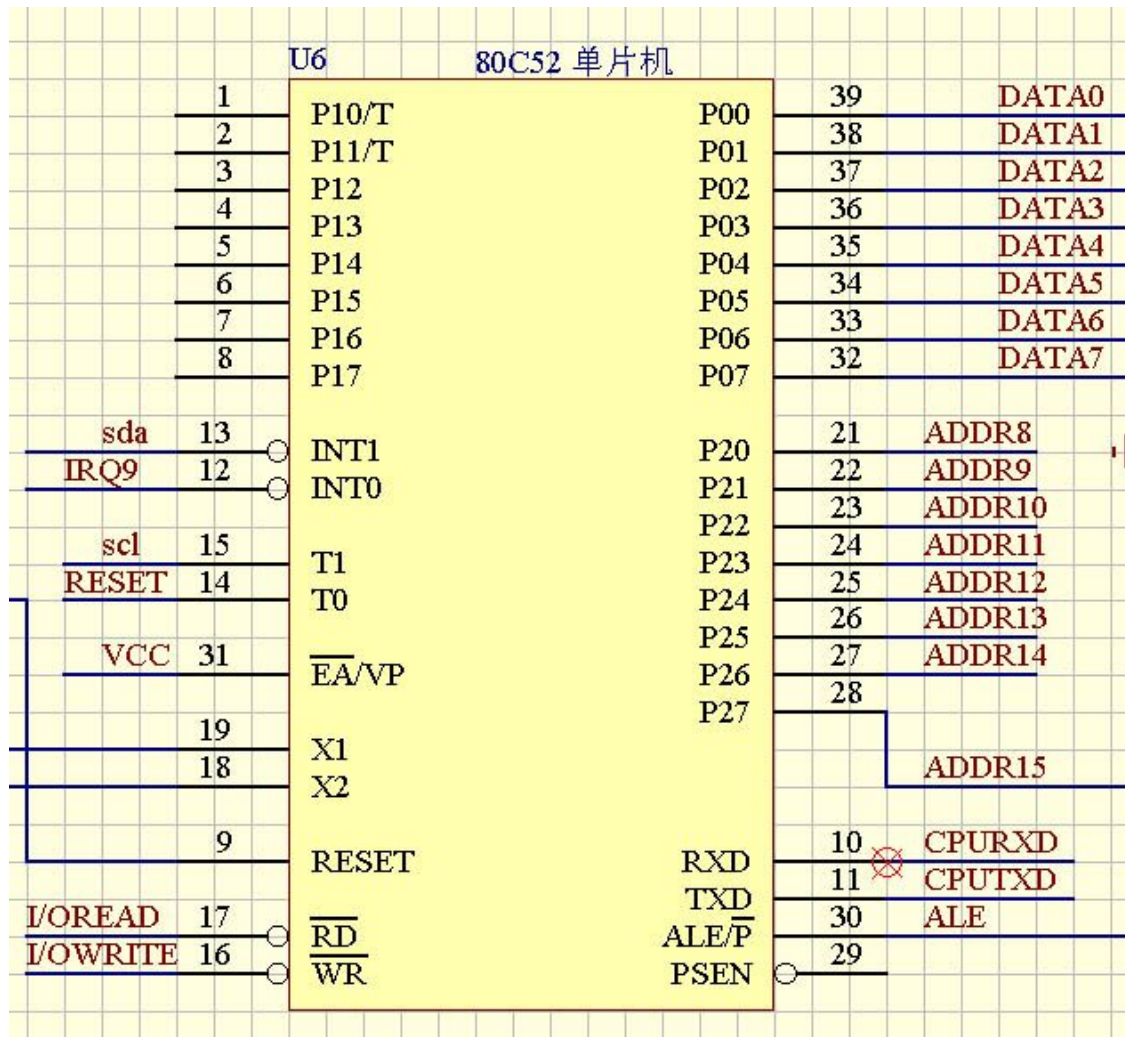
Vi điều khiển 8051 là loại vi điều khiển được sử dụng nhiều nhất trong các cuộc thi robocon từ trước tới nay. Bởi vì nó có giá thành rẻ, dễ mua và nhiều tài liệu tiếng Việt. Chính vì thế, nhiều đội robocon đã sử dụng 8051 là bộ não cho robot của mình.

Hình 5.1. Vi điều khiển 8051





Hình 5.2.Sơ đồ chân



Trong các loại điều khiển họ 8051 ,vi điều khiển 89c52 và 89s52 được nhiều đội robocon sử dụng nhiều nhất.Đặc biệt là các đội đến từ Bách Khoa Hà Nội.

Những đặc điểm của vi điều khiển 8051 ,các bạn có thể tham khảo trong cuốn giáo trình 8051 của MTC.

Một loại vi điều khiển thuộc họ 8051 khá mạnh nữa do hãng Philip sản xuất là loại P89V51RD. Chip Philip được các đội miền nam ,đặc biệt là BK HCM sử dụng rất nhiều do được tài trợ miễn phí.

Nhìn chung chip Philip có đầy đủ những tính năng của loại chip 89c51 của atmel như timer/counter , ngắt .v ..v. Bên cạnh đó , chip philip có những tính năng nổi bật hơn con 89c51 .

- Bộ nhớ ROM có dung lượng lớn

16/32/64 kb Flash rom và 1024 bytes Ram ( so với 8 k rom ,128 bytes ram của 89x) bộ nhớ của chip PHILIP

-Chức năng ISP ( In system -programming)

ISP là khả năng nạp trực tiếp chương trình vào chip ngay trên mạch mà không cần phải rút chip ra khỏi mạch .đối với vdk của atmel chỉ có con AVR ,và 89s52 mới có chức năng này.Nhờ chức năng này ,mạch nạp cho chip philip rất đơn giản dễ chế tạo

-Chức năng IAP (in-application programmable),chức năng IAP cho phép bộ nhớ flash có khả năng cấu hình lại trong khi các ứng dụng đang chạy.

-3 bộ định thời 16 bit chức năng của philip giống hệt chip 89c51

-SPI (serial peripheral interface)

Chức năng này cho phép truyền dữ liệu đồng bộ với tốc độ cao giữa chip philip và các thiết bị ngoại vi khác ,hay giữa các chip với nhau.

- PWM (pulse width modulation)

Đây là một chức năng rất đặc biệt của philip ,khi chip hoạt động ở chế độ này một chân của chip sẽ ra phát xung với tần số khác nhau (tần số này do người lập trình xác định) . Philip có chế độ băm xung 8 bit (con AVR còn hỗ trợ băm xung 10 bit ADC) với 5 chân băm xung.

-Chức năng ngắt

Philip 8 ngắt với 4 mức ưu tiên ngắt ,nó cũng có ngắt ngoài ,ngắt timer , nối tiếp như 89c ,ngoài ra còn có chức năng ngắt PCA ,UART/SPI.Có thể nói chip philip là loại chip rất ưu việt ,có thể thay thế rất tốt cho loại chip 89c52 thường được dùng trong các kì ROBOCON.các bạn có thể tham khảo

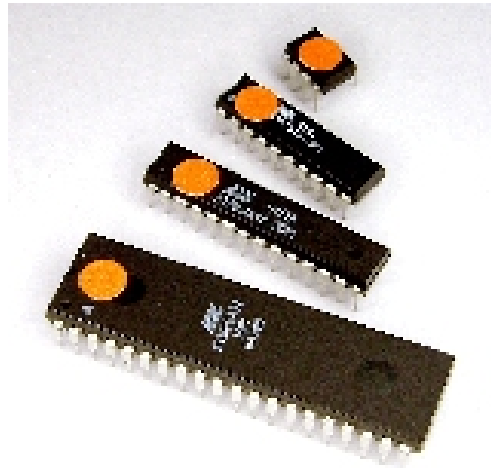
thêm datasheet của loại chip này ở trang  
<http://www.alldatasheet.com> (p89v51rd )

Trong năm 2004 ,2006 ,chip Philip được FXR và BKPRO sử dụng trong cuộc thi robocon và đã giành chức vô địch.

5.1.2.Vi điều khiển AVR

AVR là loại vi điều khiển do hãng ATMEL sản xuất .AVR có những tính năng rất mạnh so với 8051 như PWM ,ADC .AVR đã được BK-FIRE sử dụng đầu tiên vào năm 2006 và đã lọt được vào vòng chung kết toàn quốc.

Hình 5.1.Vi điều khiển AVR



Việc sử dụng AVR trong robocon có thể nói là một cải tiến mới trong công nghệ chế tạo robot theo truyền thống của các đội đến từ BKHN. Thông tin về chip AVR các bạn có thể tham khảo giáo trình Vi điều khiển nâng cao của trung tâm MTC.

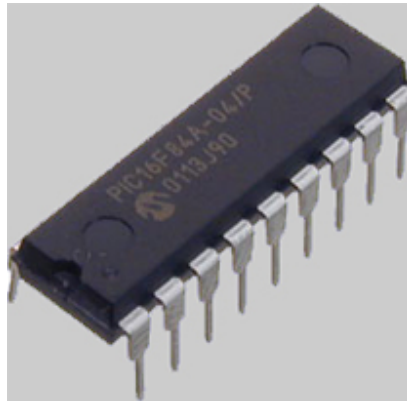
### 5.1.3.Vi điều khiển PIC.

Bên cạnh AVR ,PIC cũng là một sự lựa chọn khá tốt cho các đội robocon do những tính năng vượt trội của nó so với các loại vi điều khiển khác như khả năng chống nhiễu ,PWM ,ADC, ngắt.Hiện nay trên thtrường có khá nhiều dòng vi điều khiển PIC.VD:pic16f877,pic16f84 ,pic16f828..v.v.v

#### **Đánh giá các dòng PIC**

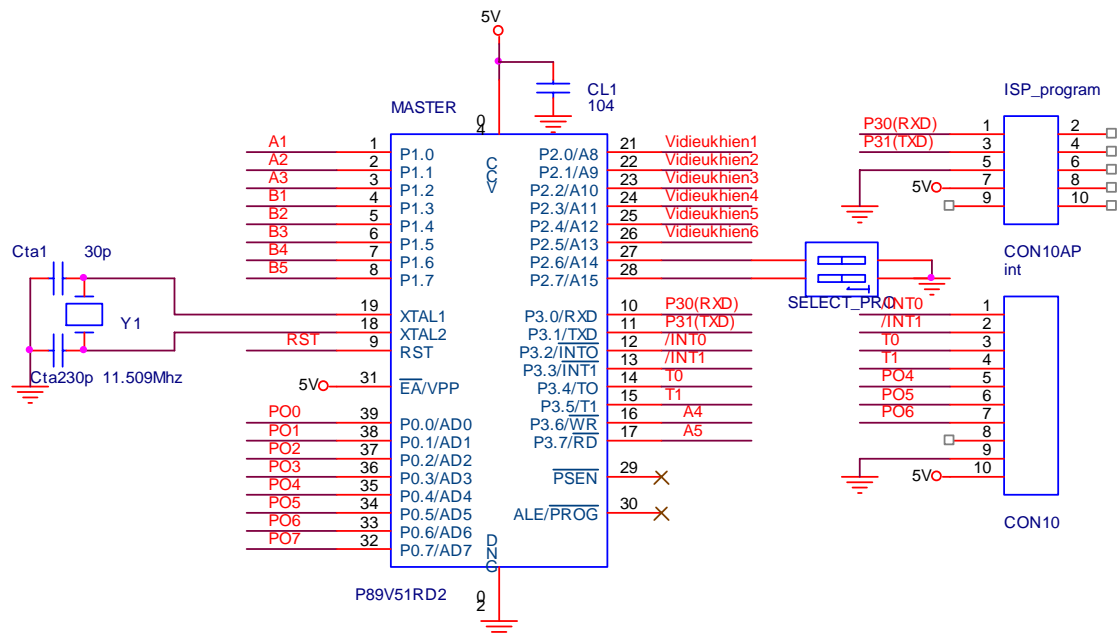
- Dòng PIC nhiều chân nhất là dòng PIC18Fxxxx, có những con số chân lên đến 80 chân
- Dòng PIC ít chân nhất là dòng PIC10Fxxx, chỉ có 6 chân
- Dòng PIC phổ biến nhất là dòng PIC16F877A (đủ mạnh về tính năng, 40 chân, bộ nhớ đủ cho hầu hết các ứng dụng thông thường)
- Dòng PIC mà chúng tôi đánh giá cao nhất là dòng PIC16F876A (28 chân, chức năng không khác gì so với PIC16F877A, nhưng nhỏ gọn hơn nhiều, và số chân cũng không quá ít như PIC16F88).
- Dòng PIC hỗ trợ giao tiếp USB là dòng PIC18F2550 và PIC18F4550
- Dòng PIC điều khiển động cơ mạnh nhất là dòng PIC18F4x31
- Khi cho rằng mình chuyên nghiệp hơn, các bạn nên dùng PIC18F458
- Dòng PIC tàng hình là dòng PIC17xxxxx, hiện nay đã không còn được sản xuất

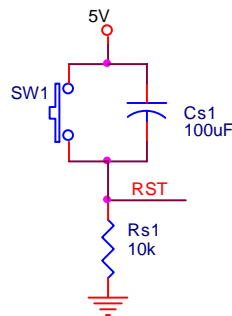
Hình 5.2. Vi điều khiển PIC



### 5.2. Sơ đồ nguyên lý

Trong giáo trình này , chúng tôi chủ yếu tập trung vào mạch vi điều khiển dùng 89s52. Bởi vì vi điều khiển 89s52 được sử dụng khá thông dụng trong các kĩ robocon. Sơ đồ nguyên lý của vi điều khiển được nói nhiều trong giáo trình vi điều khiển cơ bản (8051) do MTC soạn thảo. Các bạn có thể tham khảo kĩ hơn trong tập giáo trình này.



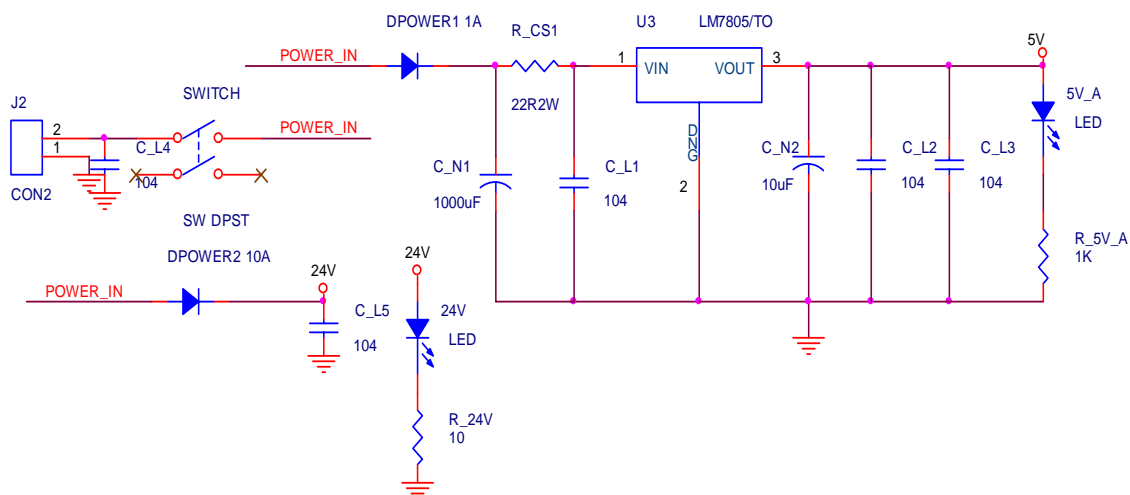


Hình 5.2. Sơ đồ nguyên lý mạch vi điều khiển.

Mạch vi điều khiển sử dụng nguồn 5 vôn. Trong robocon, mạch này đóng vai trò là khối trung tâm của robot, kết nối với tất cả các modul khác. Từ mạch vi điều khiển này, bạn có thể mở rộng ra các modul khác.

Mạch nguồn.

Mạch vi điều khiển sử dụng nguồn 5V, Tuy nhiên các acquy dùng trong robot đều là nguồn 12V hoặc 24V. Do đó, bạn không thể đấu trực tiếp nguồn ắc quy vào vi điều khiển được, cần phải thiết kế một mạch nguồn để cung cấp một nguồn điện ổn định 5V cho vi điều khiển.



Sơ đồ nguyên lý mạch nguồn dùng trong robocon.

Mạch nguồn này sử dụng IC ổn áp 7805. Đây là loại IC ổn áp: Đầu vào > 7V đầu ra 5V 500mA. Mạch ổn áp: cần cho VĐK vì nếu nguồn cho VĐK không ổn định thì sẽ treo VĐK, không chạy đúng, hoặc reset liên tục, thậm

chí là chết chíp. Trong sơ đồ trên , các tụ điện đóng vai trò lọc nhiễu , ổn áp , diode để chống ngược dòng. Ngoài ra , trong mạch nguồn , các bạn chú ý nên lắp thêm đèn báo nguồn và cầu chì để bảo vệ quá áp .

## BÀI 6: LẬP TRÌNH VÀ KỸ THUẬT DÒ ĐƯỜNG

Trong robocon , lập trình là khâu cuối cùng trong 3 bộ phận cấu thành robot. Lập trình sẽ đóng vai trò quyết định cuối cùng để đánh giá robot của bạn sẽ hoạt động tốt hay không. Robot sẽ không thể chạy tốt nếu như chương trình xử lý không hoàn hảo.

### 6.1. Các ngôn ngữ lập trình sử dụng trong robocon.

Trong robocon có 2 ngôn ngữ lập trình được sử dụng nhiều nhất là C và ASM . 2 ngôn ngữ này có những điểm mạnh và điểm yếu riêng.

Ngôn ngữ ASM có ưu điểm là gọn nhẹ , giúp người lập trình hiểu sâu về cấu trúc phần cứng của vi điều khiển . Các chương trình viết bằng ASM thường chạy nhanh và tốc độ xử lý cao. ASM đã được BK-FIRE sử dụng trong cuộc thi 2005 và hoạt động khá hiệu quả. Tuy nhiên , ASM có nhược điểm là khó học và tập lệnh nghèo nàn , không thuận tiện để lập trình các chương trình lớn.

Ngôn ngữ lập trình C có thể mạnh là dễ học , tập lệnh phong phú , và đặc điểm là ngôn ngữ lập trình có cấu trúc nên rất thuận lợi để xây dựng các chương trình lớn. Nhược điểm của C là không giúp người đọc hiểu sâu về cấu trúc phần cứng.

Nhìn chung , trong các cuộc thi robocon , ngôn ngữ C được sử dụng nhiều nhất do những ưu điểm của nó. Giáo trình này cũng hướng dẫn bạn lập trình bằng ngôn ngữ C.

### 6.2. Mã nguồn của robocon

Xin giới thiệu một đoạn mã nguồn của đội CIRTECH-45 của BKHN năm 2004

/\*

Night Lamp Saver V5.0

89C2051(ext. oscillator 680kHz) + MAC97A6 + no battery backup  
demonstration of using Micro-C and ATMEL89C2051 to build a device  
used for controlling night lamp that turn on and off night lamp  
with preset time on/off.

After reset or power failure occurred, high blink rate of led will show,  
user should press P3.0 to reset time to 18:00, low blink rate will show  
indicating normal operation.



The Saver V4.0 using Xtal 11.0592MHz produces EMI that interfere TV reception

This version the Xtal oscillator has changed to RC oscillator 680kHz.  
cputick incremental was derived from 50Hz or 20ms main frequency.

*Copyright (c) 1999 W.SIRICHOTE*

*\*/*

*#include c:\mc51\8051io.h*  
*#include c:\mc51\8051reg.h*

*/\*----- turn lamp on/off after reset time to 18:00 -----\*/*

*#define onHour1 18 /\* 18:00 turn lamp on \*/*  
*#define onMin1 00*  
*#define offHour1 18 /\* 18:01 turn off \*/*  
*#define offMin1 01*

*/\* every day turn on at 19:00 and and off at 22:00 \*/*

*#define onHour2 19*  
*#define onMin2 00*  
*#define offHour2 22*  
*#define offMin2 00*

*/\* set clock to 18:00 when press P3.0 \*/*

*#define setHour 18*  
*#define setMin 00*

*/\*-----\*/*

*extern register char cputick;*  
*unsigned register char*  
*sec25,sec50,sec,sec5,min,hour,flag1,temp,led,blink\_rate;*  
*/\* above must be defined as register for tiny model \*/*

*/\* variables description*  
*cputick increments by one every 20ms*  
*sec25 half second counter*

```
sec50  2*25Hz counter
sec    current second
sec5   5 second counter
min    current min
hour   current hour
temp   temp register
led    counter for led on duration (times cputick)
blink_rate 0 = high blink rate, 10 low blink rate

flag1  intertask signaling          mask byte

flag1.0 set every 1 second          0x01
flag1.1 set every 1 min             0x02
flag1.2 not use                     0x04
flag1.3 set every 0.5 second        0x08
flag1.4 set after P3.2 has been pressed 0x10
flag1.5 disable turn on/off 18:00-18:01 if set 0x20
flag1.6-7 not use
*/
```

```
main()
{
    cputick = 0;
    hour = 18;
    min = 0;
    sec = 0;
    sec25 = 0;
    sec50 = 0;
    flag1 = 0;
    blink_rate = 0; /* indicate reset time to 18:00 is needed */

    asm "LAMP EQU $97"; /* P1.7 */
    asm{
        SETB $AF /* setb EA */
        SETB $A8 /* enable external interrupt */
        SETB $88 /* negative edge triggering */
    }
    while(1)
    {
        while ( cputick < 1);
    }
}
```

```
    cputick = 0; /* 20ms has elapsed */

/*----- the following tasks execute every 10ms -----*/

    time();
    comparetime();
    cpubeat();
    settime();
/*    waithigh(); */
}

/*-----*/
}

time ()
/* update real-time clock, date */
{
    sec25++;
    if (sec25 >= 25) /* now 25 times means half second */
    {sec25 = 0;
        flag1 |= 0x08; /* set bit 3 every 0.5 s */
    }
    sec50++;
    if (sec50 >= 2) /* 2 * 25 * 20 ms = 1 s */
    {sec50 = 0;
        flag1 |= 0x01; /* set bit 0 */
    }
    sec++;
    if (sec >= 60)
    {sec = 0;
        flag1 |= 0x02; /* set bit 1 */
    }
    min++;
    if (min >= 60)
    {min = 0;
        hour++;
    }
    if (hour >= 24)
    {hour = 0;
    }
}
}
}
}
```

```
}  
}
```

```
comparetime()  
{
```

```
    if ((flag1 & 0x10) != 0) /* enabled only after P3.2 has been pressed */  
    {  
        compareTimeOn_Off();  
    }  
}
```

```
compareTimeOn_Off()  
{
```

```
    if ((flag1 & 0x01) != 0)  
    {  
        testOnOff();  
        if (hour == onHour2 && min == onMin2)  
            asm " CLR LAMP";  
        if (hour == offHour2 && min == offMin2)  
            asm " SETB LAMP";  
    }  
}
```

```
testOnOff()  
{
```

```
    if ((flag1 & 0x20) == 0)  
    {  
        if (hour == onHour1 && min == onMin1)  
            asm " CLR LAMP";  
        if (hour == offHour1 && min == offMin1)  
        {  
            asm " SETB LAMP";  
            flag1 |= 0x20; /* disable further test on off */  
        }  
    }  
}
```

```
cpubeat()
```

```
{  
    beat5sec();  
    livecpu();  
}
```

*beat5sec() /\* clear P3.7 every blink rate \*/*

```
{  
    if ((flag1 & 0x08) != 0)  
    {  
        flag1 &= ~0x08; /* clear bit 3 of flag1 */  
        sec5++;  
        if (sec5 > blink_rate)  
        {sec5 = 0;  
         flag1 |= 0x40; /* set bit 6 of flag1 to signal livecpu task */  
         asm " clr P3.7"; /* make led on */  
         led = 2; /* load time on duration times cputick */  
        }  
    }  
}
```

```
}
```

*livecpu()*

```
{  
    if ((flag1 & 0x40) != 0)  
    {  
        led--;  
        if (led == 0)  
        {  
            asm " setb P3.7";  
            flag1 &= ~0x40;  
        }  
    }  
}
```

*settime()*

```
{  
    if ((P3 & 0x01) == 0) /* reset time to 18:00 if P3.1 low */
```

```
{
    hour = setHour;
    min = setMin;
    sec = 0;
    sec50 = 0;
    flag1 |= 0x10; /* enable compare time on/off */
    flag1 &= ~0x20; /* reenable testOnOff after pressing set clock to 18:00
*/
    blink_rate = 10;
}
}

/*
waithigh()
{
    asm" jnb P3.2,*";
    pause(2);
    asm" jnb P3.2,*";
    pause(2);
}

pause(j)
int j;
{
    int i;
    for (i=0;i<j;i++)
        ;
}
*/
```

Mã nguồn ASM

```
;CHUONG TRINH CHO IC MASTER
; P1: CAM BIEN NHAN VAO
; P0: TRUYEN SANG SLAVER
```

```
STARTB EQU P3.7
CT_NANG EQU P3.2
CT_HATAY EQU P3.0
CT_QUA EQU P3.1
```

*CB\_L EQU P1.2*  
*CB\_R EQU P1.1*  
*MKEP\_DIR EQU P2.0*  
*MKEP\_EN EQU P2.1*  
*MNANG\_DIR EQU P2.2*  
*MNANG\_EN EQU P2.3*

*STOP\_C EQU 11111111B*  
*QPHAI\_C EQU 00011111B*  
*QTRAI\_C EQU 00101111B*  
*LUI\_C EQU 00111111B*  
*THANG\_C EQU 01001111B*  
*TINHCHINH\_C EQU 01011111B*  
*ORG 0000H*  
*LJMP INIT*  
*ORG 001BH ;DIEM NHAP VECTOR NGAT CUA TI*  
*LJMP NGATT1*  
*ORG 0030H*  
*INIT:;KHOI TAO CAC GIA TRI SAU RESET*  
*CLR TR0*  
*MOV SP,#5FH*  
*SETB STARTB          ;BIT KHOI DONG*  
*MOV P0,#0FFH*  
*MOV P1,#0FFH*  
*MOV R0,#0H    ;BIT Dem SO NGA TU*  
*MOV R1,#0H*  
*MOV P2,#0H    ;TAT CAC DONG CO*  
*MOV IE,#0H ;CHO PHEP NGAT BO DINH THOI 1*  
*MOV P3,#0FFH*  
*MOV TMOD,#00010001B*  
*CLR TR1*  
*CLR TR0*  
*CLR TF0*  
*CLR TF1*

*;=====*  
*=====*  
*=====*

*MAIN:*

```
;MOV P0,QPHAI_C  
JB STARTB,MAIN  
LAP_CHINH:  
MOV A,P1  
ANL A,#00001111B  
MOV P0,A  
LJMP LAP_CHINH  
EXIT:  
NODO: SJMP NODO
```

```
;  
=====  
=====  
=====
```

*DEMNGATU:*

```
INC R0  
;LCALL STOP_ALL  
; LCALL TRE_2S  
CJNE R0,#2,TIEP1  
LCALL DUNG_NGATU  
LCALL TRE_QUAY  
LCALL QUAYPHAI  
TIEP1:  
CJNE R0,#10,TIEP2  
LCALL DUNG_NGATU  
LCALL TRE_QUAY  
LCALL QUAYphai  
TIEP2:  
CJNE R0,#15,DITIEP  
LCALL DUNG_NGATU  
LCALL TRE_QUAY  
LCALL QUAYTRAI  
HERE20: SJMP HERE20
```

*DITIEP:*

```
TREQUAVACH: LCALL TRE0 ;CHO THOI GIAN TRE DI DI QUA  
VACH NGANG
```

*TROVE:*

```
RET ;TRO VE CHUONG TRINH
```

```
;  
=====
```



*LAYQUA:*

```
LCALL STOP_ALL
;KEPQUA
SETB MKEP_EN
LCALL TRE_quay
CLR MKEP_EN
;NANG QUA
SETB MNANG_EN
DOINANG: JB CT_NANG,DOINANG
CLR MNANG_EN
;NHA QUA
SETB MKEP_DIR
SETB MKEP_EN
LCALL TRE_quay
```

```
CLR MKEP_EN
CLR MKEP_DIR
;HA TAY
;SETB MNANG_DIR
;SETB MNANG_EN
;DOIHA: JB CT_HATAY,DOIHA
;CLR MNANG_EN
;CLR MNANG_DIR
RET
```

*;=====*

*DUNG\_NGATU:*

```
LAPNT:
JB P1.4,THOATNT
MOV A,P1
ANL A,#00001111B
MOV P0,A
LJMP LAPNT
THOATNT:
LCALL STOP_ALL
LCALL TRE_QUAY
MOV P0,#00111111B
DOINT: JNB P1.4, DOINT
LCALL STOP_ALL
MOV P0,#01011111B ;TINH CHINH
LCALL STOP_ALL
```

*RET*

;=====

*QUAYTRAI:*

*MOV P0,#00101111B ;QUAYTRAI*  
*LCALL TRE\_QUAY*  
*LAPQT:*  
*MOV A,p1*  
*ORL A,#11111001B*  
*CJNE A,#11111111B,LAPQT*  
*LCALL STOP\_ALL*  
*LCALL TRE\_100*  
*MOV P0,#00011111B ;QUAYPHAI*

*LAPQT2:*

*MOV A,p1*  
*ORL A,#11111001B*  
*CJNE A,#11111111B,LAPQT2*  
*LCALL STOP\_ALL*  
*LCALL TRE\_QUAY*  
*LCALL TRE\_QUAY*

*RET*

;=====

=====

*QUAYPHAI:*

*MOV P0,#00011111B ;QUAYTRAI*  
*LCALL TRE\_QUAY*  
*LAPQP:*  
*MOV A,p1*  
*ORL A,#11111001B*  
*CJNE A,#11111111B,LAPQP*  
*LCALL STOP\_ALL*  
*LCALL TRE\_100*  
*MOV P0,#00101111B ;QUAYPHAI*

*LAPQP2:*

*MOV A,p1*  
*ORL A,#11111001B*  
*CJNE A,#11111111B,LAPQP2*  
*LCALL STOP\_ALL*  
*LCALL TRE\_QUAY*  
*LCALL TRE\_QUAY*

```
RET
;=====
====
NGATT1:

RETI

STOP_ALL:
    MOV P0,#0FFH
RET
;=====CAC          THU          TUC
TRE=====
TRE0: ;THU TUC DE TAO TRE QUA VACH NGANG NGA TU
    mov r7,#50
    again0: mov TH0,#HIGH(-10000)
            mov TL0,#LOW(-10000)
            setb tr0
            wait0:
            MOV A,P1
            ANL A,#00001111B
            MOV P0,A
            jnb tf0,wait0
            clr tr0
            clr tf0
            djnz r7,again0
RET
TRE_500: ; 1s
    mov r7,#50
    again_1S: mov TH0,#HIGH(-10000)
              mov TL0,#LOW(-10000)
              setb tr0
              wait_1S: jnb tf0,wait_1S
              clr tr0
              clr tf0
              djnz r7,again_1S
RET
TRE_QUAY: ;
    MOV R7,#100
    again: mov TH0,#HIGH(-10000)
           mov TL0,#LOW(-10000)
```

```
    setb tr0
    wait: jnb tf0,wait
    clr tr0
    clr tf0
    djnz r7,again
RET
TRE_100: ;
    MOV R7,#10
    again: mov TH0,#HIGH(-10000)
           mov TL0,#LOW(-10000)
    setb tr0
    wait: jnb tf0,wait
    clr tr0
    clr tf0
    djnz r7,again
RET

TRE_2S: ; 1s
    mov r7,#200

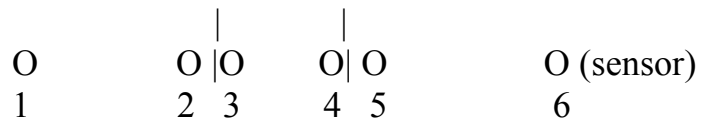
    again_2S: mov TH0,#HIGH(-10000)
              mov TL0,#LOW(-10000)
    setb tr0
    wait_2S: jnb tf0,wait_2S
    clr tr0
    clr tf0
    djnz r7,again_2S
RET
END
```

### 6.3.Kĩ thuật dò đường.

Bài học này sẽ nêu lên các phương pháp lập trình dò đường trong robocon. Việc lập trình dò đường trong robot thực chất là việc xử lý tín hiệu từ sensor và đưa ra các lệnh điều khiển động cơ để đảm bảo cho robot có thể bám theo vạch trắng trên sân.Khi lập trình ,các bạn phải xác định được tất cả các trường hợp có thể xảy ra của sensor ,từ đó đưa ra các tín hiệu điều khiển động cơ hợp lý.

Quyay về bài sensor ,ta có sơ đồ bố trí sensor

| |  
| | Vạch trắng



4 sensor 2,3,4,5 dùng để bám đường. Khi trên sân thi đấu ,giả sử khi robot lệch sang phải sensor 2 chạm vạch trắng ,sensor 4 chạm nền ,bạn phải điều chỉnh cho động cơ phải quay nhanh hơn ,động cơ trái chậm lại hay dùng để robot quay lại vị trí. Tương tự ,khi robot lệch sang trái ,bạn cũng điều khiển tương tự như vậy. Bằng cách xử lý như thế ,bạn sẽ giúp cho robot có thể bám đường một cách hiệu quả.

Khi gặp ngã tư ,hai sensor ngoài cùng 1,6 sẽ chạm vạch trắng ,có mức tín hiệu 0 đưa vào vi điều khiển .Bạn cần lập trình đếm số lần mức tín hiệu 0 vào vi điều khiển để từ đó suy ra ngã tư robot đã đi.

#### 6.4. Kỹ thuật chống nhiễu

Như đã học ở bài sensor ,khi thi đấu ảnh hưởng của đèn cao áp trên sân có thể gây ra nhiễu lên sensor dẫn tới việc nhận biết tín hiệu sai. Để chống nhiễu ,bạn có thể sử dụng chương trình để xử lý.

Có nhiều trường hợp do nhiễu nên sensor có thể nhận nhầm vạch trắng. Do đó ,bạn phải để vi điều khiển kiểm tra thành nhiều lần.

Mã nguồn .

```
void golong(unsigned char songatu)
{
    unsigned char d=0;
    unsigned char dem=0;
    motor(forward);
    while (1)
    {
        if ((out_left==nen)&&(out_right==nen)) motor(forward);
        if ((out_left==nen)&&(out_right==vach))
        {
            motor(right_stop);
            motor(left_go);
        }
        if ((out_right==nen)&&(out_left==vach))
        {
            motor(left_stop);
            motor(right_go);
        }
    }
}
```

```
// bat nga tu

if ((out_left==vach)&&(out_right==vach))
{

while ((LEFT==vach)&&(RIGHT==vach))
{
d++;
if (d==100)
{
dem++;
d=0;
if (dem<songatu)
{ h_thi(dem); motor(forward); delay(1200);}
break;//thoat khoi while
}
}
}

// chinh lech nhieu

}

//end of while (1)
}

//end of golong_ngatmo
```

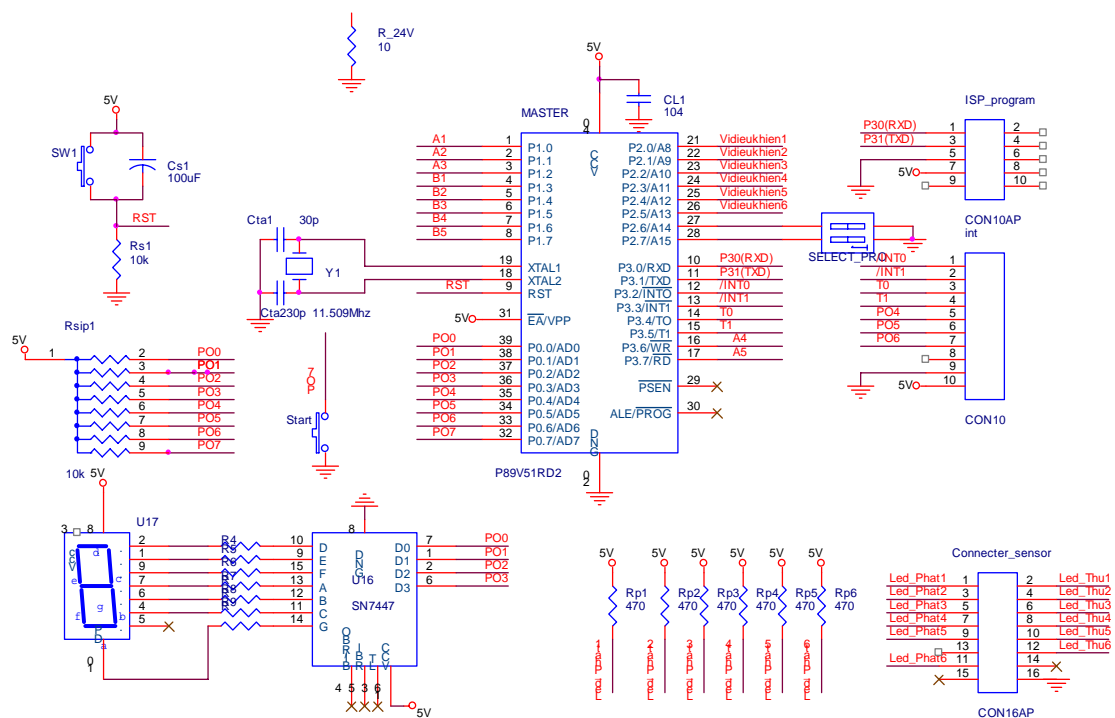
## BÀI 7: CHIẾN THUẬT THI ĐẤU

Trong khi thi đấu ,có rất nhiều các tình huống xảy ra .Do đó người lập trình cần phải xây dựng nhiều chiến thuật thi đấu khác nhau để đảm bảo có thể xử lý kịp thời mọi tình huống có thể có trên sân.

Để hiển thị chiến thuật thi đấu ,bạn có thể dùng led 7 thanh hay màn hình LCD ,một hệ thống các phím bấm để điều khiển ,để lựa chọn các giải pháp thi đấu.Thực chất ,ở phần này chúng ta đã quy về bài toán lập trình giao tiếp vi điều khiển với bàn phím và màn hình LCD ,led 7 thanh

## 7.1.Sơ đồ nguyên lý

### a)Giao tiếp với led 7 thanh



Ở sơ đồ trên ,ta sử dụng 1 led 7 thanh để hiển thị các chiến thuật thi đấu 1 ,2 ,3.v.v.2 phím bấm choice và start để lựa chọn chiến thuật.Một led 7 thanh có thể hiển thị được 10 chiến thuật thi đấu tương ứng với các số từ 0 đến 9.Ngoài ra ,led 7 thanh còn được dùng để hiển thị số ngẫu nhiên.

## 7.2.Mã nguồn

```
void main()
{
    unsigned char tam=0;
```

```
init();
P2_6=0;
P2_7=0;
h_thi(tam);
selection();

switch (select)
{
case 0:
    {
        hanhtrinh0();
    }
    break;

case 1:{
    hanhtrinh1();

    }
    break;
case 2:{
    hanhtrinh2();
    }
    break;
case 3:{
    hanhtrinh3();
    }
    break;
case 4:{
    hanhtrinh4();
    }
    break;
case 5:{
    hanhtrinh5();
    }
    break;
case 6:{
    hanhtrinh6();
    }
    break;
case 7:{
```



```
        hanhtrinh7();
    }
    break;
case 8: {
    hanhtrinh8();
    }
    break;
case 9:{
    hanhtrinh9();
    }
    break;
} //end of swith
delay(50000);
h_thi(tam);
}
        /* cac ham chuc nang */

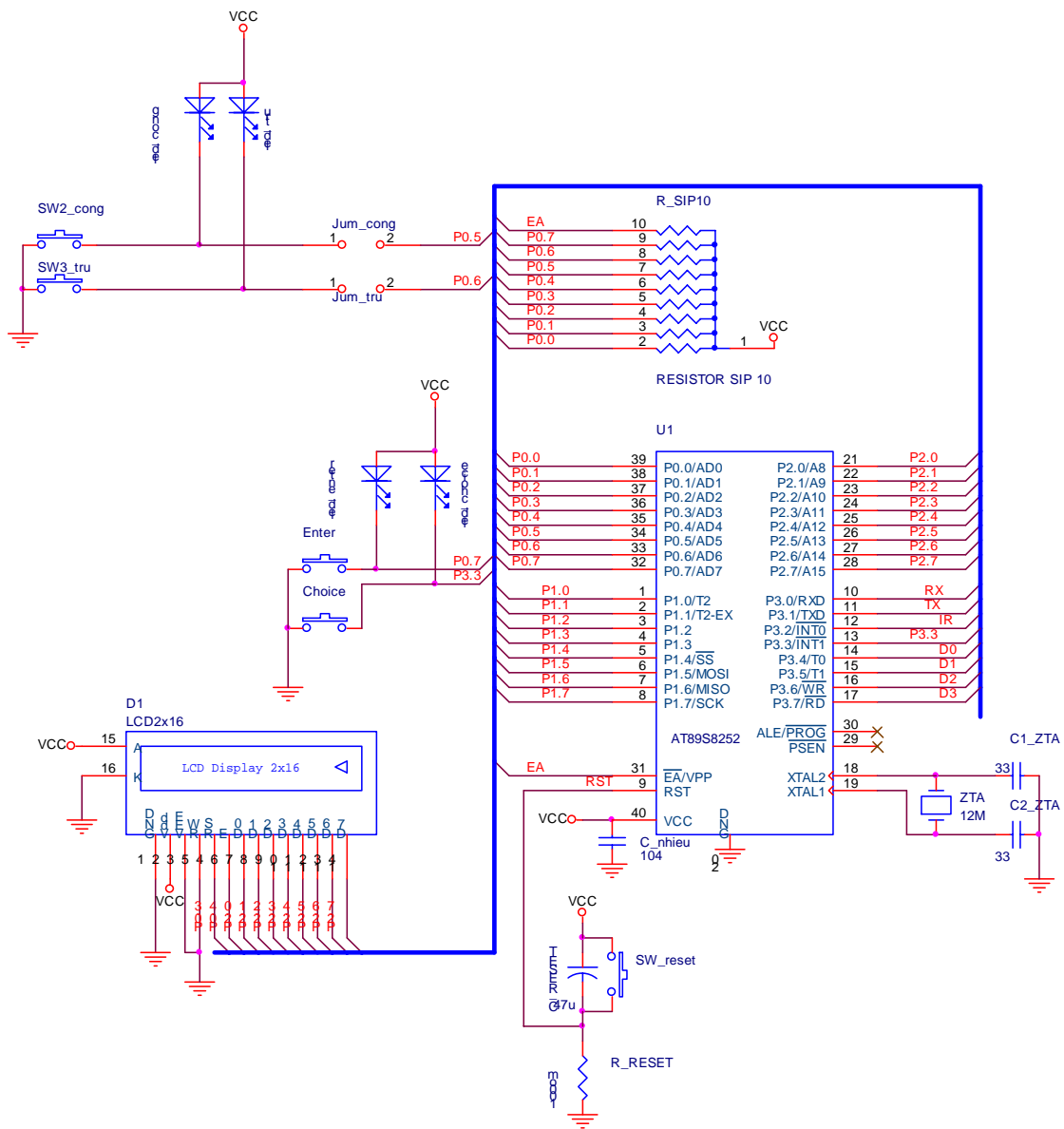
void selection()
{
    P2_6=0;
    P2_7=0;
    while (P2_7==0)
    {

        while (P2_6==1)
        {
            select++;
            if (select>9) select=0;
            h_thi(select);
            delay(12000);
        }
    }
} //end
void h_thi(unsigned char so)
{
    if (so==0) {P0_3=0;P0_2=0;P0_1=0;P0_0=0;}
    else if (so==1) {P0_3=0;P0_2=0;P0_1=0;P0_0=1;}
    else if (so==2) {P0_3=0;P0_2=0;P0_1=1;P0_0=0;}
    else if (so==3) {P0_3=0;P0_2=0;P0_1=1;P0_0=1;}
}
```

```

else if (so==4) {P0_3=0;P0_2=1;P0_1=0;P0_0=0;}
else if (so==5) {P0_3=0;P0_2=1;P0_1=0;P0_0=1;}
else if (so==6) {P0_3=0;P0_2=1;P0_1=1;P0_0=0;}
else if (so==7) {P0_3=0;P0_2=1;P0_1=1;P0_0=1;}
else if (so==8) {P0_3=1;P0_2=0;P0_1=0;P0_0=0;}
else if (so==9) {P0_3=1;P0_2=0;P0_1=0;P0_0=1;}
}
    
```

b) Sơ đồ dùng LCD





```
{
  P2=dk;   lenh ();
}

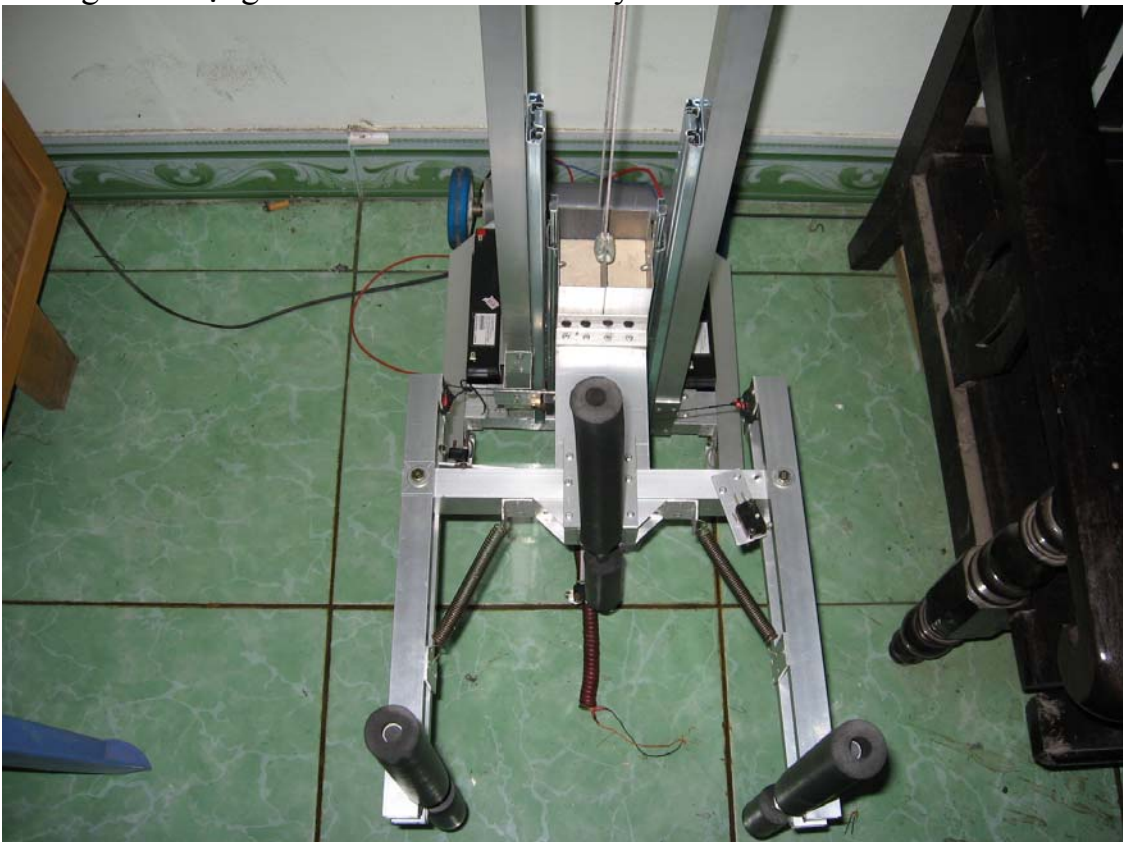
void init_LCD ()
{
  delay(400);
  LCDcontrol(0x38); //LCD 2 dong _ 5x7
  LCDcontrol(0x0C); //bat hien thi, tat con tro
  LCDcontrol(0x01); //xoa man hinh
}
  /*****/

void keyboard()
{
  unsigned char key=0;
  unsigned char test=0;
  LCDcontrol(0x01);
  choice=enter=1;
  LCDputs("Robot Ready",1);
  delay(100000);
  LCDcontrol(0x01);
  LCDputs("1:Golong 2:Around ",1);
  LCDputs("3:Turn left 4:Turn right",2);
  delay(100000);
  LCDcontrol(0x01);
  while(1)
  {
    LCDputs("1: 2: 3: 4:",2);
    while(choice==0)
    {
      delay(10000);
      key++;
      LCDcontrol(0x01);
      if (key==1)LCDputs("1 ",1);
      if (key==2)LCDputs("2 ",1);
      if (key==3)LCDputs("3 ",1);
      if (key==4)LCDputs("4 ",1);
      LCDputs("1:C 2:S 3:D 4:E",2);
      if (key>4) { LCDputs("0 ",1);key=0;}
    }
  }
}
```

```
    }  
    if (key==1)while(enter==0) golongr();  
    if (key==2)while(enter==0) Around();  
    if (key==3)while(enter==0) left();  
    if (key==4)while(enter==0) test=1;  
    if (test==1) break;  
  
} //while(1)  
  
} //keyboard
```

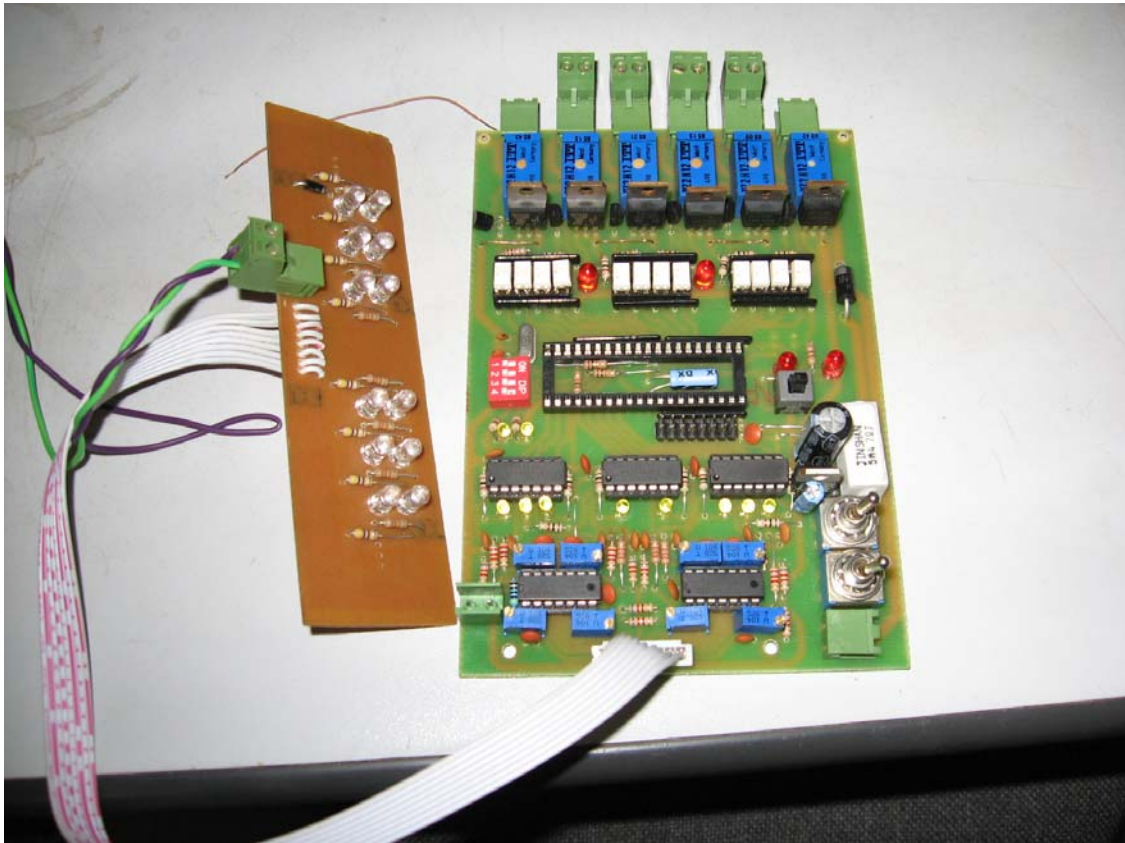
## Bài 8. Hoàn thiện robot thi đấu

Ở bài này ,chúng ta sẽ hoàn thiện một robot hoàn chỉnh để thi đấu .  
Chúng ta sử dụng robot như hình dưới đây



Robot ở hình trên là robot của một đội tham gia trong cuộc thi robocon 2007. Chúng ta sẽ sử dụng robot này để thực hành .

Phần mạch của robot ,chúng ta sử dụng mạch điều khiển động cơ bằng relay  
Như hình dưới



Đây là mạch điều khiển của đội BK-FIRE năm 2005 ,robot hoạt động khá hiệu quả .Đầu nối sensor như hình trên ,lắp đặt mạch sensor vào phần đế của robot .Robot này sử dụng bộ ắc quy 24V DC để cấp nguồn.Mạch này sử dụng 6 sensor xếp thành hàng ngang để bám đường ,6 động cơ điều khiển các cơ cấu.

Sau khi lắp đặt mạch và sensor ,đầu nối các động cơ .Ta sẽ lập trình cho để robot có thể bám theo vạch trắng có sẵn.

Trong bài thực hành này ,chúng ta sử dụng vi điều khiển 89c52 để điều khiển robot.

Đây là toàn bộ code chương trình bám đường của robot.(viết bằng ASM)

# **ROBOT VÀ ĐIỀU KHIỂN CHUYỂN ĐỘNG (2 TÍN CHỈ)**

# *Tài liệu tham khảo*

- 1, Nguyễn Thiện Phúc, Rô bốt công nghiệp. Nhà xuất bản Khoa học và Kỹ thuật, 2002.**
  - 2, Đào Văn Hiệp, Rô bốt công nghiệp. Nhà xuất bản Khoa học và Kỹ thuật, 2004.**
  - 3, Tạ Duy Liêm, Rô bốt công nghiệp. Nhà xuất bản Khoa học và Kỹ thuật .**
  - 4, Nguyễn Mạnh Tiến, Điều khiển Rô bốt công nghiệp. Nhà xuất bản Khoa học và Kỹ thuật, 2007 .**
- 
- 4, Industrial Robots, Computer Interfacing and Control Wesley E. Snyder Prentice Hall 1987.**



Nắm được các cấu trúc robot và điều khiển chuyển động tay máy. Động học và động lực học robot. Các bộ phận truyền động secvo và cảm biến. Thuật toán điều khiển robot theo quỹ đạo đặt.

# Tạo hình bề mặt trên máy công cụ điều khiển số

## 1- Khái niệm.

- Trên máy công cụ điều khiển theo chương trình số, đường tác dụng giữa dụng cụ và chi tiết được hình thành nhờ các dịch chuyển tọa độ trên nhiều trục. Để sản sinh một biên dạng gia công, giữa các chuyển động trên từng trục tọa độ riêng lẻ phải có một quan hệ hàm số (tuyến tính hoặc phi tuyến).
- Các điểm tựa phải nằm dày đặc đến mức sao cho biên dạng gia công tạo ra đủ chính xác (không có điểm nào nằm ngoài vùng dung sai cho phép).
- Giá trị tọa độ vị trí các điểm trung gian được tìm ra trong một cụm chức năng của điều khiển số- gọi là bộ nội suy.

# Tạo hình bề mặt trên máy công cụ điều khiển số

## 1- Khái niệm.

- Bộ nội suy có nhiệm vụ:
  - Tìm ra vị trí các điểm trung gian cho phép hình thành biên dạng cho trước trong giới hạn dung sai xác định.
  - Tốc độ đưa ra vị trí các điểm trung gian phải phù hợp với tốc độ chạy dao.
  - Đi tới chính xác các điểm kết thúc chương trình.
- Về mặt kỹ thuật, việc nội suy có thể được thực hiện bằng các thiết bị làm việc theo kiểu số hoặc kiểu tương tự.
  - Bộ nội suy kiểu tương tự đơn giản nhưng độ chính xác hạn chế.
  - Trên thực tế người ta dùng các bộ nội suy làm việc theo nguyên tắc số.

# Tạo hình bề mặt trên máy công cụ điều khiển số

- Nội suy số.

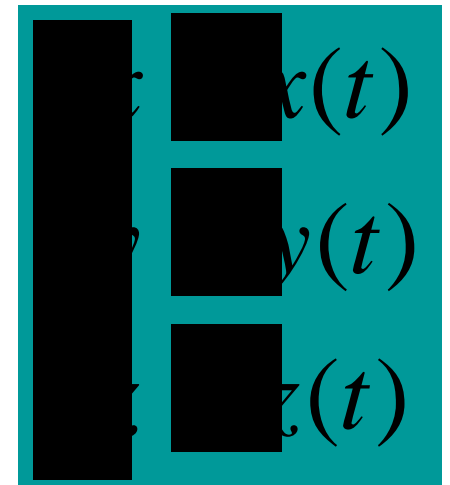
Việc tính toán các hàm số ở dạng số có thể thực hiện theo 2 phương pháp:

- Tính toán hàm số trực tiếp từ đường cong dạng hàm toán học:  $f(x,y,z)=0$

Sau mỗi bước nội suy sẽ được kiểm tra sai lệch vị trí thực so với đường cong để điều chỉnh chuyển động trên các trục.

- Tính toán theo các thông số của đường cong là hàm của thời gian thực.

- Chuyển các phương trình riêng của hệ sang phương trình vi phân để tính toán bằng số, sẽ tìm ra các giá trị chạy dao trên từng trục thông qua tổng vi phân. Phương pháp này gọi là phương pháp phân tích vi phân số DDA. (Digital- Differential- Analyse);



# Tạo hình bề mặt trên máy công cụ điều khiển số

## 2- Các dạng nội suy.

Theo biên dạng nội suy tạo ra, có:

- Nội suy tuyến tính ( thẳng).
- Nội suy phi tuyến ( tròn, parabol).

Đối với nội suy thẳng và nội suy vòng:

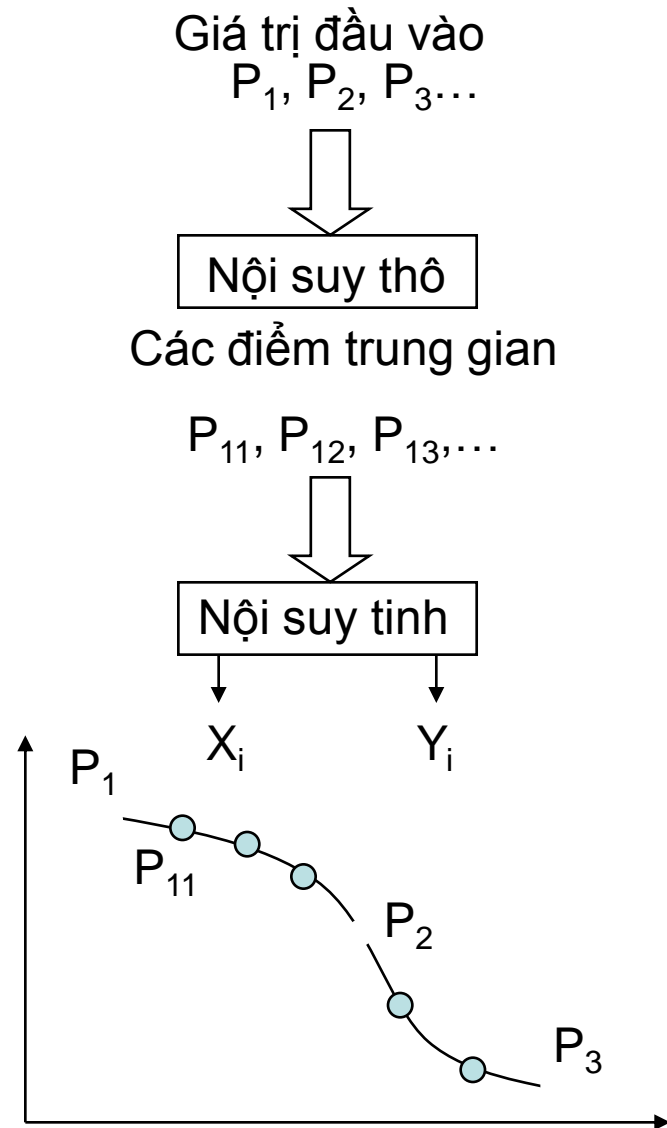
- Nội suy thẳng theo 2 trong  $n$  trục.
- Nội suy thẳng theo  $n$  trong  $n$  trục.
- Nội suy vòng theo 2 trong  $n$  trục.
- Nội suy vòng theo 2 trong  $n$  trục đồng thời với nội suy thẳng theo một trục vuông góc với mặt phẳng của đường tròn nội suy (nội suy đường xoắn vít).

# Tạo hình bề mặt trên máy công cụ điều khiển số

## 2- Phương pháp nội suy.

a-Các hệ CNC thực hiện nội suy theo hai mức:

- Nội suy thô: Xác định tọa độ của các điểm trung gian giữa điểm đầu và điểm cuối ( $P_1, P_2, P_3, \dots$ ).
- Nội suy tinh xác: Thực hiện nội suy tuyến tính giữa các điểm trung gian này.



# Tạo hình bề mặt trên máy công cụ điều khiển số

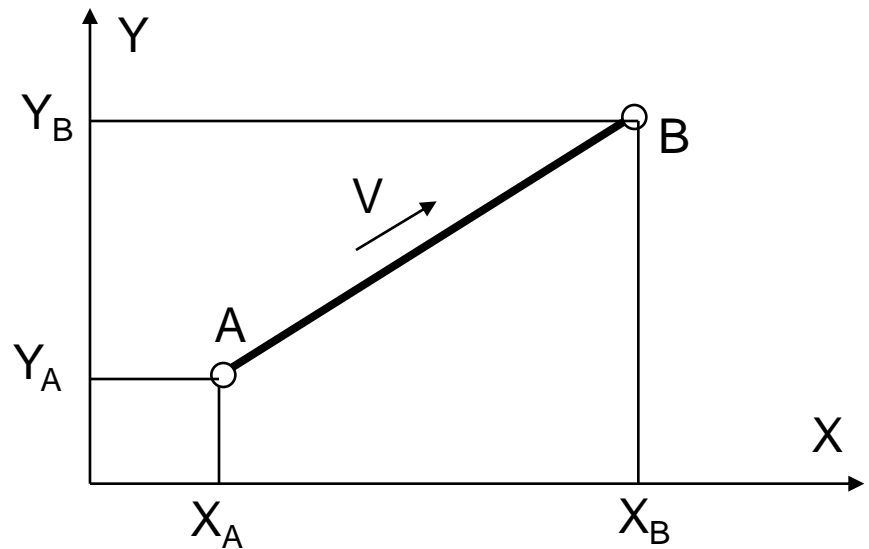
b- Phương pháp nội suy DDA.

- Phương pháp nội suy tuyến tính.

$$X(t) = X_A + \frac{X_B - X_A}{T} t$$

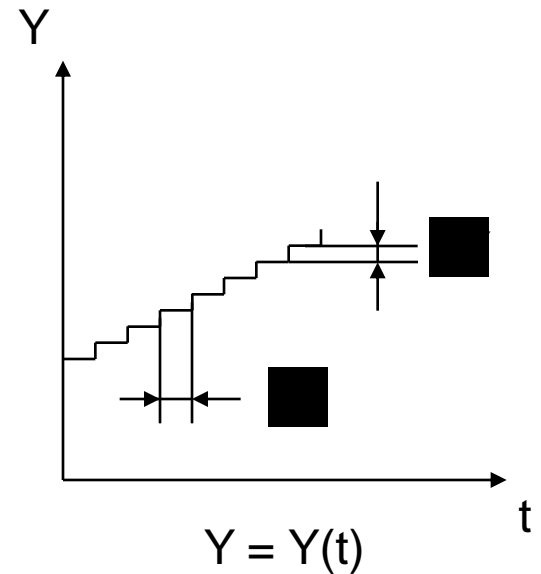
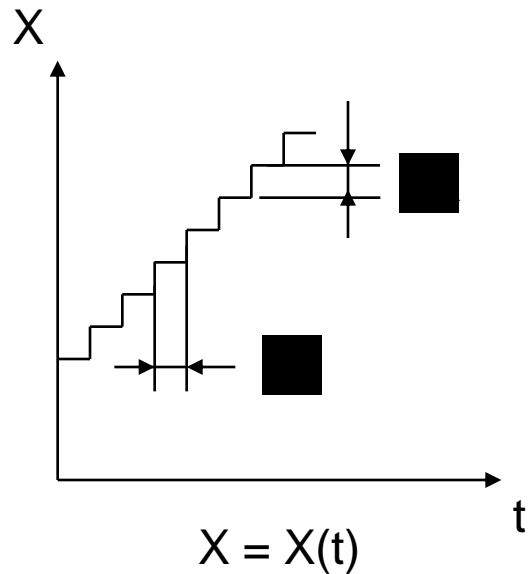
$$Y(t) = Y_A + \frac{Y_B - Y_A}{T} t$$

$$T = \frac{L_{AB}}{V} = \frac{\sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}}{V}$$



# Tạo hình bề mặt trên máy công cụ điều khiển số

Chia thời gian  $T$  thành các khoảng  $\Delta t = T/N$  đủ nhỏ.



Thay phép tích phân bằng phép cộng:

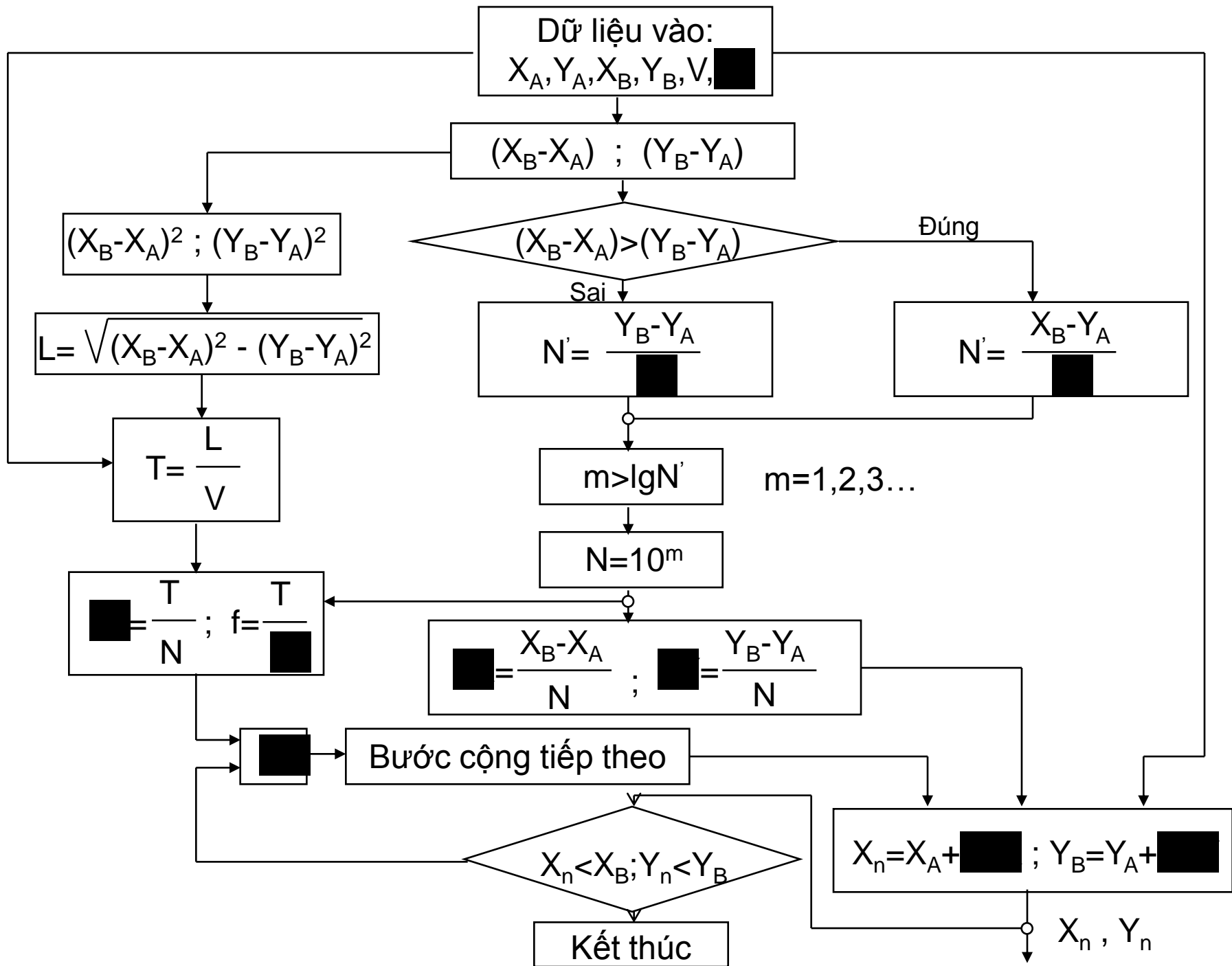
$$X(t) \approx X_A + \frac{X_B - X_A}{N} n$$

$$Y(t) \approx Y_A + \frac{Y_B - Y_A}{N} n$$

$$X_n = X_A + \frac{X_B - X_A}{N} n$$

$$Y_n = Y_A + \frac{Y_B - Y_A}{N} n$$





# • Phương pháp nội suy vòng

Phương trình đường tròn.

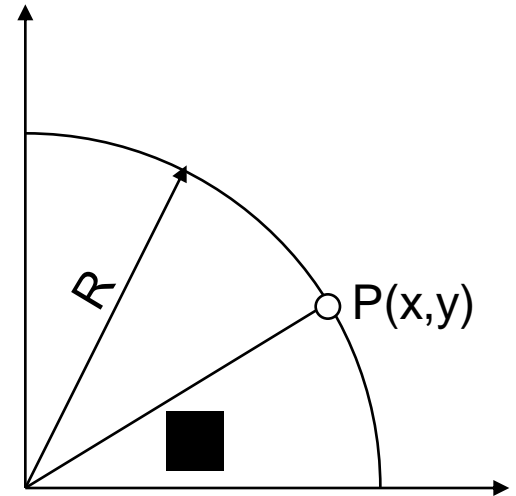
$$\begin{aligned} X &= R \cos \theta \\ Y &= R \sin \theta \end{aligned}$$

Gọi T là thời gian chạy hết toàn vòng.

$$\theta = \frac{2\pi t}{T}$$

$$\begin{aligned} X(t) &= R \cos \frac{2\pi t}{T} \\ Y(t) &= R \sin \frac{2\pi t}{T} \end{aligned}$$

$$\begin{aligned} \frac{dX}{dt} &= -\frac{2\pi}{T} R \sin \frac{2\pi t}{T} = -\frac{2\pi}{T} Y(t) \\ \frac{dY}{dt} &= \frac{2\pi}{T} R \cos \frac{2\pi t}{T} = \frac{2\pi}{T} X(t) \end{aligned}$$



$$\begin{aligned} X(t) &= X_A + \frac{2\pi}{T} \dots \\ Y(t) &= Y_A + \frac{2\pi}{T} \dots \end{aligned}$$

Với  $\Delta t = T/N$

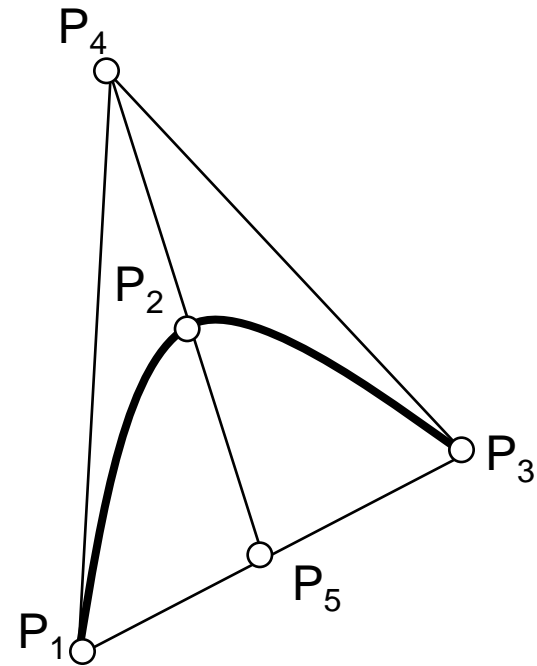
$$\begin{aligned} X(t) &= X_A + \frac{2\pi}{N} \dots(i) \\ Y(t) &= Y_A + \frac{2\pi}{N} \dots(i) \end{aligned}$$

# • Nội suy Parabol

- Chủ yếu dùng trên các máy nhiều trục(4,5...)
- Một đường parabol không gian được tạo bởi 3 điểm  $P_1, P_2, P_3$ .

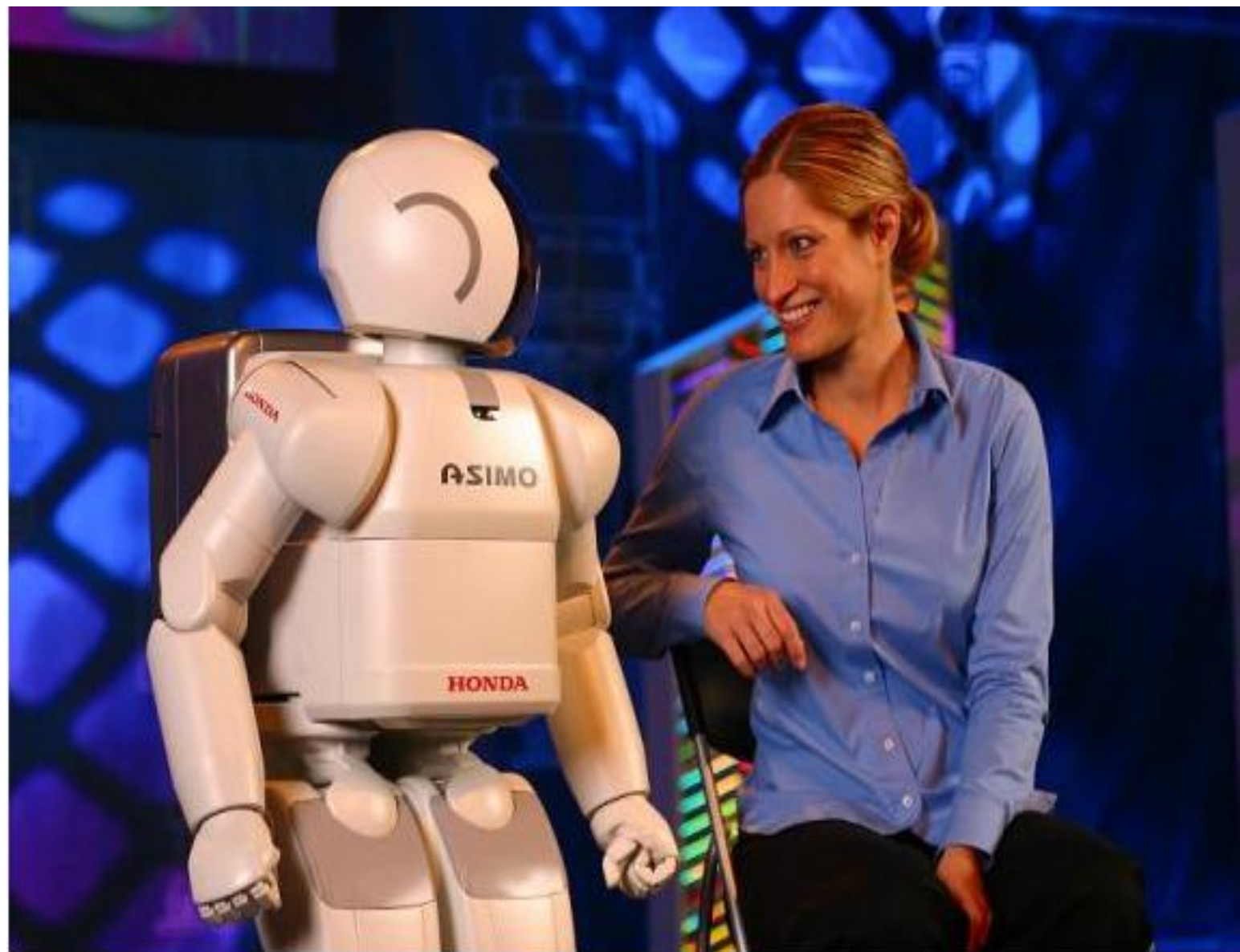
$P_2$ : là điểm giữa của  $P_1$  và  $P_5$ .

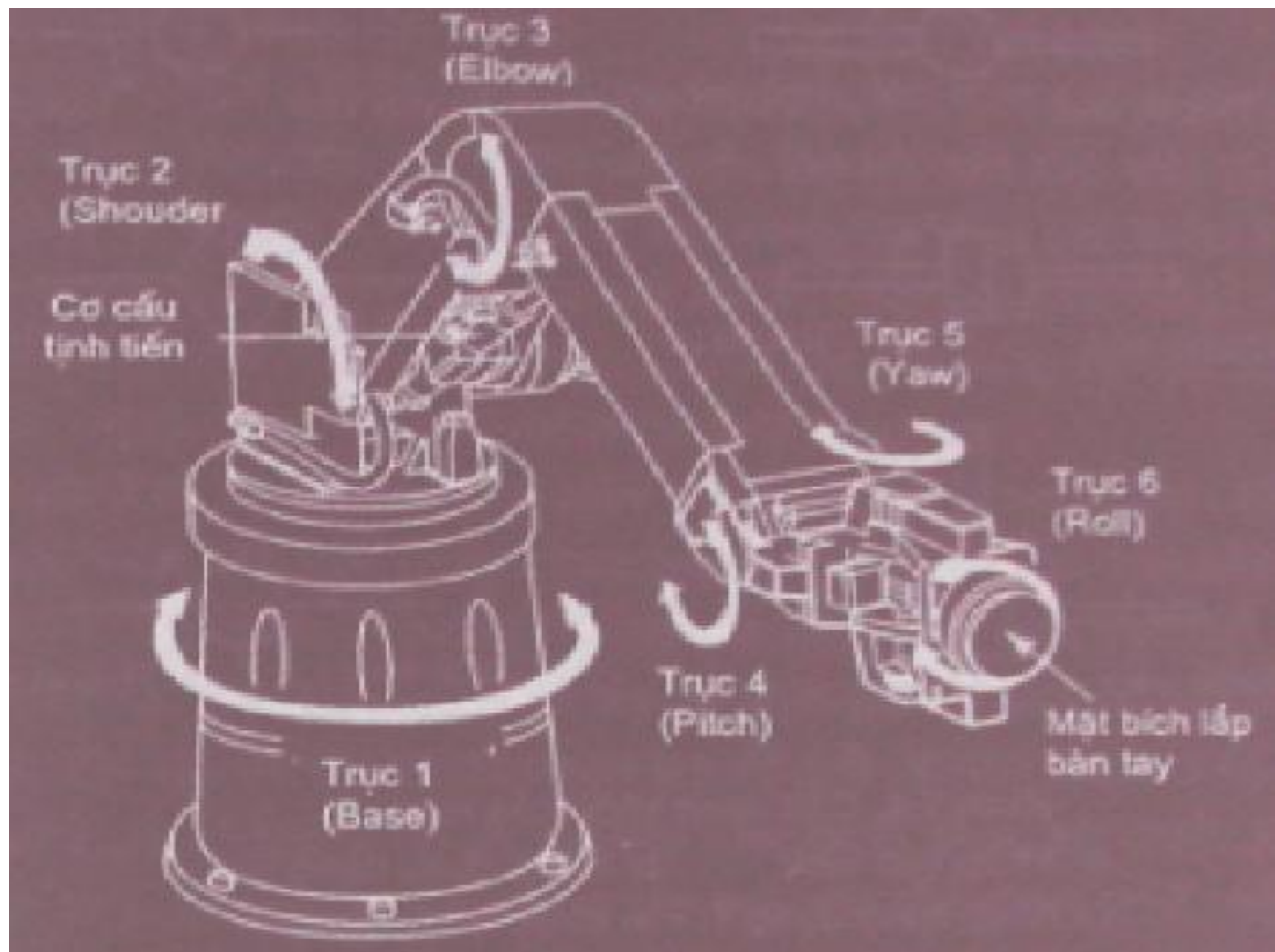
$P_5$ : là điểm giữa của  $P_1$  và  $P_3$ .



## •Nội suy ghép nối.

- Sự nối ghép và chuyển tiếp giữa các đường cong được thực hiện thông qua các tiếp tuyến.
- Với kiểu nội suy này các dạng hình học phức tạp có thể được lập trình bằng cách sử dụng các khối dữ liệu chương trình ít hơn một cách đáng kể hơn so với khi sử dụng nội suy thẳng.







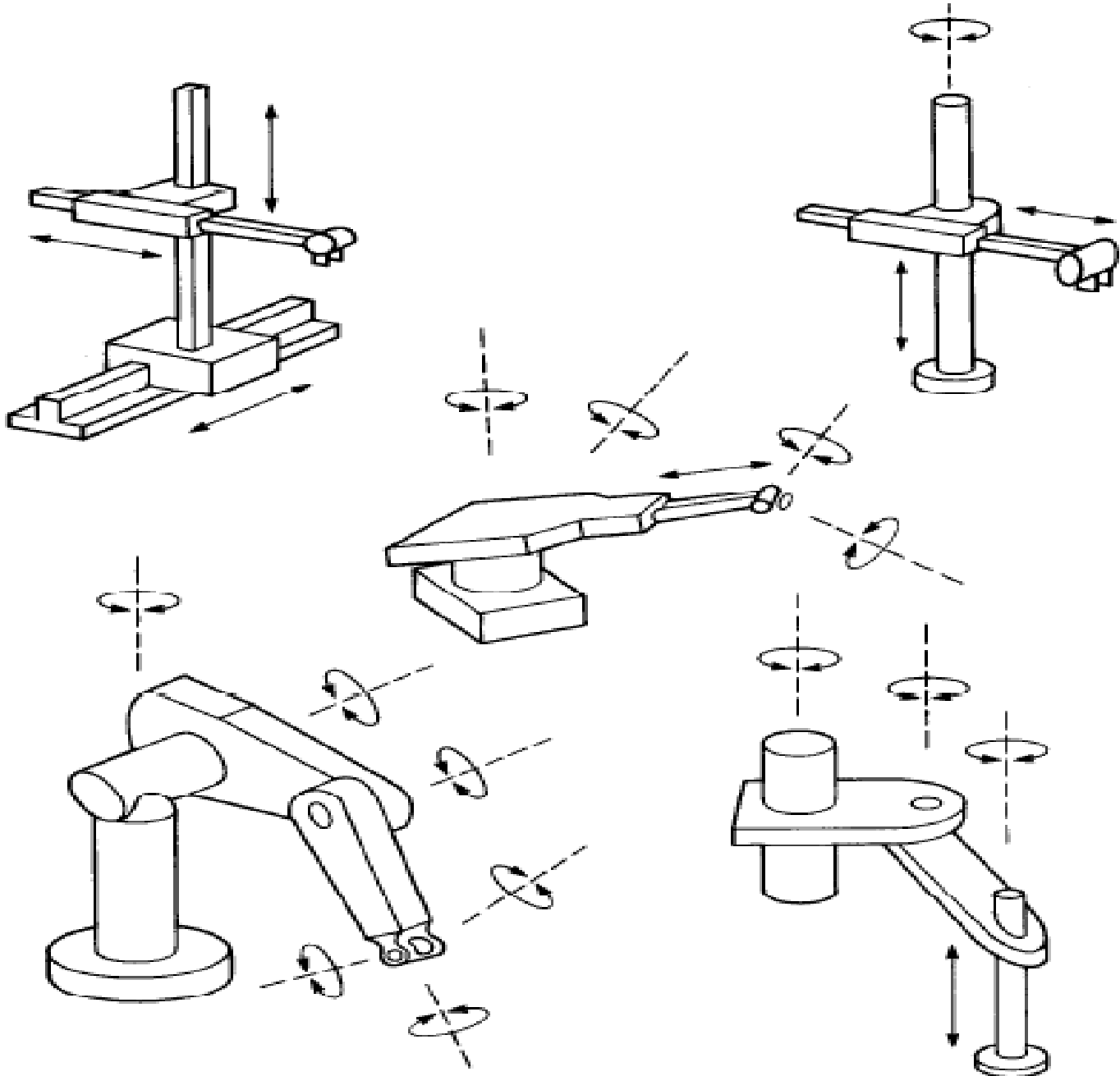


# Tự động hóa linh hoạt (FMS)



Có nhiều, robots, trong băng chuyền, máy gia công CNC, các trạm sơn, hậu cần. Tải xuống từ quản lý sản xuất, kết nối với quản trị mạng.





## ***Số lượng Robot sản xuất ở các nước phát triển***

<b>Nước SX</b>	<b>Năm 1990</b>	<b>Năm 1994</b>	<b>Năm 1998 (Dự tính)</b>
Nhật	60.118	29.756	67.000
Mỹ	4.327	7.634	11.100
Đức	5.845	5.125	8.600
Ý	2.500	2.408	4.000
Pháp	1.488	1.197	2.000
Anh	510	1.086	1.500
Hàn quốc	1.000	1.200	

## ***Các khái niệm và định nghĩa về robot công nghiệp :***

### ***Định nghĩa robot công nghiệp :***

#### ***Định nghĩa theo tiêu chuẩn AFNOR (Pháp) :***

Robot công nghiệp là một cơ cấu chuyển động tự động có thể lập trình, lặp lại các chương trình, tổng hợp các chương trình đặt ra trên các trục tọa độ; có khả năng định vị, định hướng, di chuyển các đối tượng vật chất : chi tiết, dao cụ, gá lắp . . . theo những hành trình thay đổi đã chương trình hoá nhằm thực hiện các nhiệm vụ công nghệ khác nhau.

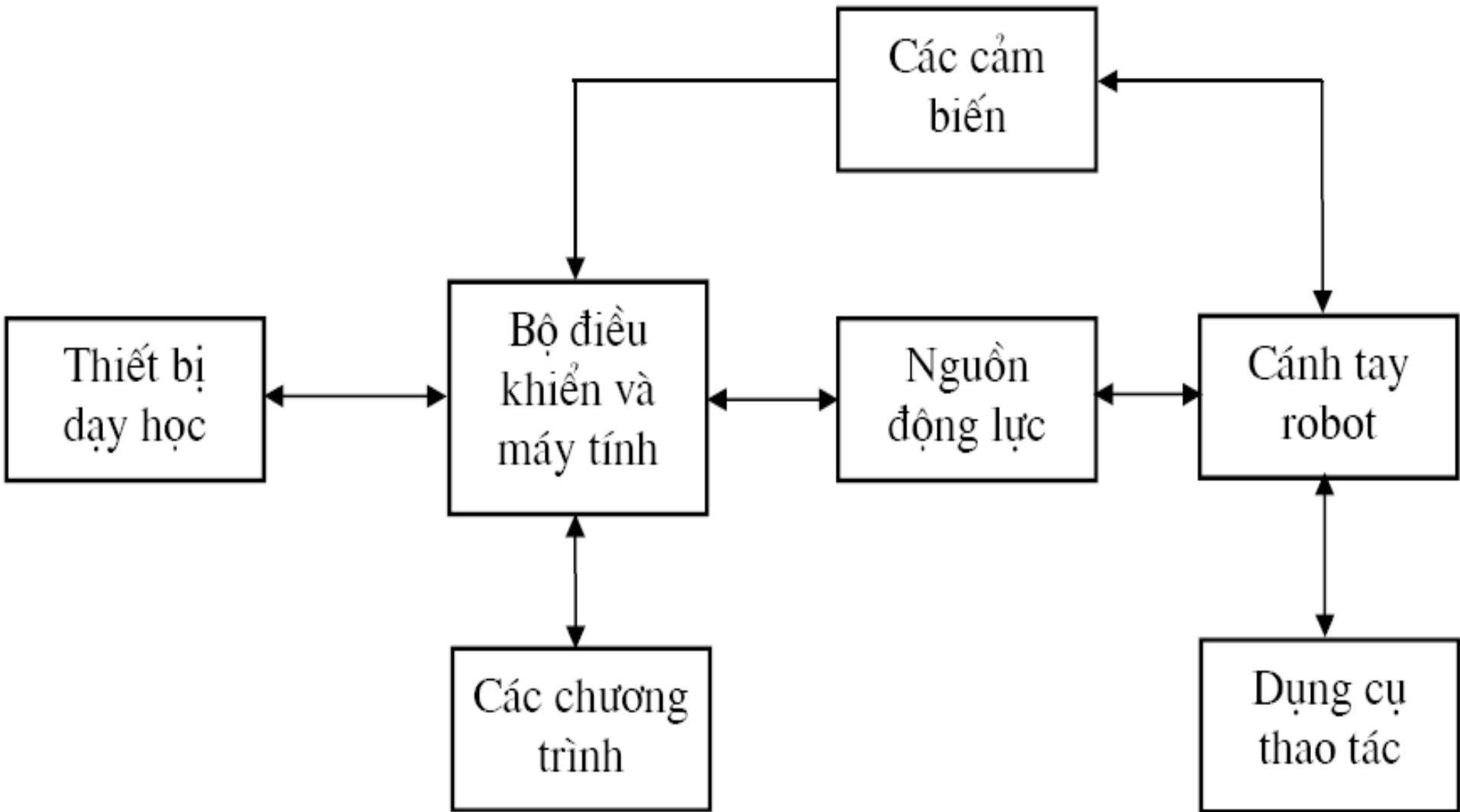
#### ***Định nghĩa theo RIA (Robot institute of America) :***

Robot là một tay máy vạn năng có thể lập lại các chương trình được thiết kế để di chuyển vật liệu, chi tiết, dụng cụ hoặc các thiết bị chuyên dùng thông qua các chương trình chuyển động có thể thay đổi để hoàn thành các nhiệm vụ khác nhau.

#### ***Định nghĩa theo ГOCT 25686-85 (Nga) :***

Robot công nghiệp là một máy tự động, được đặt cố định hoặc di động được, liên kết giữa một tay máy và một hệ thống điều khiển theo chương trình, có thể lập trình lại để hoàn thành các chức năng vận động và điều khiển trong quá trình sản xuất.

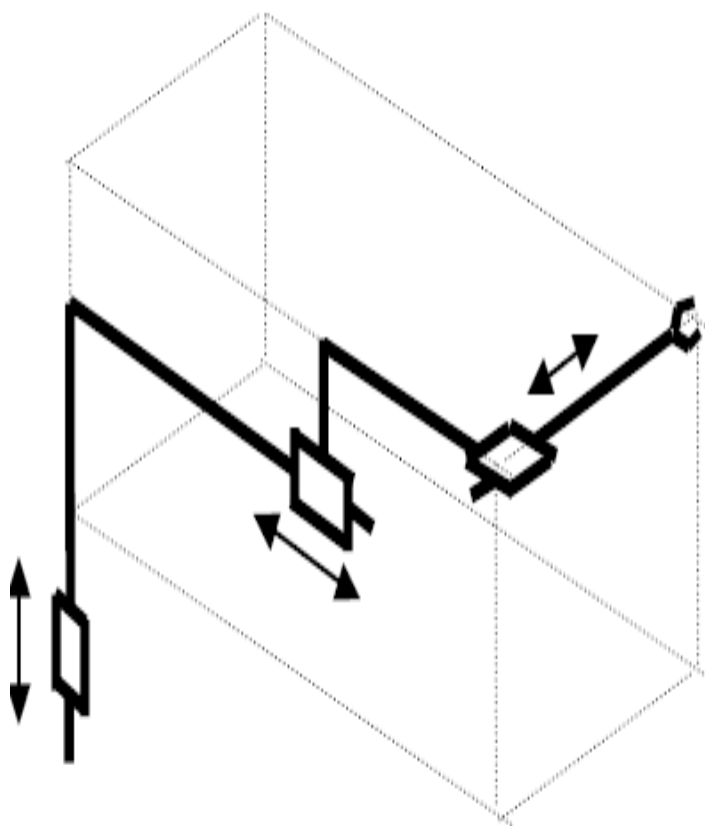
## *Cấu trúc cơ bản của robot công nghiệp :*



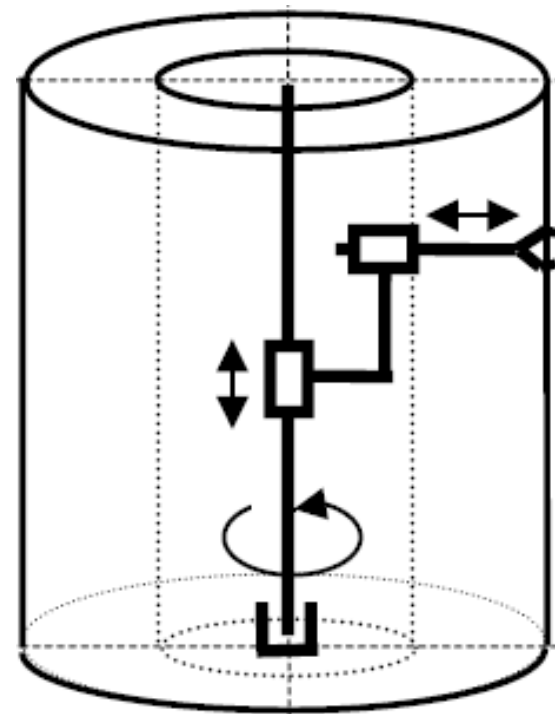
*Các thành phần chính của hệ thống robot.*

# *Kết cấu của tay máy*

*Robot kiểu tọa độ Đề các* : là tay máy có 3 chuyển động cơ bản tịnh tiến theo phương của các trục hệ tọa độ gốc (cấu hình T.T.T). Trường công tác có dạng khối chữ nhật. Do kết cấu đơn giản, loại tay máy này có độ cứng vững cao, độ chính xác cơ khí dễ đảm bảo vì vậy nó thường dùng để vận chuyển phôi liệu, lắp ráp, hàn trong mặt phẳng ...

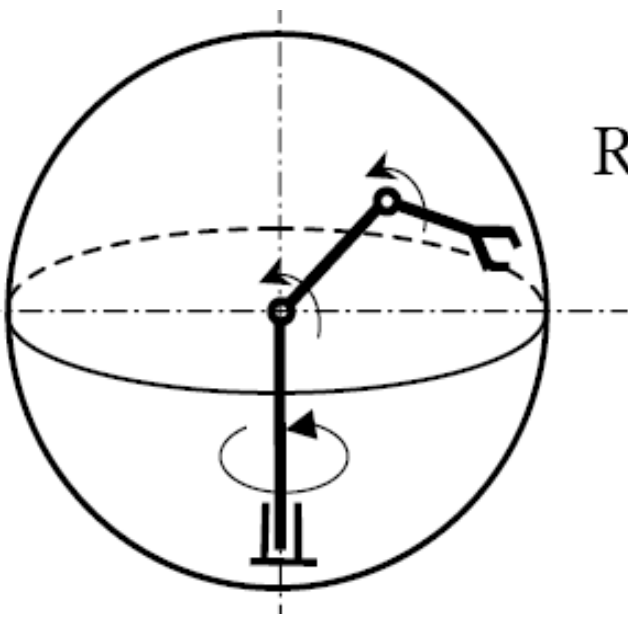


***Robot kiểu tọa độ trụ*** : Vùng làm việc của robot có dạng hình trụ rỗng. Thường khớp thứ nhất chuyển động quay. Ví dụ robot 3 bậc tự do, cấu hình R.T.T như hình vẽ. Có nhiều robot kiểu tọa độ trụ như : robot Versatran của hãng AMF (Hoa Kỳ).

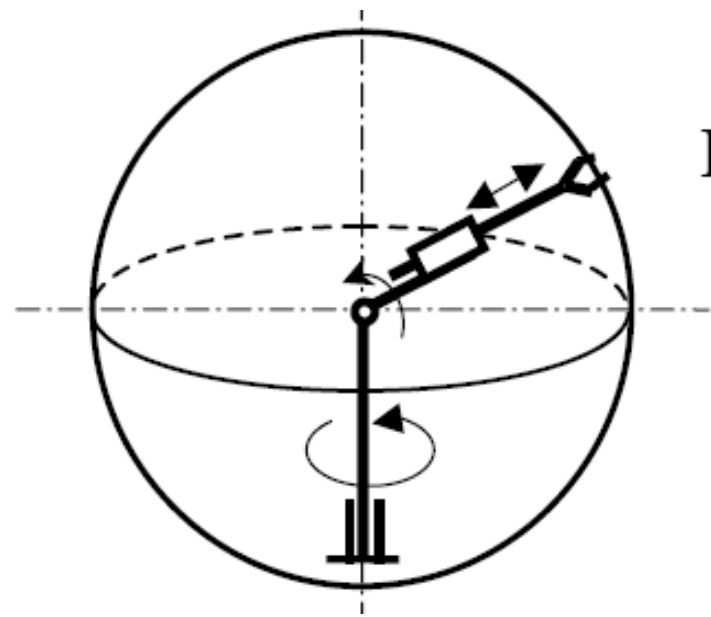


# *Robot kiểu tọa độ cầu :*

*Robot kiểu tọa độ cầu :* Vùng làm việc của robot có dạng hình cầu. thường độ cứng vững của loại robot này thấp hơn so với hai loại trên. Ví dụ robot 3 bậc tự do, cấu hình R.R.R hoặc R.R.T làm việc theo kiểu tọa độ cầu

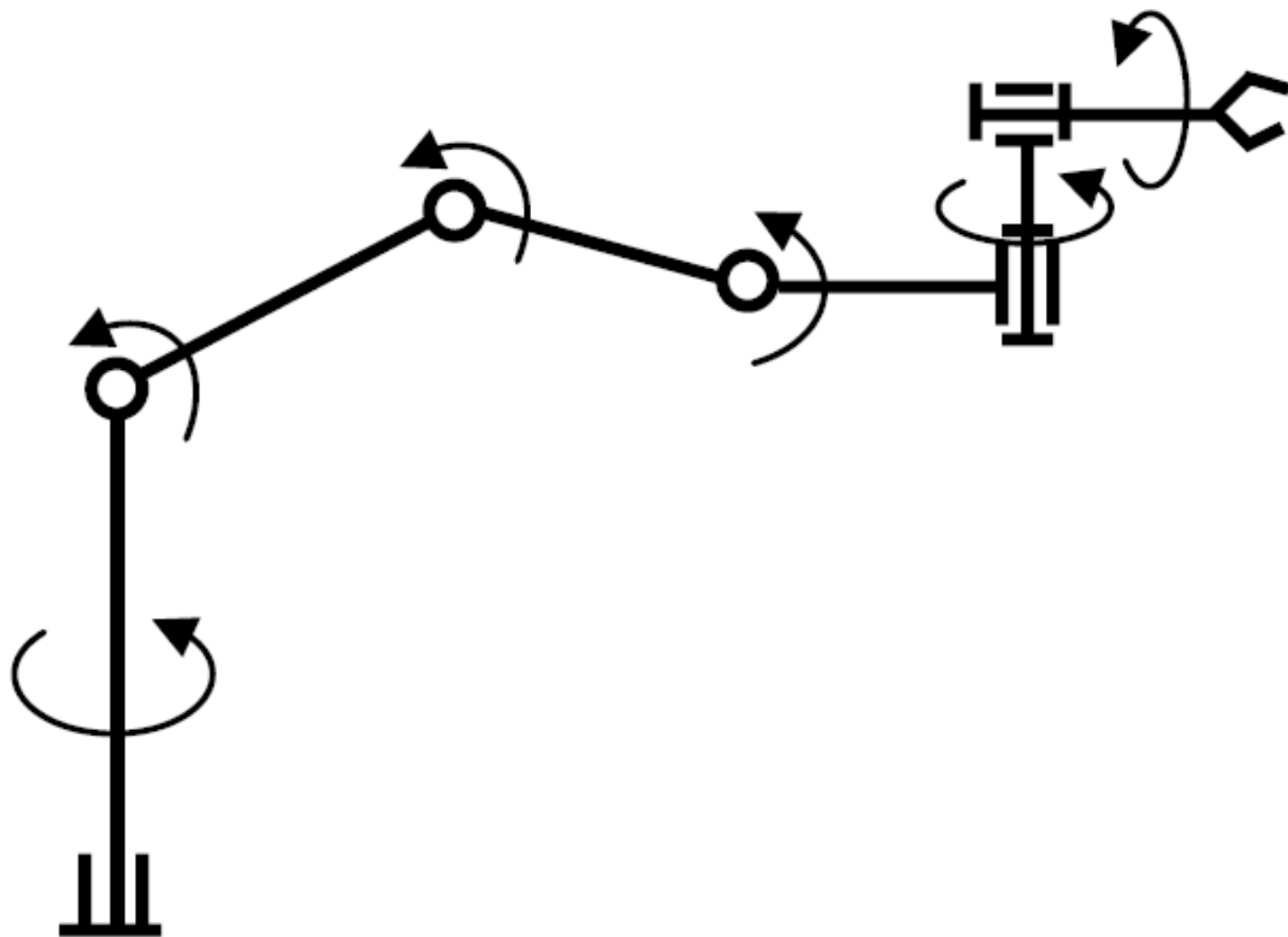


R.R.R



R.R.T

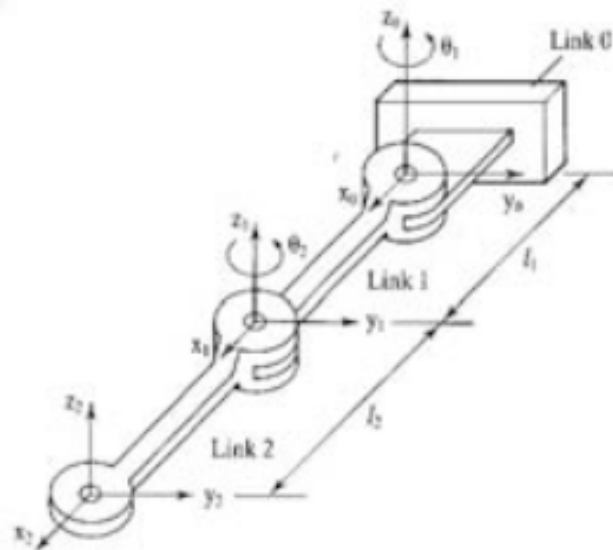
*Robot kiểu tọa độ góc (Hệ tọa độ phỏng sinh)*



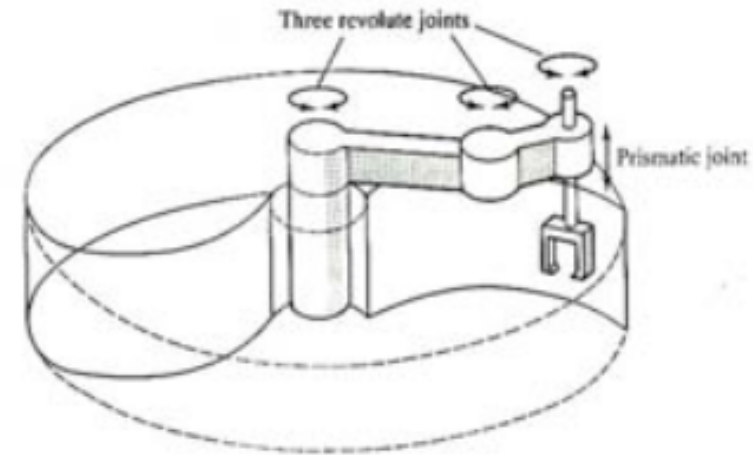




# Four-Joint (SCARA) Manipulator



Arm with 3  
revolute links



Workspace

## • Frames of Reference

- Motor coordinates
- Joint coordinates
- World coordinates
- Tool coordinates

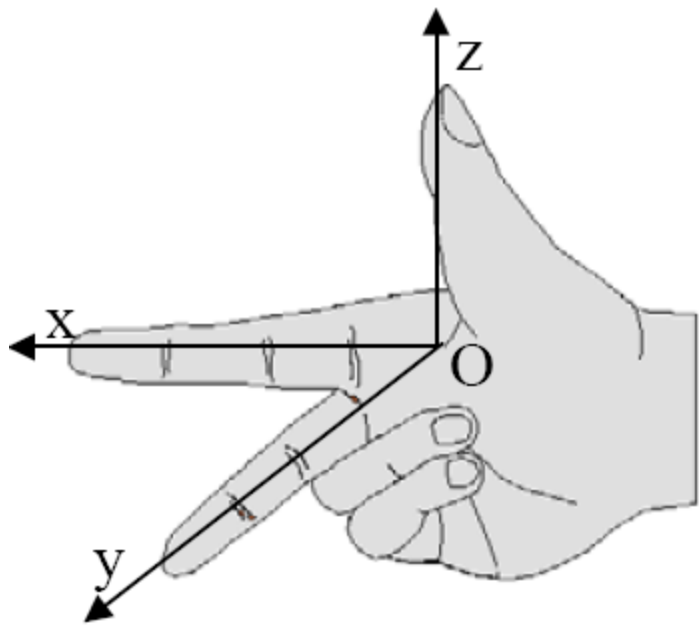
# *Bậc tự do của robot (DOF : Degrees Of Freedom)*

$$W = 6n - \sum_{i=1}^5 ip_i$$

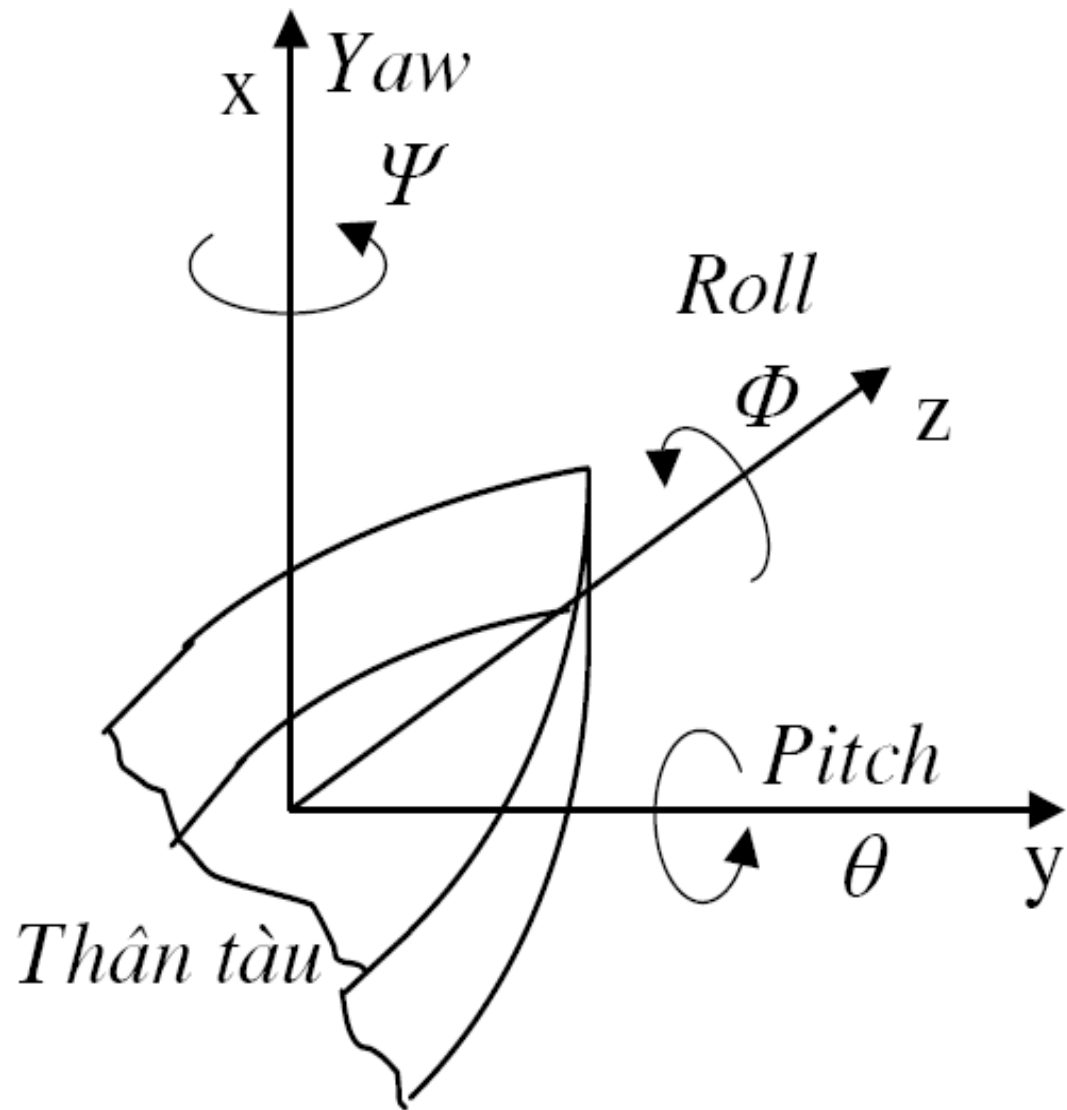
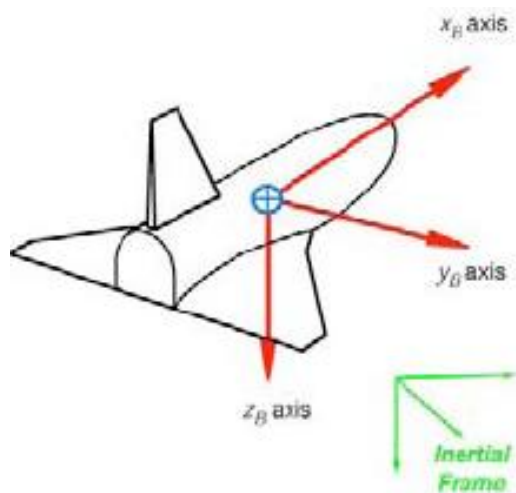
$n$  - Số khâu động;

$p_i$  - Số khớp loại  $i$  ( $i = 1, 2, \dots, 5$  : Số bậc tự do bị hạn chế)

Đối với các cơ cấu có các khâu được nối với nhau bằng khớp quay hoặc tịnh tiến (khớp động loại 5) thì số bậc tự do bằng với số khâu động. Đối với cơ cấu hở, số bậc tự do bằng tổng số bậc tự do của các khớp động.

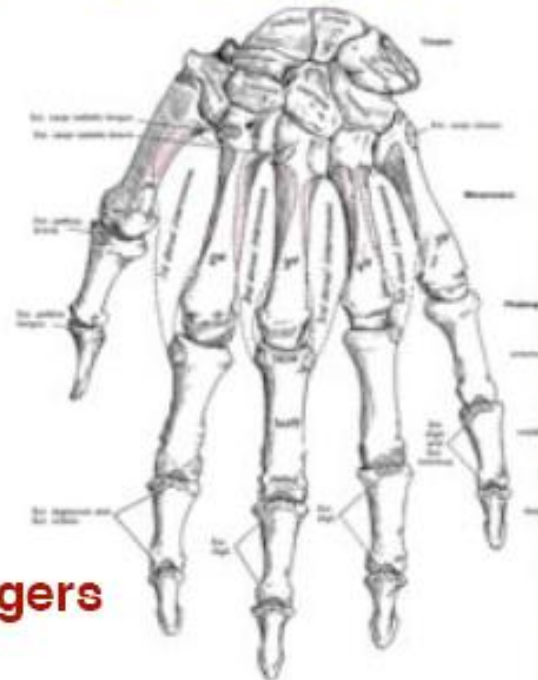
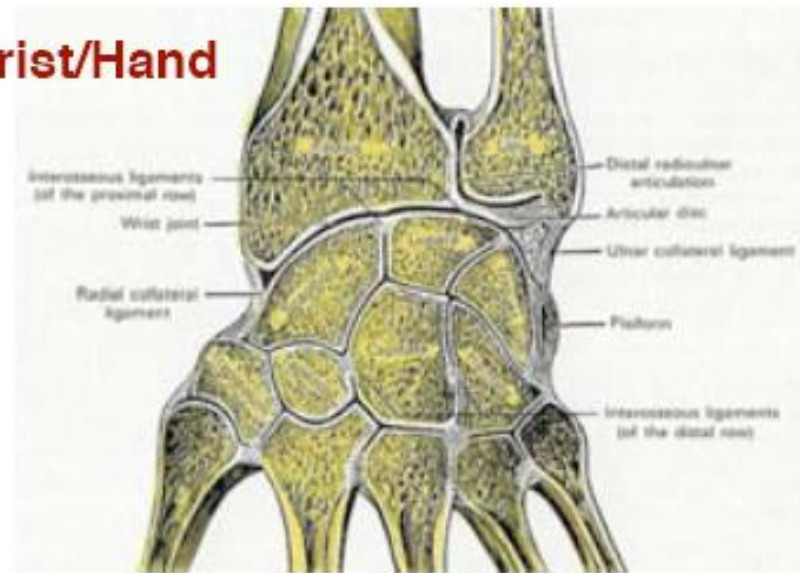


*Quy tắc bàn tay phải*



*Thân tàu*

## Wrist/Hand



## Hand/Fingers

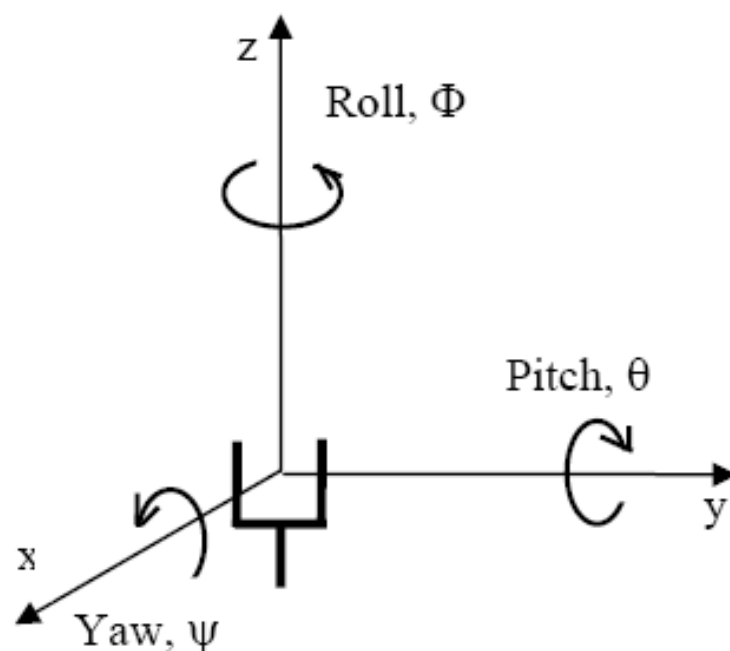
Bàn tay người có 27 khúc xương với 22 bậc tự do rất phức tạp

Cũng ở phần trước đã trình bày, ngoài ba chuyển động cơ bản để thực hiện chuyển động định vị, tay máy sẽ được bổ sung tối đa là ba chuyển động định hướng dạng ba chuyển động quay quanh ba trục vuông góc

Chuyển động xoay cổ tay (ROLL), góc quay  $\rho$

Chuyển động gập cổ tay (PITCH), góc quay  $\delta$

Chuyển động lắc cổ tay (YAW), góc quay  $\varepsilon$



*Các góc quay Roll-Pitch và Yaw của bàn tay Robot.*

$$T_6 = \begin{bmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tổng quát, ma trận  $T_6$  có thể biểu diễn gọn hơn như sau :

$$T_6 = \left[ \begin{array}{ccc|c} \text{Ma trận định hướng R} & & & \text{Vectơ vị trí p} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

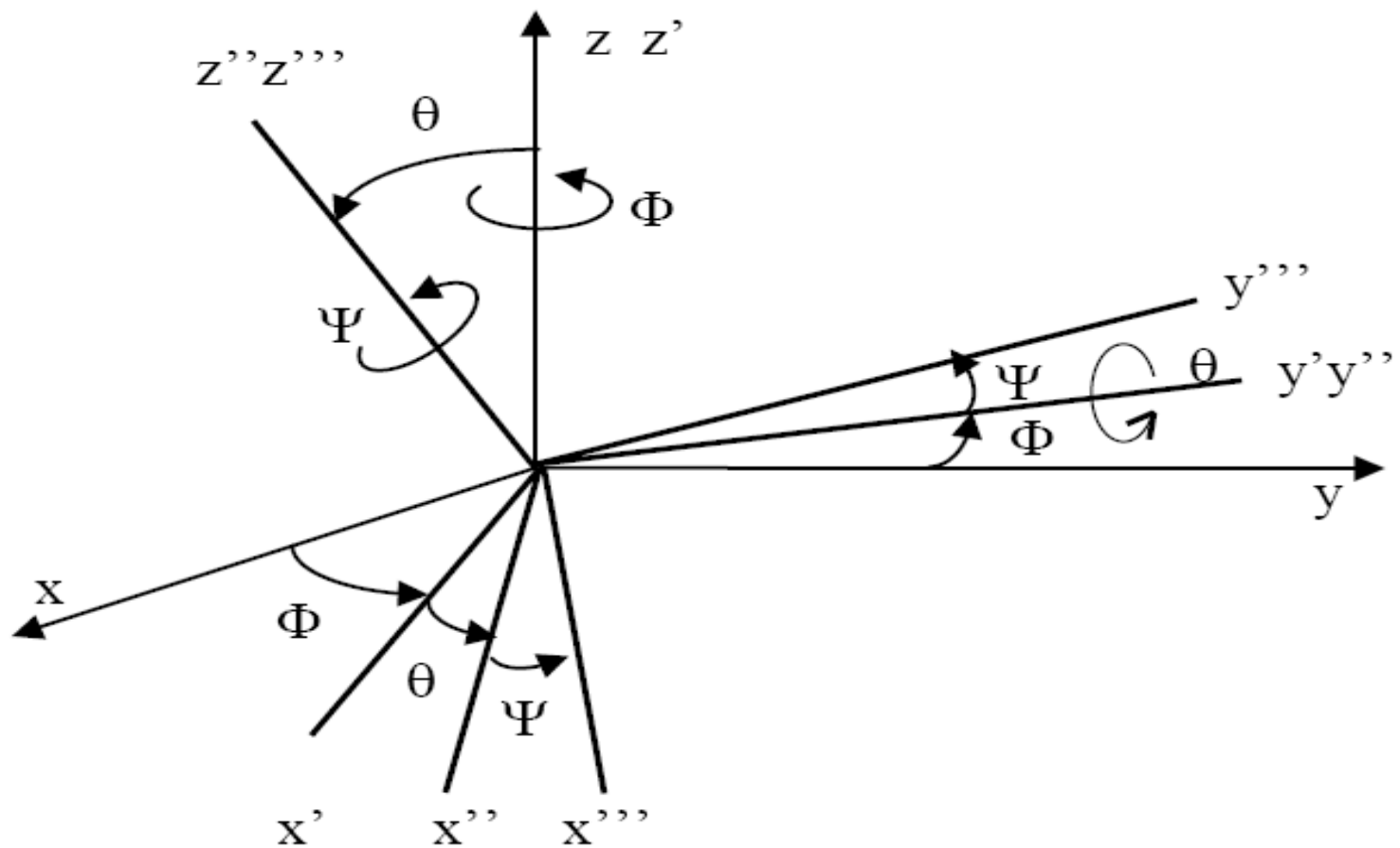
Ma trận R có kích thước 3x3, là ma trận trực giao biểu diễn hướng của bàn kẹp (khâu chấp hành cuối) đối với hệ tọa độ cơ bản. Việc xác định hướng của khâu chấp hành cuối còn có thể thực hiện theo phép quay Euler hay phép quay Roll, Pitch, Yaw.

Trên thực tế, việc định hướng thường là kết quả của phép quay xung quanh các trục  $x$ ,  $y$ ,  $z$ . Phép quay Euler mô tả khả năng định hướng bằng cách :

Quay một góc  $\Phi$  xung quanh trục  $z$ ,

Quay tiếp một góc  $\theta$  xung quanh trục  $y$  mới, đó là  $y'$ ,

cuối cùng quay một góc  $\psi$  quanh trục  $z$  mới, đó là  $z''$



Ta biểu diễn phép quay Euler bằng cách nhân ba ma trận quay với nhau

$$\text{Euler}(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \text{Rot}(y, \theta) \text{Rot}(z, \psi)$$

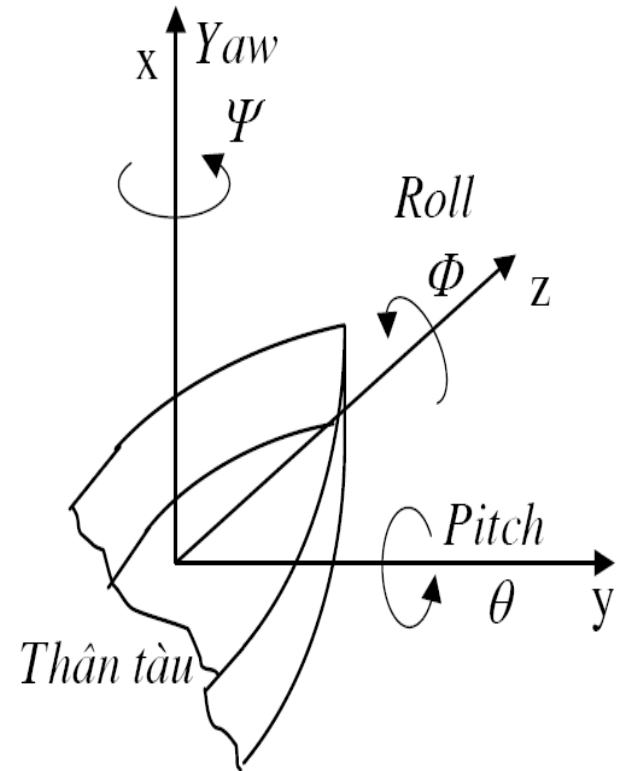
Nói chung, kết quả của phép quay phụ thuộc chặt chẽ vào thứ tự quay, tuy nhiên với phép quay Euler, nếu thực hiện theo thứ tự ngược lại, nghĩa là quay góc  $\psi$  quanh  $z$  đầu tiên, quay góc  $\theta$  quanh  $y$  và cuối cùng quay góc  $\Phi$  quanh  $z$  cũng đưa đến kết quả tương đương (trong cùng hệ qui chiếu).

$$\begin{aligned} \text{Euler}(\Phi, \theta, \psi) &= \text{Rot}(z, \Phi) \begin{bmatrix} \text{Cos}\theta & 0 & \text{sin}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sin}\theta & 0 & \text{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{cos}\psi & -\text{sin}\psi & 0 & 0 \\ \text{sin}\psi & \text{cos}\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \text{cos}\Phi & -\text{sin}\Phi & 0 & 0 \\ \text{sin}\Phi & \text{cos}\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \text{Cos}\theta \text{cos}\psi & -\text{Cos}\theta \text{sin}\psi & \text{sin}\theta & 0 \\ \text{sin}\psi & \text{cos}\psi & 0 & 0 \\ -\text{sin}\theta \text{cos}\psi & \text{sin}\theta \text{sin}\psi & \text{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \text{cos}\Phi \text{Cos}\theta \text{cos}\psi - \text{sin}\Phi \text{sin}\psi & -\text{cos}\Phi \text{Cos}\theta \text{sin}\psi - \text{sin}\Phi \text{cos}\psi & \text{cos}\Phi \text{sin}\theta & 0 \\ \text{sin}\Phi \text{Cos}\theta \text{cos}\psi + \text{cos}\Phi \text{sin}\psi & -\text{sin}\Phi \text{Cos}\theta \text{sin}\psi + \text{cos}\Phi \text{cos}\psi & \text{sin}\Phi \text{sin}\theta & 0 \\ -\text{sin}\theta \text{cos}\psi & \text{sin}\theta \text{sin}\psi & \text{cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$



# *Phép quay Roll-Pitch-Yaw*

Ta tưởng tượng, gắn hệ tọa độ xyz lên thân một con tàu. Dọc theo thân tàu là trục z, Roll là chuyển động lắc của thân tàu, tương đương với việc quay thân tàu một góc  $\Phi$  quanh trục z. Pitch là sự bồng bênh, tương đương với quay một góc  $\theta$  xung quanh trục y và Yaw là sự lệch hướng, tương đương với phép quay một góc  $\psi$  xung quanh trục x



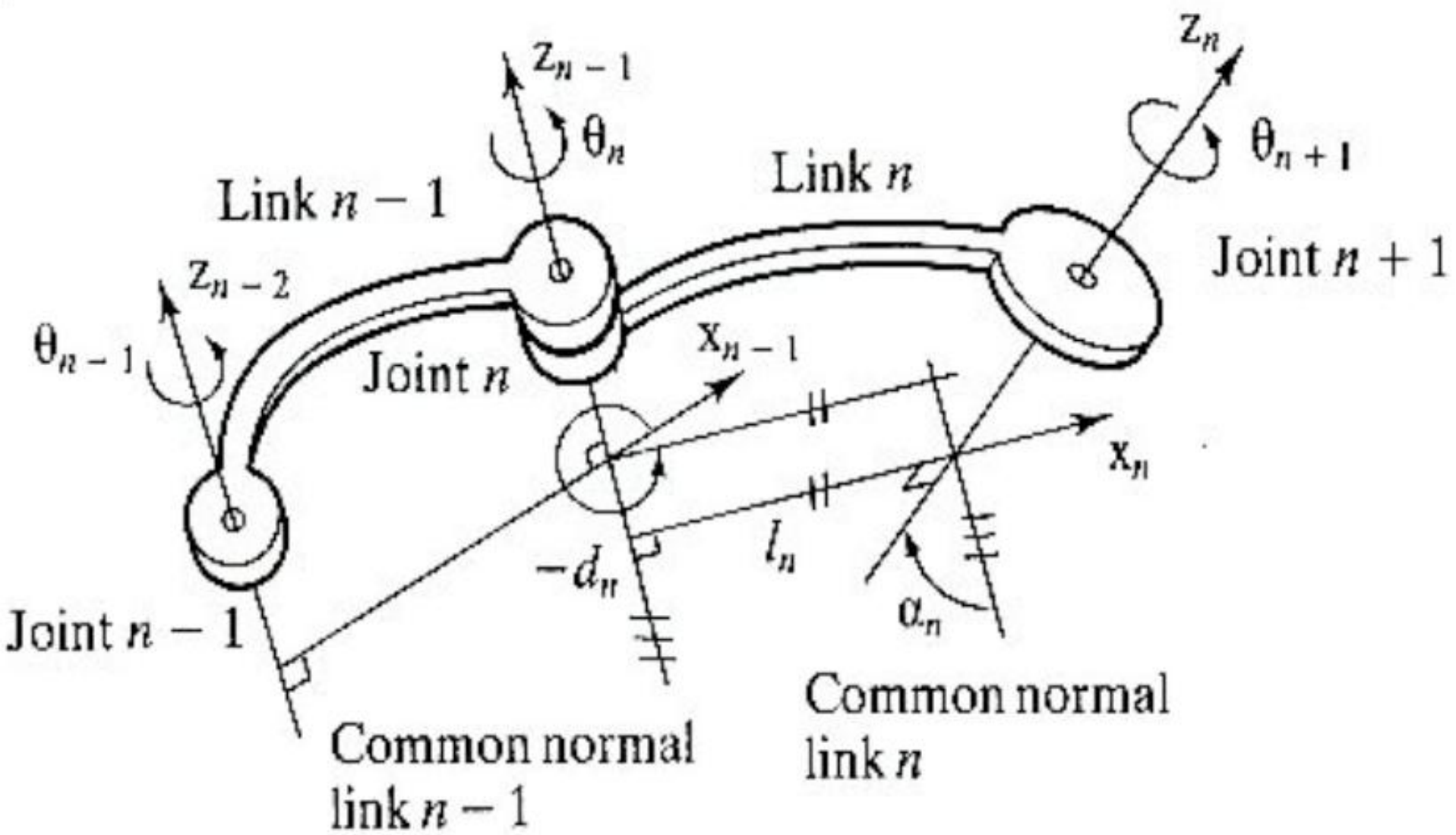
$$RPY(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \text{Rot}(y, \theta) \text{Rot}(x, \psi)$$

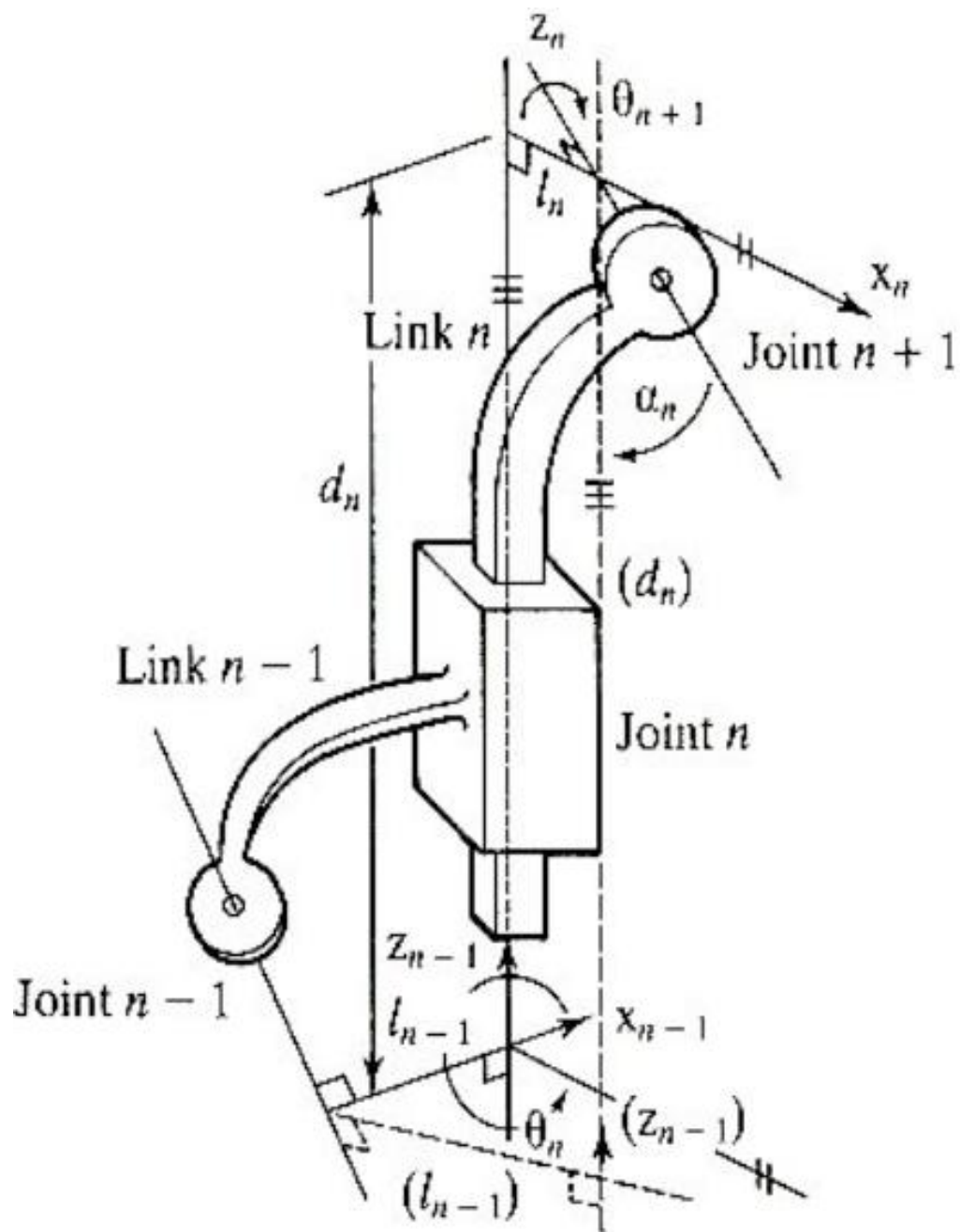
$$\text{RPY}(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

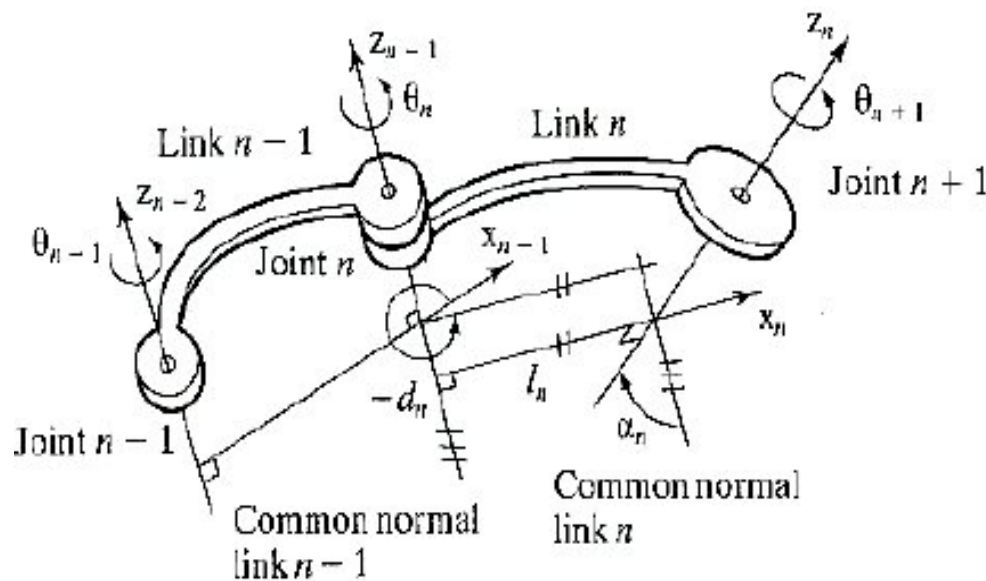
$$= \begin{bmatrix} \cos\Phi & -\sin\Phi & 0 & 0 \\ \sin\Phi & \cos\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta\sin\psi & \sin\theta\cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\Phi\cos\theta & \cos\Phi\sin\theta\sin\psi - \sin\Phi\cos\psi & \cos\Phi\sin\theta\cos\psi + \sin\Phi\sin\psi & 0 \\ \sin\Phi\cos\theta & \sin\Phi\sin\theta\sin\psi + \cos\Phi\cos\psi & \sin\Phi\sin\theta\cos\psi - \cos\Phi\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$









## 4 link parameters

- Length of the link between rotational axes,  $l$ , along the common normal (fixed)
- Twist angle between axes,  $\alpha$  (fixed)
- Angle between 2 links,  $\theta$  (revolute)
- Distance between links,  $d$  (prismatic)

- Like Euler angles, transformational effects of the 4 link parameters are defined in a specific application sequence:  $\alpha, l, d, \theta$

$$\mathbf{A}_n = \text{Rot}(z_n, \theta_n) \text{Trans}(z_n, d_n) \text{Trans}(x_n, l_n) \text{Rot}(x_n, \alpha_n)$$

- $\text{Rot}(z_n, \theta_n)$  = rotation of  $\theta_n$  rad about the  $z_n$  axis
- $\text{Trans}(x_n, l_n)$  = translation of  $l_n$  m along the  $x_n$  axis

• or

$$\mathbf{A}_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n & 0 & 0 \\ \sin \theta_n & \cos \theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_n & -\sin \alpha_n & 0 \\ 0 & \sin \alpha_n & \cos \alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & l_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & l_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

*Để mô tả mối quan hệ giữa các khâu ta gắn vào mỗi khâu một hệ tọa độ. Nguyên tắc chung để gắn hệ tọa độ lên các khâu như sau :*

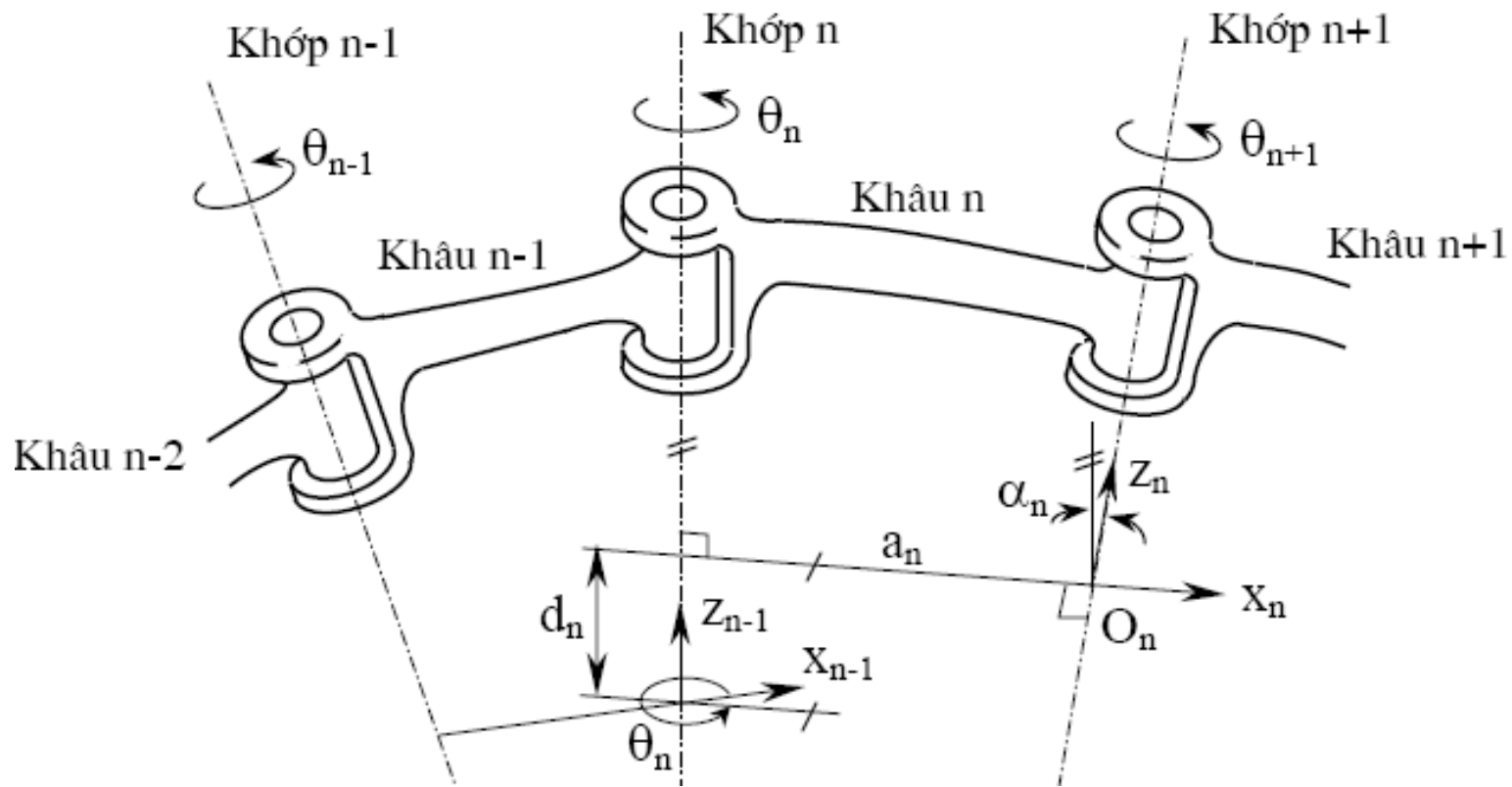
+ Gốc của hệ tọa độ gắn lên khâu thứ  $n$  đặt tại giao điểm của pháp tuyến  $a_n$  với trục khớp thứ  $n+1$ . Trường hợp hai trục khớp cắt nhau, gốc tọa độ sẽ đặt tại chính điểm cắt đó. Nếu các trục khớp song song với nhau, gốc tọa độ được chọn trên trục khớp của khâu kế tiếp, tại điểm thích hợp.

+ Trục  $z$  của hệ tọa độ gắn lên khâu thứ  $n$  đặt dọc theo trục khớp thứ  $n+1$ .

+ Trục  $x$  thường được đặt dọc theo pháp tuyến chung và hướng từ khớp  $n$  đến  $n+1$ .

Trong trường hợp các trục khớp cắt nhau thì trục  $x$  chọn theo tích vectơ  $\vec{z}_n \times \vec{z}_{n-1}$ .

Trường hợp khớp quay thì  $\theta_n$  là các biến khớp, trong trường hợp khớp tịnh tiến thì  $d_n$  là biến khớp và  $a_n$  bằng 0.



**Các thông số  $a_n$ ,  $\alpha_n$ ,  $d_n$  và  $\theta_n$  được gọi là bộ thông số DH.**

Mỗi trục sẽ có hai pháp tuyến với nó, mỗi pháp tuyến dùng cho mỗi khâu (trước và sau một khớp). Vị trí tương đối của hai khâu liên kết như thế được xác định bởi  $d_n$  là khoảng cách giữa các pháp tuyến đo dọc theo trục khớp n và  $\theta_n$  là góc giữa các pháp tuyến đo trong mặt phẳng vuông góc với trục.

$d_n$  và  $\theta_n$  thường được gọi là khoảng cách và góc giữa các khâu.



***Để mô tả mối quan hệ giữa các khâu ta gắn vào mỗi khâu một hệ tọa độ. Nguyên tắc chung để gắn hệ tọa độ lên các khâu như sau :***

+ Gốc của hệ tọa độ gắn lên khâu thứ  $n$  đặt tại giao điểm của pháp tuyến  $a_n$  với trục khớp thứ  $n+1$ . Trường hợp hai trục khớp cắt nhau, gốc tọa độ sẽ đặt tại chính điểm cắt đó. Nếu các trục khớp song song với nhau, gốc tọa độ được chọn trên trục khớp của khâu kế tiếp, tại điểm thích hợp.

+ Trục  $z$  của hệ tọa độ gắn lên khâu thứ  $n$  đặt dọc theo trục khớp thứ  $n+1$ .

+ Trục  $x$  thường được đặt dọc theo pháp tuyến chung và hướng từ khớp  $n$  đến  $n+1$ .

Trong trường hợp các trục khớp cắt nhau thì trục  $x$  chọn theo tích vectơ  $\vec{Z}_n \times \vec{Z}_{n-1}$ .

Trường hợp khớp quay thì  $\theta_n$  là các biến khớp, trong trường hợp khớp tịnh tiến thì  $d_n$  là biến khớp và  $a_n$  bằng 0.

## *Trình tự thiết lập hệ phương trình động học của robot*

### *Chọn hệ tọa độ cơ sở, gắn các hệ tọa độ mở rộng lên các khâu*

Việc gắn hệ tọa độ lên các khâu đóng vai trò rất quan trọng khi xác lập hệ phương trình động học của robot, thông thường đây cũng là bước khó nhất.

+ Giả định một vị trí ban đầu<sup>(♥)</sup> (Home Position) của robot.

+ Chọn gốc tọa độ  $O_0, O_1, \dots$

+ Các trục  $z_n$  phải chọn cùng phương với trục khớp thứ  $n+1$ .

+ Chọn trục  $x_n$  là trục quay của  $z_n$  thành  $z_{n+1}$  và góc của  $z_n$  với  $z_{n+1}$  chính là  $\alpha_{n+1}$ . Nếu  $z_n$  và  $z_{n+1}$  song song hoặc trùng nhau thì ta có thể căn cứ nguyên tắc chung hay chọn  $x_n$  theo  $x_{n+1}$ .

+ Các hệ tọa độ Oxyz phải tuân theo qui tắc bàn tay phải.

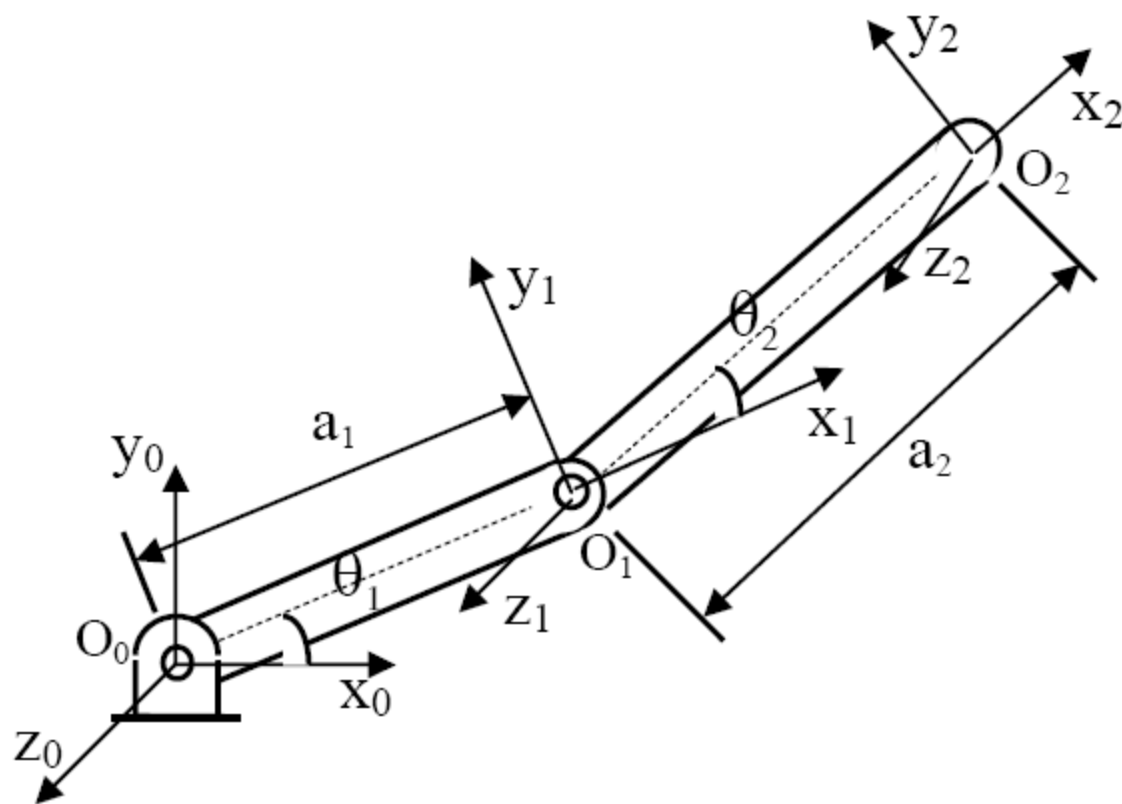
+ Khi gắn hệ tọa độ lên các khâu, phải tuân theo các phép biến đổi của ma trận  $A_n$ . đó là bốn phép biến đổi :  $A_n = \text{Rot}(z, \theta) \text{Trans}(0, 0, d) \text{Trans}(a, 0, 0) \text{Rot}(x, \alpha)$ . Nghĩa là ta coi hệ tọa độ thứ  $n+1$  là biến đổi của hệ tọa độ thứ  $n$ ; các phép quay và tịnh tiến của biến đổi này phải là một trong các phép biến đổi của  $A_n$ , các thông số DH cũng được xác định dựa vào các phép biến đổi này. Trong quá trình gắn hệ tọa độ lên các khâu, nếu xuất hiện phép quay của trục  $z_n$  đối với  $z_{n-1}$  quanh trục  $y_{n-1}$  thì vị trí ban đầu của robot đã giả định là không đúng, ta cần chọn lại vị trí ban đầu khác cho robot.

Lập bảng thông số DH (Denavit Hartenberg).

Dựa vào các thông số DH xác định các ma trận  $A_n$ .

Tính các ma trận  $T$  và viết các phương trình động học của robot

Ví dụ 1 : Xét một tay máy có hai khâu phẳng



# Bảng thông số Denavit-Hartenbert

Khâu	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	$\theta_1^*$	0	$a_1$	0
2	$\theta_2^*$	0	$a_2$	0

$${}^0A_2 = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Như vậy hệ phương trình động học thuận của robot như sau:

$$n_x = \cos (\theta_1 + \theta_2)$$

$$n_y = \sin (\theta_1 + \theta_2)$$

$$n_z = 0$$

$$o_x = -\sin (\theta_1 + \theta_2)$$

$$o_y = \cos (\theta_1 + \theta_2)$$

$$o_z = 0$$

$$a_x = 0$$

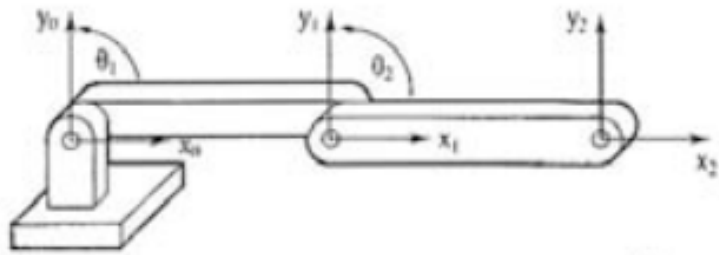
$$a_y = 0$$

$$a_z = 1$$

$$p_x = a_1 \cos \theta_1 + a_2 \cos (\theta_1 + \theta_2)$$

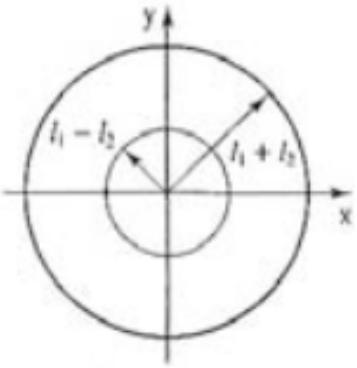
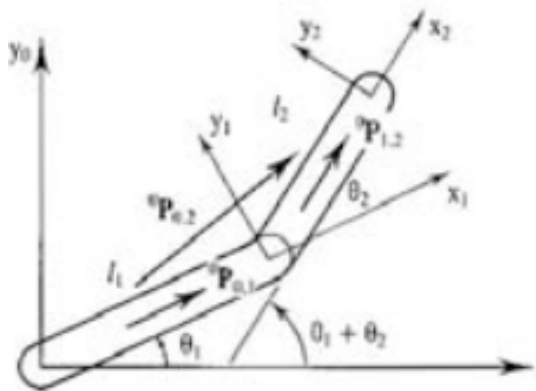
$$p_y = a_1 \sin \theta_1 + a_2 \sin (\theta_1 + \theta_2)$$

$$p_z = 0$$



Parameters for 2-link manipulator

Link variable	$\theta$	$\alpha$	$l$	$d$
1	$\theta_1$	$\theta_1$	0	$l_1$
2	$\theta_2$	$\theta_2$	0	$l_2$

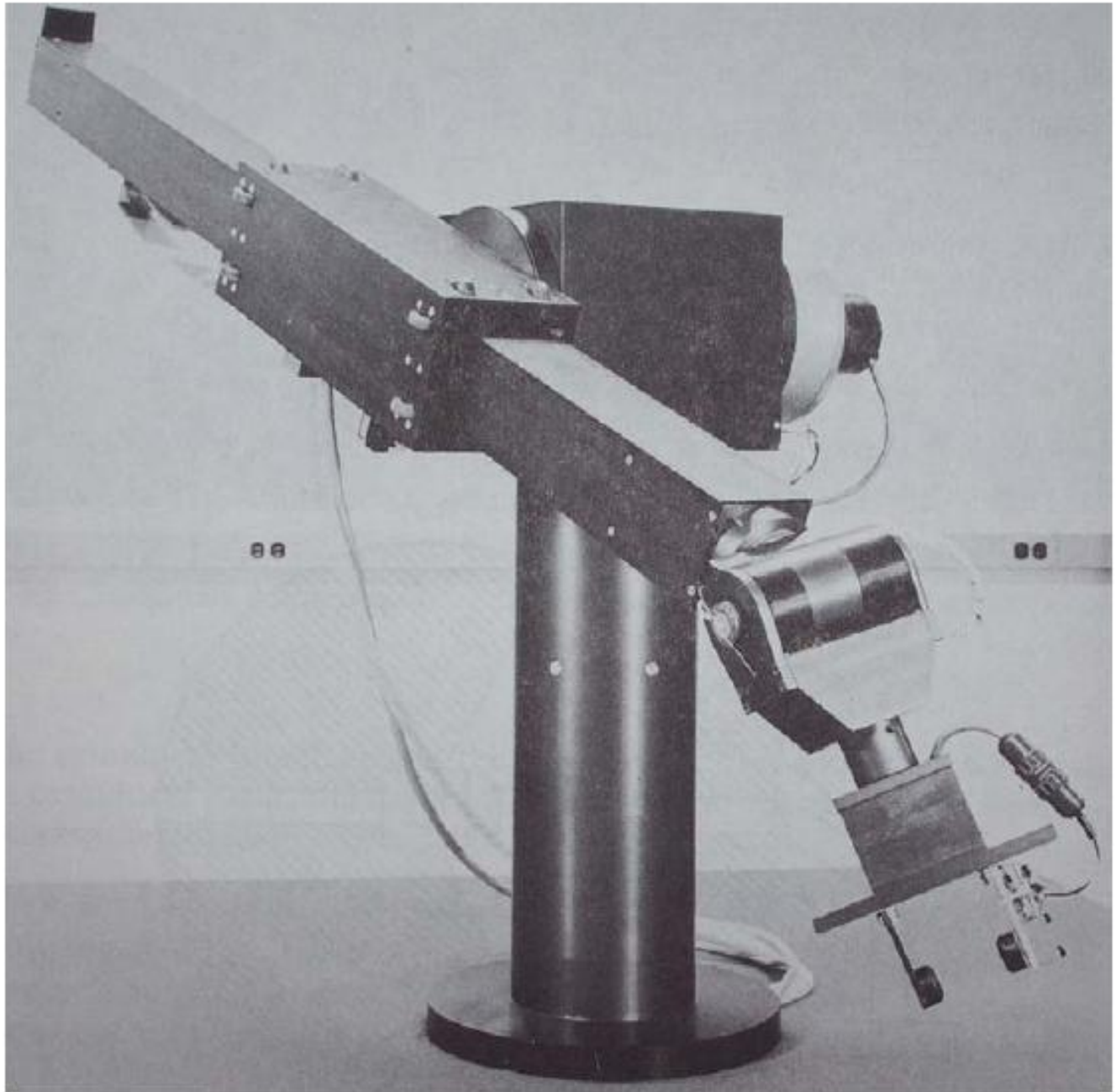


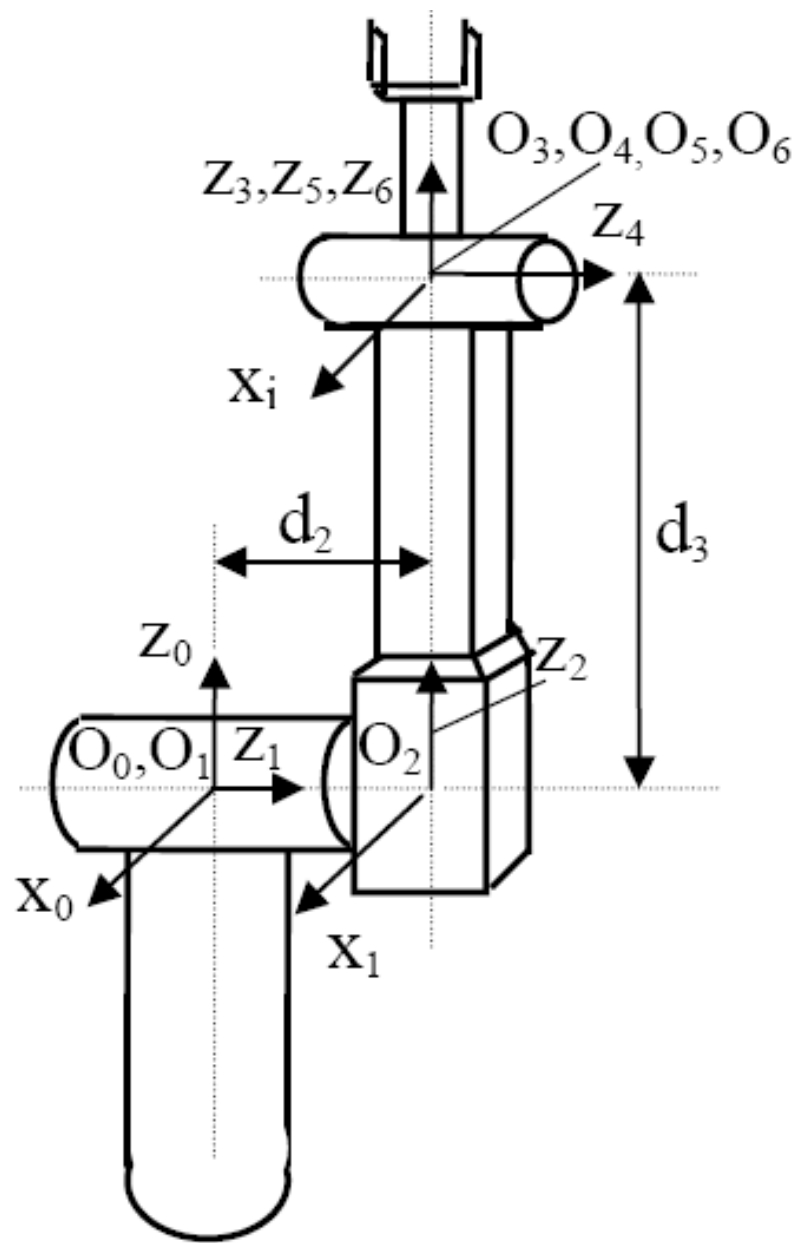
Workspace

Assignment of coordinate frames

## Hệ phương trình động học của robot STANFORD

Stanford là một robot có 6 khâu với cấu hình RRT.RRR







# Bảng thông số Denavit-Hartenbert

Khâu	$\theta_i$	$\alpha_i$	$a_i$	$d_i$
1	$\theta_1^*$	$-90^0$	0	0
2	$\theta_2^*$	$90^0$	0	$d_2$
3	0	0	0	$d_3^*$
4	$\theta_4^*$	$-90^0$	0	0
5	$\theta_5^*$	$90^0$	0	0
6	$\theta_6^*$	0	0	0

$$T_6^1 = A_2 A_3 A_4 A_5 A_6 = \begin{pmatrix} C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 & -C_2(C_4C_5S_6 - S_4C_6) + S_2S_5S_6 \\ S_2(C_4C_5C_6 - S_4S_6) + C_2S_5C_6 & -S_2(C_4C_5S_6 + S_4C_6) - C_2S_5S_6 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 \\ 0 & 0 \\ C_2C_4S_5 + S_2C_5 & S_2d_3 \\ S_2C_4S_5 - C_2C_5 & -C_2d_3 \\ S_4S_5 & d_2 \\ 0 & 1 \end{pmatrix}$$

Cuối cùng :

$$T_6 = \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 T_6^1$$

$$n_x = C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6)$$

$$n_y = S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6)$$

$$n_z = -S_2(C_4C_5C_6 - S_4S_6) + C_2S_5C_6$$

$$O_x = C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6)$$

$$O_y = S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5C_6 + C_4C_6)$$

$$O_z = S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6$$

$$a_x = C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5$$

$$a_y = S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5$$

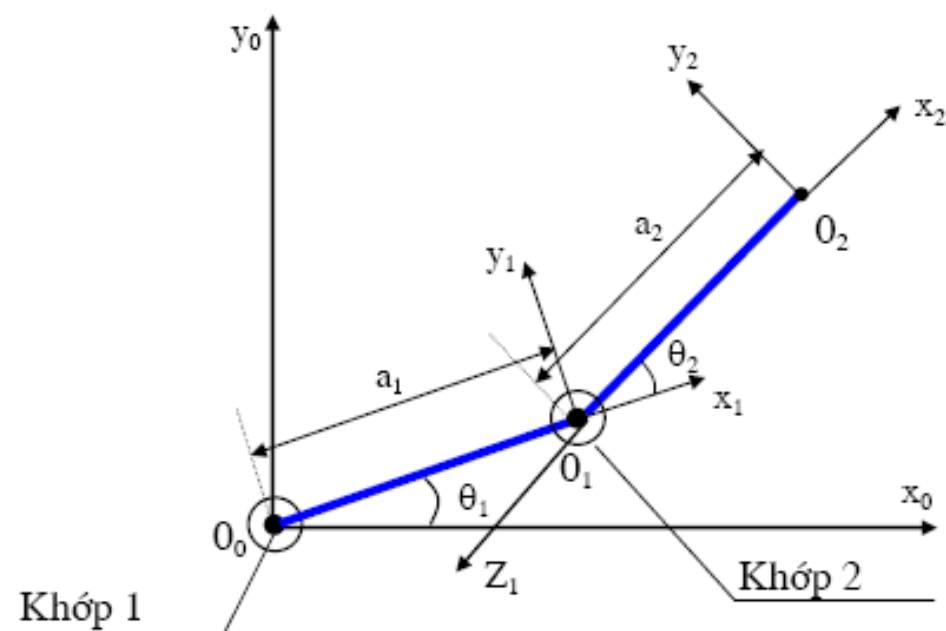
$$a_z = -S_2C_4S_5 + C_2C_5$$

$$p_x = C_1S_2d_3 - S_1d_2$$

$$p_y = S_1S_2d_3 + C_1d_2$$

$$p_z = C_2d_3$$

## Xây dựng phương trình động học thuận



$${}^0A_1 = \text{Rot}(z_0, \theta_1) \text{Trans}(a_1, 0, 0) = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \text{Rot}(z_1, \theta_2) \text{Trans}(a_2, 0, 0) = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân hai ma trận trên và rút gọn ta có:

$${}^0A_2 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Xây dựng phương trình động học ngược**

$${}^0A_2 = {}^0A_1 {}^1A_2 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$({}^0A_1)^{-1} {}^0A_2 = {}^1A_2$$

$$\text{Ta có } ({}^0A_1)^{-1} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & -a_1 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Vậy } ({}^0A_1)^{-1} {}^0A_2 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & -a_1 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix}
 n_x \cos\theta_1 + n_y \sin\theta_1 & o_x \cos\theta_1 + o_y \sin\theta_1 & a_x \cos\theta_1 + a_y \sin\theta_1 & P_x \cos\theta_1 + P_y \sin\theta_1 - a_1 \\
 -n_x \sin\theta_1 + n_y \cos\theta_1 & -o_x \sin\theta_1 + o_y \cos\theta_1 & -a_x \sin\theta_1 + a_y \cos\theta_1 & -P_x \sin\theta_1 + P_y \cos\theta_1 \\
 n_z & o_z & a_z & P_z \\
 0 & 0 & 0 & 1
 \end{bmatrix}
 =
 \begin{bmatrix}
 \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \cos\theta_2 \\
 \sin\theta_2 & \cos\theta_2 & 0 & a_2 \sin\theta_2 \\
 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1
 \end{bmatrix}$$

Cân bằng hai vế của hai ma trận tại cột thứ 4 ta được

$$\begin{cases}
 P_x \cos\theta_1 + P_y \sin\theta_1 - a_1 = a_2 \cos\theta_2 \\
 -P_x \sin\theta_1 + P_y \cos\theta_1 = a_2 \sin\theta_2
 \end{cases}$$

$$\begin{cases}
 P_x \cos\theta_1 + P_y \sin\theta_1 = a_2 \cos\theta_2 + a_1 \\
 -P_x \sin\theta_1 + P_y \cos\theta_1 = a_2 \sin\theta_2
 \end{cases}$$

$$P_x^2 + P_y^2 = a_1^2 + 2a_1a_2 \cos \theta_2 + a_2^2$$

$$\text{Vậy } \theta_2 = \arccos \left( \frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1a_2} \right)$$

$$a_1^2 + P_x^2 + P_y^2 - 2a_1(P_x \cos \theta_1 + P_y \sin \theta_1) = a_2^2$$

$$P_x \cos \theta_1 + P_y \sin \theta_1 = \frac{P_x^2 + P_y^2 + a_1^2 - a_2^2}{2a_1}$$

$$\frac{P_x}{\sqrt{P_x^2 + P_y^2}} \cos \theta_1 + \frac{P_y}{\sqrt{P_x^2 + P_y^2}} \sin \theta_1 = \frac{P_x^2 + P_y^2 + a_1^2 - a_2^2}{2a_1 \sqrt{P_x^2 + P_y^2}}$$

$$\theta_1 = \arcsin \left( \frac{P_x^2 + P_y^2 + a_1^2 - a_2^2}{2a_1 \sqrt{P_x^2 + P_y^2}} \right) - \arcsin \left( \frac{P_x}{\sqrt{P_x^2 + P_y^2}} \right)$$

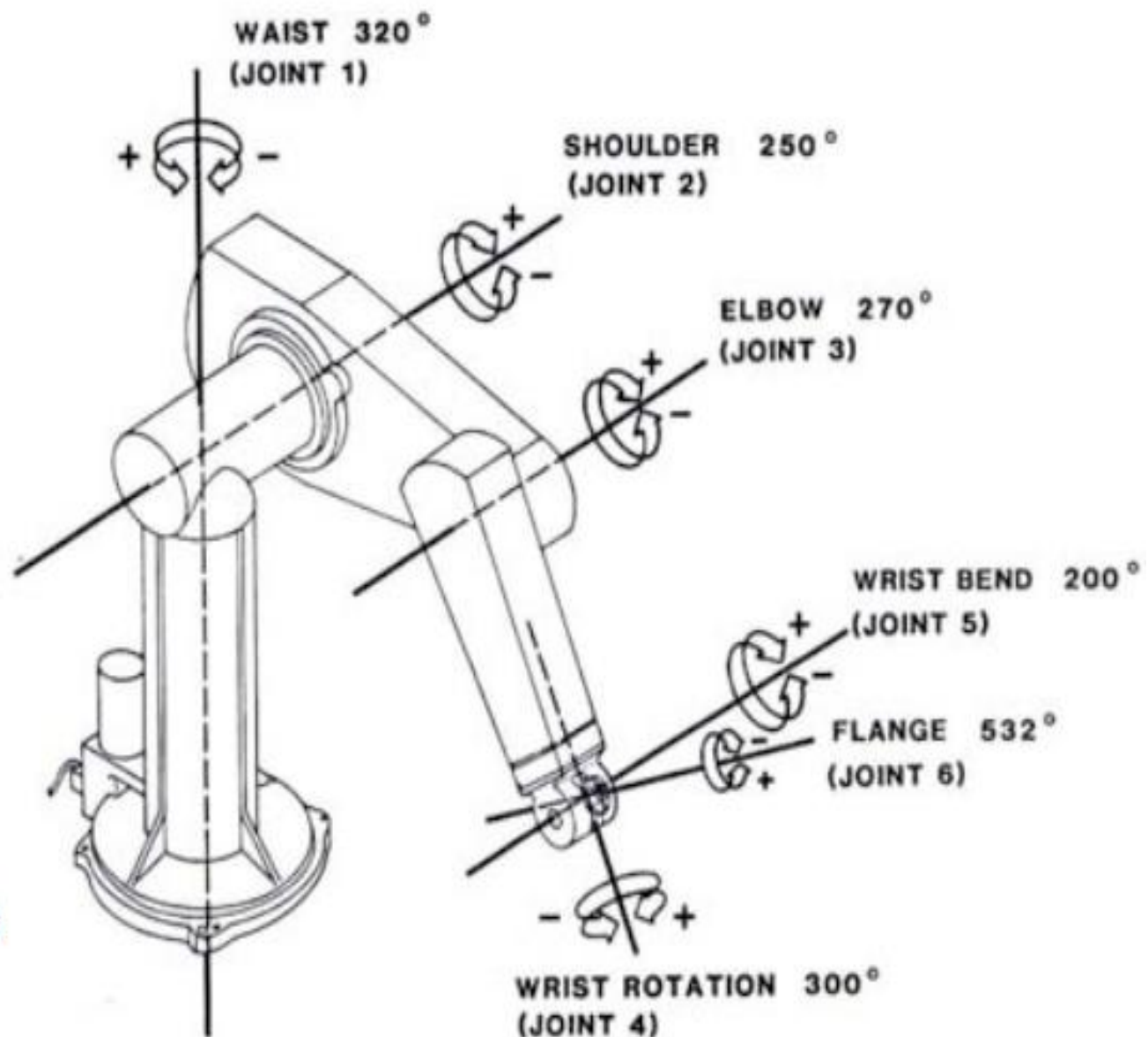


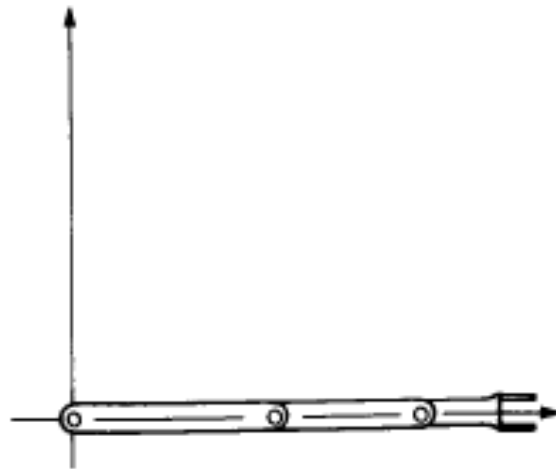
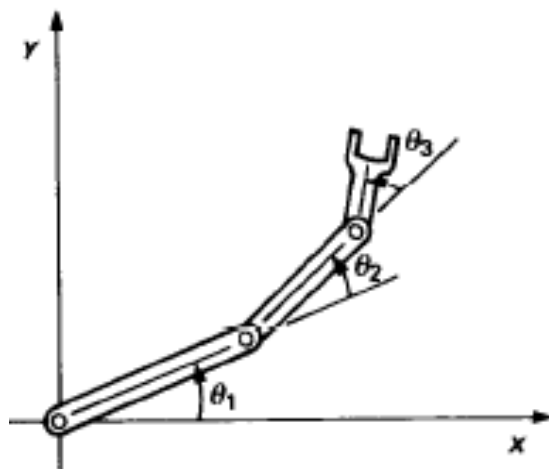
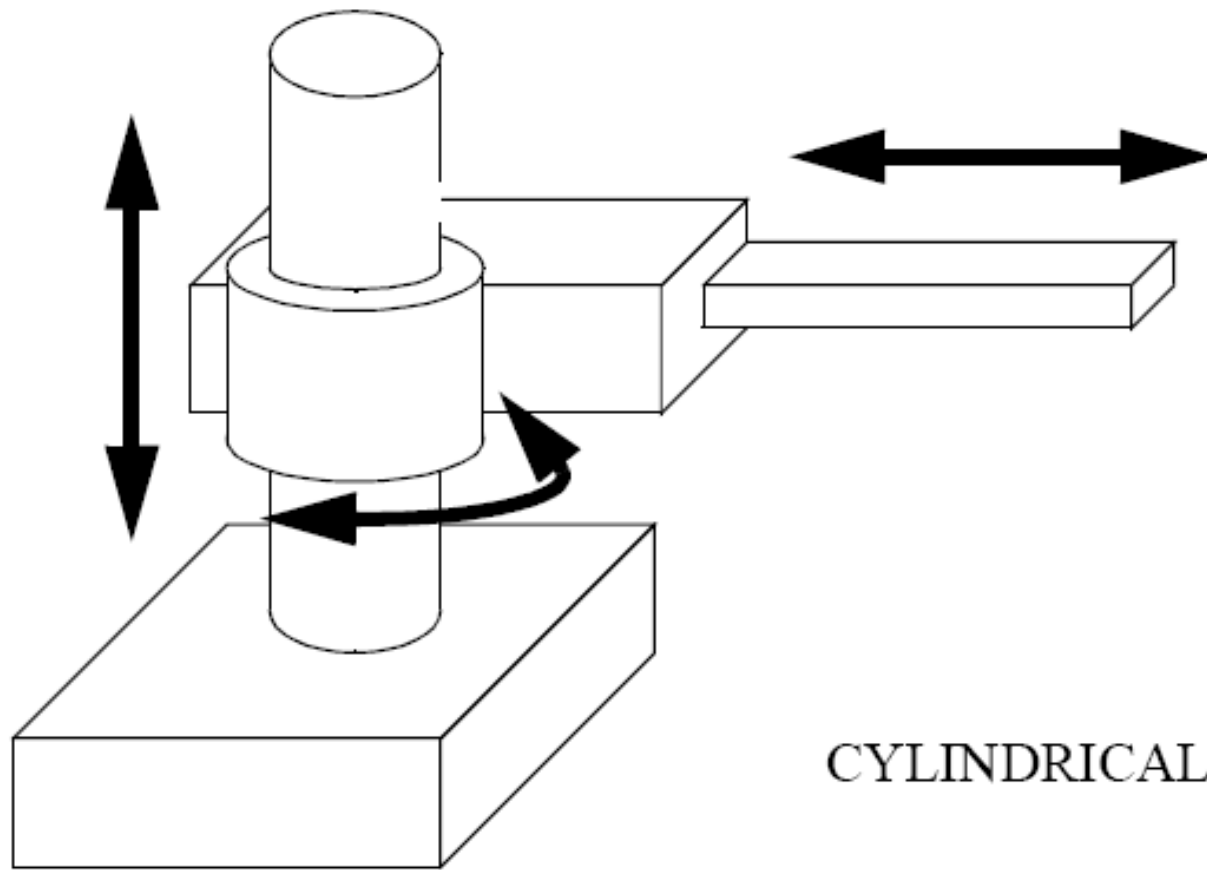
Each link type has a characteristic transformation matrix relating the proximal joint to the distal joint

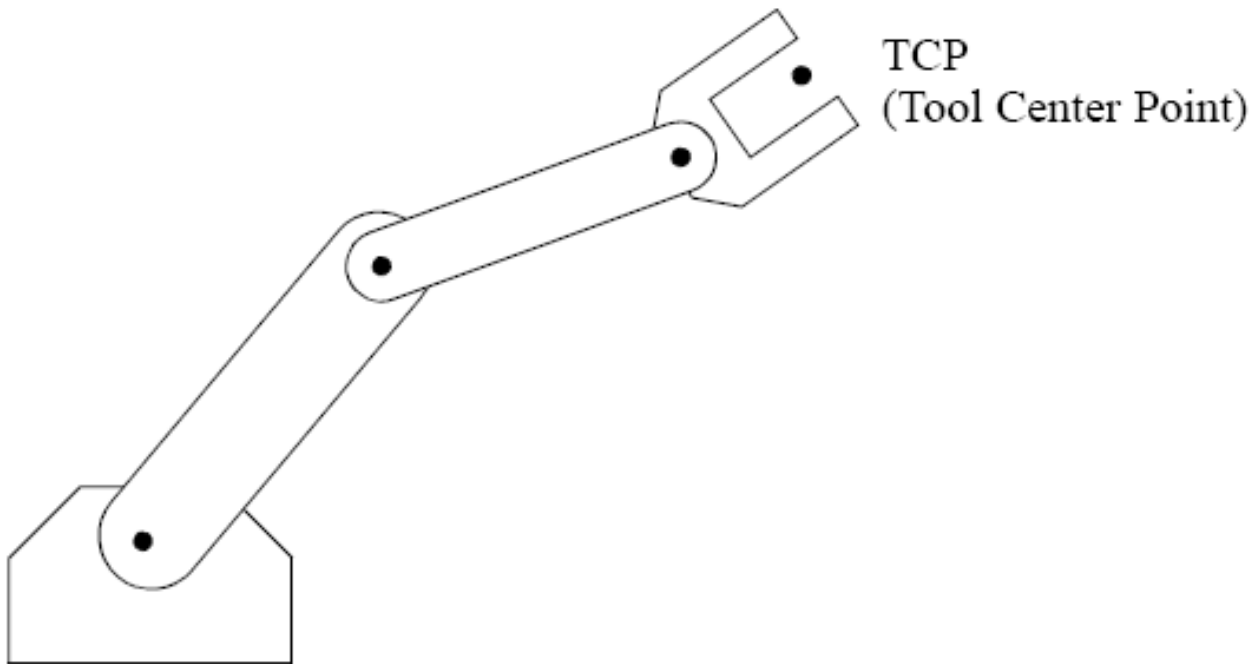
Link  $n$  has

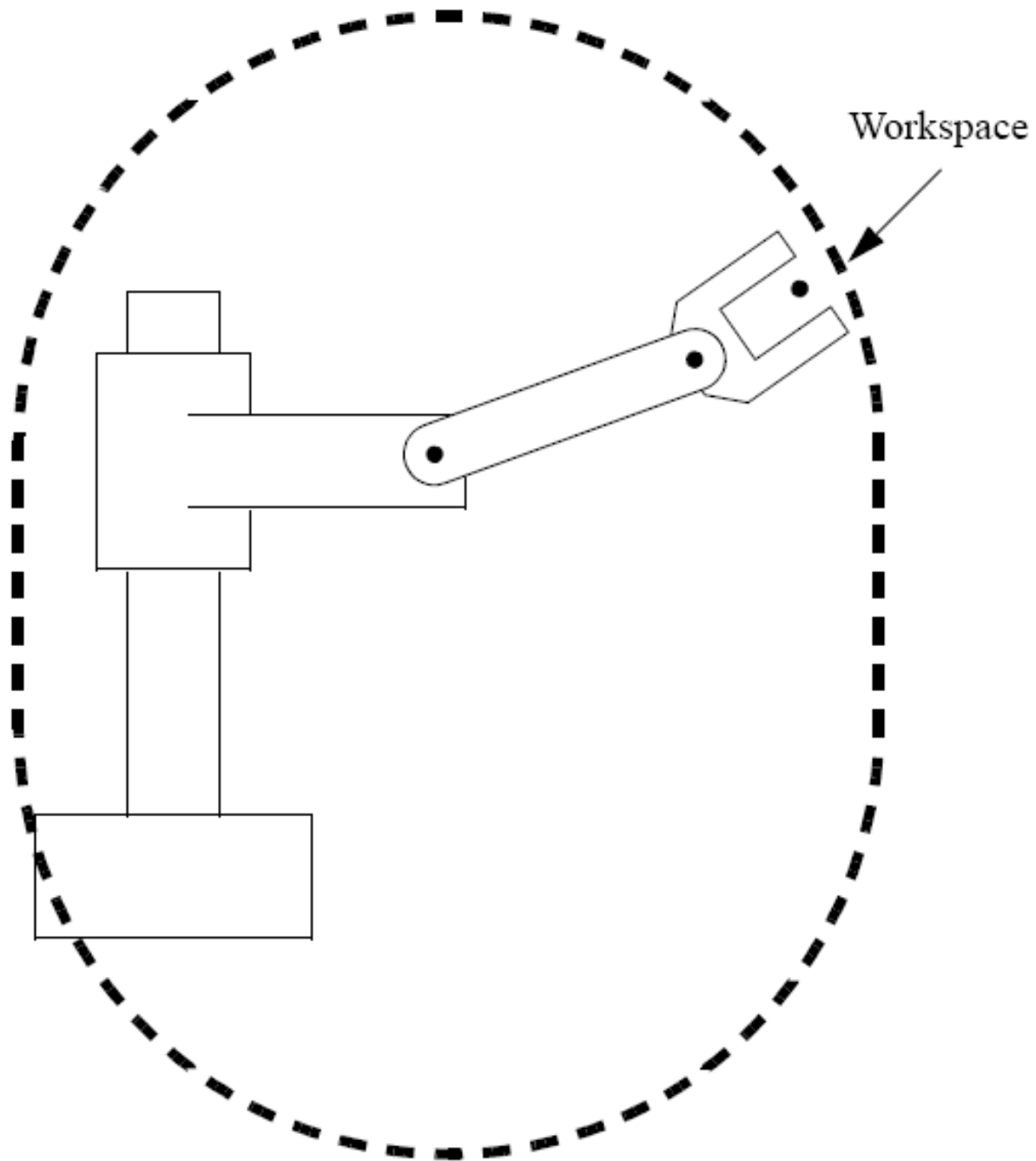
- Proximal end: Joint  $n$ , coordinate frame  $n - 1$
- Distal end: Joint  $n + 1$ , coordinate frame  $n$

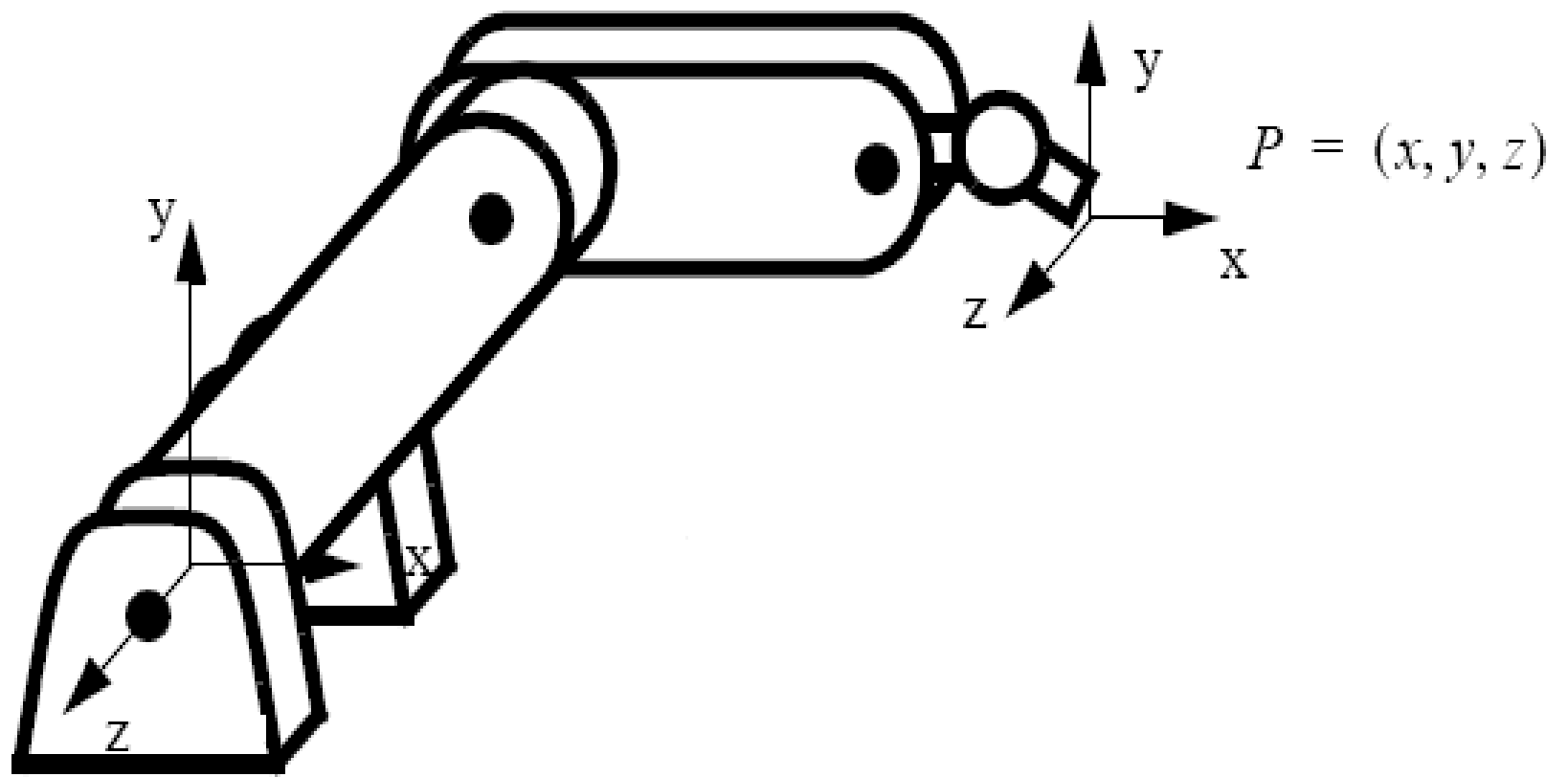
Puma geometry at right

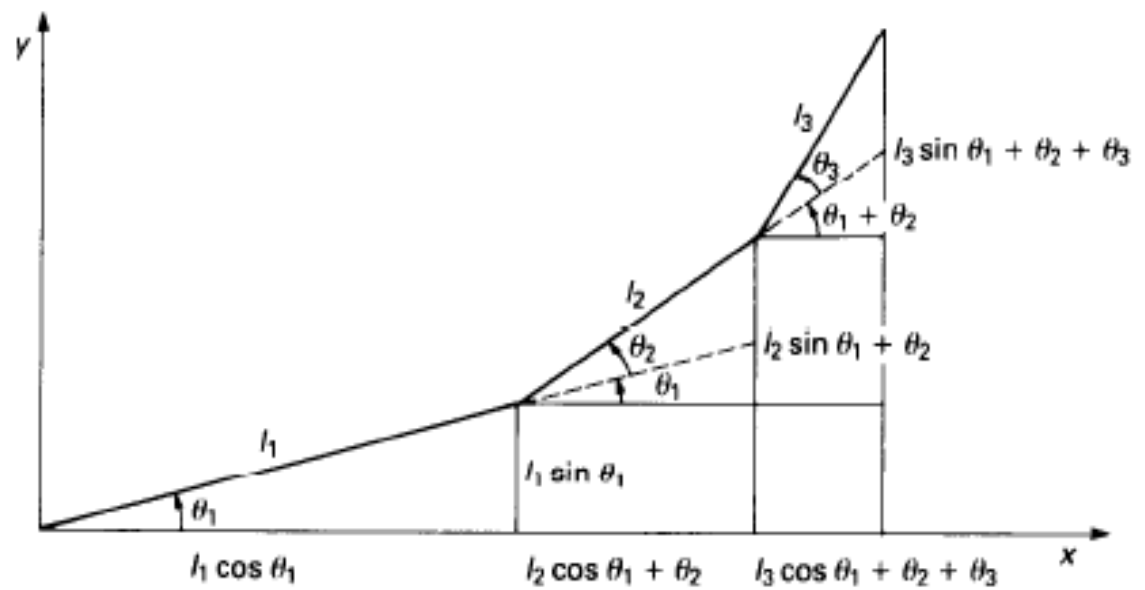
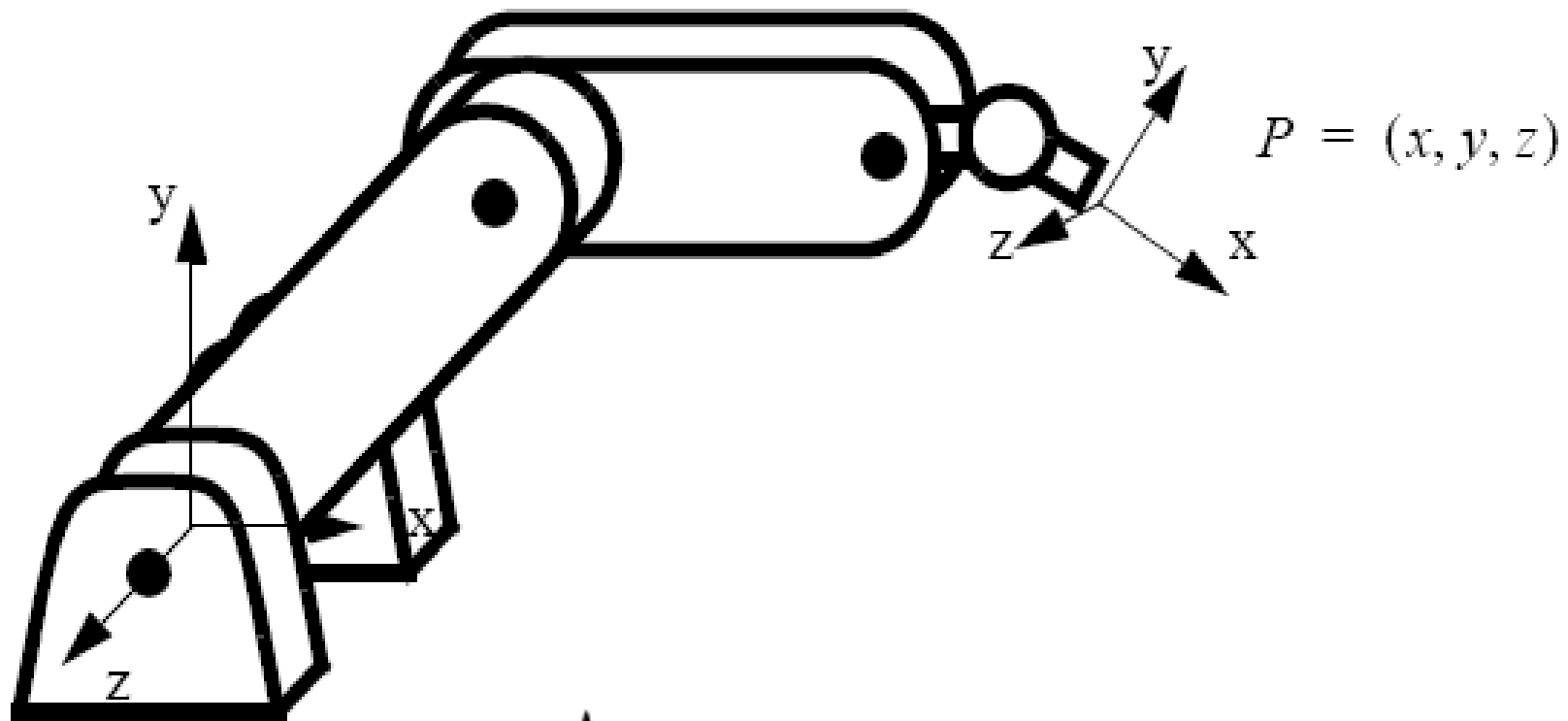


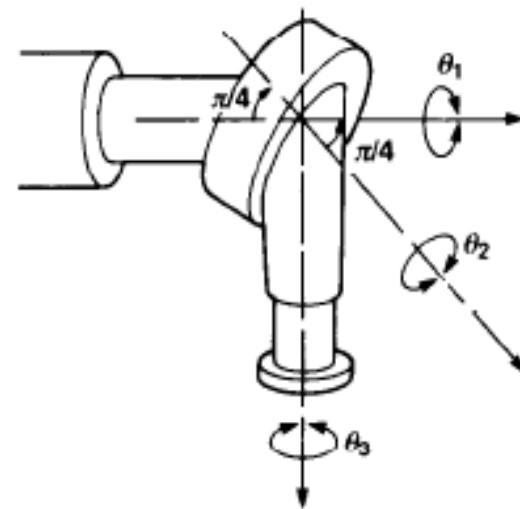
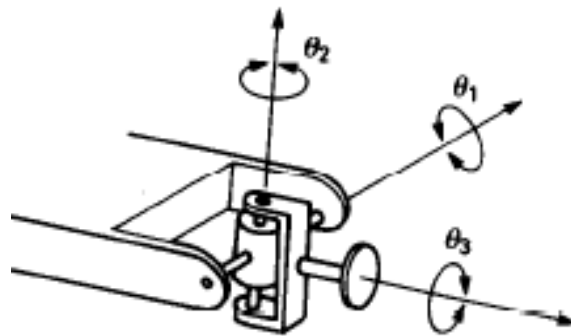
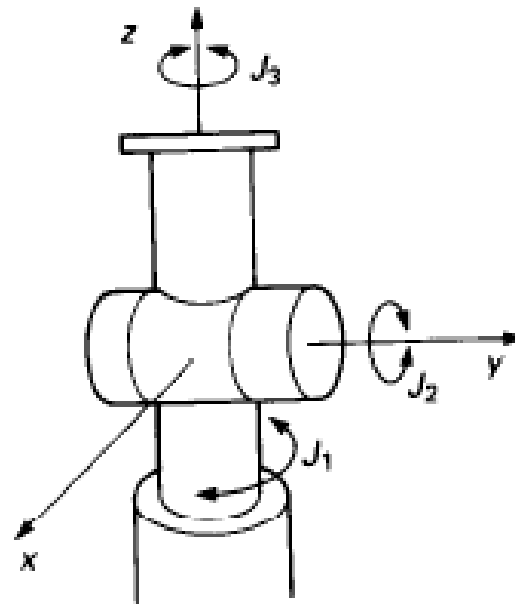
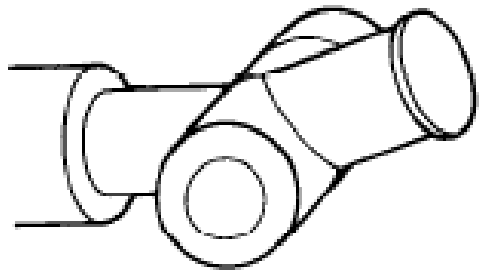


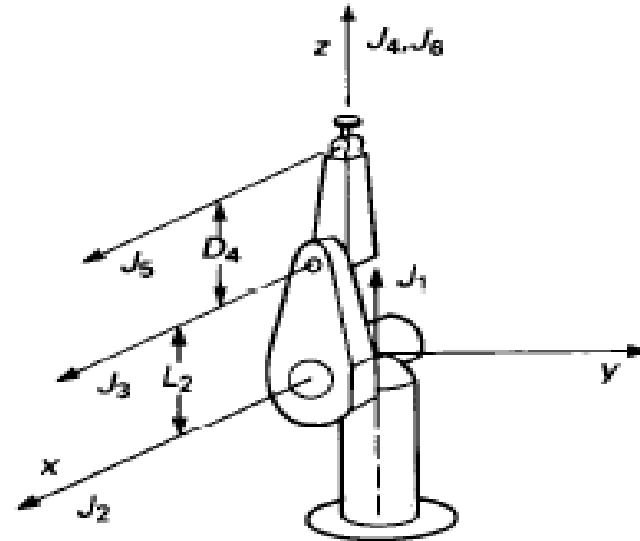
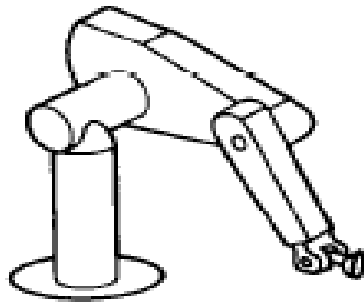








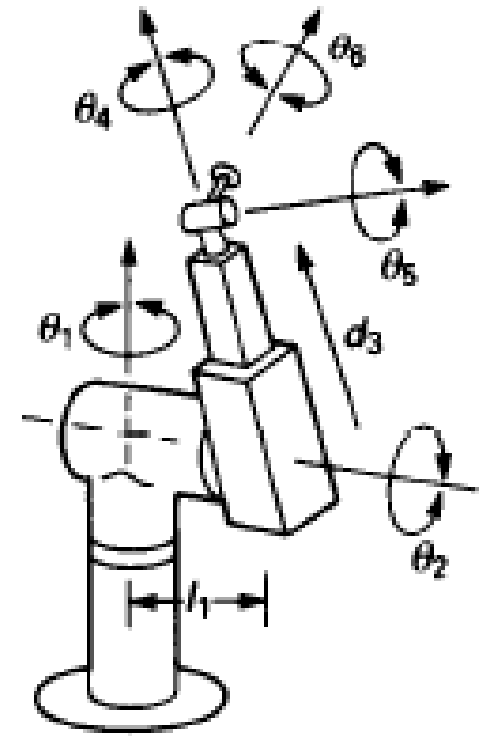
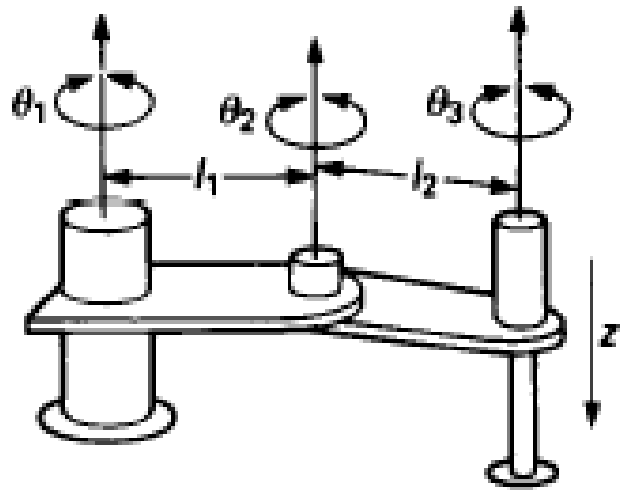




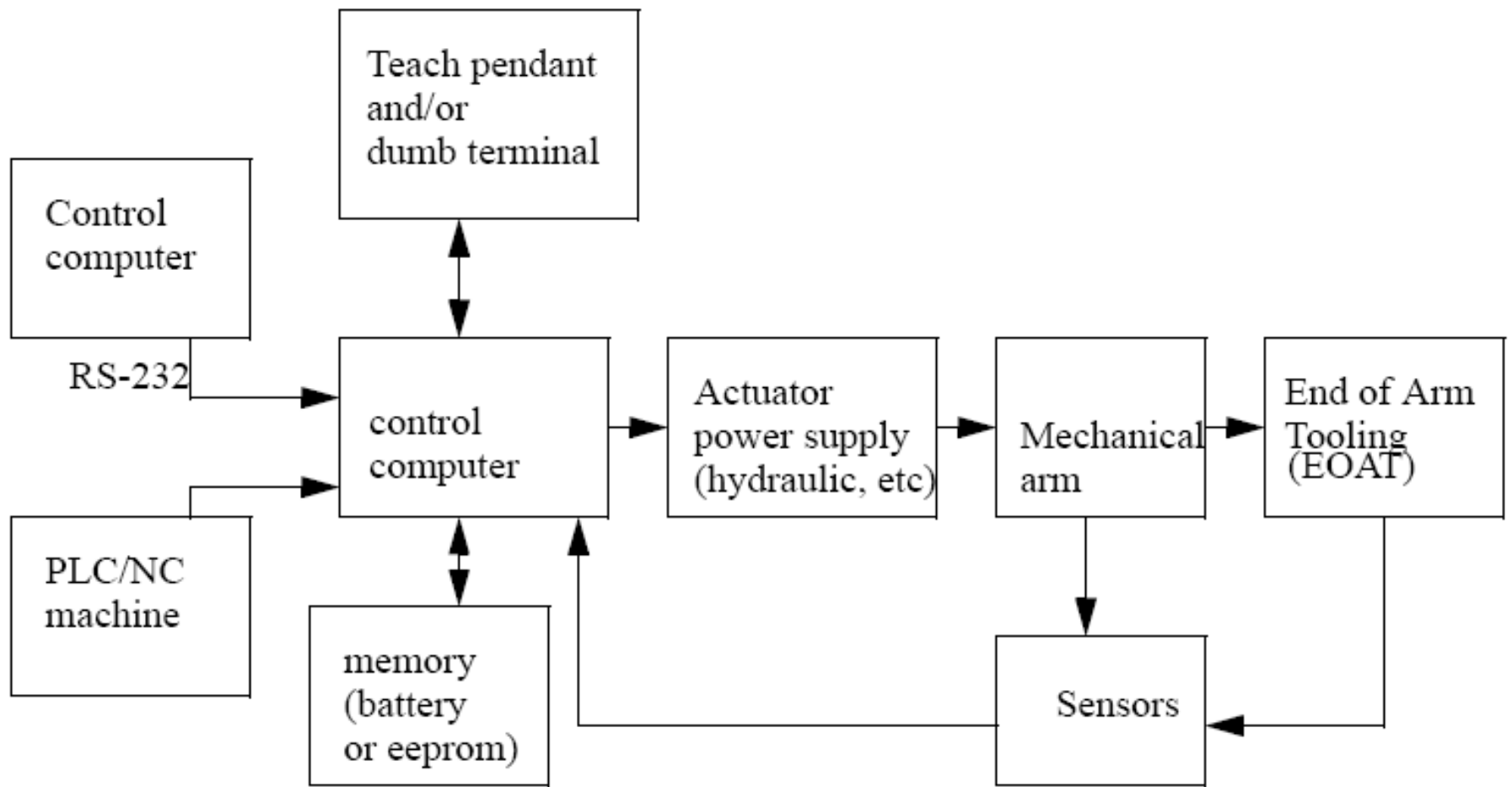
The Puma Robot and its Home Position

Link	Link length	Twist angle	Offset
$J_1—J_2$	0	$\pi/2$	—
$J_2—J_3$	$L_2$	0	(axes parallel)
$J_3—J_4$	0	$\pi/2$	$D_3$
$J_4—J_5$	0	$\pi/2$	$D_4$
$J_5—J_6$	0	$\pi/2$	0





The Scara Robot and the Stanford Manipulator



- The basic components of a robot are,

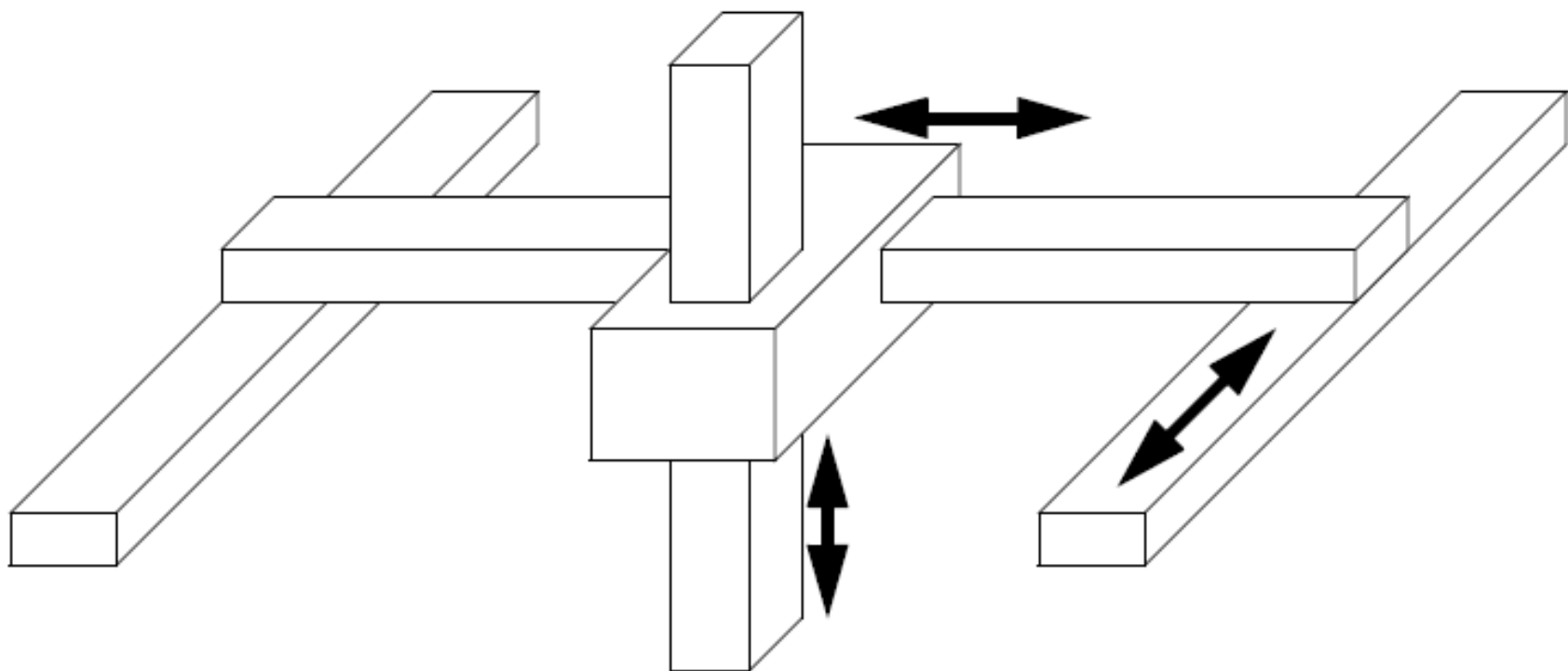
Structure - the mechanical structure (links, base, etc). This requires a great deal of mass to provide enough structural rigidity to ensure minimum accuracy under varied payloads.

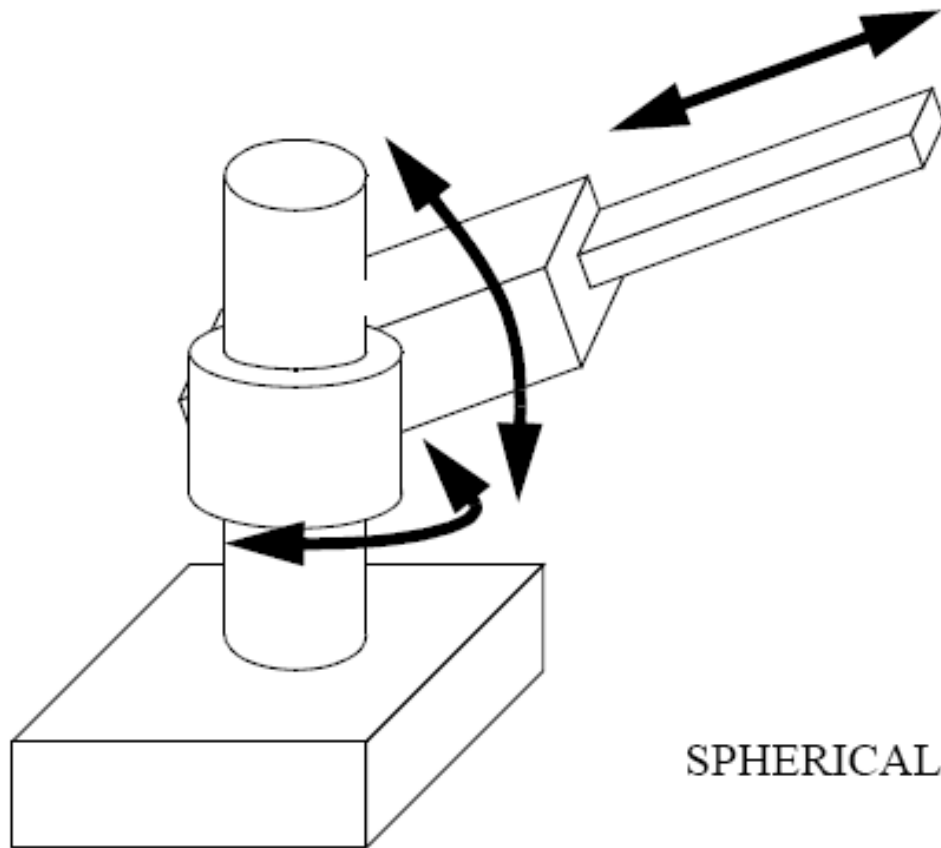
Actuators - The motors, cylinders, etc. that drive the robot joints. This might also include mechanisms for a transmission, locking, etc.

Control Computer - This computer interfaces with the user, and in turn controls the robot joints.

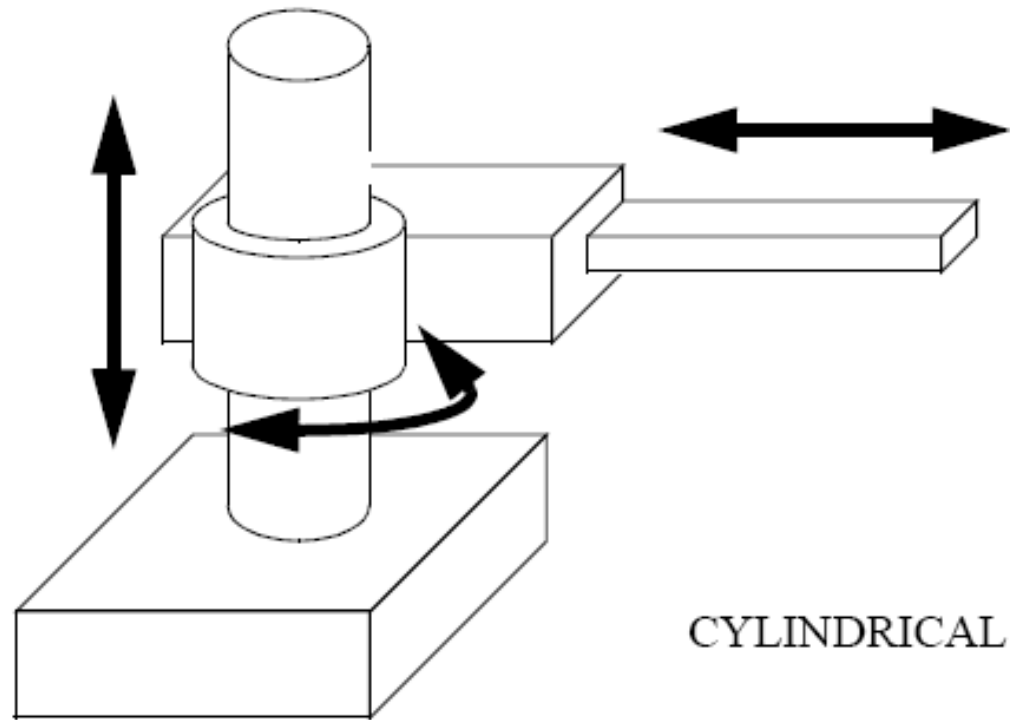
End of Arm Tooling (EOAT) - The tooling is provided by the user, and is designed for specific tasks.

Teach pendant - One popular method for programming the robot. This is a small hand held device that can direct motion of the robot, record points in motion sequences, and begin replay of sequences. More advanced pendants include more functionality.

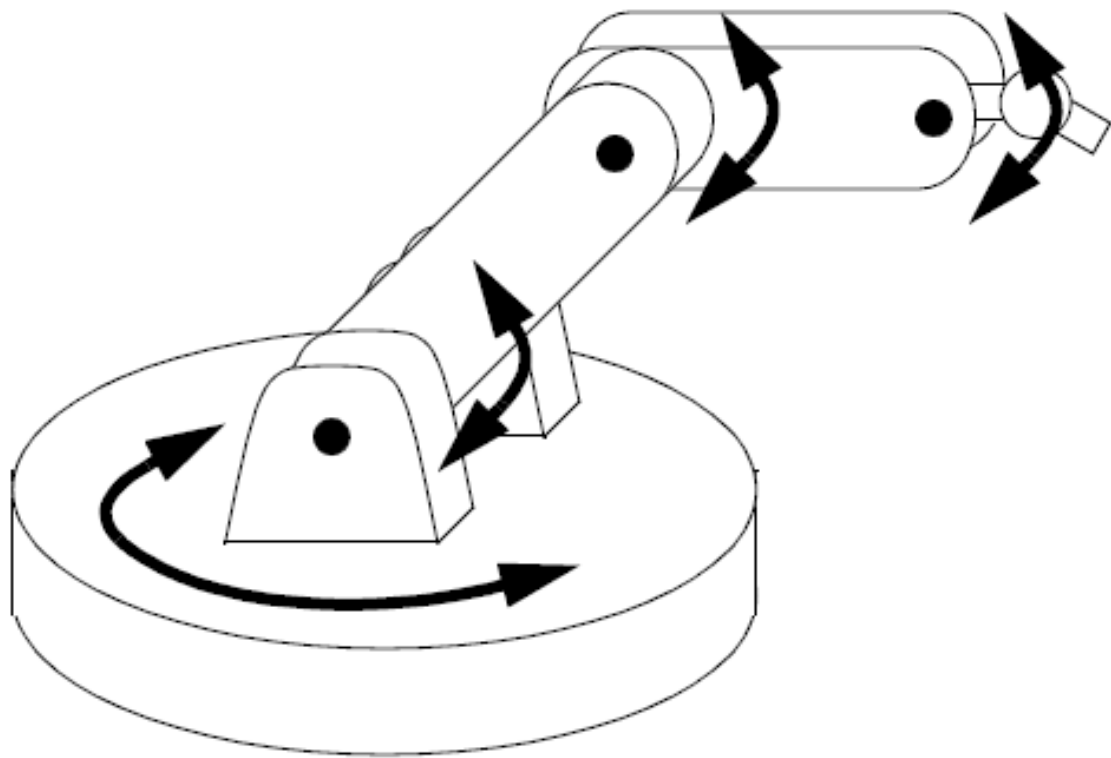




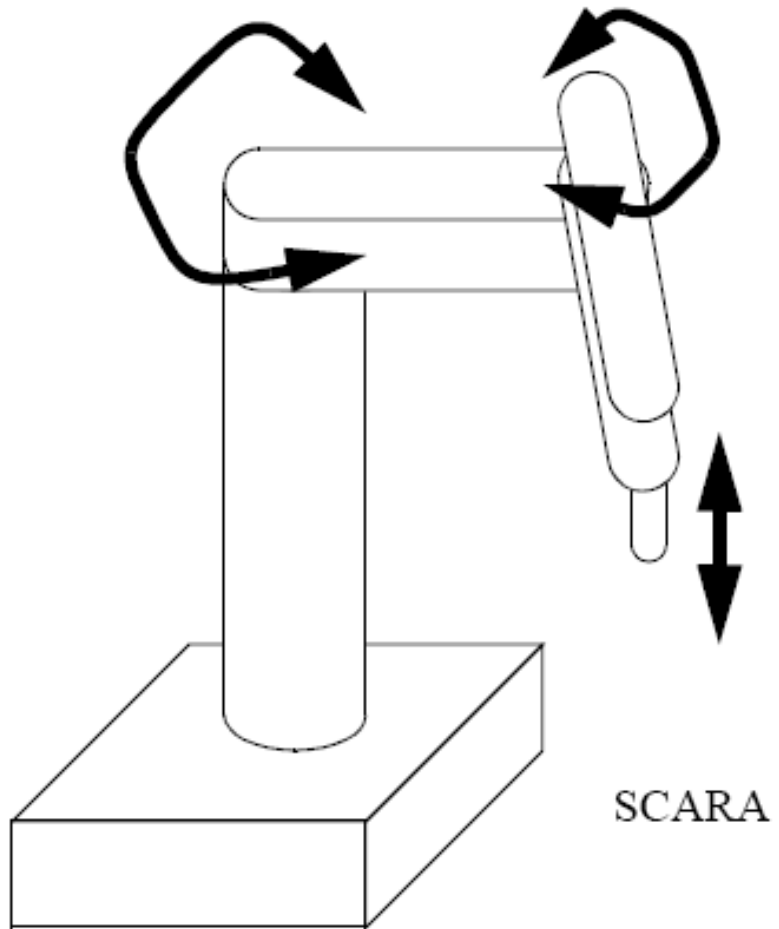
SPHERICAL



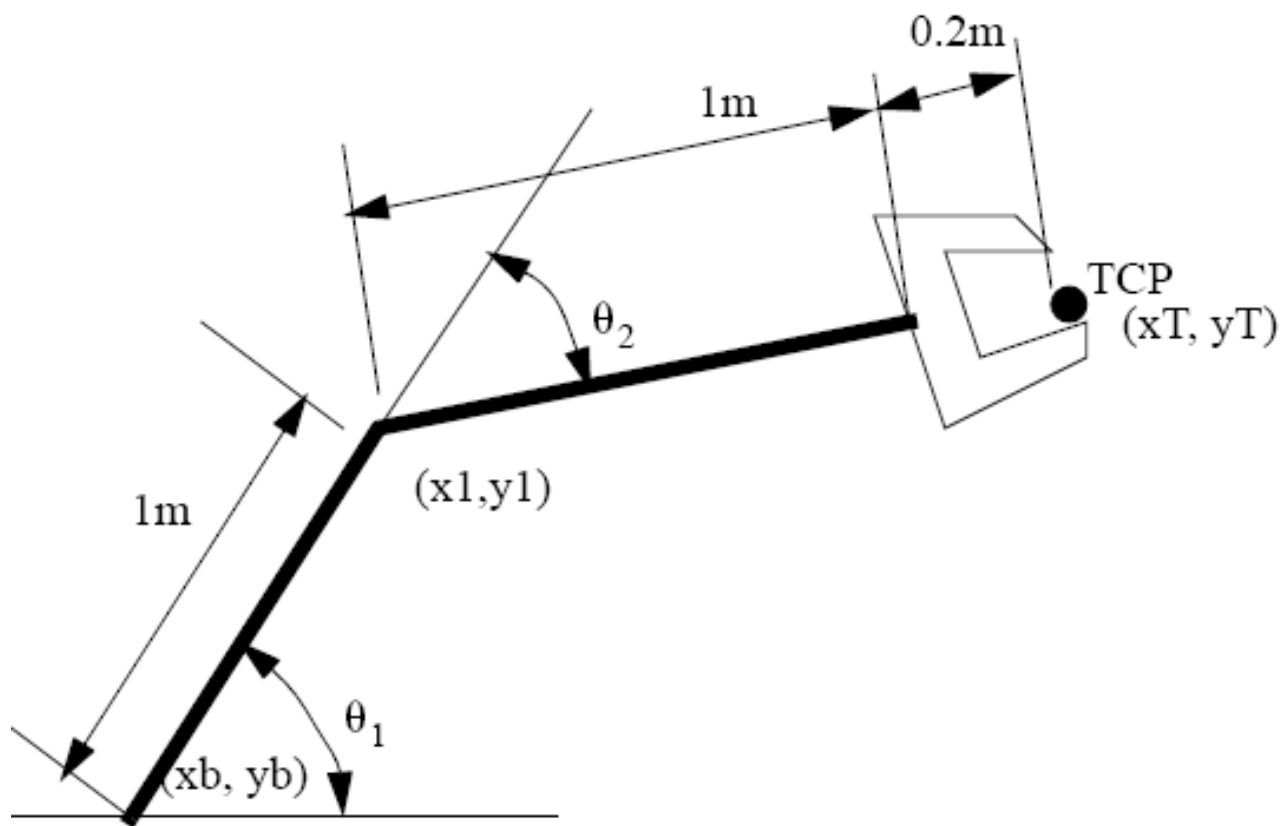
CYLINDRICAL

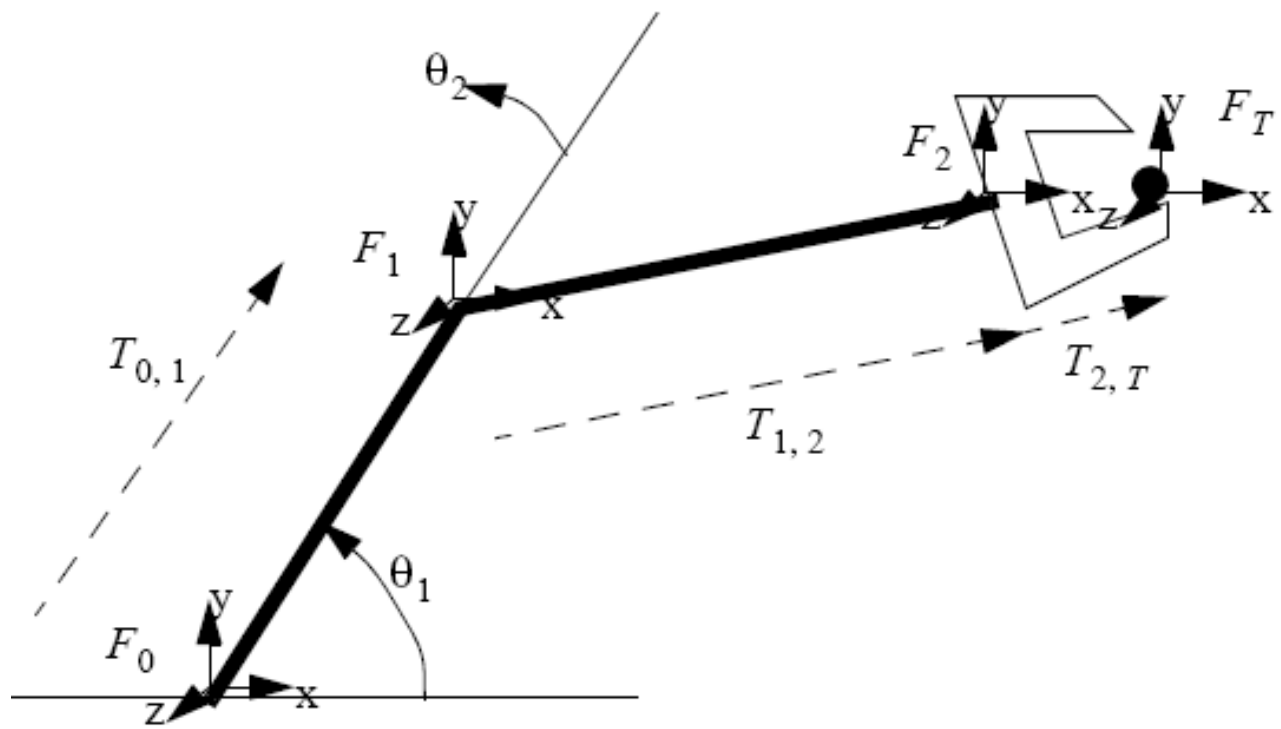


ARTICULATED/REVOLUTE/  
JOINTED SPHERICAL









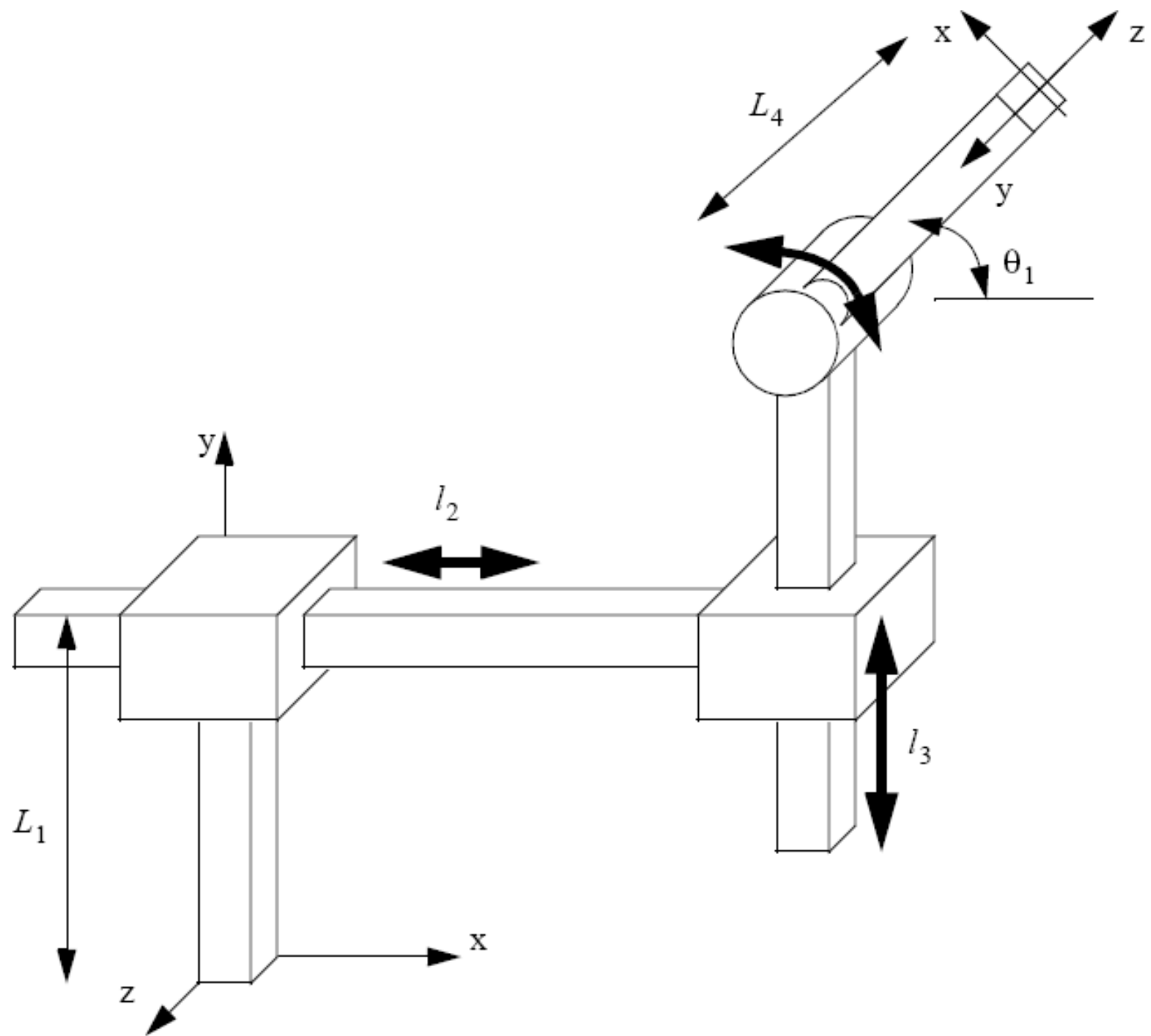
$$T_{0,1} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & \cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & \sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \theta_i = \theta_1 \\ d_i = 0 \\ a_i = 1 \\ \alpha_i = 0 \end{array}$$

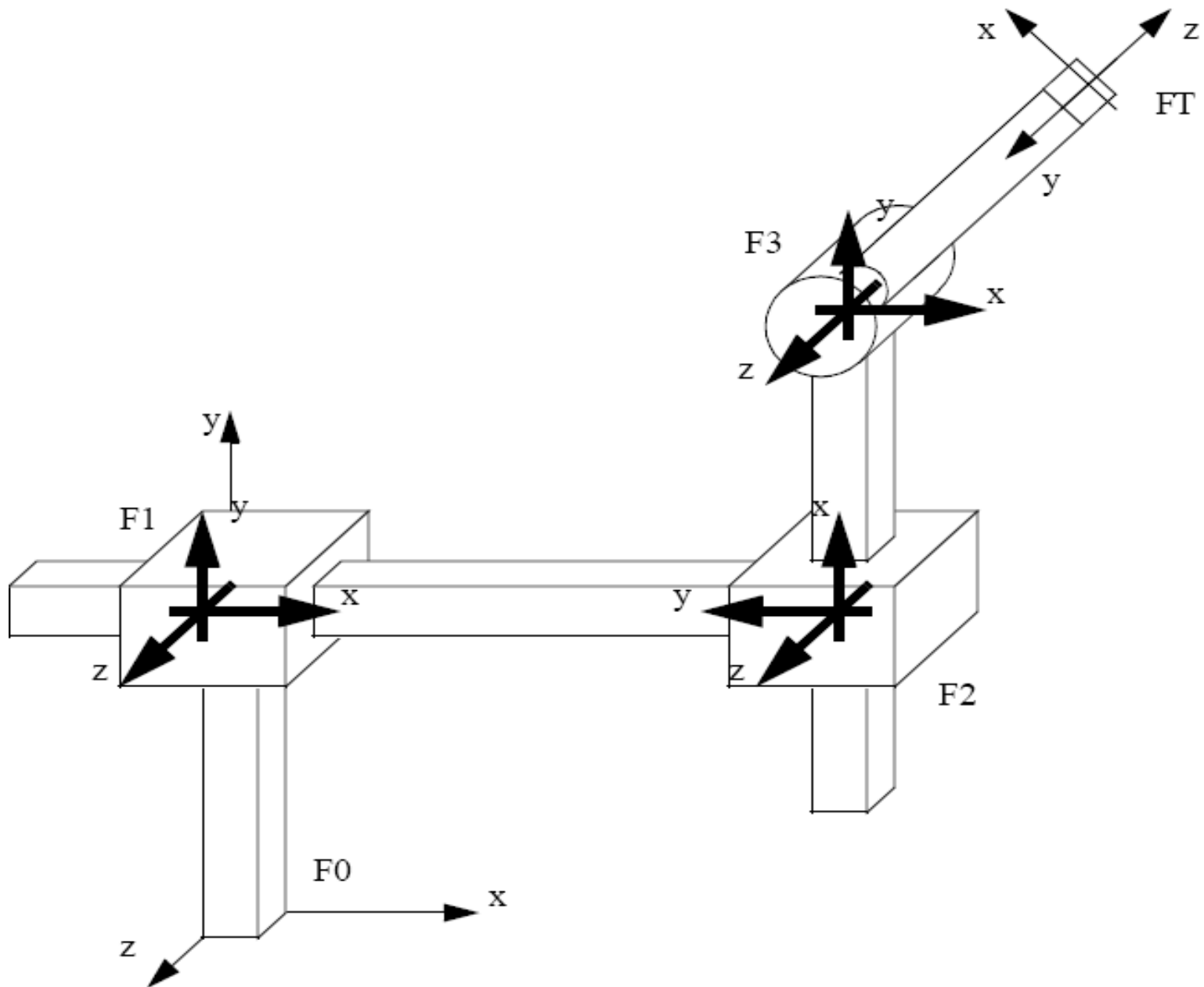
$$T_{1,2} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & \cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \theta_i = \theta_2 \\ d_i = 0 \\ a_i = 1 \\ \alpha_i = 0 \end{array}$$

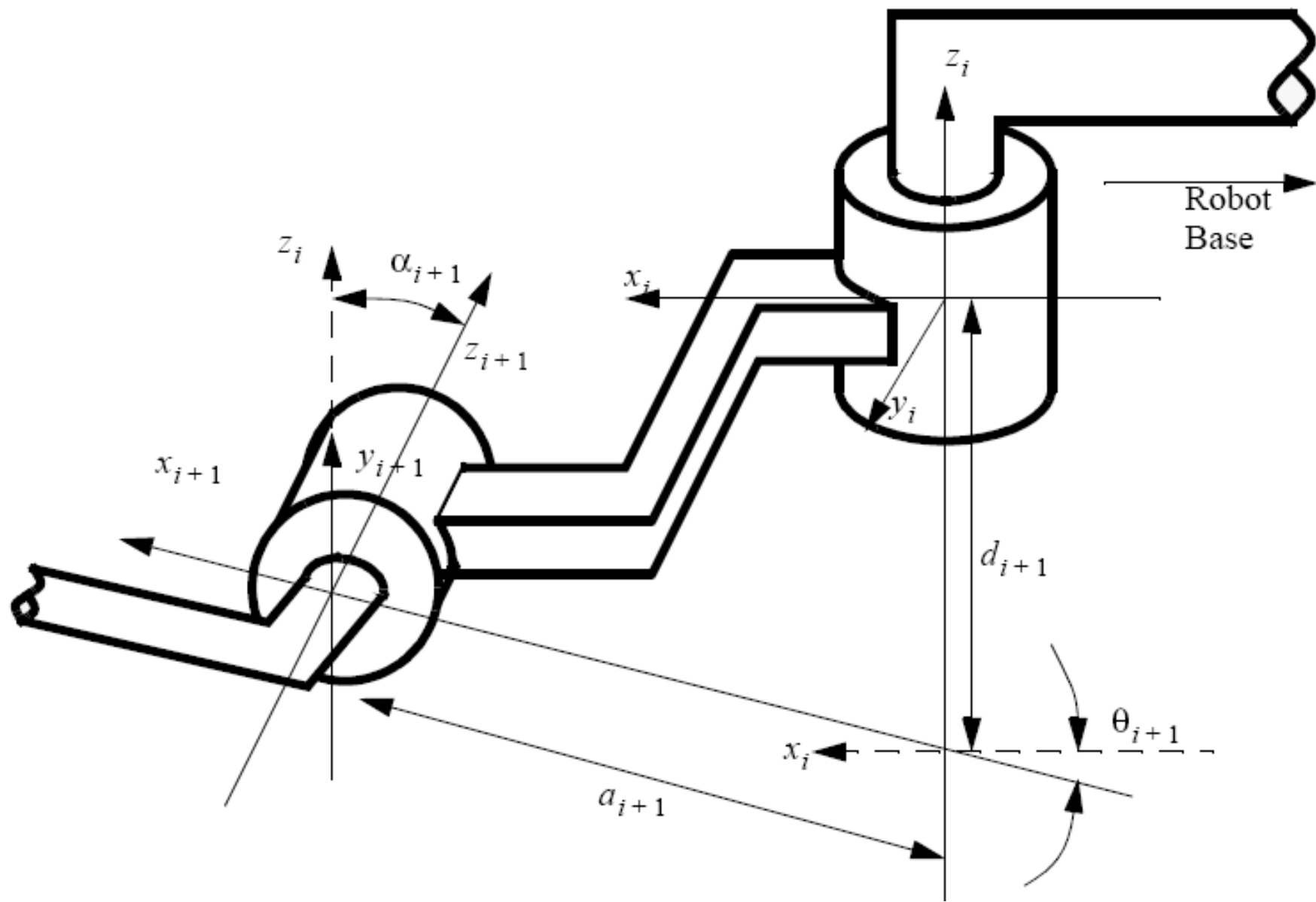
$$T_{2,T} = \begin{bmatrix} 1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \theta_i = 0 \\ d_i = 0 \\ a_i = 0.2 \\ \alpha_i = 0 \end{array}$$

$$T_{0,T} = T_{0,1}T_{1,2}T_{2,T}$$

$$T_{0, T} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) & 0 & \cos\theta_1 + 1.2\cos(\theta_1 + \theta_2) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & \sin\theta_1 + 1.2\sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$







$$T_{i-1,i} = \text{rot}(z_{i-1}, \theta_i) \text{trans}(0, 0, d_i) \text{trans}(a_i, 0, 0) \text{rot}(x_i, \alpha_i)$$

$$T_{i-1,i} = \begin{bmatrix} \cos \theta_i & \cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$