

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CÔNG NGHỆ ĐIỆN



GIÁO TRÌNH

KỸ THUẬT ROBOT

TP. Hồ Chí Minh, tháng 12 năm 2008

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CÔNG NGHỆ ĐIỆN**



GIÁO TRÌNH

KỸ THUẬT ROBOT

TP. Hồ Chí Minh, tháng 12 năm 2008

CHƯƠNG I

GIỚI THIỆU CHUNG VỀ ROBOT CÔNG NGHIỆP

1.1. Sơ lược quá trình phát triển của robot công nghiệp (IR : Industrial Robot) :

Thuật ngữ “Robot” xuất phát từ tiếng Sec (Czech) “Robota” có nghĩa là công việc tạp dịch trong vở kịch Rossum’s Universal Robots của Karel Capek, vào năm 1921. Trong vở kịch này, Rossum và con trai của ông ta đã chế tạo ra những chiếc máy gần giống với con người để phục vụ con người. Có lẽ đó là một gợi ý ban đầu cho các nhà sáng chế kỹ thuật về những cơ cấu, máy móc bắt chước các hoạt động cơ bắp của con người.

Đầu thập kỷ 60, công ty Mỹ AMF (American Machine and Foundry Company) quảng cáo một loại máy tự động vạn năng và gọi là “Người máy công nghiệp” (Industrial Robot). Ngày nay người ta đặt tên người máy công nghiệp (hay robot công nghiệp) cho những loại thiết bị có dáng dấp và một vài chức năng như tay người được điều khiển tự động để thực hiện một số thao tác sản xuất.

Về mặt kỹ thuật, những robot công nghiệp ngày nay, có nguồn gốc từ hai lĩnh vực kỹ thuật ra đời sớm hơn đó là các cơ cấu điều khiển từ xa (Teleoperators) và các máy công cụ điều khiển số (NC - Numerically Controlled machine tool).

Các cơ cấu điều khiển từ xa (hay các thiết bị kiểu chủ-tớ) đã phát triển mạnh trong chiến tranh thế giới lần thứ hai nhằm nghiên cứu các vật liệu phóng xạ. Người thao tác được tách biệt khỏi khu vực phóng xạ bởi một bức tường có một hoặc vài cửa quan sát để có thể nhìn thấy được công việc bên trong. Các cơ cấu điều khiển từ xa thay thế cho cánh tay của người thao tác; nó gồm có một bộ kẹp ở bên trong (tớ) và hai tay cầm ở bên ngoài (chủ). Cả hai, tay cầm và bộ kẹp, được nối với nhau bằng một cơ cấu sáu

bậc tự do để tạo ra các vị trí và hướng tùy ý của tay cầm và bộ kẹp. Cơ cấu dùng để điều khiển bộ kẹp theo chuyển động của tay cầm.

Vào khoảng năm 1949, các máy công cụ điều khiển số ra đời, nhằm đáp ứng yêu cầu gia công các chi tiết trong ngành chế tạo máy bay. Những robot đầu tiên thực chất là sự nối kết giữa các khâu cơ khí của cơ cấu điều khiển từ xa với khả năng lập trình của máy công cụ điều khiển số.

Dưới đây chúng ta sẽ đi qua một số thời điểm lịch sử phát triển của người máy công nghiệp. Một trong những robot công nghiệp đầu tiên được chế tạo là robot Versatran của công ty AMF, Mỹ. Cũng vào khoảng thời gian này ở Mỹ xuất hiện loại robot Unimate -1900 được dùng đầu tiên trong kỹ nghệ ô tô.

Tiếp theo Mỹ, các nước khác bắt đầu sản xuất robot công nghiệp : Anh -1967, Thụy Điển và Nhật -1968 theo bản quyền của Mỹ; CHLB Đức - 1971; Pháp - 1972; ở Ý - 1973. . .

Tính năng làm việc của robot ngày càng được nâng cao, nhất là khả năng nhận biết và xử lý. Năm 1967 ở trường Đại học tổng hợp Stanford (Mỹ) đã chế tạo ra mẫu robot hoạt động theo mô hình “mắt-tay”, có khả năng nhận biết và định hướng bàn kẹp theo vị trí vật kẹp nhờ các cảm biến. Năm 1974 Công ty Mỹ Cincinnati đưa ra loại robot được điều khiển bằng máy vi tính, gọi là robot T3 (The Tomorrow Tool : Công cụ của tương lai). Robot này có thể nâng được vật có khối lượng đến 40 Kg.

Có thể nói, Robot là sự tổ hợp khả năng hoạt động linh hoạt của các cơ cấu điều khiển từ xa với mức độ “tri thức” ngày càng phong phú của hệ thống điều khiển theo chương trình số cũng như kỹ thuật chế tạo các bộ cảm biến, công nghệ lập trình và các phát triển của trí khôn nhân tạo, hệ chuyên gia ...

Trong những năm sau này, việc nâng cao tính năng hoạt động của robot không ngừng phát triển. Các robot được trang bị thêm các loại cảm biến khác nhau để nhận biết môi trường chung quanh, cùng với những

thành tựu to lớn trong lĩnh vực Tin học - Điện tử đã tạo ra các thế hệ robot với nhiều tính năng đặc biệt, Số lượng robot ngày càng gia tăng, giá thành ngày càng giảm. Nhờ vậy, robot công nghiệp đã có vị trí quan trọng trong các dây chuyền sản xuất hiện đại.

Một vài số liệu về số lượng robot được sản xuất ở một vài nước công nghiệp phát triển như sau :

(Bảng 1.1)

Nước sản xuất	Năm 1990	Năm 1994	Năm 1998 (Dự tính)
Nhật	60 118	29756	67 000
Mỹ	4 327	7634	11 100
Đức	5 845	5125	8 600
Ý	2 500	2408	4 000
Pháp	1 488	1197	2 000
Anh	510	1086	1 500
Hàn Quốc	1 000	1200	

Mỹ là nước đầu tiên phát minh ra robot, nhưng nước phát triển cao nhất trong lĩnh vực nghiên cứu chế tạo và sử dụng robot lại là Nhật.

1.2. Ứng dụng robot công nghiệp trong sản xuất :

Từ khi mới ra đời robot công nghiệp được áp dụng trong nhiều lĩnh vực dưới góc độ thay thế sức người. Nhờ vậy các dây chuyền sản xuất được tổ chức lại, năng suất và hiệu quả sản xuất tăng lên rõ rệt.

Mục tiêu ứng dụng robot công nghiệp nhằm góp phần nâng cao năng suất dây chuyền công nghệ, giảm giá thành, nâng cao chất lượng và khả năng cạnh tranh của sản phẩm đồng thời cải thiện điều kiện lao động. Đạt được các mục tiêu trên là nhờ vào những khả năng to lớn của robot như : làm việc không biết mệt mỏi, rất dễ dàng chuyển nghề một cách thành thạo, chịu được phóng xạ và các môi trường làm việc độc hại, nhiệt độ cao, “cảm

thấy” được cả từ trường và “nghe” được cả siêu âm ... Robot được dùng thay thế con người trong các trường hợp trên hoặc thực hiện các công việc tuy không nặng nhọc nhưng đơn điệu, dễ gây mệt mỏi, nhầm lẫn.

Trong ngành cơ khí, robot được sử dụng nhiều trong công nghệ đúc, công nghệ hàn, cắt kim loại, sơn, phun phủ kim loại, tháo lắp vận chuyển phôi, lắp ráp sản phẩm . . .

Ngày nay đã xuất hiện nhiều dây chuyền sản xuất tự động gồm các máy CNC với Robot công nghiệp, các dây chuyền đó đạt mức tự động hoá cao, mức độ linh hoạt cao . . . ở đây các máy và robot được điều khiển bằng cùng một hệ thống chương trình.

Ngoài các phân xưởng, nhữ máy, kỹ thuật robot cũng được sử dụng trong việc khai thác thăm lục địa và đại dương, trong y học, sử dụng trong quốc phòng, trong chinh phục vũ trụ, trong công nghiệp nguyên tử, trong các lĩnh vực xã hội...

Rõ ràng là khả năng làm việc của robot trong một số điều kiện vượt hơn khả năng của con người; do đó nó là phương tiện hữu hiệu để tự động hoá, nâng cao năng suất lao động, giảm nhẹ cho con người những công việc nặng nhọc và độc hại. Nhược điểm lớn nhất của robot là chưa linh hoạt như con người, trong dây chuyền tự động, nếu có một robot bị hỏng có thể làm ngừng hoạt động của cả dây chuyền, cho nên robot vẫn luôn hoạt động dưới sự giám sát của con người.

1.3. Các khái niệm và định nghĩa về robot công nghiệp :

1.3.1. Định nghĩa robot công nghiệp:

Hiện nay có nhiều định nghĩa về Robot, có thể điểm qua một số định nghĩa

như sau :

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 toạ 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 GOCT 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.

Có thể nói Robot công nghiệp là một máy tự động linh hoạt thay thế từng phần hoặc toàn bộ các hoạt động cơ bắp và hoạt động trí tuệ của con người trong nhiều khả năng thích nghi khác nhau.

Robot công nghiệp có khả năng chương trình hoá linh hoạt trên nhiều trục chuyển động, biểu thị cho số bậc tự do của chúng. Robot công nghiệp được trang bị những bàn tay máy hoặc các cơ cấu chấp hành, giải quyết những nhiệm vụ xác định trong các quá trình công nghệ : hoặc trực tiếp tham gia thực hiện các nguyên công (son, hàn, phun phủ, rót kim loại vào khuôn đúc, lắp ráp máy . . .) hoặc phục vụ các quá trình công nghệ (tháo lắp chi tiết gia công, dao cụ, đồ gá . . .) với những thao tác cầm nắm, vận chuyển và trao đổi các đối tượng với các trạm công nghệ, trong một hệ thống máy tự động linh hoạt, được gọi là "*Hệ thống tự động linh hoạt robot hoá*" cho phép thích ứng nhanh và thao tác đơn giản khi nhiệm vụ sản xuất thay đổi.

1.3.2. Bậc tự do của robot (DOF : Degrees Of Freedom) :

Bậc tự do là số khả năng chuyển động của một cơ cấu (chuyển động quay hoặc tịnh tiến). Để dịch chuyển được một vật thể trong không gian, cơ cấu chấp hành của robot phải đạt được một số bậc tự do. Nói chung cơ hệ của robot là một cơ cấu hờ, do đó bậc tự do của nó có thể tính theo công thức :

$$w = 6n - \sum_{i=1}^5 ip_i \quad (1.1)$$

ở đây:

n - Số khâu động;

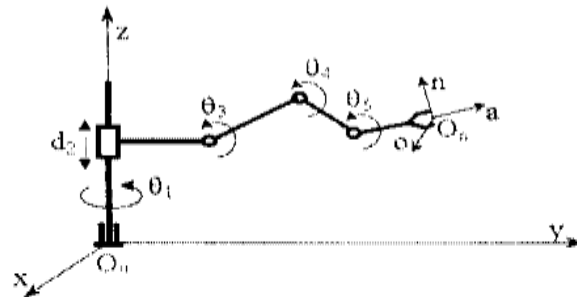
p_i - Số khớp loại i ($i = 1, 2, \dots, 5$: Số bậc tự do bị hạn chế).

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 hờ, số bậc tự do bằng tổng số bậc tự do của các khớp động.

Để định vị và định hướng khâu chấp hành cuối một cách tùy ý trong không gian 3 chiều robot cần có 6 bậc tự do, trong đó 3 bậc tự do để định vị và 3 bậc tự do để định hướng. Một số công việc đơn giản nâng hạ, sắp xếp... có thể yêu cầu số bậc tự do ít hơn. Các robot hàn, sơn... thường yêu cầu 6 bậc tự do. Trong một số trường hợp cần sự khéo léo, linh hoạt hoặc khi cần phải tối ưu hoá quỹ đạo,... người ta dùng robot với số bậc tự do lớn hơn 6.

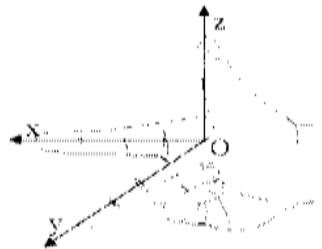
1.3.3. Hệ tọa độ (Coordinate frames) :

Mỗi robot thường bao gồm nhiều khâu (links) liên kết với nhau qua các khớp (joints), tạo thành một xích động học xuất phát từ một khâu cơ bản (base) đứng yên. Hệ tọa độ gắn với khâu cơ bản gọi là hệ tọa độ cơ bản (hay hệ tọa độ chuẩn). Các hệ tọa độ trung gian khác gắn với các khâu động gọi là hệ tọa độ suy rộng. Trong từng thời điểm hoạt động, các tọa độ suy rộng xác định cấu hình của robot bằng các chuyển dịch dãi hoặc các chuyển dịch góc của các khớp tịnh tiến hoặc khớp quay (hình 1.1). Các tọa độ suy rộng còn được gọi là biến khớp.



Hình 1.1 : Các toạ độ suy rộng của robot.

Các hệ toạ độ gắn trên các khâu của robot phải tuân theo qui tắc bên tay phải : Dùng tay phải, nắm hai ngón tay út vỗ áp út vào lòng bốn tay, xoay 3 ngón : cái, trỏ vỗ giữa theo 3 phương vuông góc nhau, nếu chọn ngón cái lờ phương vỗ chiều của trục z, thì ngón trỏ chỉ phương, chiều của trục x vỗ ngón giữa sẽ biểu thị phương, chiều của trục y (hình 1.2).



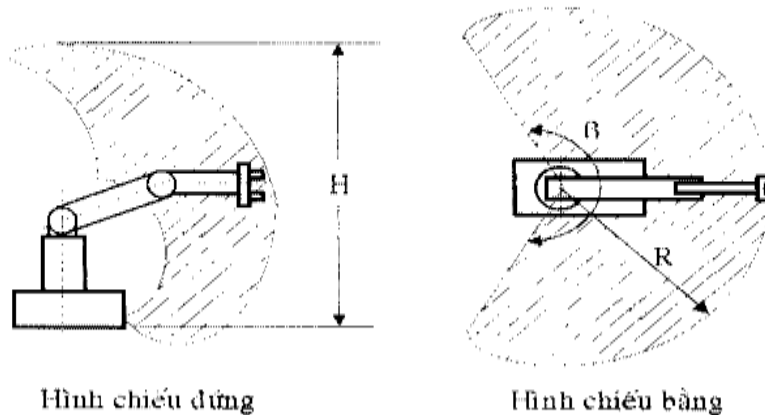
Hình 1.2 : Qui tắc bàn tay phải

Trong robot ta thường dùng chữ O với chỉ số n để chỉ hệ toạ độ gắn trên khâu thứ n. Như vậy hệ toạ độ cơ bản (Hệ toạ độ gắn với khâu cố định) sẽ được ký hiệu lờ \$O_0\$; hệ toạ độ gắn trên các khâu trung gian tương ứng sẽ lờ \$O_1, O_2, \dots, O_{n-1}\$, Hệ toạ độ gắn trên khâu chấp hỡnh cuối ký hiệu lờ \$O_n\$.

1.3.4. Trường công tác của robot (Workspace or Range of motion):

Trường công tác (hay vùng làm việc, không gian công tác) của robot là toàn bộ thể tích được quét bởi khâu chấp hành cuối khi robot thực hiện tất cả các chuyển động có thể. Trường công tác bị ràng buộc bởi các thông

số hình học của robot cũng như các rãnh buộc cơ học của các khớp; ví dụ, một khớp quay có chuyển động nhỏ hơn một góc 360° . Người ta thường dùng hai hình chiếu để mô tả trường công tác của một robot (hình 1.3).

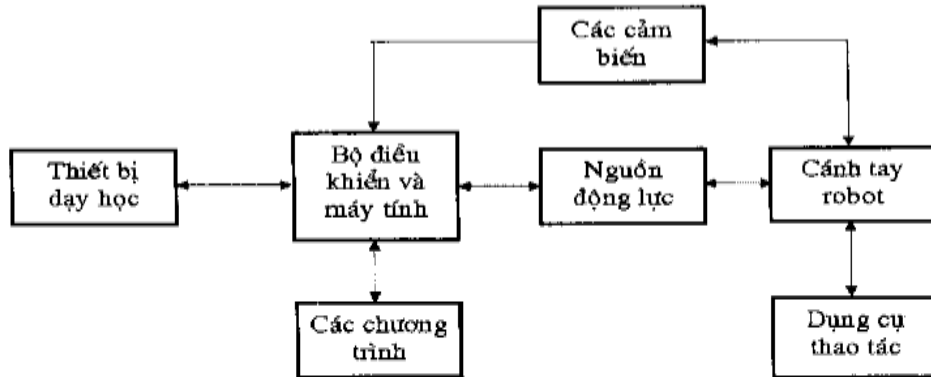


Hình 1.3 : Biểu diễn trường công tác của robot.

1.4. Cấu trúc cơ bản của robot công nghiệp :

1.4.1. Các thành phần chính của robot công nghiệp :

Một robot công nghiệp thường bao gồm các thành phần chính như : cánh tay robot, nguồn động lực, dụng cụ gắn lên khâu chấp hành cuối, các cảm biến, bộ điều khiển , thiết bị dạy học, máy tính ... các phần mềm lập trình cũng nên được coi là một thành phần của hệ thống robot. Mối quan hệ giữa các thành phần trong robot như hình 1.4.



Hình 1.4 : Các thành phần chính của hệ thống robot.

Cánh tay robot (tay máy) là kết cấu cơ khí gồm các khâu liên kết với nhau bằng các khớp động để có thể tạo nên những chuyển động cơ bản của robot.

Nguồn động lực là các động cơ điện (một chiều hoặc động cơ bước), các hệ thống xy lanh khí nén, thủy lực để tạo động lực cho tay máy hoạt động.

Dụng cụ thao tác được gắn trên khâu cuối của robot, dụng cụ của robot có thể có nhiều kiểu khác nhau như : dạng bàn tay để nắm bắt đối tượng hoặc các công cụ làm việc như mỏ hàn, đá mài, đầu phun sơn ...

Thiết bị dạy - học (Teach-Pendant) dùng để dạy cho robot các thao tác cần thiết theo yêu cầu của quá trình làm việc, sau đó robot tự lặp lại các động tác đã được dạy để làm việc (phương pháp lập trình kiểu dạy học).

Các phần mềm để lập trình và các chương trình điều khiển robot được cài đặt trên máy tính, dùng điều khiển robot thông qua bộ điều khiển (Controller). Bộ điều khiển còn được gọi là Modul điều khiển (hay Unit, Driver), nó thường được kết nối với máy tính. Một modul điều khiển có thể còn có các cổng Vào - Ra (I/O port) để làm việc với nhiều thiết bị khác nhau như các cảm biến giúp robot nhận biết trạng thái của bản thân, xác

định vị trí của đối tượng làm việc hoặc các dò tìm khác; điều khiển các băng tải hoặc cơ cấu cấp phối hoạt động phối hợp với robot...

1.4.2. Kết cấu của tay máy :

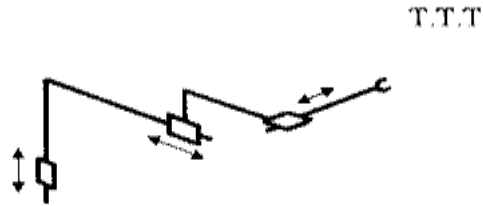
Như đã nói trên, tay máy là thành phần quan trọng, nó quyết định khả năng làm việc của robot. Các kết cấu của nhiều tay máy được phỏng theo cấu tạo và chức năng của tay người; tuy nhiên ngày nay, tay máy được thiết kế rất đa dạng, nhiều cánh tay robot có hình dáng rất khác xa cánh tay người. Trong thiết kế và sử dụng tay máy, chúng ta cần quan tâm đến các thông số hình - động học, là những thông số liên quan đến khả năng làm việc của robot như : tầm với (hay trường công tác), số bậc tự do (thể hiện sự khéo léo linh hoạt của robot), độ cứng vững, tải trọng vật nâng, lực kẹp .

- Chuyển động tịnh tiến theo hướng x,y,z trong không gian Descarte, thông thường tạo nên các hình khối, các chuyển động này thường ký hiệu là T (Translation) hoặc P (Prismatic).

- Chuyển động quay quanh các trục x,y,z ký hiệu là R (Roatation).

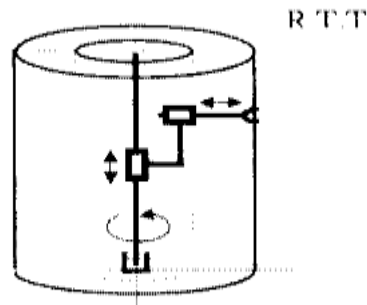
Tuỳ thuộc vào số khâu và sự tổ hợp các chuyển động (R và T) mà tay máy có các kết cấu khác nhau với vùng làm việc khác nhau. Các kết cấu thường gặp của là Robot là robot kiểu tọa độ Đề các, tọa độ trụ, tọa độ cầu, robot kiểu SCARA, hệ tọa độ góc (phồng sinh) ...

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 phối liệu, lắp ráp, hàn trong mặt phẳng (hình 1.5).



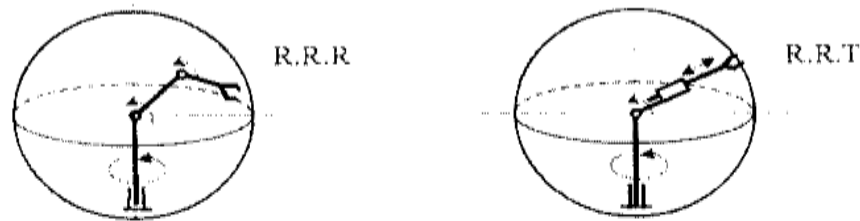
Hình 1.5 : Robot kiểu toạ độ Đê các

Robot kiểu toạ 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 1.6. Có nhiều robot kiểu toạ độ trụ như : robot Versatran của hãng AMF (Hoa Kỳ).



Hình 1.6 : Robot kiểu toạ độ trụ

Robot kiểu toạ 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 toạ độ cầu (hình 1.7).

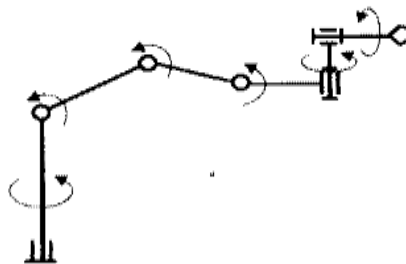


Hình 1.7 : Robot kiểu tọa độ cầu

Robot kiểu tọa độ góc (Hệ tọa độ phỏng sinh) : Đây là kiểu robot được dùng nhiều hơn cả. Ba chuyển động đầu tiên là các chuyển động quay, trục quay thứ nhất vuông góc với hai trục kia. Các chuyển động định hướng khác cũng là các chuyển động quay. Vùng làm việc của tay máy này gần giống một phần khối cầu. Tất cả các khâu đều nằm trong mặt phẳng thẳng đứng nên các tính toán cơ bản là bởi toán phẳng. ưu điểm nổi bật của các loại robot hoạt động theo hệ tọa độ góc là gọn nhẹ, tức là có vùng làm việc tương đối lớn so với kích cỡ của bản thân robot, độ linh hoạt cao.

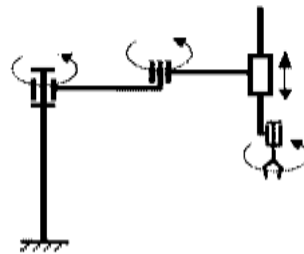
Các robot hoạt động theo hệ tọa độ góc như : Robot PUMA của hãng Unimation - Nokia (Hoa Kỳ - Phần Lan), IRb-6, IRb-60 (Thụy Điển), Toshiba, Mitsubishi, Mazak (Nhật Bản) .V.V...

Ví dụ một robot hoạt động theo hệ tọa độ góc (Hệ tọa độ phỏng sinh), có cấu hình RRR.RRR :



Hình 1.8 : Robot hoạt động theo hệ tọa độ góc.

Robot kiểu SCARA : Robot SCARA ra đời vào năm 1979 tại trường đại học Yamanashi (Nhật Bản) là một kiểu robot mới nhằm đáp ứng sự đa dạng của các quá trình sản xuất. Tên gọi SCARA là viết tắt của "Selective Compliant Articulated Robot Arm" : Tay máy mềm dẻo tùy ý. Loại robot này thường dùng trong công việc lắp ráp nên SCARA đôi khi được giải thích là từ viết tắt của "Selective Compliance Assembly Robot Arm". Ba khớp đầu tiên của kiểu Robot này có cấu hình R.R.T, các trục khớp đều theo phương thẳng đứng. Sơ đồ của robot SCARA như hình 1.9.



Hình 1.9 : Robot kiểu SCARA

1.5. Phân loại Robot công nghiệp :

Robot công nghiệp rất phong phú đa dạng, có thể được phân loại theo các cách sau:

1.4.1. Phân loại theo kết cấu :

Theo kết cấu của tay máy người ta phân thành robot kiểu tọa độ Đề các, Kiểu tọa độ trụ, kiểu tọa độ cầu, kiểu tọa độ góc, robot kiểu SCARA như đã trình bày ở trên.

1.4.2. Phân loại theo hệ thống truyền động :

Có các dạng truyền động phổ biến là :

Hệ truyền động điện : Thường dùng các động cơ điện 1 chiều (DC : Direct Current) hoặc các động cơ bước (step motor). Loại truyền động này dễ điều khiển, kết cấu gọn.

Hệ truyền động thủy lực : có thể đạt được công suất cao, đáp ứng những điều kiện làm việc nặng. Tuy nhiên hệ thống thủy lực thường có kết cấu công kênh, tồn tại độ phi tuyến lớn khó xử lý khi điều khiển.

Hệ truyền động khí nén : có kết cấu gọn nhẹ hơn do không cần dẫn ngược nhưng lại phải gắn liền với trung tâm tạo ra khí nén. Hệ này làm việc với công suất trung bình và nhỏ, kém chính xác, thường chỉ thích hợp với các robot hoạt động theo chương trình định sẵn với các thao tác đơn giản “nhấc lên - đặt xuống” (Pick and Place or PTP : Point To Point).

1.4.3. Phân loại theo ứng dụng :

Dựa vào ứng dụng của robot trong sản xuất có Robot sơn, robot hàn, robot lắp ráp, robot chuyển phôi .v.v...

1.4.4. Phân loại theo cách thức và đặc trưng của phương pháp điều khiển :

Có robot điều khiển hở (mạch điều khiển không có các quan hệ phản hồi), Robot điều khiển kín (hay điều khiển servo) : sử dụng cảm biến, mạch phản hồi để tăng độ chính xác và mức độ linh hoạt khi điều khiển.

Ngoài ra còn có thể có các cách phân loại khác tùy theo quan điểm và mục đích nghiên cứu.

CHƯƠNG II

CÁC PHÉP BIẾN ĐỔI THUẦN NHẤT

(Homogeneous Transformation)

Khi xem xét, nghiên cứu mối quan hệ giữa robot và vật thể ta không những cần quan tâm đến *vị trí* (Position) tuyệt đối của điểm, đường, mặt của vật thể so với điểm tác động cuối (End effector) của robot mà còn cần quan tâm đến vấn đề *định hướng* (Orientation) của khâu chấp hờnh cuối khi vận động hoặc định vị tại một vị trí.

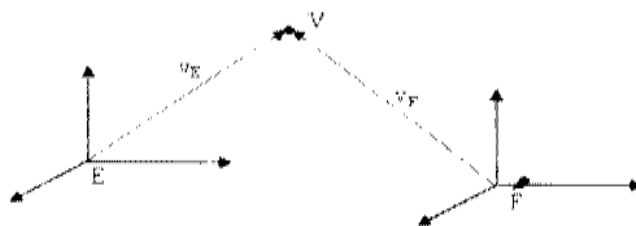
Để mô tả quan hệ về vị trí và hướng giữa robot và vật thể ta phải dùng đến các phép biến đổi thuần nhất.

Chương này cung cấp những hiểu biết cần thiết trước khi đi vào giải quyết các vấn đề liên quan tới động học và động lực học robot.

2.1. Hệ tọa độ thuần nhất :

Để biểu diễn một điểm trong không gian ba chiều, người ta dùng Vector điểm (*Point vector*). Vector điểm thường được ký hiệu bằng các chữ viết thường như u, v, x_1, \dots để mô tả vị trí của điểm U, V, X_1, \dots

Tùy thuộc vào hệ qui chiếu được chọn, trong không gian 3 chiều, một điểm V có thể được biểu diễn bằng nhiều vector điểm khác nhau :



Hình 2.2 : Biểu diễn 1 điểm trong không gian

v_E và v_F là hai vector khác nhau mặc dù cả hai vector cùng mô tả điểm V . Nếu i, j, k là các vec tơ đơn vị của một hệ tọa độ nào đó, chẳng hạn trong E , ta có :

$$\vec{v} = a\vec{i} + b\vec{j} + c\vec{k}$$

với a, b, c là tọa độ vị trí của điểm V trong hệ đó.

Nếu quan tâm đồng thời vấn đề định vị và định hướng, ta phải biểu diễn vector \vec{v} trong không gian bốn chiều với suất vector là một ma trận cột :

$$v = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} \quad \text{Trong đó:} \quad \frac{x}{w} = a; \quad \frac{y}{w} = b; \quad \frac{z}{w} = c$$

với w là một hằng số thực nào đó.

w còn được gọi là *hệ số tỉ lệ*, biểu thị cho chiều thứ tư ngầm định, Nếu w = 1 dễ thấy :

$$\frac{x}{w} = \frac{x}{1} = x = a \quad \frac{y}{w} = \frac{y}{1} = y = b \quad \frac{z}{w} = \frac{z}{1} = z = c$$

Trong trường hợp này thì các tọa độ biểu diễn bằng với tọa độ vật lý của điểm trong không gian 3 chiều, hệ tọa độ sử dụng w=1 được gọi là *hệ tọa độ thuần nhất*.

Với w = 0 ta có : $\frac{x}{w} = \frac{y}{w} = \frac{z}{w} = \infty$

Giới hạn ∞ thể hiện *hướng* của các trục tọa độ.

Nếu w là một hằng số nào đó $\neq 0$ và 1 thì việc biểu diễn điểm trong không gian tương ứng với hệ số tỉ lệ w :

Ví dụ : $\vec{v} = 3\vec{i} + 4\vec{j} + 5\vec{k}$

với w = 1 (trường hợp thuần nhất) :

$$v = [3 \ 4 \ 5 \ 1]^T$$

với w=-10 biểu diễn tương ứng sẽ là :

$$v = [-30 \ -40 \ -50 \ -10]^T$$

Ký hiệu $[\dots]^T$ (Chữ T viết cao lên trên để chỉ phép chuyển đổi vector hàng thành vector cột).

Theo cách biểu diễn trên đây, ta qui ước :

$[0 \ 0 \ 0 \ 0]^T$ là vector không xác định

$[0 \ 0 \ 0 \ n]^T$ với $n \neq 0$ là vector không, trùng với gốc tọa độ

$[x \ y \ z \ 0]^T$ là vector chỉ hướng

$[x \ y \ z \ 1]^T$ là vector điểm trong hệ tọa độ thuần nhất.

2.2. Nhắc lại các phép tính về vectơ và ma trận :

2.2.1. Phép nhân vectơ :

Cho hai vectơ :

$$\vec{a} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}$$

$$\vec{b} = b_x \vec{i} + b_y \vec{j} + b_z \vec{k}$$

Ta có tích vô hướng:

$$a.b = a_x b_x + a_y b_y + a_z b_z$$

Và tích vector :

$$\vec{a} \times \vec{b} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = (a_y b_z - a_z b_y) \vec{i} + (a_z b_x - a_x b_z) \vec{j} + (a_x b_y - a_y b_x) \vec{k}$$

2.2.2 Các phép tính về ma trận:

a/ Phép cộng, trừ ma trận :

Cộng (trừ) ma trận A và B cùng bậc sẽ có ma trận C cùng bậc với các phần tử c_{ij} bằng tổng (hiệu) của các phần tử a_{ij} và b_{ij} (với mọi i,j)

$$A + B = C \text{ Ví } c_{ij} = a_{ij} + b_{ij}.$$

$$A - B = C \text{ Ví } c_{ij} = a_{ij} - b_{ij}.$$

Phép cộng, trừ ma trận có các tính chất giống phép cộng số thực.

b/ Tích của hai ma trận : Tích của ma trận A (kích thước m x n) với ma trận B (kích thước n x p) là ma trận C có kích thước m x p.

Ví dụ :

Cho hai ma trận :

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \text{ và } B = \begin{vmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{vmatrix}$$

$$C = A.B = \begin{vmatrix} 1.1+2.3+3.5 & 1.2+2.4+3.6 \\ 4.1+5.3+6.5 & 4.2+5.4+6.6 \\ 7.1+8.3+9.5 & 7.2+8.4+9.6 \end{vmatrix} = \begin{vmatrix} 22 & 28 \\ 49 & 64 \\ 76 & 100 \end{vmatrix}$$

Phép nhân hai ma trận không có tính giao hoán, nghĩa là : $A . B \neq B . A$

Ma trận đơn vị I (Identity Matrix) giao hoán được với bất kỳ ma trận nào :

$$I.A = A.I$$

Phép nhân ma trận tuân theo các qui tắc sau :

$$1. (k.A).B = k.(A.B) = A.(k.B)$$

$$2. A.(B.C) = (A.B).C$$

$$3. (A + B).C = A.C + B.C$$

$$4. C.(A + B) = C.A + C.B$$

c/ Ma trận nghịch đảo của ma trận thuận nhất :

Một ma trận thuận nhất là ma trận 4 x 4 có dạng :

$$T = \begin{bmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ma trận nghịch đảo của T ký hiệu là T^{-1} :

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -p.n \\ O_x & O_y & O_z & -p.O \\ a_x & a_y & a_z & -p.a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-1)$$

Trong đó p.n là tích vô hướng của vector p và n. Nghĩa là :

$$p.n = p_x n_x + p_y n_y + p_z n_z$$

$$\text{tương tự : } p.O = p_x O_x + p_y O_y + p_z O_z$$

$$\text{và } p.a = p_x a_x + p_y a_y + p_z a_z$$

Ví dụ : tìm ma trận nghịch đảo của ma trận biến đổi thuận nhất :

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Giải : áp dụng công thức (2-1), ta có :

$$H^{-1} = \begin{bmatrix} 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & -2 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Chúng ta kiểm chứng rằng đây chính là ma trận nghịch đảo bằng các nhân ma trận H với H^{-1} :

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 2 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & -2 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Phương pháp tính ma trận nghịch đảo này nhanh hơn nhiều so với phương pháp chung; tuy nhiên nó không áp dụng được cho ma trận 4x4 bất kỳ mà kết quả chỉ đúng với ma trận thuận nhất.

d/ *Vết của ma trận :*

Vết của ma trận vuông bậc n là tổng các phần tử trên đường chéo :

$$\text{Trace(A) hay Tr(A)} = \sum_{i=1}^n a_{ii}$$

Một số tính chất quan trọng của vết ma trận :

1/ $\text{Tr(A)} = \text{Tr(A}^T)$

2/ $\text{Tr(A+B)} = \text{Tr(A)} + \text{Tr(B)}$

3/ $\text{Tr(A.B)} = \text{Tr(B.A)}$

4/ $\text{Tr(ABC}^T) = \text{Tr(CB}^T\text{A}^T)$

e/ *Đạo hàm và tích phân ma trận :*

Nếu các phần tử của ma trận A là hàm nhiều biến, thì các phần tử của ma trận đạo hàm bằng đạo hàm riêng của các phần tử ma trận A theo biến tương ứng.

Ví dụ:

Cho

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

thì

$$dA = \begin{bmatrix} \frac{\partial a_{11}}{\partial t} & \frac{\partial a_{12}}{\partial t} & \frac{\partial a_{13}}{\partial t} & \frac{\partial a_{14}}{\partial t} \\ \frac{\partial a_{21}}{\partial t} & \frac{\partial a_{22}}{\partial t} & \frac{\partial a_{23}}{\partial t} & \frac{\partial a_{24}}{\partial t} \\ \frac{\partial a_{31}}{\partial t} & \frac{\partial a_{32}}{\partial t} & \frac{\partial a_{33}}{\partial t} & \frac{\partial a_{34}}{\partial t} \\ \frac{\partial a_{41}}{\partial t} & \frac{\partial a_{42}}{\partial t} & \frac{\partial a_{43}}{\partial t} & \frac{\partial a_{44}}{\partial t} \end{bmatrix} dt$$

Tương tự, phép tích phân của ma trận A là một ma trận, có :

$$\int A(t)dt = \{ \int a_{ij}(t)dt \}$$

2.3. Các phép biến đổi

Cho u là vectơ biểu diễn điểm cần biến đổi, h là vectơ dẫn được biểu diễn bằng một ma trận H gọi là ma trận chuyển đổi. Ta có :

$$v = H.u$$

v là vectơ biểu diễn điểm sau khi đã biến đổi.

2.3.1. Phép biến đổi tịnh tiến (Translation) :

Giả sử cần tịnh tiến một điểm hoặc một vật thể theo vectơ dẫn

$\vec{h} = a\vec{i} + b\vec{j} + c\vec{k}$. Trước hết ta có định nghĩa của ma trận chuyển đổi H :

$$H = \text{Trans}(a,b,c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Gọi u là vectơ biểu diễn điểm cần tịnh tiến : $u = [x \ y \ z \ w]^T$

Thì v là vectơ biểu diễn điểm đã biến đổi tịnh tiến được xác định bởi :

$$v = H.u = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x+aw \\ y+bw \\ z+cw \\ w \end{bmatrix} = \begin{bmatrix} x/w+a \\ y/w+b \\ z/w+c \\ 1 \end{bmatrix}$$

Như vậy bản chất của phép biến đổi tịnh tiến là phép cộng vectơ giữa vectơ biểu diễn điểm cần chuyển đổi và vectơ dẫn.

Ví dụ:

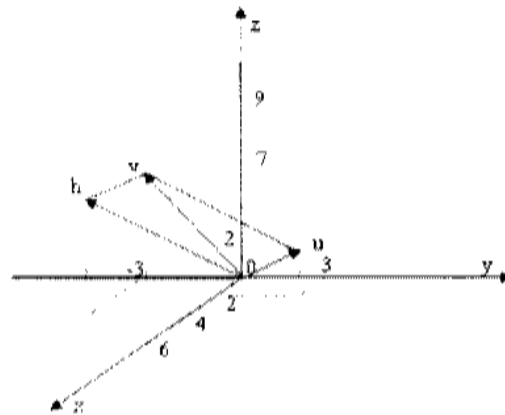
$$\vec{u} = 2\vec{i} + 3\vec{j} + 2\vec{k}$$

$$\vec{h} = 4\vec{i} - 3\vec{j} + 7\vec{k}$$

thì

$$\mathbf{v} = \mathbf{H}\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2+4 \\ 3-3 \\ 2+7 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 9 \\ 1 \end{bmatrix}$$

và viết là: $\mathbf{v} = \text{Trans}(a,b,c)\mathbf{u}$



Hình 2.4: Phép biến đổi tịnh tiến trong không gian

2.3.2. Phép quay (Rotation) quanh các trục tọa độ :

Giả sử ta cần quay một điểm hoặc một vật thể xung quanh trục tọa độ nào đó với góc quay θ° , ta lần lượt có các ma trận chuyển đổi như sau :

$$\text{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$\text{Rot}(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$\text{Rot}(z, \theta^\circ) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Ví dụ : Cho điểm U biểu diễn bởi $\vec{u} = 7\vec{i} + 3\vec{j} + 2\vec{k}$ quay xung quanh z một góc $\theta = 90^\circ$ (hình 2.5). Ta có:

$$\mathbf{v} = \text{Rot}(z, 90^\circ)\mathbf{u} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 7 \\ 2 \\ 1 \end{bmatrix}$$

Nếu cho điểm đã biến đổi tiếp tục quay xung quanh y một góc 90° ta có :

$$\mathbf{w} = \text{Rot}(y, 90^\circ)\mathbf{v} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ 7 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix}$$

Và có thể biểu diễn :

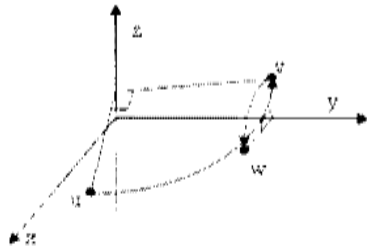
$$\mathbf{w} = \text{Rot}(y, 90^\circ) \cdot \text{Rot}(z, 90^\circ) \cdot \mathbf{u} = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 1 \end{bmatrix}$$

Chú ý : Nếu đổi thứ tự quay ta sẽ được $w' \neq w$ (hình 2.6), cụ thể : cho U quay quanh y trước 1 góc 90° , ta có :

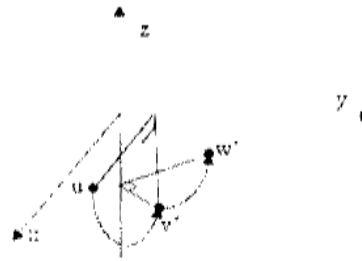
$$\mathbf{v}' = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ -7 \\ 1 \end{bmatrix} = \text{Rot}(y, 90^\circ) \cdot \mathbf{u}$$

Sau đó cho điểm vừa biến đổi quay quanh z một góc 90° , ta được :

$$\text{Rõ ràng : } \text{Rot}(y, 90^\circ) \cdot \text{Rot}(z, 90^\circ) \mathbf{u} \neq \text{Rot}(z, 90^\circ) \cdot \text{Rot}(y, 90^\circ) \mathbf{u}$$



Hình 2.5
 $w = Rot(y, 90^\circ). Rot(z, 90^\circ)u$



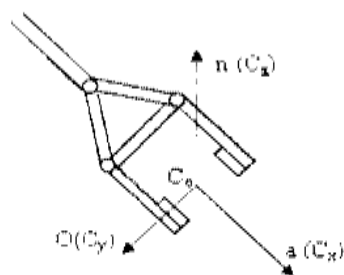
Hình 2.6
 $w' = Rot(z, 90^\circ). Rot(y, 90^\circ)u$

2.3.3. Phép quay tổng quát :

Trong mục trên, ta vừa nghiên cứu các phép quay cơ bản xung quanh các trục tọa độ x,y,z của hệ tọa độ chuẩn O(x,y,z). Trong phần này, ta nghiên cứu phép quay quanh một vectơ k bất kỳ một góc θ . Ràng buộc duy nhất là vectơ k phải trùng với gốc của một hệ tọa độ xác định trước.

Ta hãy khảo sát một hệ tọa độ C, gắn lên điểm tác động cuối (bàn tay) của robot, hệ C được biểu diễn bởi :

$$C = \begin{bmatrix} C_x & C_y & C_z & C_0 \\ n_x & O_x & a_x & 0 \\ n_y & O_y & a_y & 0 \\ n_z & O_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Hình 2.7 : Hệ tọa độ gắn trên khâu chấp hành cuối (bàn tay)

Khi gắn hệ tọa độ này lên bàn tay robot (hình 2.7), các vectơ đơn vị được biểu thị như sau :

a : là vectơ có hướng tiếp cận với đối tượng (approach);

O: là vectơ có hướng mà theo đó các ngón tay nắm vào khi cầm nắm đối tượng (Occupation);

n : Vectơ pháp tuyến với (O,a) (Normal).

Bây giờ ta hãy coi vectơ bất kỳ k (mà ta cần thực hiện phép quay quanh nó một góc θ) là một trong các vectơ đơn vị của hệ C.

Chẳng hạn :
$$\vec{k} = a_x \vec{i} + a_y \vec{j} + a_z \vec{k}$$

Lúc đó, phép quay $\text{Rot}(k, \theta)$ sẽ trở thành phép quay $\text{Rot}(C_z, \theta)$.

Nếu ta có T mô tả trong hệ gốc trong đó k là vectơ bất kỳ, thì ta có X mô tả trong hệ C với k là một trong các vectơ đơn vị. Từ điều kiện biến đổi thuận nhất, T và X có liên hệ :

$$T = C.X$$

hay
$$X = C^{-1}.T$$

Lúc đó các phép quay dưới đây là đồng nhất :

$$\text{Rot}(k, \theta) = \text{Rot}(C_z, \theta)$$

hay là
$$\text{Rot}(k, \theta).T = C.\text{Rot}(z, \theta).X = C.\text{Rot}(z, \theta).C^{-1}.T$$

Vậy
$$\text{Rot}(k, \theta) = C.\text{Rot}(z, \theta).C^{-1} \quad (2.6)$$

Trong đó $\text{Rot}(z, \theta)$ là phép quay cơ bản quanh trục z một góc θ , có thể sử dụng công thức (2.5) như đã trình bày.

C^{-1} là ma trận nghịch đảo của ma trận C. Ta có :

$$C^{-1} = \begin{bmatrix} n_x & n_y & n_z & 0 \\ O_x & O_y & O_z & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thay các ma trận vào vế phải của phương trình (2.6) :

$$\text{Rot}(k,\theta) = \begin{bmatrix} n_x & O_x & a_x & 0 \\ n_y & O_y & a_y & 0 \\ n_z & O_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & n_y & n_z & 0 \\ O_x & O_y & O_z & 0 \\ a_x & a_y & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nhân 3 ma trận này với nhau ta được :

Trên đây ta đã nghiên cứu các bài toán thuận, nghĩa là chỉ định trục quay và góc quay trước- xem xét kết quả biến đổi theo các phép quay đã chỉ định.

Ngược lại với bài toán trên, giả sử ta đã biết kết quả của một phép biến đổi nào đó, ta phải đi tìm trục quay k và góc quay θ tương ứng. Giả sử kết quả của phép biến đổi thuận nhất $R=Rot(k, \theta)$, xác định bởi :

$$R = \begin{bmatrix} n_x & O_x & a_x & 0 \\ n_y & O_y & a_y & 0 \\ n_z & O_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ta cần xác định trục quay k và góc quay θ . Ta đã biết $Rot(k, \theta)$ được định nghĩa bởi ma trận (2.6) , nên :

$$\begin{bmatrix} n_x & O_x & a_x & 0 \\ n_y & O_y & a_y & 0 \\ n_z & O_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} k_x k_x \text{vers}\theta + \cos\theta & k_x k_y \text{vers}\theta - k_x \sin\theta & k_x k_z \text{vers}\theta + k_x \sin\theta & 0 \\ k_x k_y \text{vers}\theta + k_y \sin\theta & k_y k_y \text{vers}\theta + \cos\theta & k_y k_z \text{vers}\theta - k_y \sin\theta & 0 \\ k_x k_z \text{vers}\theta + k_z \sin\theta & k_y k_z \text{vers}\theta + k_z \sin\theta & k_z k_z \text{vers}\theta + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

Bước 1 : Xác định góc quay θ .

* Cộng đường chéo của hai ma trận ở hai vế ta có :

$$\begin{aligned} n_x + O_y + a_z + 1 &= k_x^2 \text{vers}\theta + \cos\theta + k_y^2 \text{vers}\theta + \cos\theta + k_z^2 \text{vers}\theta + \cos\theta + 1 \\ &= (1 - \cos\theta)(k_x^2 + k_y^2 + k_z^2) + 3\cos\theta + 1 \\ &= 1 - \cos\theta + 3\cos\theta + 1 \\ &= 2(1 + \cos\theta) \\ \Rightarrow \cos\theta &= (n_x + O_y + a_z + 1)/2 \end{aligned}$$

* Tính hiệu các phần tử tương đương của hai ma trận, chẳng hạn :

$$\left. \begin{aligned} O_z - a_y &= 2k_x \sin\theta \\ a_x - n_z &= 2k_y \sin\theta \\ n_y - O_x &= 2k_z \sin\theta \end{aligned} \right\} \tag{2.10}$$

Bình phương hai vế của các phương trình trên rồi cộng lại ,ta có :

$$(O_z - a_y)^2 + (a_x - n_z)^2 + (n_y - O_x)^2 = 4 \sin^2\theta$$

$$\Rightarrow \sin\theta = \pm \frac{1}{2} \sqrt{(O_z - a_y)^2 + (a_x - n_z)^2 + (n_y - O_x)^2}$$

Với $0 \leq \theta \leq 180^\circ$:

$$\text{tg}\theta = \frac{\sqrt{(O_z - a_y)^2 + (a_x - n_z)^2 + (n_y - O_x)^2}}{(n_x + O_y + a_z + 1)}$$

Và trục k được định nghĩa bởi :

$$k_x = \frac{O_z - a_y}{2\sin\theta}; \quad k_y = \frac{a_x - n_z}{2\sin\theta}; \quad k_z = \frac{n_y - O_x}{2\sin\theta} \quad (2.11)$$

Đề ý rằng với các công thức (2.8) :

- Nếu $\theta = 0^\circ$ thì k_x, k_y, k_z có dạng $\frac{0}{0}$. Lúc này phải chuẩn hoá k sao cho

$$|k| = 1$$

- Nếu $\theta = 180^\circ$ thì k_x, k_y, k_z có dạng $a \neq \frac{0}{0}$. Lúc này k không xác định

được, ta phải dùng cách tính khác cho trường hợp này :

Xét các phần tử tương đương của hai ma trận (2.9) :

$$n_x = k_x^2 \text{vers}\theta + \cos\theta$$

$$O_y = k_y^2 \text{vers}\theta + \cos\theta$$

$$a_z = k_z^2 \text{vers}\theta + \cos\theta$$

Từ đây ta suy ra :

$$k_x = \pm \sqrt{\frac{n_x - \cos\theta}{\text{vers}\theta}} = \pm \sqrt{\frac{n_x - \cos\theta}{1 - \cos\theta}}$$

$$k_y = \pm \sqrt{\frac{O_y - \cos\theta}{\text{vers}\theta}} = \pm \sqrt{\frac{O_y - \cos\theta}{1 - \cos\theta}}$$

$$k_z = \pm \sqrt{\frac{a_z - \cos\theta}{\text{vers}\theta}} = \pm \sqrt{\frac{a_z - \cos\theta}{1 - \cos\theta}}$$

Trong khoảng $90^\circ \leq \theta \leq 180^\circ$ $\sin\theta$ luôn luôn dương

Dựa vào hệ phương trình (2.10) ta thấy k_x, k_y, k_z luôn có cùng dấu với vế trái.

Ta dùng hàm Sgn(x) để biểu diễn quan hệ “cùng dấu với x”, như vậy :

$$\left. \begin{aligned} k_x &= \text{Sgn}(O_z - a_y) \sqrt{\frac{n_x - \cos\theta}{1 - \cos\theta}} \\ k_y &= \text{Sgn}(a_x - n_z) \sqrt{\frac{O_y - \cos\theta}{1 - \cos\theta}} \\ k_z &= \text{Sgn}(n_y - O_x) \sqrt{\frac{a_z - \cos\theta}{1 - \cos\theta}} \end{aligned} \right\} \quad (2.12)$$

Hệ phương trình (2.12) chỉ dùng để xác định xem trong các k_x, k_y, k_z thành phần nào có giá trị lớn nhất. Các thành phần còn lại nên tính theo thành phần có giá trị lớn nhất để xác định k được thuận tiện. Lúc đó dùng phương pháp cộng các cặp còn lại của các phần tử đối xứng qua đường chéo ma trận chuyển đổi (2.9) :

$$\begin{aligned} n_y + O_x &= 2k_x k_y \text{vers} \theta = 2k_x k_y (1 - \cos \theta) \\ O_x + a_y &= 2k_y k_z \text{vers} \theta = 2k_y k_z (1 - \cos \theta) \\ a_x + n_z &= 2k_z k_x \text{vers} \theta = 2k_z k_x (1 - \cos \theta) \end{aligned} \tag{2.13}$$

Giả sử theo hệ (2.12) ta có k_x là lớn nhất, lúc đó k_y, k_z sẽ tính theo k_x bằng hệ (2.13); cụ thể là:

$$\begin{aligned} k_y &= \frac{n_y + O_x}{2k_x(1 - \cos \theta)} \\ k_z &= \frac{a_x + n_z}{2k_x(1 - \cos \theta)} \end{aligned}$$

Ví dụ : Cho $R = \text{Rot}[y, 90^\circ] \text{Rot}[z, 90^\circ]$. Hãy xác định k và θ để $R = \text{Rot}[k, \theta]$. Ta đã biết :

$$R = \text{Rot}(y, 90^\circ) \cdot \text{Rot}(z, 90^\circ) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Ta có } \cos \theta = (n_x + O_y + a_z - 1) / 2 = (0 + 0 + 0 - 1) / 2 = -1/2$$

$$\begin{aligned} \sin \theta &= \frac{1}{2} \sqrt{(O_z - a_y)^2 + (a_x - n_z)^2 + (n_y - O_x)^2} \\ &= \frac{1}{2} \sqrt{(1 - 0)^2 + (1 - 0)^2 + (1 - 0)^2} = \frac{\sqrt{3}}{2} \end{aligned}$$

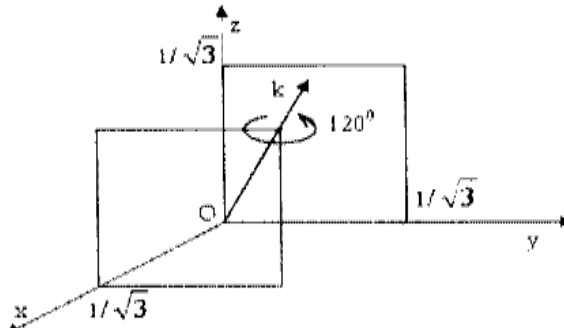
$$\Rightarrow \text{tg} \theta = -\sqrt{3} \text{ và } \theta = 120^\circ$$

Theo (2.12), ta có :

$$k_x = k_y = k_z = + \sqrt{\frac{0+1}{2}} = \frac{1}{\sqrt{3}}$$

Vậy : $R = \text{Rot}(y, 90^\circ) \cdot \text{Rot}(z, 90^\circ) = \text{Rot}(\bar{k}, 120^\circ)$; với :

$$\bar{k} = \frac{1}{\sqrt{3}} \bar{i} + \frac{1}{\sqrt{3}} \bar{j} + \frac{1}{\sqrt{3}} \bar{k}$$

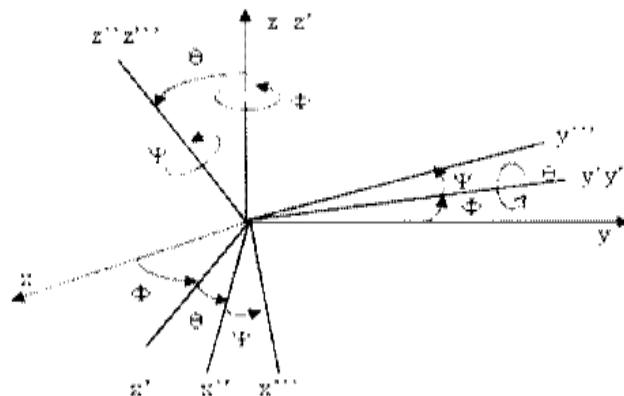


Hình 2.8 : Tìm góc quay và trục quay tương đương

2.3.5. Phép quay Euler :

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 Φ xung quanh trục z,
- Quay tiếp một góc θ xung quanh trục y mới, đó là y' ,
- Cuối cùng quay một góc ψ quanh trục z mới, đó là z'' (Hình 2.9).



Hình 2.9 : Phép quay Euler

Ta biểu diễn phép quay Euler bằng cách nhân ba ma trận quay với nhau :

$$\text{Euler}(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \text{Rot}(y, \theta) \text{Rot}(z, \psi) \quad (2.14)$$

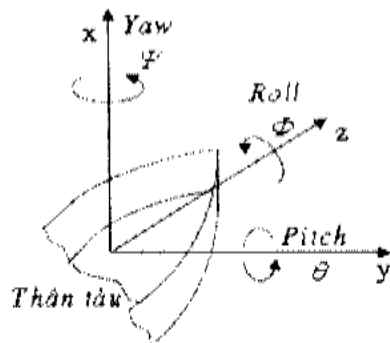
Nói chung, kết quả của phép quay phụ thuộc chặt chẽ vào thứ tự quay, tuy nhiên, ở phép quay Euler, nếu thực hiện theo thứ tự ngược lại, nghĩa là quay góc ψ quanh z rồi tiếp đến quay góc θ quanh y và cuối cùng quay góc Φ quanh z cũng đưa đến kết quả tương tự (Xét trong cùng hệ qui chiếu).

$$\begin{aligned} \text{Euler}(\Phi, \theta, \psi) &= \text{Rot}(z, \Phi) \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \Phi & -\sin \Phi & 0 & 0 \\ \sin \Phi & \cos \Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta \cos \psi & -\cos \theta \sin \psi & \sin \theta & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \Phi \cos \theta \cos \psi - \sin \Phi \sin \psi & -\cos \Phi \cos \theta \sin \psi - \sin \Phi \cos \psi & \cos \Phi \sin \theta & 0 \\ \sin \Phi \cos \theta \cos \psi + \cos \Phi \sin \psi & -\sin \Phi \cos \theta \sin \psi + \cos \Phi \cos \psi & \sin \Phi \sin \theta & 0 \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15) \end{aligned}$$

2.3.6. Phép quay Roll-Pitch-Yaw :

Một phép quay định hướng khác cũng thường được sử dụng là phép quay Roll-Pitch và Yaw.

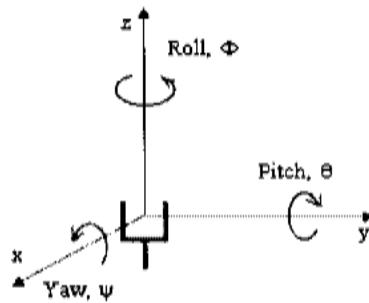
Ta tưởng tượng, gắn hệ tọa độ xyz lên thân một con tàu. Dọc theo thân tàu là trục z , Roll là chuyển động lắc của thân tàu, tương đương với việc quay thân tàu một góc Φ quanh trục z . Pitch là sự bồng bênh, tương đương với quay một góc θ xung quanh trục y và Yaw là sự lệch hướng, tương đương với phép quay một góc ψ xung quanh trục x (Hình 2.10).



Hình 2.10: Phép quay Roll-Pitch-Yaw

Các phép quay áp dụng cho khâu chấp hành cuối của robot như hình 2.11. Ta xác định thứ tự quay và biểu diễn phép quay như sau :

$$RPY(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \text{Rot}(y, \theta) \text{Rot}(x, \psi) \tag{2.16}$$



Hình 2.11 : Các góc quay Roll-Pitch và Yaw của bàn tay Robot.

nghĩa là, quay một góc ψ quanh trục x, tiếp theo là quay một góc θ quanh trục y và sau đó quay một góc Φ quanh trục z.

Thực hiện phép nhân các ma trận quay, các chuyển vị Roll, Pitch và Yaw được biểu thị như sau :

$$\begin{aligned}
 \text{RPY}(\Phi, \theta, \psi) &= \text{Rot}(z, \Phi) \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos\Phi & -\sin\Phi & 0 & 0 \\ \sin\Phi & \cos\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta\sin\psi & \sin\theta\cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos\Phi\cos\theta & \cos\Phi\sin\theta\sin\psi - \sin\Phi\cos\psi & \cos\Phi\sin\theta\cos\psi + \sin\Phi\sin\psi & 0 \\ \sin\Phi\cos\theta & \sin\Phi\sin\theta\sin\psi + \cos\Phi\cos\psi & \sin\Phi\sin\theta\cos\psi - \cos\Phi\sin\psi & 0 \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.17}
 \end{aligned}$$

2.4. Biến đổi hệ tọa độ và mối quan hệ giữa các hệ tọa độ biến đổi :

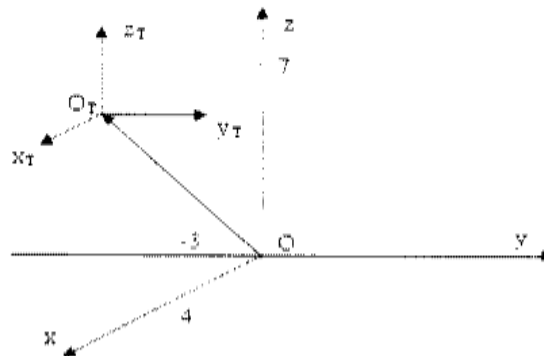
2.4.1 Biến đổi hệ tọa độ :

Giả sử cần tịnh tiến gốc tọa độ Đề cát $O(0, 0, 0)$ theo một vector dẫn

$\vec{h} = 4\vec{i} - 3\vec{j} + 7\vec{k}$ (hình 2.12) . Kết quả của phép biến đổi là :

$$\mathbf{O}_T = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -3 \\ 7 \\ 1 \end{bmatrix}$$

Nghĩa là gốc ban đầu có tọa độ $O(0, 0, 0)$ đã chuyển đổi đến gốc mới O_T có tọa độ $(4, -3, 7)$ so với hệ tọa độ cũ.



Hình 2.12 : Phép biến đổi tịnh tiến hệ tọa độ

Tuy nhiên trong phép biến đổi này các trục tọa độ của O_T vẫn song song và đồng hướng với các trục tọa độ của O .

Nếu ta tiếp tục thực hiện các phép biến đổi quay :

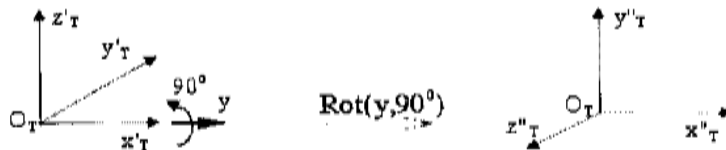
$$\text{Rot}(y,90^\circ)\text{Rot}(z,90^\circ).O_T$$

ta sẽ có một hệ tọa độ hoàn toàn mới, cụ thể tại gốc tọa độ mới (4,-3,7) khi cho hệ O_T quay quanh z một góc 90° (chiều quay dương qui ước là ngược chiều kim đồng hồ), ta có :

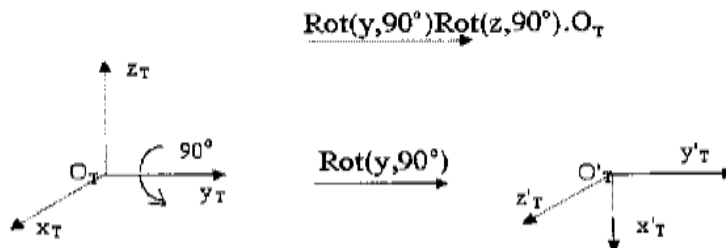


Ta tiếp tục quay hệ O_T quanh trục y (trục y của hệ tọa độ gốc) một góc 90° ,

Ta có :



Ví dụ trên đây ta đã chọn **Hệ tạo độ cơ sở làm hệ qui chiếu** và thứ tự thực hiện các phép biến đổi là từ **Phải sang Trái**. Nếu thực hiện các phép biến đổi theo thứ tự ngược lại từ **Trái sang Phải** thì **hệ qui chiếu được chọn là các hệ tọa độ trung gian**. Xét lại ví dụ trên :



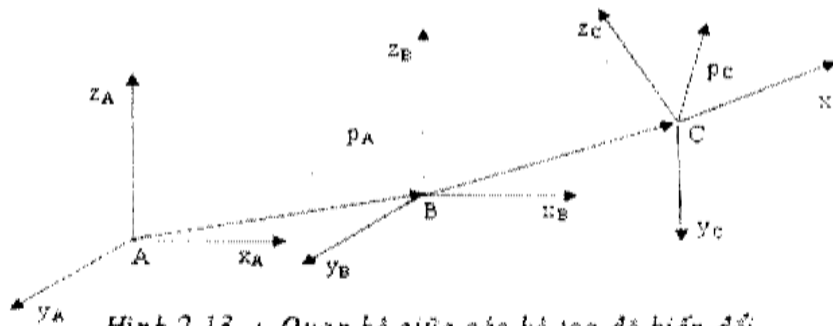
Ta tiếp tục quay hệ O'_T quanh trục z (Bây giờ là trục z'_T của hệ toạ độ mới) một góc 90° :



Như vậy kết quả của hai phương pháp quay là giống nhau, nhưng về ý nghĩa vật lý thì khác nhau.

2.4.2. Quan hệ giữa các hệ toạ độ biến đổi :

Giả sử ta có 3 hệ toạ độ A, B, C; Hệ B có quan hệ với hệ A qua phép biến đổi và hệ C có quan hệ với hệ B qua phép biến đổi ${}^B T_C$. Ta có điểm P trong hệ C ký hiệu P_C , ta tìm mối quan hệ của điểm P trong hệ A, tức là tìm P_A (Hình 2.13) :



Hình 2.13 : Quan hệ giữa các hệ toạ độ biến đổi.

Chúng ta có thể biến đổi p_C thành p_B như sau :

$$p_B = {}^B T_C p_C \tag{2.18}$$

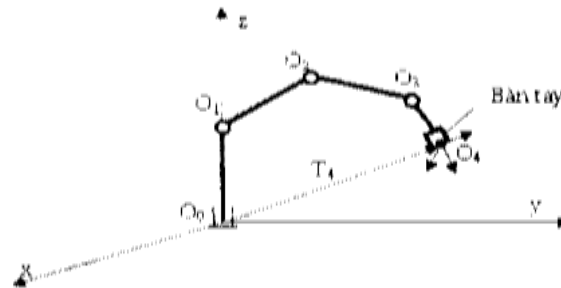
Sau đó biến đổi p_B thành p_A như sau :

$$p_A = {}^A T_B p_B \tag{2.19}$$

Kết hợp (2.18) và (2.19) ta có :

$$p_A = {}^A T_B {}^B T_C p_C \tag{2.20}$$

Qua ví dụ trên ta thấy có thể mô tả mối quan hệ giữa hệ tọa độ gắn trên điểm tác động cuối với hệ tọa độ cơ bản, thông qua mối quan hệ của các hệ tọa độ trung gian gắn trên các khâu của robot, bằng ma trận T như hình 2.14.



Hình 2.14 : Hệ tọa độ cơ bản (base) và các hệ tọa độ trung gian của Robot.

2.5. Mô tả một vật thể :

Các vật thể là đối tượng làm việc của robot rất đa dạng và phong phú, tuy nhiên có thể dựa vào những đặc điểm hình học để mô tả chúng. Ta có thể chia hình dáng vật thể thành 3 nhóm chính sau :

- Nhóm vật thể tròn xoay (Rotative)
- Nhóm vật thể có góc cạnh (Prismatic)
- Nhóm vật thể có cấu trúc hỗn hợp (Kombination)

Nhóm vật thể tròn xoay có các giá trị đặc trưng là tọa độ tâm và bán kính mặt cong.

Nhóm vật thể có góc cạnh đặc trưng bằng tọa độ của các điểm giới hạn.

Nhóm còn lại có các giá trị đặc trưng hỗn hợp.

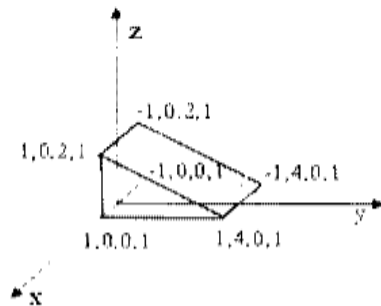
Tuy nhiên, đối với hoạt động cầm nắm đối tượng và quá trình vận động của robot việc mô tả vật thể cần phải gắn liền với các phép biến đổi thuận nhất. Ta xét ví dụ sau đây : Cho một vật hình lăng trụ đặt trong hệ tọa độ chuẩn O(xyz) như hình 2.15.

Ta thực hiện các phép biến đổi sau :

$$H = \text{Trans}(4,0,0)\text{Rot}(y,90^{\circ})\text{Rot}(z,90^{\circ})$$

Với vị trí của vật thể, ta có ma trận tọa độ của 6 điểm đặc trưng mô tả nó là :

①	②	③	④	⑤	⑥
1	-1	-1	1	1	-1
0	0	0	0	4	4
0	0	2	2	0	0
1	1	1	1	1	1



Hình 2.15 : Mô tả vật thể

Sau khi thực hiện các phép biến đổi :

- Quay vật thể quanh trục z một góc 90° (Hình 2.16),
- Cho vật thể quay quanh trục y một góc 90° (Hình 2.17),
- Tiếp tục tịnh tiến vật thể dọc theo trục x một đoạn bằng 4 đơn vị (hình 2.18)

ta xác định được ma trận tọa độ các điểm giới hạn của vật thể ở vị trí đã được biến đổi như sau (các phép quay đã chọn hệ qui chiếu là hệ tọa độ gốc) :

Một chuyển vị có thể là kết quả liên tiếp của nhiều phép biến đổi quay và tịnh tiến. Tuy nhiên ta cần lưu ý đến thứ tự của các phép biến đổi, nếu thay đổi thứ tự thực hiện có thể dẫn đến các kết quả khác nhau.

Bài tập chương II :

Bài 1 : Cho điểm A biểu diễn bởi vector điểm $v=[2\ 4\ 1\ 1]^T$. Tịnh tiến điểm A theo vector dẫn $h = [1\ 2\ 1\ 1]^T$, sau đó tiếp tục quay điểm đã biến đổi quanh trục x một góc 90° . Xác định vector biểu diễn điểm A sau hai phép biến đổi.

Bài 2 : Viết ma trận biến đổi thuần nhất biểu diễn các phép biến đổi sau :

$$H = \text{Trans}(3,7,9)\text{Rot}(x,-90^\circ)\text{Rot}(z,90^\circ)$$

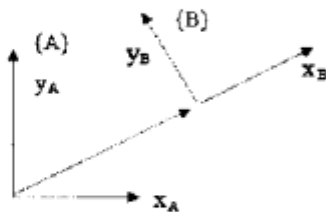
Bài 3 : Cho ma trận biến đổi thuần nhất A, tìm ma trận nghịch đảo A^{-1} và kiểm chứng.

$$A = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 2 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bài 4 : Hình vẽ 2-19 mô tả hệ tọa độ {B} đã được quay đi một góc 30° xung quanh trục z_A , tịnh tiến dọc theo trục x_A 4 đơn vị và tịnh tiến dọc theo y_A 3 đơn vị.

(a) Mô tả mối qua hệ của {B} đối với {A} : ${}^A T_B$?

(b) Tìm mối quan hệ ngược lại ${}^B T_A$?



Hình 2.19 : Quan hệ {A} và {B}

Bài 5: Cho $k = \frac{1}{\sqrt{3}}(1,1,1)^T, \theta = 90^\circ$. Tìm ma trận $R = \text{Rot}(k, \theta)$.

Bài 6: Xác định các góc quay Euler, và các góc quay RPY khi biết ma trận T_6

:

$$T_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & -1 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Bài 7: Một vật thể đặt trong một hệ tọa độ tham chiếu được xác định bởi phép biến đổi :

$${}^U T_P = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 2 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Một robot mù hệ tọa độ chuẩn có liên hệ với hệ tọa độ tham chiếu bởi phép biến đổi:

$${}^U T_R = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Chúng ta muốn đặt bàn tay của robot lên vật thể, đó là làm cho hệ tọa độ gắn trên bàn tay trùng với hệ tọa độ của vật thể. Tìm phép biến đổi ${}^R T_H$ (biểu diễn mối quan hệ bàn tay và hệ tọa độ gốc của robot) để thực hiện điều nói trên.

CHƯƠNG III

PHƯƠNG TRÌNH ĐỘNG HỌC CỦA ROBOT (Kinematic Equations)

3.1. Dẫn nhập :

Bất kỳ một robot nào cũng có thể coi là một tập hợp các khâu (links) gắn liền với các khớp (joints). Ta hãy đặt trên mỗi khâu của robot một hệ toạ độ. Sử dụng các phép biến đổi thuần nhất có thể mô tả vị trí tương đối và hướng giữa các hệ toạ độ này. Denavit, J. đã gọi biến đổi thuần nhất mô tả quan hệ giữa một khâu và một khâu kế tiếp là một ma trận A. Nói đơn giản hơn, một ma trận A là một mô tả biến đổi thuần nhất bởi phép quay và phép tịnh tiến tương đối giữa hệ toạ độ của hai khâu liền nhau. A1 mô tả vị trí và hướng của khâu đầu tiên; A2 mô tả vị trí và hướng của khâu thứ hai so với khâu thứ nhất. Như vậy vị trí và hướng của khâu thứ hai so với hệ toạ độ gốc được biểu diễn bởi ma trận :

$$T2 = A1.A2$$

Cũng như vậy, A3 mô tả khâu thứ ba so với khâu thứ hai và :

$$T3 = A1.A2.A3 ;$$

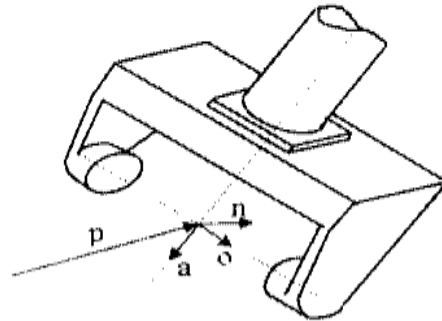
v.v...

Cũng theo Denavit, tích của các ma trận A được gọi là ma trận T, thường có hai chỉ số: trên và dưới. Chỉ số trên chỉ hệ toạ độ tham chiếu tới, bỏ qua chỉ số trên nếu chỉ số đó bằng 0. Chỉ số dưới thường dùng để chỉ khâu chấp hành cuối. Nếu một robot có 6 khâu ta có :

$$T6 = A1.A2.A3.A4.A5.A6 \quad (3.1)$$

T6 mô tả mối quan hệ về hướng và vị trí của khâu chấp hành cuối đối với hệ toạ độ gốc. Một robot 6 khâu có thể có 6 bậc tự do và có thể được định vị trí và định hướng trong trường vận động của nó (range of motion). Ba bậc tự do xác định vị trí thuần tuý và ba bậc tự do khác xác

định hướng mong muốn. T6 sẽ là ma trận trình bày cả hướng và vị trí của robot. Hình 3.1 mô tả quan hệ đó với bàn tay máy. Ta đặt gốc tọa độ của hệ mô tả tại điểm giữa của các ngón tay. Gốc tọa độ này được mô tả bởi vector p (xác định vị trí của bàn tay). Ba vector đơn vị mô tả hướng của bàn tay được xác định như sau :



Hình 3.1 : Các vector định vị trí và định hướng của bàn tay máy

* Vector có hướng mà theo đó bàn tay sẽ tiếp cận đến đối tượng, gọi là vector a (approach).

* Vector có hướng mà theo đó các ngón tay của bàn tay nắm vào nhau khi cầm nắm đối tượng, gọi là vector o (Occupation).

* Vector cuối cùng là vector pháp tuyến n (normal), do vậy ta có :

$$n = a \times o$$

Chuyển vị T6 như vậy sẽ bao gồm các phần tử :

$$T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tổng quát, ma trận T6 có thể biểu diễn gọn hơn như sau :

$$T_6 = \left[\begin{array}{ccc|c} \text{Ma trận định hướng R} & & & \text{Vector vị trí p} \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

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.

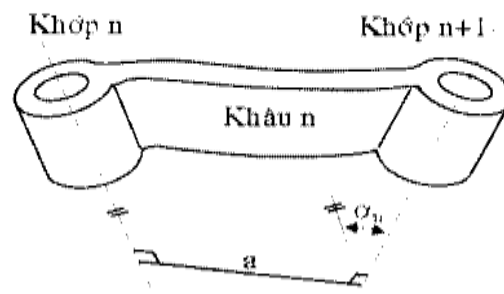
Vector điểm p có kích thước 3x1, biểu diễn mối quan hệ tọa độ vị trí của của gốc hệ tọa độ gắn trên khâu chấp hành cuối đối với hệ tọa độ cơ bản.

3.2. Bộ thông số Denavit-Hartenberg (DH) :

Một robot nhiều khâu cấu thành từ các khâu nối tiếp nhau thông qua các khớp động. Gốc chuẩn (Base) của một robot là khâu số 0 và không tính vào số các khâu. Khâu 1 nối với khâu chuẩn bởi khớp 1 và không có khớp ở đầu mút của khâu cuối cùng. Bất kỳ khâu nào cũng được đặc trưng bởi hai kích thước :

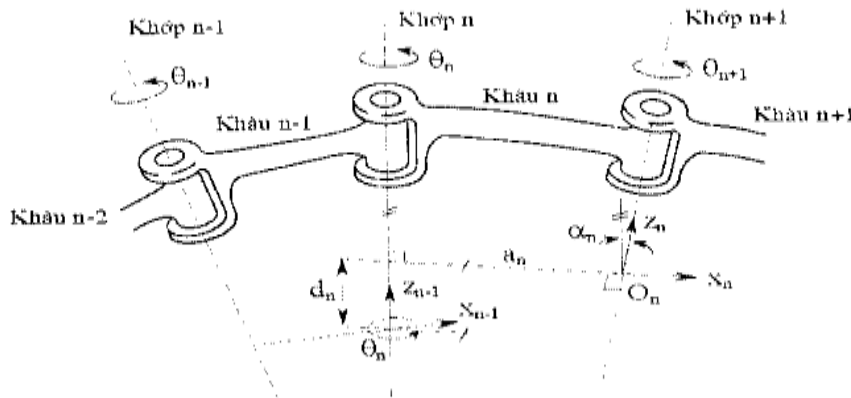
Độ dài pháp tuyến chung : a_n .

Góc giữa các trục trong mặt phẳng vuông góc với a_n : α_n .



Hình 3.5 : Chiều dài và góc xoắn của 1 khâu.

Thông thường, người ta gọi a_n là chiều dài và α_n là góc xoắn của khâu (Hình 3.5). Phổ biến là hai khâu liên kết với nhau ở chính trục của khớp (Hình 3.6).



Hình 3.6 : Các thông số của khâu : θ , d , a và α .

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 θ_n là góc giữa các pháp tuyến đo trong mặt phẳng vuông góc với trục.

d_n và θ_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 :

+ Góc 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 $n+1$.

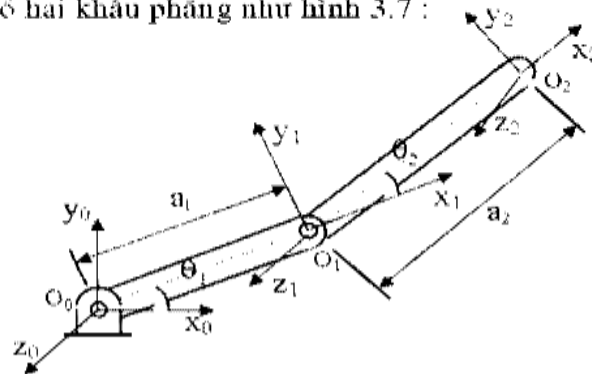
+ Trục x thường được đặt dọc theo pháp tuyến chung và hướng

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 θ_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 , α_n , d_n và θ_n được gọi là bộ thông số DH.

Ví dụ 1 : Xét một tay máy có hai khâu phẳng như hình 3.7 :



Hình 3.7 : Tay máy có hai khâu phẳng (vị trí bất kỳ).

Ta gắn các hệ tọa độ lên các khâu như hình vẽ : trục z_0, z_1 và z_2 vuông góc với tờ giấy. Hệ tọa độ cơ sở là $O_0x_0y_0z_0$, chiều của x_0 hướng từ O_0 đến O_1 . Sau khi thiết lập hệ tọa độ cơ sở, Hệ tọa độ $o_1x_1y_1z_1$ có hướng như hình vẽ, O_1 đặt tại tâm trục khớp 2. Hệ tọa độ $O_2x_2y_2z_2$ có gốc O_2 đặt ở điểm cuối của khâu 2.

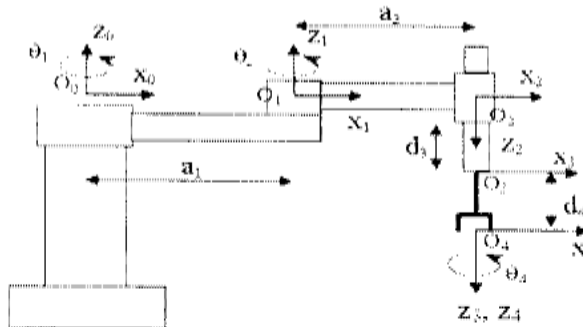
Bảng thông số Denavit-Hartenbert của tay máy này như sau :

Khâu	θ_i	α_i	a_i	d_i
1	θ_1^*	0	a_1	0
2	θ_2^*	0	a_2	0

Trong đó θ_i là các biến khớp (dùng dấu * để ký hiệu các biến khớp).

Ví dụ 2 : Xem sơ đồ robot SCARA có 4 khâu như hình 3.8 :

Đây là robot có cấu hình kiểu RRTR, bàn tay có chuyển động xoay xung quanh trục đứng. Hệ tọa độ gắn lên các khâu như hình vẽ.



Hình 3.8 : Robot SCARA và các hệ tọa độ (vị trí ban đầu).

Đối với tay máy này các trục khớp đều song song nhau, để tiện lợi tất cả các gốc tọa độ đặt tại tâm các trục khớp. Trục x_0 nằm trong mặt phẳng tờ giấy. Các hệ tọa độ khác như hình vẽ. Bảng thông số DH của robot SCARA như sau:

Khâu	θ_i	α_i	a_i	d_i
1	θ_1^*	0	a_1	0
2	θ_2^*	180°	a_2	0
3	0	0	0	d_3^*
4	θ_4^*	0	0	d_4

*: là các biến khớp

3.3. Đặc trưng của các ma trận A :

Trên cơ sở các hệ tọa độ đã ấn định cho tất cả các khâu liên kết của robot, ta có thể thiết lập mối quan hệ giữa các hệ tọa độ nối tiếp nhau (n-1), (n) bởi các phép quay và tịnh tiến sau đây :

- Quay quanh z_{n-1} một góc θ_n
- Tịnh tiến dọc theo z_{n-1} một khoảng d_n
- Tịnh tiến dọc theo $x_{n-1} = x_n$ một đoạn a_n
- Quay quanh x_n một góc xoắn α_n

Bốn phép biến đổi thuận nhất này thể hiện quan hệ của hệ tọa độ thuộc khâu thứ n so với hệ tọa độ thuộc khâu thứ n-1 và tích của chúng được gọi là ma trận A :

$$A_n = \text{Rot}(z, \theta) \text{Trans}(0, 0, d) \text{Trans}(a, 0, 0) \text{Rot}(x, \alpha) \quad (3.4)$$

$$A_n = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_n = \begin{pmatrix} \cos\theta & -\sin\theta \cos\alpha & \sin\theta \sin\alpha & a \cos\theta \\ \sin\theta & \cos\theta \cos\alpha & -\cos\theta \sin\alpha & a \sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

Đối với khớp tịnh tiến ($a = 0$ và $\theta_i = 0$) thì ma trận A có dạng :

$$A_n = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.6)$$

Đối với một khâu đi theo một khớp quay thì d , a và α là hằng số. Như vậy ma trận A của khớp quay là một hàm số của biến khớp θ .

Đối với một khâu đi theo một khớp tịnh tiến thì θ , α là hằng số. Ma trận A của khớp tịnh tiến là một hàm số của biến số d .

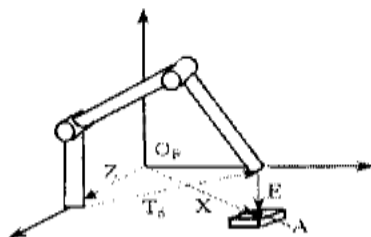
Nếu các biến số được xác định thì giá trị của các ma trận A theo đó cũng được xác định.

3.4. Xác định T_6 theo các ma trận A_n :

Ta đã biết : $T_6 = A_1 A_2 A_3 A_4 A_5 A_6$

Trong đó T_6 được miêu tả trong hệ tọa độ gốc (hệ tọa độ gắn với khâu cơ bản cố định của robot). Nếu mô tả T_6 theo các hệ tọa độ trung gian thứ $n-1$ thì :

$${}^{n-1}T_6 = \prod_{i=n}^6 A_i$$

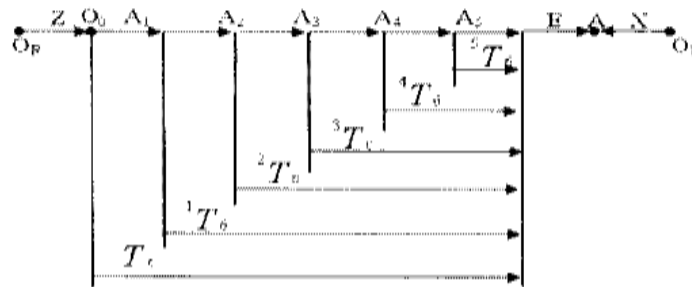


Hình 3.9 : Vật thể và Robot

Trong trường hợp tổng quát, khi xét quan hệ của robot với các thiết bị khác, nếu hệ tọa độ cơ bản của robot có liên hệ với một hệ tọa độ nào đó bởi phép biến đổi Z, Khâu chấp hành cuối lại có gắn một công cụ, có quan hệ với vật thể bởi phép biến đổi E (hình 3.9) thì vị trí và hướng của điểm cuối của công cụ, khảo sát ở hệ tọa độ tham chiếu mô tả bởi X sẽ được xác định bởi :

$$X = Z T_6 E$$

Quan hệ này được thể hiện trên toán đồ sau :



Hình 3.10 : Toán đồ chuyển vị của robot.

Từ toán đồ này ta có thể rút ra : $T_6 = Z^{-1} X E^{-1}$

(Z^{-1} và E^{-1} là các ma trận nghịch đảo).

3.5. Trình tự thiết lập hệ phương trình động học của robot :

Để thiết lập hệ phương trình động học của robot, ta tiến hành theo các bước sau :

1. 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. Nguyên tắc gắn hệ tọa độ lên các khâu đã được trình bày một cách tổng quát trong phần 3.5. Trong thực tế, các trục khớp của robot thường song song hoặc vuông góc với nhau, đồng thời thông qua các phép biến đổi của ma trận A ta có thể xác định các hệ tọa độ gắn trên các khâu của robot theo trình tự sau :

- + Giả định một vị trí ban đầu^(*) (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 α_{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.

2. Lập bảng thông số DH (Denavit Hartenberg).

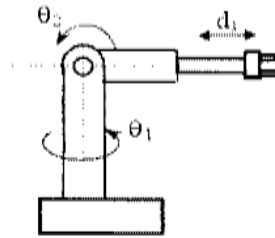
3. Dựa vào các thông số DH xác định các ma trận A_n .

4. Tính các ma trận T và viết các phương trình động học của robot.

(*) Vị trí ban đầu là vị trí mà các biến nhận giá trị ban đầu, thường bằng 0.

Ví dụ sau đây trình bày chi tiết của các bước khi thiết lập hệ phương trình động học của robot :

Cho một robot có ba khâu, cấu hình RRT như hình 3.11. Hãy thiết lập hệ phương trình động học của robot.

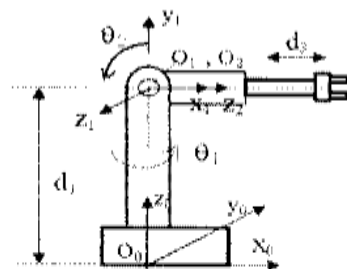


Hình 3.11 : Robot RRT

1. Gắn hệ tọa độ lên các khâu :

Ta giả định vị trí ban đầu và chọn gốc tọa độ O_0 của robot như hình 3.12. Các trục z đặt cùng phương với các trục khớp.

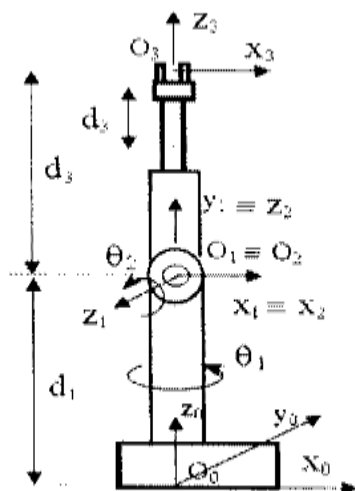
Ta thấy trục z_1 đã quay tương đối một góc 90° so với trục z_0 , đây chính là phép quay quanh trục x_0 một góc α_1 (phép biến đổi $Rot(x_0, \alpha_1)$ trong biểu thức tính A_n). Nghĩa là trục x_0 vuông góc với z_0 và z_1 . Ta chọn chiều của x_0 từ trái sang phải thì góc quay $\alpha_1 = 90^\circ$ (chiều dương ngược chiều kim đồng hồ). Đồng thời ta cũng thấy gốc O_1 đã tịnh tiến một đoạn dọc theo z_0 , so với O_0 , đó chính là phép biến đổi $Trans(0, 0, d_1)$ (tịnh tiến dọc theo z_0 một đoạn d_1); các trục y_0 , và y_1 xác định theo quy tắc bàn tay phải (Hình 3.12) .



Hình 3.12 : Gắn các hệ tọa độ O_0 và O_1

Tiếp tục chọn gốc tọa độ O_2 đặt trùng với O_1 vì trục khớp thứ ba và trục khớp thứ hai cắt nhau tại O_1 (như hình 3.12). Trục z_2 cùng phương với trục khớp thứ ba, tức là đã quay đi một góc 90° so với z_1 quanh trục y_1 ; phép biến đổi này không có trong biểu thức tính A_n nên không dùng được, ta cần chọn lại vị trí ban đầu của robot (thay đổi vị trí của khâu thứ 3) như hình

3.13.



Hình 3.13 : Hệ tọa độ gắn lên các khâu

Theo hình 3.13, O_2 vẫn được đặt trùng với O_1 , trục z_2 có phương thẳng đứng, nghĩa là ta đã quay trục z_1 thành z_2 quanh trục x_1 một góc -90° (tức $\alpha_2 = -90^\circ$).

Đầu cuối của khâu thứ 3 không có khớp, ta đặt O_3 tại điểm giữa của các ngón tay, và trục z_3, x_3 chọn như hình vẽ, như vậy ta đã tịnh tiến gốc tọa độ dọc theo z_2 một đoạn d_3 (Phép biến đổi $\text{Trans}(0,0,d_3)$), vì đây là khâu tịnh tiến nên d_3 là biến .

Như vậy việc gắn các hệ tọa độ lên các khâu của robot đã hoàn thành. Thông qua các phân tích trên đây, ta có thể xác định được các thông số DH của robot.

2. Lập bảng thông số DH :

Khâu	θ_i	α_i	a_i	d_i
1	θ_1^*	90	0	d_1
2	θ_2^*	-90	0	0
3	0	0	0	d_3^*

3. Xác định các ma trận A :

Ma trận An có dạng :

$$A_n = \begin{pmatrix} \cos\theta & -\sin\theta \cos\alpha & \sin\theta \sin\alpha & 0 \\ \sin\theta & \cos\theta \cos\alpha & -\cos\theta \sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Với qui ước viết tắt : $C_1 = \cos\theta_1$; $S_1 = \sin\theta_1$; $C_2 = \cos\theta_2 \dots$

$$A_1 = \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4. Tính các ma trận biến đổi thuần nhất T :

+ Ma trận ${}^2T_3 = A_3$

+ Ma trận ${}^1T_3 = A_2 \cdot {}^2T_3$

$${}^1T_3 = \begin{pmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_2 & 0 & -S_2 & -S_2 \cdot d_3 \\ S_2 & 0 & C_2 & C_2 \cdot d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

+ Ma trận $T_3 = A_1 \cdot {}^1T_3$

$$T_3 = \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_2 & 0 & -S_2 & -S_2 \cdot d_3 \\ S_2 & 0 & C_2 & C_2 \cdot d_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} C_1C_2 & -S_1 & -C_1S_2 & -C_1S_2d_3 \\ S_1d_2 & C_1 & -S_1S_2 & -S_1S_2d_3 \\ S_2 & 0 & C_2 & C_2d_3 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

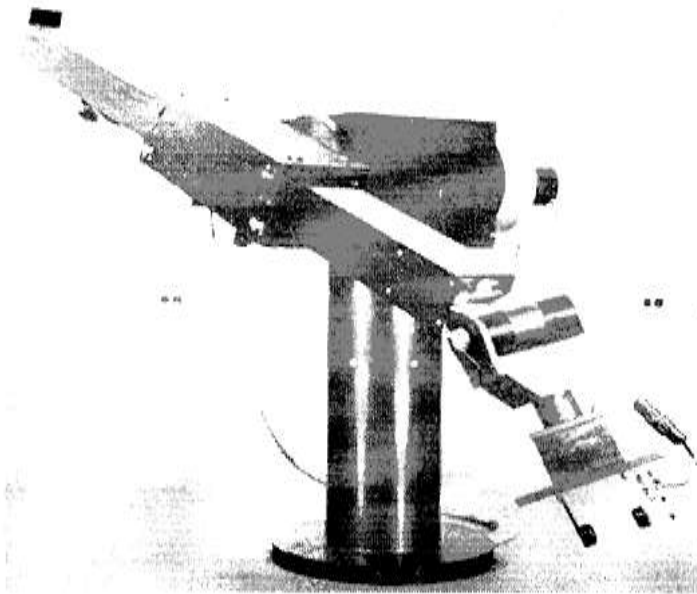
Ta có hệ phương trình động học của robot như sau :

$$\begin{array}{lll} n_x = C_1C_2; & n_y = S_1C_2; & n_z = S_2 \\ O_x = -S_1; & O_y = C_1; & O_z = 0; \\ a_x = -C_1S_2; & a_y = -S_1S_2; & a_z = C_2; \\ p_x = -C_1S_2d_3 & p_y = -S_1S_2d_3 & p_z = C_2d_3 + d_1; \end{array}$$

(Ta có thể kiểm tra kết quả tính toán bằng cách dựa vào tọa độ vị trí p_x, p_y, p_z , đã tính so với cách tính hình học trên hình vẽ).

3.9. Hệ phương trình động học của robot STANFORD :

Stanford là một robot có 6 khâu với cấu hình RRT.RRR (Khâu thứ 3 chuyển động tịnh tiến, năm khâu còn lại chuyển động quay). Kết cấu của robot Stanford như hình 3.14 :



Hình 3.14 Robot Stanford

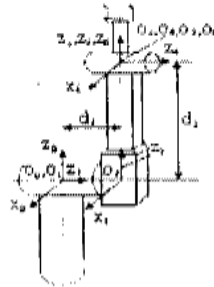
Trên hình 3.15 trình bày mô hình của robot Stanford với việc gắn các hệ tọa độ lên từng khâu. Để đơn giản trong khi viết các phương trình động học của robot, ta qui ước cách viết tất các hàm lượng giác như sau :

$$\begin{array}{l} C_1 = \cos\theta_1; \\ S_1 = \sin\theta_1; \end{array}$$

$$C_{12} = \cos(\theta_1 + \theta_2);$$

$$S_{12} = \sin(\theta_1 + \theta_2)$$

$$S_{234} = \sin(\theta_2 + \theta_3 + \theta_4) \dots$$



Hình 3.13: Hệ tọa độ của Robot Stanford

Hệ tọa độ gắn lên các khâu của robot như hình 3.15. (Khâu cuối có chiều dài và khoảng cách bằng không, để có thể gắn các loại công cụ khác nhau nên chọn $O_6 \equiv O_5$).

Bảng thông số DH (Denavit-Hartenberg) của robot Stanford như sau :

Khâu	θ_i	α_i	a_i	d_i
1	θ_1^*	-90°	0	0
2	θ_2^*	90°	0	d_2
3	0	0	0	d_3^*
4	θ_4^*	-90°	0	0
5	θ_5^*	90°	0	0
6	θ_6^*	0	0	0

(* : Các biến khớp).

Các ma trận A của robot Stanford được xác định như sau :

$$A_1 = \begin{pmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2 = \begin{pmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_5 = \begin{pmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_6 = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tích của các ma trận chuyển vị A đối với robot Stanford được bắt đầu ở khâu 6 và chuyển dần về gốc; theo thứ tự này ta có :

$$T_6^5 = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^4 = A_5 A_6 = \begin{pmatrix} C_5 C_6 & -C_5 S_6 & S_5 & 0 \\ S_5 C_6 & -S_5 S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^3 = A_4 A_5 A_6 = \begin{pmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & 0 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & 0 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^2 = A_3 A_4 A_5 A_6 = \begin{pmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & 0 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & 0 \\ -S_5 C_6 & S_5 S_6 & C_5 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^1 = A_2 A_3 A_4 A_5 A_6 = \begin{pmatrix} C_2(C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6 & -C_2(C_4 C_5 S_6 - S_4 C_6) + S_2 S_5 S_6 \\ S_2(C_4 C_5 C_6 - S_4 S_6) + C_2 S_5 C_6 & -S_2(C_4 C_5 S_6 + S_4 C_6) - C_2 S_5 S_6 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 \\ 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} C_2 C_4 S_5 + S_2 C_5 & S_2 d_3 \\ S_2 C_4 S_5 - C_2 C_5 & -C_2 d_3 \\ S_4 S_5 & d_2 \\ 0 & 1 \end{pmatrix}$$

Cuối cùng :

$$T_6 = \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 T_6^1$$

Để tính T_6 , ta phải nhân A_1 với T_6^1 sau đó cân bằng các phần tử của ma trận T_6 ở hai vế ta được một hệ thống các phương trình sau :

$$n_x = C_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] - S_1 (S_4 C_5 C_6 + C_4 S_6)$$

$$n_y = S_1 [C_2 (C_4 C_5 C_6 - S_4 S_6) - S_2 S_5 C_6] + C_1 (S_4 C_5 C_6 + C_4 S_6)$$

$$n_z = -S_2 (C_4 C_5 C_6 - S_4 S_6) + C_2 S_5 C_6$$

$$O_x = C_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] - S_1 (-S_4 C_5 S_6 + C_4 C_6)$$

$$O_y = S_1 [-C_2 (C_4 C_5 S_6 + S_4 C_6) + S_2 S_5 S_6] + C_1 (-S_4 C_5 S_6 + C_4 C_6)$$

$$O_z = S_2 (C_4 C_5 S_6 + S_4 C_6) + C_2 S_5 S_6$$

$$a_x = C_1 (C_2 C_4 S_5 + S_2 C_5) - S_1 S_4 S_5$$

$$a_y = S_1 (C_2 C_4 S_5 + S_2 C_5) + C_1 S_4 S_5$$

$$a_z = -S_2 C_4 S_5 + C_2 C_5$$

$$p_x = C_1 S_2 d_3 - S_1 d_2$$

$$p_y = S_1 S_2 d_3 + C_1 d_2$$

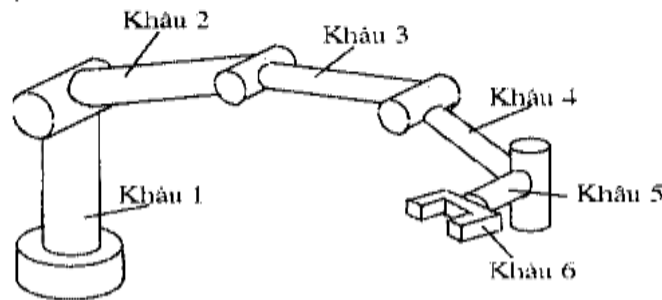
$$p_z = C_2 d_3$$

Nếu ta biết được các giá trị của biến khớp, thì vị trí và hướng của bụn tay robot sẽ tìm được bằng cách xác định các giá trị các phần tử của T_6 theo các phương trình trên.

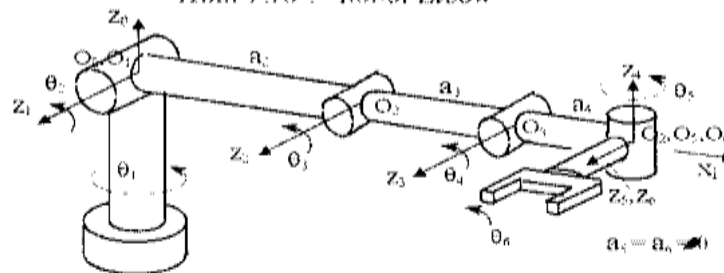
Các phương trình trên gọi là hệ phương trình động học thuận của robot Stanford.

3.10. Hệ phương trình động học của robot ELBOW :

Để hiểu rõ hơn về cách thiết lập hệ phương trình động học của robot, ta xét thêm trường hợp robot Elbow.



Hình 1.16 : Robot Elbow



Hình 1.17 : Vị trí ban đầu của robot Elbow và các hệ tọa độ

Bộ thông số DH của robot Elbow

Khâu	θ_i^*	α_i	a_i	d_i
1	θ_1	90°	0	0
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	-90°	a_4	0
5	θ_5	90°	0	0
6	θ_6	0	0	0

(* : các biến khớp)

Các ma trận A của robot Elbow được xác định như sau :

$$\begin{aligned}
 A_1 &= \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_2 &= \begin{pmatrix} C_2 & -S_2 & 0 & C_2 a_2 \\ S_2 & C_2 & 0 & S_2 a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_3 &= \begin{pmatrix} C_3 & -S_3 & 0 & C_3 a_3 \\ S_3 & C_3 & 0 & S_3 a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_4 &= \begin{pmatrix} C_4 & 0 & -S_4 & C_4 a_4 \\ S_4 & 0 & C_4 & S_4 a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_5 &= \begin{pmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & A_6 &= \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Ta xác định các ma trận T theo các hệ tọa độ lần lượt từ khâu cuối trở về gốc :

$$T_6^5 = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^4 = A_5 A_6 = \begin{pmatrix} C_5 C_6 & -C_5 S_6 & S_5 & 0 \\ S_5 C_6 & -S_5 S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^3 = A_4 A_5 A_6 = \begin{pmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & C_4 a_4 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & S_4 a_4 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_6^2 = A_3 A_4 A_5 A_6 = \begin{pmatrix} C_3 C_4 C_5 C_6 - S_3 S_4 S_6 & -C_3 C_4 C_5 S_6 - S_3 S_4 C_6 & C_3 C_4 S_5 & C_3 C_4 a_4 + C_3 a_3 \\ S_3 C_4 C_5 C_6 + C_3 S_4 S_6 & -S_3 C_4 C_5 S_6 + C_3 S_4 C_6 & S_3 C_4 S_5 & S_3 C_4 a_4 + S_3 a_3 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned}
 T_6^1 &= A_2 A_3 A_4 A_5 A_6 = \\
 &\begin{pmatrix} C_{234} C_5 C_6 - S_{234} S_6 & -C_{234} C_5 S_6 - S_{234} C_6 & C_{234} S_5 & C_{234} a_4 + C_{23} a_3 + C_2 a_2 \\ S_{234} C_5 C_6 + C_{234} S_6 & -S_{234} C_5 S_6 + C_{234} C_6 & S_{234} S_5 & S_{234} a_4 + S_{23} a_3 + S_2 a_2 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Cuối cùng :

$$T_6 = \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 T_6^1$$

Để tính T_6 , ta phải nhân A_1 với T_6^1 sau đó cân bằng các phần tử của ma trận T_6 ta được một hệ thống các phương trình sau :

$$\begin{aligned}
 n_x &= C_1 (C_{234} C_5 C_6 - S_{234} S_6) - S_1 S_5 C_6 \\
 n_y &= S_1 (C_{234} C_5 C_6 - S_{234} S_6) + C_1 S_5 C_6
 \end{aligned}$$

$$\begin{aligned}
 n_z &= S_{234} C_5 C_6 + C_{234} S_6 \\
 O_x &= -C_1 (C_{234} C_5 S_6 + S_{234} C_6) + S_1 S_5 S_6 \\
 O_y &= -S_1 (C_{234} C_5 S_6 + S_{234} C_6) - C_1 S_5 S_6 \\
 O_z &= -S_{234} C_5 S_6 + C_{234} C_6 \\
 a_x &= C_1 C_{234} S_5 + S_1 C_5 \\
 a_y &= S_1 C_{234} S_5 - C_1 C_5 \\
 a_z &= S_{234} S_5 \\
 p_x &= C_1 (C_{234} a_4 + C_{23} a_3 + C_2 a_2) \\
 p_y &= S_1 (C_{234} a_4 + C_{23} a_3 + C_2 a_2) \\
 p_z &= S_{234} a_4 + S_{23} a_3 + S_2 a_2
 \end{aligned}$$

Cột đầu tiên của ma trận T_6 có thể được xác định bởi tích vector : $\vec{h} = \vec{O} \times \vec{a}$.

3.11. Kết luận :

Trong chương này chúng ta đã nghiên cứu việc dùng các phép biến đổi thuận nhất để mô tả vị trí và hướng của khâu chấp hành cuối của robot thông qua việc xác lập các hệ tọa độ gắn lên các khâu và các thông số DH. Phương pháp này có thể dùng cho bất cứ robot nào với số khâu (khớp) tùy ý. Trong quá trình xác lập các hệ tọa độ mở rộng ta cũng xác định được vị trí dùng của mỗi robot. Tuy thuộc kết cấu của robot cũng như công cụ gắn lên khâu chấp hành cuối mà ta có thể đưa các thông số của khâu chấp hành cuối vào phương trình động học hay không. Việc tính toán các ma trận T để thiết lập hệ phương trình động học của robot thường tốn nhiều thời gian và dễ nhầm lẫn, ta có thể lập trình trên máy tính để tính toán (ở dạng ký hiệu) nhằm nhanh chóng xác định các ma trận A_n và thiết lập hệ phương trình động học của robot .

Thiết lập hệ phương trình động học của robot là bước rất quan trọng để có thể dựa vào đó lập trình điều khiển robot. **Bài toán này thường được gọi là bài toán động học thuận robot.** Việc giải hệ phương trình động học của robot được gọi là bài toán động học ngược, nhằm xác định giá trị của các biến khớp theo các thông số đã biết của khâu chấp hành cuối; vấn đề này ta sẽ nghiên cứu trong chương tiếp theo.

Bài tập chương III :

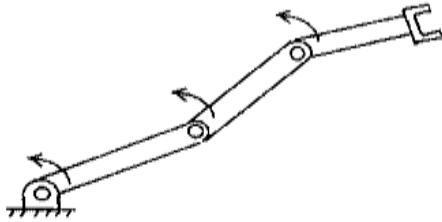
Bài 1 : Cho ma trận :

$$T_6 = \begin{pmatrix} ? & 0 & -1 & 0 \\ ? & 0 & 0 & 1 \\ ? & -1 & 0 & 2 \\ ? & 0 & 0 & 1 \end{pmatrix}$$

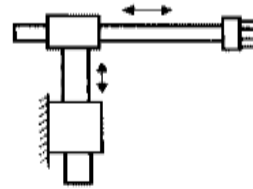
là ma trận biểu diễn hướng và vị trí của khâu chấp hành cuối. Tìm các phần tử được đánh dấu ?

Bài 2 : Cho một robot có 3 khâu phẳng như hình 3.18, cấu hình RRR. Thiết lập hệ phương trình động học của robot.

Bài 3 : Cho một robot có 2 khâu tịnh tiến như hình 3.19, cấu hình TT. Thiết lập hệ phương trình động học của robot.



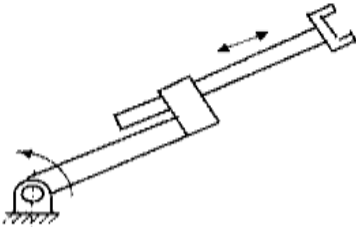
Hình 3.18 : Robot cấu hình RRR



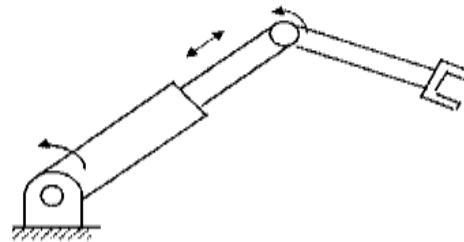
Hình 3.19 : Robot cấu hình TT

Bài 4 : Cho một robot có 2 khâu phẳng như hình 3.20, cấu hình RT. Thiết lập hệ phương trình động học của robot.

Bài 5 : Cho một robot có 3 khâu như hình 3.21, cấu hình RTR. Thiết lập hệ phương trình động học của robot.

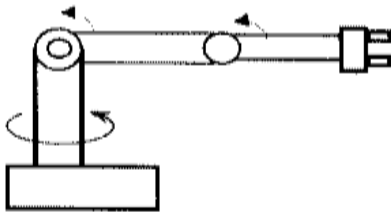


Hình 3.20 : Robot cấu hình RT

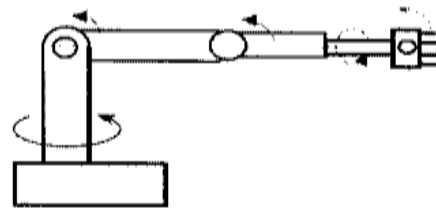


Hình 3.21 : Robot cấu hình RTR

Bài 6 : Cho một robot có 3 khâu như hình 3.22, cấu hình RRR. Thiết lập hệ phương trình động học của robot.



Hình 3.22 : Robot cấu hình RRR



Hình 3.23 : Robot cấu hình RRRRR

Bài 7 : Cho một robot có 5 khâu như hình 3.23, cấu hình RRRRR. Thiết lập hệ phương trình động học của robot.

CHƯƠNG IV

GIẢI PHƯƠNG TRÌNH ĐỘNG HỌC ROBOT HAY PHƯƠNG TRÌNH ĐỘNG HỌC NGƯỢC (Invers Kinematic Equations)

Trong phần trước, ta đã nghiên cứu việc thiết lập hệ phương trình động học của robot thông qua ma trận T_6 bằng phương pháp gắn các hệ tọa độ lên các khâu và xác định các thông số DH. Ta cũng đã xét tới các phương pháp khác nhau để mô tả hướng của khâu chấp hành cuối như các phép quay Euler, phép quay Roll-Pitch và Yaw .v.v... Trong chương này chúng ta sẽ tiến hành giải hệ phương trình động học đã thiết lập ở chương trước nhằm xác định các biến trong bộ thông số Denavit - Hartenberg khi đã biết ma trận vector cuối T_6 . Kết quả của việc giải hệ phương trình động học đóng vai trò hết sức quan trọng trong việc điều khiển robot. Thông thường, điều ta biết là các vị trí và hướng mà ta muốn robot phải dịch chuyển tới và điều ta cần biết là mối quan hệ giữa các hệ tọa độ trung gian để phối hợp tạo ra chuyển động của robot, hay nói cách khác đó chính là giá trị của các biến khớp ứng với mỗi tọa độ và hướng của khâu chấp hành cuối hoặc công cụ gắn lên khâu chấp hành cuối, muốn vậy ta phải giải hệ phương trình động học của robot. Việc nhận được lời giải của bài toán động học ngược là vấn đề khó mà ta sẽ nghiên cứu trong chương này. Nhiệm vụ của bài toán là xác định tập nghiệm $(\theta_1, \theta_2, \dots, \theta_6, d_i^*)$ khi đã biết hình thể của robot thông qua vector cuối T_6 (khái niệm “hình thể” của robot bao gồm khái niệm về vị trí và hướng của khâu chấp hành cuối : Configuration = Position + Orientation).

Cũng cần lưu ý rằng, đa số các robot có bộ Teach pendant là thiết bị dạy học, có nhiệm vụ điều khiển robot đến các vị trí mong muốn trong động trình đầu tiên (điều khiển điểm: Point to point), các chuyển động này sẽ được ghi lại vào bộ nhớ trung tâm (CPU) của robot hoặc máy tính

điều khiển robot, sau đó robot có thể thực hiện lại đúng các động tác đã được học. Trong quá trình hoạt động của robot, nếu dạng quỹ đạo đường đi không quan trọng thì không cần lời giải của bài toán động học ngược.

4.1. Các điều kiện của bài toán động học ngược :

Việc giải bài toán động học ngược của robot cần thoả mãn các điều kiện sau :

4.1.1. Điều kiện tồn tại nghiệm :

Điều kiện này nhằm khẳng định: Có ít nhất một tập nghiệm ($\theta_1, \theta_2, \dots, \theta_6, d_i^*$) sao cho robot có hình thể cho trước. (“Hình thể” là khái niệm mô tả tương minh của vectơ cuối T6 cả về vị trí và hướng).

4.1.2. Điều kiện duy nhất của tập nghiệm :

Trong khi xác định các tập nghiệm cần phân biệt rõ hai loại nghiệm :

+ Nghiệm toán (Mathematical Solution): Các nghiệm này thoả mãn các phương trình cho trước của T6.

+ Nghiệm vật lý (Physical Solution) : là các tập con của nghiệm toán, phụ thuộc vào các giới hạn vật lý (giới hạn về góc quay, kích thước ...) nhằm xác định tập nghiệm duy nhất.

Việc giải hệ phương trình động học có thể được tiến hành theo hai phương pháp cơ bản sau :

+ Phương pháp giải tích (Analytical Method) : tìm ra các công thức hay các phương trình toán giải tích biểu thị quan hệ giữa các giá trị của không gian biến trục và các thông số khác của bộ thông số DH.

+ Phương pháp số (Numerical Method) : Tìm ra các giá trị của tập nghiệm bằng kết quả của một quá trình lặp.

4.2. Lời giải của phép biến đổi Euler :

Trong chương 3 ta đã nghiên cứu về phép biến đổi Euler để mô tả hướng của khâu chấp hành cuối :

$$\text{Euler}(\Phi, \theta, \psi) = \text{Rot}(z, \Phi) \text{Rot}(y, \theta) \text{Rot}(z, \psi)$$

Tập nghiệm muốn tìm là các góc Φ, θ, ψ khi đã biết ma trận biến đổi đồng nhất T6 (còn gọi là ma trận vectơ cuối), Nếu ta có các giá trị số

của các phần tử trong ma trận T_6 thì có thể xác định được các góc Euler Φ, θ, ψ thích hợp.

Như vậy ta có :

$$\text{Euler}(\Phi, \theta, \psi) = T_6 \tag{4-1}$$

Vế trái của phương trình (4-1) đã được biểu diễn bằng công thức (3-4) , nên ta có:

$$\begin{pmatrix} \cos\Phi\cos\theta\cos\psi - \sin\Phi\sin\psi & -\cos\Phi\cos\theta\sin\psi - \sin\Phi\cos\psi & \cos\Phi\sin\theta & 0 \\ \sin\Phi\cos\theta\cos\psi + \cos\Phi\sin\psi & -\sin\Phi\cos\theta\sin\psi + \cos\Phi\cos\psi & \sin\Phi\sin\theta & 0 \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4-2}$$

Lần lượt cho cân bằng các phần tử tương ứng của hai ma trận trong phương trình (4-2) ta có các phương trình sau :

$$n_x = \cos\Phi\cos\theta\cos\psi - \sin\Phi\sin\psi \tag{4.3}$$

$$n_y = \sin\Phi\cos\theta\cos\psi + \cos\Phi\sin\psi \tag{4-4}$$

$$n_z = -\sin\theta\cos\psi \tag{4-5}$$

$$O_x = -\cos\Phi\cos\theta\sin\psi - \sin\Phi\cos\psi \tag{4-6}$$

$$O_y = -\sin\Phi\cos\theta\sin\psi + \cos\Phi\cos\psi \tag{4-7}$$

$$O_z = \sin\theta\sin\psi \tag{4-8}$$

$$a_x = \cos\Phi\sin\theta \tag{4-9}$$

$$a_y = \sin\Phi\sin\theta \tag{4-10}$$

$$a_z = \cos\theta \tag{4-11}$$

Ta thử giải hệ phương trình này để tìm Φ, θ, ψ như sau :

Từ (4-11) ta có $\theta = \cos^{-1}(a_z)$ (4-12)

Từ (4-9) ta có $\Phi = \cos^{-1}(a_x / \sin\theta)$ (4-13)

Từ (4-5) và (4-12) ta có $\psi = \cos^{-1}(-n_z / \sin\theta)$ (4-14)

Trong đó ta đã dùng ký hiệu \cos^{-1} thay cho hàm arccos.

Nhưng các kết quả đã giải ở trên chưa dùng được vì các lý do dưới đây :

+ Hàm arccos không chỉ biểu hiện cho một góc chưa xác định mà về độ chính xác nó lại phụ thuộc vào chính góc đó, nghĩa là :

$\cos\theta = \cos(-\theta)$: θ chưa được xác định duy nhất.

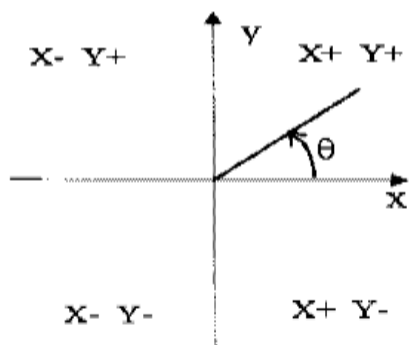
$$\frac{d(\rho \cos\theta)}{d\theta} = 0 : \theta \text{ xác định không chính xác.}$$

$d\theta \in (0, 180)$

+ Trong lời giải đối với Φ và ψ một lần nữa chúng ta lại dùng hàm arccos và chia cho $\sin\theta$, điều này dẫn tới sự mất chính xác khi θ có giá trị lân cận 0.

+ Các phương trình (4-13) và (4-14) không xác định khi $\theta = 0$ hoặc $\theta = \pm 180^\circ$.

Do vậy chúng ta cần phải cẩn thận hơn khi chọn lời giải. Để xác định các góc khi giải bài toán ngược của robot ta phải dùng hàm arctg2 (y,x) (hàm arctang hai biến). Hàm arctg2 nhằm mục đích xác định được góc thực – duy nhất khi xét dấu của hai biến y và x. Hàm số trả về giá trị góc trong khoảng $-\pi \leq \theta < \pi$.



Hình 4.1 : Hàm $arctg2(y,x)$

Ví dụ :

$$arctg2(-1/-1) = -135^\circ,$$

trong khi $arctg2(1/1) = 45^\circ$

Hàm này xác định ngay cả khi x hoặc y bằng 0 và cho kết quả đúng. (Trong một số ngôn ngữ lập trình như Matlab, turbo C++, Maple hàm $arctg2(y,x)$ đã có sẵn trong thư viện).

Để có thể nhận được những kết quả chính xác của bài toán Euler,

ta thực hiện thủ thuật toán học sau : Nhân T6 với ma trận quay nghịch đảo $Rot(z, \Phi)^{-1}$, ta có:

$$Rot(z, \Phi)^{-1} T6 = Rot(y, \theta) Rot(z, \psi) \tag{4-15}$$

Vế trái của phương trình (4-15) là một hàm số của ma trận T và góc quay Φ . Ta thực hiện phép nhân ma trận ở vế phải của (4-15), tìm ra các phần tử của ma trận có giá trị bằng 0 hoặc bằng hằng số, cho các phần tử này cân bằng với những phần tử tương ứng của ma trận ở vế trái, cụ thể từ (4-15) ta có :

$$\begin{pmatrix} \cos\Phi & \sin\Phi & 0 & 0 \\ -\sin\Phi & \cos\Phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \text{Cos}\theta\text{cos}\psi & -\text{Cos}\theta \sin\psi & \sin\theta & 0 \\ \sin\psi & \text{cos}\psi & 0 & 0 \\ -\sin\theta \text{cos}\psi & \sin\theta \sin\psi & \text{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4-16}$$

Tích hai ma trận ở vế trái của phương trình (4-16) là một ma trận mà có thể được viết gọn lại bằng các ký hiệu sau :

$$\begin{pmatrix} f_{11}(n) & f_{11}(O) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(O) & f_{12}(a) & f_{12}(p) \\ f_{13}(n) & f_{13}(O) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Trong đó :

$$f_{11} = \cos\Phi x + \sin\Phi y \tag{4-17}$$

$$f_{12} = -\sin\Phi x + \cos\Phi y \tag{4-18}$$

$$f_{13} = z \tag{4-19}$$

và x, y, z là các phần tử của vectơ xác định bởi các dữ kiện f_{11} , f_{12} , f_{13} , ví dụ :

$$f_{11}(n) = \cos\Phi n_x + \sin\Phi n_y$$

$$f_{12}(O) = -\sin\Phi O_x + \cos\Phi O_y$$

$$f_{13}(a) = a_z$$

Như vậy phương trình (4-16) có thể được viết thành :

$$\begin{pmatrix} f_{11}(n) & f_{11}(O) & f_{11}(a) & 0 \\ f_{12}(n) & f_{12}(O) & f_{12}(a) & 0 \\ f_{13}(n) & f_{13}(O) & f_{13}(a) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \text{Cos}\theta\text{cos}\psi & -\text{Cos}\theta \sin\psi & \sin\theta & 0 \\ \sin\psi & \text{cos}\psi & 0 & 0 \\ -\sin\theta \text{cos}\psi & \sin\theta \sin\psi & \text{Cos}\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Trong đó f_{11} , f_{12} , f_{13} đã được định nghĩa ở (4-17), (4-18) và (4-19).

Khi tính toán về trái, ta chú ý rằng p_x , p_y , p_z bằng 0 vì phép biến đổi Euler chỉ toàn phép quay không chứa một phép biến đổi tịnh tiến nào, nên $f_{11}(p) = f_{12}(p) = f_{13}(p) = 0$. Từ phương trình (4-20), cho cân bằng phần tử ở hàng 2 cột 3 ta có :

$$f_{12}(a) = -\sin\Phi a_x + \cos\Phi a_y = 0.$$

Cộng hai vế với $\sin\Phi a_x$ và chia cho $\cos\Phi a_x$ ta có :

$$\operatorname{tg}\Phi = \frac{\sin \Phi}{\cos \Phi} = \frac{a_y}{a_x}$$

Góc Φ có thể xác định bằng hàm arctg hai biến :

$$\Phi = \operatorname{arctg}2(a_y, a_x).$$

Ta cũng có thể giải phương trình (4-21) bằng cách cộng hai vế với $-\cos\Phi a_y$ rồi chia hai vế cho $-\cos\Phi a_x$, triệt tiêu $-a_x$ ở vế trái và $\cos\Phi$ ở vế phải, ta có :

$$\operatorname{tg}\Phi = \frac{\sin \Phi}{\cos \Phi} = \frac{-a_y}{-a_x}$$

Trong trường hợp này góc Φ tìm được là :

$$\Phi = \operatorname{arctg}2(-a_y, -a_x).$$

Như vậy phương trình (4-21) có một cặp nghiệm cách nhau 180° (đây là nghiệm toán) và ta có thể viết :

$$\Phi = \operatorname{arctg}2(a_y, a_x) \text{ và } \Phi = \Phi + 180^\circ.$$

(Hiệu theo cách viết khi lập trình trên máy tính).

Nếu cả a_x và a_y đều bằng 0 thì góc Φ không xác định được. Điều đó xảy ra khi bàn tay chỉ thẳng lên trên hoặc xuống dưới và cả hai góc Φ và ψ tương ứng với cùng một phép quay. Điều này được coi là một phép suy biến (degeneracy), trong trường hợp này ta cho $\Phi = 0$.

Với giá trị của Φ nhận được, các phần tử ma trận ở vế bên trái của phương trình (4-20) sẽ được xác định. Tiếp tục so sánh các phần tử của hai ma trận ta có :

$$f_{11}(a) = \cos\Phi a_x + \sin\Phi a_y = \sin\theta.$$

$$\text{Và } f_{13}(a) = a_z = \cos\theta.$$

$$\text{Vậy : } \theta = \arctg2(\cos\Phi a_x + \sin\Phi a_y, a_z)$$

Khi cả hai hàm sin và cos đều được xác định như trường hợp trên, thì góc thường được xác định duy nhất và không xảy ra trường hợp suy biến như góc Φ trước đây. Cũng từ phương trình (4-20) ta có :

$$f_{12}(n) = -\sin\Phi n_x + \cos\Phi n_y = \sin\psi$$

$$f_{12}(O) = -\sin\Phi O_x + \cos\Phi O_y = \cos\psi$$

$$\text{Vậy : } \psi = \arctg2(-\sin\Phi n_x + \cos\Phi n_y, -\sin\Phi O_x + \cos\Phi O_y)$$

Tóm lại, nếu cho trước một phép biến đổi đồng nhất dưới dạng các phép quay, ta có thể xác định các góc Euler tương ứng là :

$$\Phi = \arctg2(a_y, a_x) \text{ và } \Phi = \Phi + 180^0$$

$$\theta = \arctg2(\cos\Phi a_x + \sin\Phi a_y, a_z)$$

$$\psi = \arctg2(-\sin\Phi n_x + \cos\Phi n_y, -\sin\Phi O_x + \cos\Phi O_y)$$

4.3. Lời giải của phép biến đổi Roll, Pitch và Yaw :

Phép biến đổi Roll, Pitch và Yaw đã được định nghĩa :

$$\text{RPY}(\Phi, \theta, \psi) = \text{Rot}(z, \Phi)\text{Rot}(y, \theta)\text{Rot}(x, \psi)$$

Việc giải phương trình : $T_6 = \text{RPY}(\Phi, \theta, \psi)$ sẽ xác định được các góc Φ, θ và ψ .

Cách giải được tiến hành tương tự như khi thực hiện lời giải cho phép quay Euler. Nhân T_6 với ma trận nghịch đảo $\text{Rot}(z, \Phi)^{-1}$, ta có :

$$\text{Rot}(z, \Phi)^{-1}T_6 = \text{Rot}(y, \theta)\text{Rot}(x, \psi)$$

Hay là :

$$\text{Giáo trình} \left(\begin{array}{cccc} f_{11}(n) & f_{11}(O) & f_{11}(a) & 0 \\ f_{12}(n) & f_{12}(O) & f_{12}(a) & 0 \\ f_{13}(n) & f_{13}(O) & f_{13}(a) & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \left(\begin{array}{cccc} \cos\theta & \sin\theta \sin\psi & \sin\theta \cos\psi & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ -\sin\theta & \cos\theta \sin\psi & \cos\theta \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{array} \right) \quad 63$$

Trong đó :

$$f_{11} = \cos\Phi x + \sin\Phi y$$

$$f_{12} = -\sin\Phi x + \cos\Phi y$$

$$f_{13} = z$$

Cân bằng phần tử ở hàng 2 cột 1 : $f_{12}(n) = 0$, ta có :

$$-\sin\Phi x + \cos\Phi y = 0$$

Phương trình này cho ta hai nghiệm như đã biết :

$$\Phi = \arctg2(n_x, n_y)$$

và

$$\Phi = \Phi + 180^\circ$$

Tiếp tục cân bằng các phần tử tương ứng của hai ma trận ta có :

$$-\sin\theta = n_z$$

$$\cos\theta = \cos\Phi n_x + \sin\Phi n_y$$

do vậy :

$$\theta = \arctg2(-n_z, \cos\Phi n_x + \sin\Phi n_y)$$

Ngoài ra ta còn có :

$$-\sin\psi = -\sin\Phi a_x + \cos\Phi a_y$$

$$\cos\psi = -\sin\Phi O_x + \cos\Phi O_y$$

Nên :

$$\psi = \arctg2(\sin\Phi a_x - \cos\Phi a_y, -\sin\Phi O_x + \cos\Phi O_y)$$

Như vậy ta đã xác định được các góc quay Roll, Pitch và Yaw theo các phần tử của ma trận T_6 .

4.4. Giải bài toán động học ngược của robot Stanford :

Hệ phương trình động học của robot Stanford đã được thiết lập trong chương III, Ta có :

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \quad (4-23)$$

Liên tục nhân (4-23) với các ma trận A nghịch đảo, ta được :

$$A_1^{-1}T_6 = {}^1T_6 \tag{4-24}$$

$$A_2^{-1}A_1^{-1}T_6 = {}^2T_6 \tag{4-25}$$

$$A_3^{-1}A_2^{-1}A_1^{-1}T_6 = {}^3T_6 \tag{4-26}$$

$$A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}T_6 = {}^4T_6 \tag{4-27}$$

$$A_5^{-1}A_4^{-1}A_3^{-1}A_2^{-1}A_1^{-1}T_6 = {}^5T_6 \tag{4-28}$$

Các phần tử ở vế trái của các phương trình này là hàm số của các phần tử T_6 và các biến khớp của (n-1) khớp đầu tiên. Trong khi đó các phần tử của ma trận vế bên phải hoặc bằng 0, bằng hằng số hoặc là hàm số của các biến khớp thứ n đến khớp thứ 6. Từ mỗi phương trình ma trận, cho cân bằng các phần tử tương ứng chúng ta nhận được 12 phương trình. Mỗi phương trình có các phần tử của 4 vector n, O, a, p.

Từ phương trình (4-24), ta có :

$$A_1^{-1}T_6 = \begin{pmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_x & O_x & a_x & p_x \\ n_y & O_y & a_y & p_y \\ n_z & O_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} f_{11}(n) & f_{11}(O) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(O) & f_{12}(a) & f_{12}(p) \\ f_{13}(n) & f_{13}(O) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Trong đó : $f_{11} = C_1 x + S_1 y$

$$f_{12} = -z$$

$$f_{13} = -S_1 x + C_1 y$$

Vế bên phải của (4-24) là :

$${}^1T_6 = \begin{pmatrix} C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 & -C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6 & C_2C_4S_5 + S_2C_5 & S_2d_1 \\ S_2(C_4C_5C_6 - S_4S_6) + C_2S_5C_6 & -S_2(C_4C_5S_6 + S_4C_6) - C_2S_5S_6 & S_2C_4S_5 - C_2C_5 & -C_2d_1 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Các phần tử của ma trận vế phải đều là hàm số của $\theta_2, d_3, \theta_4, \theta_5, \theta_6$ ngoại trừ phần tử ở hàng 3 cột 4, đó là :

$$f_{13}(p) = d_2$$

hay :

$$-S_1p_x + C_1p_y = d_2$$

Để giải phương trình ở dạng này ta có thể thay thế bởi các hàm lượng giác sau đây:

$$p_x = r \cos\Phi$$

$$p_y = r \sin\Phi$$

Trong đó : $r = +\sqrt{p_x^2 + p_y^2}$

$$\Phi = \text{arctg2}(p_x, p_y)$$

Thế p_x và p_y vào phương trình $-S_1 p_x + C_1 p_y = d_2$ ta có :

$$\sin\Phi \cos\theta_1 - \cos\Phi \sin\theta_1 = d_2 / r \text{ Với } 0 < d_2 / r \leq 1$$

Hay là : $\sin(\Phi - \theta_1) = d_2 / r$ Với $0 < \Phi - \theta_1 < \pi$

Từ đó ta có :

$$\cos(\Phi - \theta_1) = \pm\sqrt{1 - (d_2 / r)^2}$$

Trong đó dấu trừ phù hợp với hình thể vai trái của robot và dấu cộng phù hợp với hình thể vai phải của robot. Cuối cùng :

$$\theta_1 = \text{arctg2}(p_y, p_x) - \text{arctg2}(d_2, \pm\sqrt{1 - (d_2 / r)^2}) \quad (4-29)$$

Nếu tính được θ_1 thì về trái của phương trình (4-24) được xác định.

Cho cân bằng các phần tử ở hàng 1 cột 4 và hàng 2 cột 4, ta có :

$$S_2 d_3 = C_1 p_x + S_1 p_y$$

$$-C_2 d_3 = -p_z$$

d_3 là dịch chuyển dài của khớp tịnh tiến, $d_3 > 0$, nên ta có :

$$\theta_2 = \text{arctg2}(C_1 p_x + S_1 p_y, p_z) \quad (4-30)$$

Từ phương trình (4-25) : $A_2^{-1} A_1^{-1} T_6 = A_2^{-1} {}^1 T_6 = {}^2 T_6$, ta có

$$\begin{pmatrix} f_{21}(n) & f_{21}(O) & f_{21}(a) & 0 \\ f_{22}(n) & f_{22}(O) & f_{22}(a) & 0 \\ f_{23}(n) & f_{23}(O) & f_{23}(a) & f_{23}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & C_4 S_5 & 0 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & S_4 S_5 & 0 \\ -S_5 C_6 & S_5 S_6 & C_5 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4-31)$$

Trong đó : $f_{21} = C_2(C_1 x + S_1 y) - S_2 z$

$$f_{22} = -S_1 x + C_1 y$$

$$f_{23} = S_2(C_1 x + S_1 y) + C_2 z$$

Từ cân bằng phần tử ở hàng 3 cột 4 ta có :

$$d_3 = S_2(C_1 p_x + S_1 p_y) + C_2 p_z \quad (4-32)$$

- Từ phương trình (4-27) ta có : $A_4^{-1} A_3^{-1} {}^2 T_6 = {}^4 T_6$

Thực hiện phép nhân các ma trận ở vế trái, và biểu diễn ở dạng rút gọn sau

Trong đó :

$$f_{41} = C_4[C_2(C_1 x + S_1 y) - S_2 z] + S_4(-S_1 x + C_1 y)$$

$$f_{42} = -S_2(-S_1 x + C_1 y) - C_2 z$$

$$f_{43} = -S_4[C_2(C_1 x + S_1 y) + S_2 z] + C_4(-S_1 x + C_1 y)$$

Cân bằng phần tử hàng 3, cột 3 ta được một hàm số của θ_4 , đó là :

$$f_{43}(a) = 0.$$

Hay :

$$-S_4[C_2(C_1 a_x + S_1 a_y) + S_2 a_z] + C_4(-S_1 a_x + C_1 a_y) = 0$$

Đây là phương trình lượng giác có dạng : $-\sin\Phi a_x + \cos\Phi a_y = 0$. Như đã giải trong các phần trước đây, phương trình này có hai nghiệm :

$$\left. \begin{aligned} \theta_4 &= \arctg2(-S_1 a_x + C_1 a_y, C_2(C_1 a_x + S_1 a_y) + S_2 a_z) \\ \text{và} \quad \theta_4 &= \theta_4 + 180^0 \end{aligned} \right\} \quad (4-33)$$

Nếu các yếu tố tử số và mẫu số của (4-33) tiến tới 0 thì robot rơi vào tình trạng suy biến như trường hợp đã nói ở mục 4.2.

Ta cũng có thể tìm giá trị của góc quay θ_4 bằng cách cân bằng các phần tử hàng 1 cột 3 và hàng 2 cột 3 của phương trình ma trận (4-31) , ta có :

$$C_4 S_5 = C_2(C_1 a_x + S_1 a_y) - S_2 a_z$$

$$S_4 S_5 = -S_1 a_x + C_1 a_y$$

$$\text{Với } \theta_5 > 0 \text{ ta được } \theta_4 = \arctg(-S_1 a_x + C_1 a_y, C_2(C_1 a_x + S_1 a_y) + S_2 a_z)$$

$$\text{Với } \theta_5 < 0 \text{ ta được } \theta_4 = \theta_4 + 180^0$$

đúng như kết quả đã tìm (4-33).

Khi $S_5 = 0, \theta_5 = 0$. Robot có suy biến do cả hai trục của khớp 4 và 6 nằm thẳng hàng ($z_3 \equiv z_5$). ở vị trí này chỉ có tổng $\theta_4 + \theta_6$ là có ý nghĩa.

Khi $\theta_5 = 0$, ta có thể tự do chọn một giá trị của θ_4 . Thường giá trị hiện hành được sử dụng.

Từ vế phải của phương trình $A_4^{-1}A_5^{-1}T_6 = {}^4T_6 = A_5A_6$ ta có thể có các phương trình của S_5, C_5, S_6 và C_6 bằng cách cân bằng các phần tử thích hợp.

Chẳng hạn khi cân bằng các phần tử của ma trận hàng 1 cột 3 và hàng 2 cột 3 ta có :

$$S_5 = C_4 [C_2(C_1 a_x + S_1 a_y) - S_2 a_z] + S_4(-S_1 a_x + C_1 a_y)$$

$$C_5 = S_2 (C_1 a_x + S_1 a_y) + C_2 a_z$$

Từ đó suy ra :

$$\theta_5 = \arctg2(C_4 [C_2(C_1 a_x + S_1 a_y) - S_2 a_z] + S_4(-S_1 a_x + C_1 a_y), S_2 (C_1 a_x + S_1 a_y) + C_2 a_z) \quad (4-34)$$

Các phương trình có liên quan đến θ_6 nằm ở cột 1 của phương trình ma trận, đó là các thành phần của vectơ n của T_6 . Vectơ này thường không có ý nghĩa trong tính toán, vì nó luôn có thể được xác định bằng tích vectơ của hai vectơ O và a như đã nói trước đây ($\vec{h} = \vec{O} \times \vec{a}$). Do đó ta phải tìm cách khác để xác định θ_6 .

Thực hiện phép nhân các ma trận ở vế trái của phương trình (4-28)

$A_5^{-1} T_6 = {}^5T_6 = A_6$ biểu diễn dưới dạng ký hiệu ta có:

$$\begin{pmatrix} f_{51}(n) & f_{51}(O) & 0 & 0 \\ f_{52}(n) & f_{52}(O) & 0 & 0 \\ f_{53}(n) & f_{53}(O) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4-35)$$

Trong đó :

$$f_{51} = C_5 \{ C_4 [C_2(C_1 x + S_1 y) - S_2 z] + S_4(-S_1 x + C_1 y) \} + S_5[-S_2 (C_1 x + S_1 y) - C_2 z]$$

$$f_{52} = -S_4 [C_2(C_1 x + S_1 y) - S_2 z] + C_4[-S_1 x + C_1 y]$$

$$f_{53} = S_5 \{ C_4 [C_2(C_1 x + S_1 y) - S_2 z] + S_4(-S_1 x + C_1 y) \} + C_5[S_2 (C_1 x + S_1 y) - C_2 z]$$

Cho cân bằng các phần tử ở hàng 1 cột 2 và hàng 2 cột 2 ta nhận được các giá trị của S_6 và C_6 :

$$S_6 = -C_5 \{ C_4 [C_2(C_1 O_x + S_1 O_y) - S_2 O_z] + S_4(-S_1 O_x + C_1 O_y) \} + S_5[S_2 (C_1 O_x + S_1 O_y) + C_2 O_z]$$

$$C_6 = -S_4 [C_2(C_1 O_x + S_1 O_y) - S_2 O_z] + C_4[-S_1 O_x + C_1 O_y]$$

$$\text{Từ đó ta xác định được : } \theta_6 = \arctg2(S_6, C_6) \quad (4-36)$$

Các biểu thức (4-29), (4-30), (4-32), (4-33), (4-34) và (4-36) xác định tập nghiệm khi giải bài toán ngược của robot Stanford.

4.5. Giải bài toán động học ngược của robot ELBOW :

Để tiếp tục làm quen với việc giải hệ phương trình động học, chúng ta nghiên cứu phép giải bài toán động học ngược của robot Elbow. Hệ phương trình động học thuận của robot Elbow đã được xác định trong chương III.

Trước hết ta khảo sát phương trình :

$$A_1^{-1}T_6 = {}^1T_6 = A_2 A_3 A_4 A_5 A_6$$

Tương tự như đã làm, ta xác định các phần tử ma trận của hai vế như sau :

$$\begin{pmatrix} f_{11}(a) & f_{11}(0) & f_{11}(a) & f_{11}(p) \\ f_{12}(a) & f_{12}(0) & f_{12}(a) & f_{12}(p) \\ f_{13}(a) & f_{13}(0) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_{234}C_5C_6 - S_{234}S_6 & -C_{234}C_5S_6 - S_{234}C_6 & C_{234}S_5 & C_{234}a_4 + C_{233}a_3 + C_2a_2 \\ S_{234}C_5C_6 + C_{234}S_6 & -S_{234}C_5S_6 + C_{234}C_6 & S_{234}S_5 & S_{234}a_4 + S_{233}a_3 + S_2a_2 \\ -S_5C_6 & S_5S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4-37}$$

Trong đó : $f_{11} = C_1 x + S_1 y$

$f_{12} = z$

$f_{13} = S_1 x + C_1 y$

Ta đã ký hiệu : $C_{234} = \cos(\theta_2 + \theta_3 + \theta_4)$

$S_{234} = \sin(\theta_2 + \theta_3 + \theta_4)$

Cho cân bằng phần tử ở hàng 3 cột 4, ta có :

$S_1 p_x + C_1 p_y = 0$

Suy ra :

$\theta_1 = \arctg2(p_y, p_x)$

và $\theta_1 = \theta_1 + 180^\circ$ (4-38)

Trong trường hợp robot Elbow, ba khớp kế tiếp đều song song và không có kết quả nạo nhận được từ phép nhân với những ma trận nghịch đảo A_i^{-1} . Cho đến khâu thứ 4 thì phép nhân với ma trận nghịch đảo mới có ý nghĩa.

$$A_4^{-1} A_3^{-1} A_2^{-1} T_6 = T_6 = A_5 A_6$$

Khi xác định các phần tử ma trận của hai vế ta được :

$$\begin{pmatrix} f_{41}(n) & f_{41}(O) & f_{41}(a) & f_{41}(p) - C_{34}a_2 - C_4a_3 - a_4 \\ f_{42}(n) & f_{42}(O) & f_{42}(a) & 0 \\ f_{43}(n) & f_{43}(O) & f_{43}(a) & f_{43}(p) + S_{34}a_2 + S_4a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4-39)$$

Trong đó : $f_{41} = C_{234}(C_1 x + S_1 y) + S_{234} z$

$$f_{42} = -S_1 x + C_1 y$$

$$f_{43} = -S_{234}(C_1 x + S_1 y) + C_{234} z$$

Cân bằng phần tử hàng 3 cột 3 ta được một phương trình cho θ_{234} :

$$-S_{234}(C_1 a_x + S_1 a_y) + C_{234} a_z = 0$$

Suy ra :

$$\theta_{234} = \arctg2(a_z, C_1 a_x + S_1 a_y)$$

$$\text{và } \theta_{234} = \theta_{234} + 180^0 \quad (4-40)$$

Bây giờ ta trở lại phương trình (4-37). Cân bằng các phần tử ma trận ở hàng 1 cột 4 và hàng 2 cột 4, ta có :

$$C_1 p_x + S_1 p_y = C_{234} a_4 + C_{23} a_3 + C_2 a_2 \quad (a)$$

$$p_z = S_{234} a_4 + S_{23} a_3 + S_2 a_2 \quad (b)$$

Ta gọi : $p'_x = C_1 p_x + S_1 p_y - C_{234} a_4$ (c)

$$p'_y = p_z - S_{234} a_4 \quad (d)$$

Đem (a) + (c) và (b) + (d) ta được ;

$$p'_x = C_{23} a_3 + C_2 a_2 \quad (e)$$

$$p'_y = S_{23} a_3 + S_2 a_2 \quad (g)$$

Bình phương hai vế và cộng hai phương trình (e) và (g), ta có :

$$p'^2_x = (C_{23} a_3 + C_2 a_2)^2$$

$$p'^2_y = (S_{23} a_3 + S_2 a_2)^2$$

$$p'^2_x + p'^2_y = (S_{23}^2 + C_{23}^2) a_3^2 + (S_2^2 + C_2^2) a_2^2 + 2 a_2 a_3 (C_{23} C_2 + S_{23} S_2)$$

Ta có $C_{23} C_2 + S_{23} S_2 = \cos(\theta_2 + \theta_3 - \theta_2) = \cos\theta_3 = C_3$. Nên suy ra :

$$C_3 = (p'_x{}^2 + p'_y{}^2 - a_3^2 - a_2^2) / 2a_2a_3$$

Trong khi có thể tìm θ_3 từ hàm arccos, ta vẫn nên tìm một giá trị S_3 và dùng hàm arctg² như thường lệ :

$$\text{Ta có : } S_3 = \pm\sqrt{1 - C_3^2}$$

Cặp nghiệm ứng với hai dấu +,- phù hợp với hình thể của robot lúc nâng vai lên và hạ vai xuống :

$$\theta_3 = \text{arctg2}(S_3, C_3) \quad (4-41)$$

Để tìm S_2 và C_2 ta giải hệ phương trình (e),(g).

$$\text{Từ (e)} \Rightarrow (C_2C_3 - S_2S_3)a_3 + C_2a_2 = p'_x$$

$$\text{Từ (g)} \Rightarrow (S_2C_3 - C_2S_3)a_3 + S_2a_2 = p'_y$$

Khai triển và rút gọn :

$$\begin{aligned} \text{Từ (g)} \Rightarrow & (C_3a_3 + a_2)C_2 - S_3a_3.S_2 = p'_x \\ \text{Ta có :} & S_3a_3.C_2 + (C_3a_3 + a_2)S_2 = p'_y \end{aligned}$$

$$\Delta = \begin{vmatrix} C_3a_3 + a_2 & -S_3a_3 \\ S_3a_3 & C_3a_3 + a_2 \end{vmatrix}$$

$$\Delta_c = \begin{vmatrix} p'_x & -S_3a_3 \\ p'_y & C_3a_3 + a_2 \end{vmatrix} \quad \Delta_s = \begin{vmatrix} C_3a_3 + a_2 & p'_x \\ S_3a_3 & p'_y \end{vmatrix}$$

$$C_2 = \frac{\Delta_c}{\Delta} = \frac{(C_3a_3 + a_2)p'_y - S_3a_3p'_x}{(C_3a_3 + a_2)^2 + (S_3a_3)^2}$$

$$S_2 = \frac{\Delta_s}{\Delta} = \frac{(C_3a_3 + a_2)p'_x + S_3a_3p'_y}{(C_3a_3 + a_2)^2 + (S_3a_3)^2}$$

Do mẫu số dương và bằng nhau, nên ta có :

$$\theta_2 = \text{arctg2}(S_2, C_2)$$

$$\theta_2 = \text{arctg2}((C_3a_3 + a_2)p'_y - S_3a_3p'_x, (C_3a_3 + a_2)p'_x + S_3a_3p'_y) \quad (4-42)$$

Đến đây θ_4 được xác định bởi :

$$\theta_4 = \theta_{234} - \theta_3 - \theta_2 \quad (4-$$

43)

Các phương trình dùng để tính θ_5 được thiết lập từ sự cân bằng các phần tử ma trận hàng 1 cột 3 và hàng 2 cột 3 của phương trình 4T_6 (4-39) :

$$S_5 = C_{234}(C_1a_x + S_1a_y) + S_{234}a_z$$

$$C_5 = S_1 a_x - C_1 a_y$$

Suy ra :

$$\theta_5 = \arctg2(C_{234}(C_1 a_x + S_1 a_y) + S_{234} a_z, S_1 a_x - C_1 a_y) \quad (4-44)$$

Để tìm θ_6 , ta tiếp tục nhân A_5^{-1} với 4T_6 , ta được : $A_5^{-1} \cdot {}^4T_6 = A_6$.

Viết tích ma trận về trái ở dạng ký hiệu :

$$\begin{pmatrix} f_{51}(n) & f_{51}(0) & 0 & 0 \\ f_{52}(n) & f_{52}(0) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Trong đó : $f_{51} = C_5[C_{234}(C_1 x + S_1 y) + S_{234} z] - S_5(S_1 x + C_1 y)$

$$f_{52} = -S_{234}(C_1 x + S_1 y) + C_{234} z$$

Cho cân bằng các phần tử ma trận tương ứng, ta có :

$$S_6 = -C_5[C_{234}(C_1 O_x + S_1 O_y) + S_{234} O_z] - S_5(S_1 O_x + C_1 O_y)$$

$$C_6 = -S_{234}(C_1 O_x + S_1 O_y) + C_{234} O_z$$

Vậy :

$$\theta_6 = \arctg2(S_6, C_6) \quad (4-45)$$

Các phương trình (4-38), (4-41), (4-42), (4-43), (4-44) và (4-45) xác định tập nghiệm khi giải bài toán động học ngược của robot Elbow.

4.6. Kết luận :

Phương pháp giải bài toán động học ngược đưa ra trong chương này sử dụng các hàm lượng giác tự nhiên. Các góc thường được xác định thông qua hàm arctang hai biến. Phương pháp này được đưa ra bởi Pieper và áp dụng tốt với những robot đơn giản, Thường ta nhận được nghiệm ở dạng công thức đơn giản.

Khi giải bài toán động học ngược có thể xảy ra hiện tượng “suy biến” . Khi có nhiều hơn một tập nghiệm đối với bài toán ngược để xác định vị trí và hướng của bàn tay, thì cánh tay được gọi là suy biến.

Dùng phương pháp Pieper, các nghiệm nhận được thường có 4 dạng công thức, mỗi dạng có một ý nghĩa động học riêng. Dạng thứ nhất

:

$$- \sin\Phi ax + \cos\Phi ay = 0$$

Dạng này cho ta một cặp nghiệm cách nhau 180°, nó mô tả hai hình thê tương ứng của robot. Nếu các tử số và mẫu số đều bằng không, robot bị suy biến, lúc đó robot mất đi một bậc tự do.

Dạng thứ hai :

$$-S_1p_x + C_1p_y = d_2$$

Dạng này cũng cho ta cặp nghiệm sai khác nhau 180°, một lần nữa lại tồn tại khả năng suy biến khi tử số và mẫu số bằng 0. Robot ở trường hợp này thường có một khớp tịnh tiến và độ dài tịnh tiến được coi là > 0 .

Dạng thứ ba :

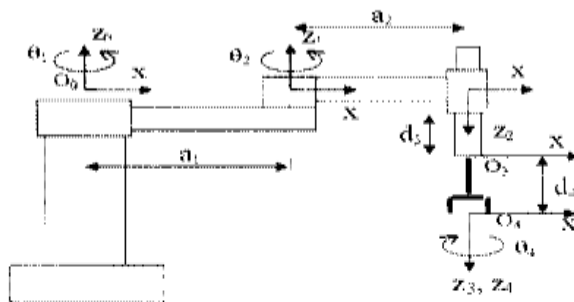
$$C_1p_x + S_1p_y = S_2d_3$$

và dạng thứ tư :

$$- C_2d_3 = - p_z$$

Bài tập chương IV

Thiết lập hệ phương trình động học của robot SCARA (hình 4.4) và giải bài toán động học ngược của nó.



Hình 4.4 : Robot SCARA

CHƯƠNG V

NGÔN NGỮ LẬP TRÌNH ROBOT (Robot Programming Languages)

5.1. Giới thiệu chung về lập trình điều khiển robot :

Lập trình điều khiển robot thể hiện mối quan hệ giữa người điều khiển và robot công nghiệp. Tính phức tạp của việc lập trình cũng tăng khi các ứng dụng công nghiệp đòi hỏi sử dụng đồng thời nhiều robot với các máy tự động khả lập trình khác tạo nên hệ thống sản xuất tự động linh hoạt.

Robot khác với các máy tự động cố định ở tính “linh hoạt”, nghĩa là có thể lập trình được (Programmable : khả lập trình). Không những chỉ có các chuyển động của robot mà ngay cả việc sử dụng các cảm biến cũng như những thông tin quan hệ với máy tự động khác trong phân xưởng cũng có thể lập trình. Robot có thể dễ dàng thích nghi với sự thay đổi của nhiệm vụ sản xuất bằng cách thay đổi chương trình điều khiển nó.

Khi xem xét vấn đề lập trình cho robot, chúng ta nên nhớ rằng robot là một thành phần của một quá trình được tự động hoá. Thuật ngữ, workcell được dùng để mô tả một tập hợp các thiết bị mà nó bao gồm một hoặc nhiều robot, hệ thống băng chuyền, các cơ cấu cấp phôi, và đồ gá. ở mức cao hơn, Workcell có thể được liên kết trong mạng lưới các phân xưởng vì thế máy tính điều khiển trung tâm có thể điều khiển toàn bộ các hoạt động của phân xưởng. Vì vậy, việc lập trình điều khiển robot trong thực tế sản xuất cần phải được xem xét trong mối quan hệ rộng hơn.

Để bước đầu làm quen với việc lập trình điều khiển robot, chương này cũng giới thiệu tóm tắt phương pháp lập trình điều khiển robot TERGAN-45 thông qua ngôn ngữ ASPECT của phần mềm Procomm Plus for Window.

5.2. Các mức lập trình điều khiển robot :

Người sử dụng có thể có nhiều kiểu giao diện lập trình điều khiển robot. Trước sự phát triển nhanh chóng của các loại máy vi tính dùng trong công nghiệp và các ngôn ngữ lập trình ngày càng có nhiều tiện ích cao, việc lập trình điều khiển robot ngày càng dễ dàng và thuận tiện hơn.

5.2.1. Lập trình kiểu “Dạy - Học” :

Các robot thế hệ đầu tiên đã được lập trình bằng một phương pháp mà chúng ta gọi là : dạy bằng chỉ dẫn (Teach by showing), robot được điều khiển để di chuyển đến các điểm mong muốn và các vị trí đó được ghi lại trong bộ nhớ của máy tính, sau đó các dữ liệu sẽ được đọc tuần tự và robot thực hiện lại các động tác đã được học. Để dạy robot, người sử dụng có thể hướng dẫn robot bằng tay hoặc thông qua một thiết bị dạy học gọi là Teach pendant. Thiết bị dạy học gồm một hộp nhỏ cầm tay (teaching box) có các nút bấm và card điều khiển mà nó cho phép điều khiển các khớp của robot đạt được các giá trị mong muốn.

5.2.2. Dùng các ngôn ngữ lập trình :

Cùng với quá trình phát triển ngày càng rõ hơn và mạnh hơn của máy tính,, chương trình điều khiển robot được phát triển theo hướng viết các chương trình bằng các ngôn ngữ lập trình của máy tính. Thường các ngôn ngữ lập trình này có những đặc điểm mà chúng ta có thể ứng dụng để viết các phần mềm hay chương trình điều khiển robot, và chúng được gọi là “ngôn ngữ lập trình robot”. Hầu hết các hệ thống điều khiển dùng ngôn ngữ lập trình robot vẫn duy trì kiểu giao diện Teach pendant (dạy- học).

Ngôn ngữ lập trình robot có nhiều dạng khác nhau. Chúng ta phân chúng thành ba loại như sau :

a) Ngôn ngữ robot chuyên dùng : những ngôn ngữ lập trình robot này được xây dựng bằng cách tạo ra một ngôn ngữ mới hoàn toàn. Cú pháp (Syntax) và ngữ nghĩa (Semantics) của các ngôn ngữ này cần phải rất đơn giản vì người lập trình cho các ứng dụng công nghiệp không phải là một

chuyên gia về lập trình. Ví dụ như ngôn ngữ VAL (VAL 2) được dùng để điều khiển các robot công nghiệp của hãng Unimation (Hoa Kỳ); hoặc một ngôn ngữ robot chuyên dùng khác gọi là AL được xây dựng ở Đại học Stanford (Hoa Kỳ)...

b) Tạo ra các thư viện robot cho một ngôn ngữ lập trình cấp cao đã có sẵn : Những ngôn ngữ lập trình robot này được xây dựng bằng cách dựa trên các ngôn ngữ lập trình cấp cao thông dụng (ví dụ như Pascal) và thêm vào một thư viện các thủ tục và hàm đặc biệt dùng cho robot. Khi viết các chương trình Pascal để điều khiển robot, người sử dụng sẽ gọi các hàm hoặc thủ tục đã định nghĩa trước trong thư viện để xử lý các nội dung có liên quan đến việc tính toán hoặc điều khiển robot.

Ví dụ PASRO (Pascal for Robot) là một thư viện dùng cho lập trình robot, cung cấp nhiều thủ tục và hàm đặc biệt để tính toán và điều khiển robot dùng trong môi trường ngôn ngữ Turbo Pascal, hoặc PASRO/C là phát triển của PASRO, nhưng được viết trên cơ sở của ngôn ngữ Turbo C.

c) Tạo ra các thư viện robot cho một ngôn ngữ hoặc phần mềm đa dụng (Robot library for a new general - purpose language) : Những ngôn ngữ lập trình robot này được xây dựng bằng cách sử dụng các ngôn ngữ hoặc phần mềm dùng chung cho nhiều mục đích như là một chương trình cơ bản, sau đó cung cấp thêm một thư viện chứa các thủ tục đặc biệt dùng cho robot. Ví dụ như ngôn ngữ lập trình robot AML của hãng IBM và RISE của hãng Silma, ngôn ngữ Aspect của phần mềm Procomm Plus ...

5.2.3. Ngôn ngữ lập trình theo nhiệm vụ (Task-level programming language)

Mức thứ ba của phương pháp lập trình robot là tạo ra các ngôn ngữ lập trình theo nhiệm vụ. Những ngôn ngữ này cho phép người sử dụng ra các lệnh để robot thực hiện một công việc mong muốn một cách trực tiếp mà không cần xác định một cách chi tiết các hoạt động của robot như các

ngôn ngữ lập trình thông thường. Một hệ thống lập trình robot theo nhiệm vụ phải có khả năng thể hiện nhiều công việc một cách tự động. Chẳng hạn, nếu một chỉ thị “Grasp the bolt” (cầm lấy bulong) được tạo ra, hệ thống phải vạch ra được một quỹ đạo của tay máy mà nó tránh được sự va chạm với bất kỳ chướng ngại vật nào chung quanh, chọn được vị trí tốt nhất để cầm lấy bulong một cách tự động. Ngược lại, trong ngôn ngữ lập trình robot thông thường tất cả những sự lựa chọn này phải được thực hiện bởi người lập trình. Trong thực tế, ngôn ngữ lập trình theo nhiệm vụ chưa được dùng trong sản xuất, nó còn là một lĩnh vực đang được nghiên cứu.

Sau đây ta sẽ nghiên cứu một phần mềm đa dụng dùng truyền dữ liệu và điều khiển thiết bị có thể dùng để điều khiển robot.

5.3. Giới thiệu tóm tắt phần mềm Procomm Plus For Windows :

Procomm Plus là phần mềm dùng để truyền dữ liệu và điều khiển trực tiếp các thiết bị qua cổng COM của máy tính cá nhân. Với Procomm Plus ta có thể sử dụng máy tính như một Terminal hoặc thực hiện các Scrip files viết bằng ngôn ngữ lập trình Aspect.

Để chạy phần mềm Procomm Plus ở chế độ Terminal ta có thể dùng một số cách sau :

a) Sử dụng Desktop Windows : ấn đúp chuột trên biểu tượng của Procomm Plus terminal Windows.

b) Từ mục Run... trong Start của Windows, gõ lệnh : pw3 , chọn OK.

c) Vào Start của Windows, chọn Programs, chọn Procomm Plus 3, chọn Data Terminal...

Menu chính của Procomm Plus có nhiều tiện ích, rất thuận tiện khi điều khiển các thiết bị giao diện với máy tính kiểu RS 232.

Cửa sổ chính của phần mềm Procomm plus ở chế độ Terminal như hình 5.1.



Hình 5.1: Cửa sổ chính của Procomm Plus Windows, Version 3.0

Menu chính : Cung cấp các tiện ích cần thiết trong quá trình sử dụng, menu chính có các menu kéo xuống (Pulldown) tương tự như nhiều phần mềm thông dụng khác. Nội dung của Menu chính có thể thay đổi được theo mục đích sử dụng.

Một số nội dung của menu chính có thể dùng trong quá trình điều khiển robot như sau :

Menu Setup : Dùng để xác định cấu hình của Terminal Windows và chế độ giao diện giữa máy tính với thiết bị. Trong menu này còn có thể sử dụng mục con Action Bars để chọn file chứa nội dung của thanh công cụ và cho thể hiện trên màn hình.

Menu Data : Trong menu này ta có thể dùng các menu con sau :

- + Clear screen (Alt+C): Xoá màn hình nhập xuất dữ liệu;
- + Reset terminal (Alt+U): Xoá màn hình và bộ đệm (buffer) của

Procomm.


Menu Scripts : Trong menu này ta có thể dùng các menu con sau :

+ Start scrips (Alt+.) : Thực hiện một Aspect scrips file, có tên được thể hiện trên thanh công cụ.

+ Run... (Alt+F5) : Mở hộp hội thoại Run ASPECT file , chúng ta có thể chọn tên file, thực hiện việc dịch các file nguồn trước khi chạy chương trình.

+ Compile / Edit... (Alt+F3) : Mở hộp hội thoại soạn thảo và dịch các file nguồn.

+ Start recorder... : bắt đầu tự động tạo ra một scrips file bằng các ghi lại tất cả các lệnh thể hiện trên màn hình (nhập từ bàn phím). Khi chọn mục này sẽ xuất hiện mục Stop recorder, dùng khi muốn kết thúc việc ghi tự động scrips file.

Có thể chọn mục này trên thanh công cụ bằng cách ấn chuột vào biểu tượng .

Menu Tools : Trong menu này ta có thể dùng các menu con sau :

+ Action bar Edition : Dùng để soạn thảo hay thay đổi nội dung thanh công cụ cho phù hợp với mục đích sử dụng.

+ Aspect Editor : Mở cửa sổ để soạn thảo script files bằng ngôn ngữ Aspect, chúng ta có thể tạo mới, xem hoặc sửa đổi nội dung của một file (dạng Text).

+ Dialog Editor : Mở cửa sổ soạn thảo hộp hội thoại, cho phép ta tạo ra các hộp hội thoại bằng phương pháp trực quan (Visual).

Thanh công cụ (Tool bar) : có nhiều Icon (biểu tượng) giúp người sử dụng có thể thực hiện nhanh một công việc bằng cách bấm chuột trên biểu tượng tương ứng, thay vì phải vào Menu chính. Nội dung của thanh công cụ cũng có thể thay đổi dễ dàng để phù hợp với mục đích sử dụng (mục Action bar Edition).

Cửa sổ nhập - xuất dữ liệu : là phần màn hình để người sử dụng nhập vào các dữ liệu, các lệnh điều khiển và các thể hiện thông báo trả về từ các thiết bị được điều khiển.

Meta Keys : Dùng để cài đặt sẵn các ứng dụng thường hay thực hiện. Nội dung của các Meta Keys có thể thay đổi được để phù hợp với từng mục đích sử dụng. Khi muốn thực hiện một công việc đã gán cho Meta Key chỉ cần ấn chuột vào Meta key đó. Muốn soạn thảo hay thay đổi nhiệm vụ của Meta Keys ta thực hiện như sau :

Cách 1 : ấn phím ALT+M .

Cách 2 : Chọn mục Meta Keys Editor từ Tool menu .

Dòng chọn nhanh kiểu giao diện : Cho phép người sử dụng chọn nhanh kiểu thông số giao diện giữa thiết bị điều khiển và máy tính như : cổng giao diện, tốc độ truyền thông tin, kiểu xử lý dữ liệu ... bằng cách ấn nút chuột trái lên các mục.

5.4. Ngôn ngữ lập trình ASPECT trong Procomm :


5.4.1. Giới thiệu :

Một ASPECT script file là một file dạng text được tạo ra để chứa các lệnh được thực hiện bởi Procomm Plus.

Giống như nhiều ngôn ngữ lập trình khác, ASPECT yêu cầu phải dịch chương trình soạn thảo. Một script file chưa dịch, hay còn gọi là file nguồn, có đuôi là .was (Windows Aspect Source); còn một script file đã dịch có đuôi là .wax (Windows Aspect eXecutable). Khi một script đã được dịch, thì các dữ liệu và các câu lệnh chứa trong file nguồn sẽ được chuyển sang mã mà Procomm có thể đọc và xử lý một cách nhanh chóng. Sau khi dịch thì file dịch (.wax) có kích thước nhỏ hơn so với file nguồn.

Tóm lại : một script file phải được dịch trước khi có thể thực hiện. Một file đã được dịch không thể dịch ngược trở lại thành file nguồn.

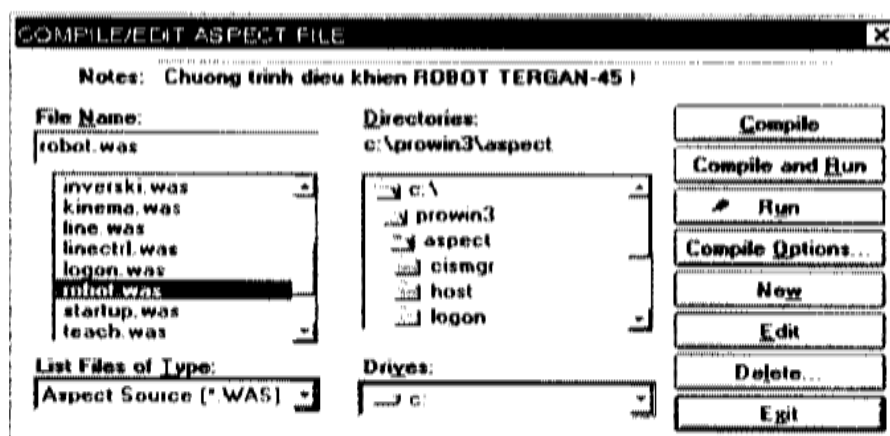
Chúng ta có thể tạo mới và soạn thảo file nguồn (.was) bằng trình ASPECT Editor hay bất kỳ một trình soạn thảo dạng text nào khác, nhưng phải đặt tên tệp có đuôi là . was.

Để tạo mới một file nguồn hoặc thay đổi bổ sung nội dung của một file đã có, từ menu chính của Procomm, chọn Scripts | Compile/Edit... hoặc ấn chuột vào biểu tượng  trên thanh công cụ. Hộp hội thoại dùng để soạn thảo và dịch các script files như hình 5.3.

Muốn tạo một file mới ta chọn nút lệnh New; muốn sửa đổi nội dung một file đã có (tên file đã chọn trước trong mục File name) ta chọn nút lệnh Edit; muốn thoát khỏi của sổ soạn thảo ta chọn nút lệnh Exit.

Khi chọn nút lệnh New hoặc Edit, trên màn hình sẽ xuất hiện của sổ soạn thảo để ta viết hoặc sử đổi chương trình.

Sau khi soạn thảo xong, muốn ghi vào đĩa ta chọn File | Save hoặc File | Save as ... Ta cũng có thể chọn biểu tượng “Ghi và dịch” (Save and Compile) trên thanh công cụ để ghi vào đĩa đồng thời dịch thành file .wax.




Hình 5.3 : Cửa sổ soạn thảo và dịch các script file

Để chạy một Aspect script file có thể thực hiện bằng nhiều cách :

+ Chọn mục Script trên Menu chính, tiếp theo chọn mục Run... Lúc này sẽ xuất hiện hộp hội thoại để chọn file muốn thực hiện.

+ Ấn chuột trên mục Script file của thanh công cụ, sau đó chọn tên file muốn thực hiện.

Nếu một file đã chạy, tên vẫn còn trong mục Script file, muốn chạy lại thì ấn chuột vào biểu tượng  trên thanh công cụ.

+ Có thể chạy một script file từ cửa sổ Compile/Edit ASPECT file (Chọn mục RUN) (hình 5.3).

5.4.2. Kiểu dữ liệu và khai báo biến trong ASPECT :

a) Kiểu dữ liệu : ASPECT cung cấp các kiểu dữ liệu như sau :

integer (kiểu nguyên) : Có giá trị từ -32768 đến 32767.

float (kiểu số thực) : Có giá trị từ 2.22507385072014e-308 đến 1.797693134862315e+308.

long (kiểu nguyên dài) : Có giá trị từ -2147483648 đến 2147483647.

String (kiểu chuỗi) : Có thể chứa từ 0 đến 256 ký tự.

Tất cả tên của các phần tử trong ASPECT, như tên từ lệnh, tên hàm và thủ tục, tên nhãn (label) và biến ... có chiều dài không quá 30 ký tự.

b) Các loại biến : Trong ASPECT có các loại biến sau :

+ Biến hệ thống : Biến hệ thống là các biến “chỉ đọc” (read-only) mà ASPECT và Procomm Plus có thể ấn định các giá trị đặc biệt.

Ví dụ : chúng ta không thể thay đổi giá trị của biến hệ thống \$ROW mà nó luôn luôn bằng vị trí dòng hiện tại của con trỏ trên màn hình, ta chỉ có thể đọc giá trị của nó bất kỳ nơi nào trong chương trình và xử lý khi cần thiết. Biến hệ thống luôn có dấu \$ ở đầu.

+ Biến do người dùng định nghĩa , có hai loại :

- Biến toàn cục (Global variables) : Biến toàn cục có thể được định nghĩa ở bất kỳ nơi nào trong chương trình nhưng phải ở bên ngoài các khối Thủ tục và Hàm. Phổ biến , các biến toàn cục thường được khai báo ở đầu chương trình. Biến toàn cục có thể được tham chiếu đến từ bất cứ hàm hay thủ tục nào của chương trình, Nếu một thủ tục hoặc hàm làm thay đổi giá trị

của một biến toàn cục thì giá trị đó vẫn được duy trì cho đến khi nào có một lệnh khác làm thay đổi giá trị của nó.

- Biến địa phương (Local variables) : Không giống như biến toàn cục, biến địa phương chỉ được tham khảo đến trong phạm vi của thủ tục và hàm mà nó được định nghĩa. Giá trị của nó sẽ bị xoá khi ra khỏi thủ tục và hàm đó. Ta có thể đặt tên các biến địa phương giống nhau trong các thủ tục và hàm khác nhau của chương trình, nhưng điều đó không có nghĩa là giá trị của biến được ghi nhớ giữa các thủ tục hoặc hàm khác nhau.

+ Tham biến (Parameter variables):

Bất cứ thủ tục nào, ngoại trừ chương trình chính (Proc main) đều có thể khai báo (định nghĩa) đến 12 tham biến. Các tham biến tương tự như các biến địa phương, nghĩa là nó chỉ được tham chiếu đến trong phạm vi thủ tục hoặc hàm mà nó được định nghĩa, tuy nhiên khác với biến địa phương, các tham biến nhận các giá trị ban đầu một cách tự động khi các thủ tục hoặc hàm được gọi, các giá trị sử dụng được cung cấp bởi câu lệnh gọi. Các tham biến phải được khai báo ở đầu mỗi thủ tục hoặc hàm, trước bất cứ lệnh nào hoặc các biến địa phương. Một tham biến được khai báo mà chúng sẽ được gọi bởi các thủ tục hoặc hàm.

c) Khai báo (định nghĩa) các biến : Tất cả các loại biến dùng trong chương trình phải được khai báo (định nghĩa) trước. Nếu các biến có cùng kiểu dữ liệu, ta có thể khai báo trên một dòng cách nhau bởi dấu phẩy (,).

Ví dụ :

Integer sokhop, Tong, i = 1

Float Goc

Integer A[4][4]

Trong ví dụ trên ta khai báo các biến : sokhop, Tong, i là các biến nguyên, trong đó biến i được gán giá trị ban đầu là 1. Goc là biến thực. A là biến mảng (array) có kích thước 4x4 , các phần tử của mảng kiểu nguyên.

Cách khai báo tham biến trong thủ tục và hàm như sau :

param (kiểu dữ liệu) (tên) [, tên] . . .

Ví dụ : param Integer X, Y, Z

Chương trình ví dụ :

; Ví dụ về khai báo biến.

Proc main ; Chương trình chính.

integer A,B,C ; Khai báo 3 biến nguyên.

integer Tong ; Tổng của 3 số (biến nguyên).

A=2, B=4, C=8 ; Gán giá trị cho các biến.

Tong = Sum(A,B,C) ; Gọi hàm Sum để cộng các số.

Usermsg " Tong = %d." Tong ; Cho hiện tổng của các số lên màn hình

Endproc ; Hết chương trình chính.

Func Sum : Integer ; Định nghĩa hàm Sum để tính tổng.

Param integer X, Y, Z ; Khai báo các tham biến kiểu nguyên.

integer Tong ; Khai báo biến Tong (biến địa phương).

Tong= X+Y+Z ; Tổng của 3 số.

return Tong ; Trả về giá trị của tổng của 3 số.

Endfunc ; hết phần định nghĩa hàm

(Ghi chú : dấu ";" dùng để ghi chú trong chương trình, các nội dung sau dấu ";" không được dịch).

5.4.3. Cấu trúc của chương trình :

Cấu trúc chương trình của một ASPECT script file gần giống như một file viết bằng ngôn ngữ Pascal, nghĩa là có một chương trình chính và các thủ tục hoặc hàm khác. Chỗ khác nhau cơ bản là chương trình chính được viết trước, chương trình chính có thể gọi đến các hàm hoặc thủ tục được định nghĩa sau đó.

Trong chương trình chính không được khai báo các tham biến. Khi thực hiện chương trình, nó sẽ lần lượt thực hiện các lệnh từ dòng đầu tiên đến hết chương trình.

Khi kết thúc một hàm hoặc thủ tục được gọi, nó tự động trả về dòng lệnh tiếp theo. Cấu trúc chung của một chương trình như sau :

; Dòng đầu tiên dùng ghi chú về nội dung chương trình, dòng này sẽ thể hiện trong

; hộp hội thoại Compile/Edit để người sử dụng dễ nhận biết về nội dung của chương

; trình.

Proc main ; bắt đầu chương trình chính

(Khai báo biến)

(các câu lệnh thể hiện nội dung chương trình)

.....

Endproc ; hết chương trình chính.

Proc (tên thủ tục) ; Bắt đầu một thủ tục

(khai báo các tham biến nếu có)

(khai báo các biến địa phương)

(các câu lệnh thể hiện nội dung thủ tục)

.....

Endproc ; hết một thủ tục

Func (tên hàm) ; Bắt đầu một hàm

(khai báo các tham biến nếu có)

(khai báo các biến địa phương)

(các câu lệnh thể hiện nội dung của hàm)

.....

return (biến) ; trả giá trị của biến về thủ tục gọi

Endproc ; kết thúc hàm

5.4.4. Một số phép tính dùng trong ASPECT :

ASPECT sử dụng nhiều phép tính số học và logic khác nhau, dưới đây giới thiệu một số phép tính hay dùng :

+, -, *, / Phép toán cộng, trừ, nhân, chia.

>, <, >=, <= Lớn hơn, nhỏ hơn, lớn hơn hoặc bằng, nhỏ hơn hoặc bằng.

!= Khác với

! NOT

&& AND

|| OR

++, -- Tăng hoặc giảm một đơn vị.

?: Thực hiện một điều kiện

.v.v...

Ví dụ 1 : Cho A=2, B=4

A+B-- = 6 : A được cộng với B trước, và rồi B giảm đi 1 (B=3).

A+ --B = 5 : Trước tiên B giảm đi 1, sau đó cộng A với B.

Ví dụ 2 :

Proc main

integer A,B,C,D

integer Tong

A=2, B=4

C=A+B

Tong = A+ --B

D=(tong < C) ? tong : C ; nếu Tong < C thì D=Tong, nếu sai D=C

Usermsg " D = %d , C = %d" D,C

Endproc

Kết quả D = 5 và C = 6.

5.4.5. Một số từ lệnh trong ASPECT hay dùng khi điều khiển robot:

Ngôn ngữ ASPECT có hơn 600 từ lệnh, dùng với nhiều mục đích khác nhau.

Phần này chỉ giới thiệu sơ lược một số lệnh hay dùng khi lập trình điều khiển robot.

Người đọc có thể sử dụng mục Help trên menu cửa sổ soạn thảo để biết thêm chi tiết.

* Các lệnh căn bản :

call :

Gọi một thủ tục hoặc hàm từ chương trình chính hoặc từ một thủ tục khác.

Cú pháp :

Khi gọi một hàm :

call <tên> [WITH <danh sách tham biến>] [INTO <biến>]

Khi gọi một thủ tục :

call <tên> [WITH <danh sách tham biến>]

tên : tên thủ tục hoặc hàm được gọi.

Danh sách tham biến : Tên các tham biến trong thủ tục hoặc hàm.

INTO <biến> : Chỉ dùng khi gọi một hàm, biến sẽ chứa giá trị trả lại của hàm.

case/endcase :

Câu lệnh lựa chọn, dùng với từ lệnh Switch.

Cú pháp :

switch <biến> (string | integer | long)

case <giá trị so sánh> (string | integer | long)

...

[exitswitch] ; thoát khỏi khối lệnh switch không

... điều kiện.

[endcase]

[default] ; thực hiện khi các tr-ờng hợp so sánh

... đều không đúng.

endcase

endswitch

Ví dụ :

proc main

integer Alpha = 2 ; gán giá trị ban đầu cho biến Alpha=2.

switch Alpha ; tìm giá trị của biến số

case 0 ; Tr-ờng hợp biến có giá trị bằng 0.

usermsg "Alpha = 0" ; Xuất kết quả trên cửa sổ mơn hình.

Endcase ; Hết tr-ờng hợp so sánh thứ nhất.

case 1 ; tương tự như trên . . .

usermsg "Alpha = 1"

endcase

case 2

usermsg "Alpha = 2"

endcase

endswitch ; luôn đi kèm với switch để kết thúc khối lệnh

switch.

endproc

if / endif : Câu lệnh điều kiện.

Cú pháp :

if <điều kiện 1>

...

[elseif <điều kiện 2>

...

[else]

...

endif ; kết thúc khối lệnh if.

(Lệnh này gần giống nh- lệnh if trong Pascal, không có từ then).

while/endwhile :

Lặp lại một số câu lệnh cho đến khi điều kiện kiểm tra lờ sai.

Ví dụ :

proc main

integer SoLanLap = 0 ; Biến nguyên dùng để đếm số lần

lặp

while (SoLanLap++) < 3 ; Mỗi lần lặp biến tăng giá trị thêm một

endwhile ; Kết thúc khối lệnh while.

usermsg "Toi da lap %d lan" SoLanLap

endproc

for/endifor : Câu lệnh lặp theo một số lần nhất định

Cú pháp :

for <biến đếm>=<giá trị ban đầu> UPTO | DOWNTO <giá trịcuối>

[BY <b-ớc>]

.

[exitfor] ; Chuyển điều khiển thoát khỏi câu lệnh lặp for

.

; đến dòng lệnh sau endifor

endifor

Return :

Thoát khỏi thủ tục hoặc hàm hiện tại, tiếp tục ở câu lệnh tiếp theo của thủ tục đó gọi.

* Câu lệnh khác :

transmit : Gửi một dòng ký tự (lệnh) đến cổng đang hoạt động.

Ví dụ :

```
proc main
transmit "B-250~C-200~F-240~~P+200" ; Chuyển lệnh điều khiển
robot TG-45
```

```
endproc
```

Pause : Tạm dừng thực hiện chương trình trong một số giây qui định.

Cú pháp :

```
pause <số giây | FOREVER>
```

Ví dụ :

Pause 5 : tạm dừng thực hiện chương trình 5 giây

Pause Forever : Dừng với thời gian khựng xác định. Lệnh Pause có thể được huỷ bỏ khi ấn Ctl-Break.

Ký tự ~ thay cho lệnh pause với giá trị dừng bằng 0,5 giây.

Ví dụ : Transmit "B+200~~E-100"

Sau khi truyền lệnh B+200 sẽ tạm dừng 1 giây (2 ký tự ~) mới truyền tiếp lệnh E-100.

chdir : Thay đổi đường dẫn đến một ổ đĩa hoặc thư mục khác.

Cú pháp : chdir <"Tên đường dẫn">

Ví dụ :

```
Chdir "C:\procom3\Robot"
```

copyfile : Copy một file theo đường dẫn hoặc với một tên khác.

Cú pháp : copyfile <"file nguồn"> <"file đích">

Ví dụ:

```
copy "C:\Procom3\ aspect\ robot.was" "C:\ tam\ robot1.txt"
```

delfile : Xóa một file theo chỉ định.

Cú pháp : delfile <"tên file">

mkdir : Tạo một thư mục mới.

Cú pháp : mkdir <"tên thư mục">

rmdir : Xóa một thư mục (trống)

Cú pháp : rmdir <"tên thư mục">

rename : Đổi tên một file.

Cú pháp : rename <"tên file cũ"> <"tên file mới">

Fopen : Mở một file để đọc hoặc ghi.

Cú pháp : fopen <số hiệu file> <"tên file"> READ | WRITE |

READWRITE | CREATE | APPEND | READAPPEND

Các tùy chọn : READ : chỉ đọc; READWRITE : có thể đọc và ghi;

CREATE : Tạo mới; APPEND : Ghi tiếp vào cuối file;

READAPPEND : Có thể đọc và ghi tiếp vào cuối file.

Fclose : Đóng một file đã mở.

Cú pháp : Fclose <số hiệu file>

fputs : Ghi một chuỗi ký tự lên file.

Cú pháp : fputs <số hiệu file> <"chuỗi ký tự">

Ví dụ :

```
proc main
```

```
string Fname = "Vidu.txt" ; Tên file được mở.
```

```
if fopen 0 Fname CREATE ; Tạo mới và mở một file có tên
```

```
"Vidu.txt"
```

```
fputs 0 "Day la file moi duoc mo !" ; Ghi một chuỗi lên file.
```

```
fclose 0 ; Đóng file đã được tạo mới và mở.
```

```
else
```

```
errmsg "Couldn't open file ``%s``." Fname
```

```
endif
```

```
endproc
```

feof : Kiểm tra điều kiện đã ở cuối một file.

Cú pháp : feof <số hiệu file> [biến nguyên]

[biến nguyên] : có giá trị 0 nếu chưa kết thúc file, bằng 1 nếu đã kết thúc file.

Fgets : Đọc một dòng ký tự từ một tệp đã mở ghi vào một biến.

Cú pháp : fgets <số hiệu file> <tên biến kiểu string>

Ví dụ :

```
proc main
```

```
string Fname = "Vidu.txt" ; Tên file cần đọc
```

```
string chuoi ; Biến chuoi nhận giá trị đọc từ file.
```

```
if fopen 0 Fname READ ; Mở file chỉ để đọc (số hiệu file id=0).
```

```
while not feof 0 ; Lặp lại khi chưa kết thúc file.
```

```
fgets 0 FInput ; Đọc một dòng từ file
```

```
usermsg FInput ; Thể hiện dòng đã đọc
```

```
endwhile
```

```
fclose 0 ; Đóng file
```

```
else
```

```
errmsg "Can't open ``%s`` for input." Fname ; báo lỗi nếu file
```

không tồn tại.

```
endif
```

```
endproc
```

usesmsg : thể hiện một dòng thông báo hay kết quả trên cửa sổ.

Cú pháp : usermsg <:dòng thông báo> [biến1, ...]

Xem các ví dụ trên.

termwrites : Viết một dòng ký tự lên của sổ nhập xuất dữ liệu.

Cú pháp : termwrites <biến hoặc "dòng ký tự">

Run : Thực hiện một chương trình bên ngoài (đuôi COM, EXE hoặc BAT).

Cú pháp : run <"tên chương trình">

Ví dụ :


```
proc main
string Prog = "C:\ windows\ pbrush.exe" ; Chương trình cần thực
hiện.
```

```
run Prog ; Thực hiện chương trình PaintBrush của Windows.
```

```
Endproc
```

Người các từ lệnh đã giới thiệu trên, còn có rất nhiều lệnh khác..., người sử dụng có thể tham khảo trực tiếp trong mục HELP của cử sở soạn thảo khi cần thiết.

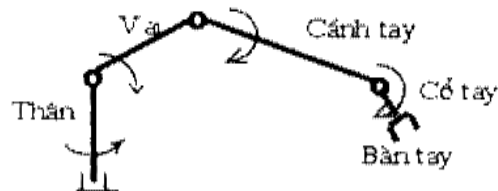
Ngôn ngữ ASPECT không có sẵn các hàm toán học như sin, cos, ... nên khi muốn thực hiện các tính toán phức tạp ta phải dùng các phần mềm khác.

5.5. Lập trình điều khiển robot TERGAN - 45 :

Như đã nói trên, để điều khiển robot TERGAN-45 ta có thể dùng phần mềm Procomm Plus for Windows điều khiển trực tiếp hoặc viết các chương trình bằng ngôn ngữ ASPECT.

5.5.1. Giới thiệu robot TERGAN 45 (TG-45):

TERGAN 45 là một loại robot dùng để dạy học do Pháp sản xuất. Đây là loại robot toàn khớp quay, có 4 bậc tự do. Đi kèm với robot gồm có một bộ nguồn và một môđun điều khiển. Môđun điều khiển cho phép điều khiển robot trên các Terminal hoặc máy tính có giao diện kiểu RS-232. Cấu hình của robot như hình 5.2 :



Hình 5.3 : Sơ đồ động Robot TG-45

Các khớp quay của robot được dẫn động bằng các động cơ điện một chiều có gắn các potentiometer, ngoài ra để đóng mở bàn tay của robot người ta dùng truyền động vít-me có gắn cử hành trình, vận tốc đóng mở các ngón tay có thể điều chỉnh được.

Các góc quay giới hạn của các khâu trên robot là :

- + Chuyển động của thân 261° .
- + Chuyển động của vai 85° .
- + Chuyển động của cánh tay 249° .
- + Chuyển động của cổ tay 180° .

Tốc độ truyền thông tin qua modul điều khiển từ 50 đến 9600 bauds với bộ vi xử lý 8 bits, Stop bit là 1 hoặc 2. Điện áp nguồn cung cấp là 110V/220V, 50HZ. Điện áp điều khiển $\pm 12V$.

Trên modul điều khiển có thêm các đầu vào và ra để giao diện với các thiết bị khác (như các cảm biến, điều khiển bằng tay nhỏ, ...). Modul điều khiển robot TG-45 được thiết kế giao diện với máy tính bằng các lệnh cơ bản sau :

- B \pm XXX : Điều khiển thân (Base),
- E \pm XXX : Điều khiển vai (épaule),
- C \pm XXX: Điều khiển cánh tay (Coude),
- F \pm XXX : Điều khiển cổ tay (Poignet),
- P \pm XXX : Đóng mở bàn tay (Pince),
- S \pm XXX : Điều khiển các tín hiệu ra,
- I \pm XXX : Điều khiển các tín hiệu vào.

Chiều dài của các lệnh điều khiển là 5 ký tự mã ASCII. Ký hiệu XXX biểu diễn các chữ số từ 000 đến 511.

Ví dụ :

Lệnh B-200 sẽ điều khiển thân robot quay sang phải một góc :

$$\theta_1 = (261^\circ/2) \times 200 / 511 \approx 51^\circ 04'$$

Lệnh C+200 sẽ điều khiển cánh tay robot quay lên phía trên một góc

:

$$\theta_3 = (249^\circ/2) \times 200 / 511 \approx 48^\circ 43' \text{ (so với vai).}$$

Lệnh P+200 sẽ đóng bàn tay (dùng khi muốn nắm một vật) , vận tốc đóng mở thay đổi được theo giá trị từ 001 đến 511. Ví dụ P+100 sẽ đóng chậm hơn P+200.

Các lệnh được chuyển đến từ máy tính sẽ được môđun điều khiển xử lý sau đó trả lại các thông báo thực hiện (message) trên màn hình.

5.5.2. Điều khiển trực tiếp robot TG-45 nhờ phần mềm Procomm :

ở chế độ TERMINAL của Procomm Plus ta có thể điều khiển trực tiếp robot Tergan 45 bằng cách gõ trực tiếp các lệnh làm quay các khớp của robot, ví dụ :

B+200

C-250

E-100

F-250

P+200

Ta có thể ghi lại các lệnh vừa nhập vào một file để thực hiện lại sau này, nếu trước khi nhập các lệnh ta chọn mục START RECORDER trên menu hoặc Icon tương ứng.

5.5.3. Viết chương trình điều khiển robot TERGAN-45 :

Ta có thể điều khiển robot Tergan-45 bằng cách viết các chương trình bằng ngôn ngữ ASPECT. Một chương trình ví dụ đơn giản như sau :

```
proc main
```

```
transmit "E-100~B-250~F-180~C-200~B-300~~~~P+150~~~~"
```

```
transmit "E+000~C-150~B+300~~~C-180~~~~~P-200~~~~"
```

```
transmit "E+200~B-400~E-000~C-300~F-080~B-450~P+150~"
transmit "C-260~E+100~B+300~P-200~"
transmit "F+200~C-130~B-350~F-300~E-180~B-400~P+100~"
transmit "E+200~B+300~E-100~F-230~P-200~"
transmit "C-000~F-000~E-000~B-000~P-200~"
pause 50
clear
termwrites "Da thuc hien xong, xin cho lenh !"
endproc
```

Khi soạn thảo xong chương trình ta phải đặt tên và ghi vào đĩa, ví dụ tên chương trình là DEMO.WAS. Sau đó ta phải dịch chương trình để tạo ra file DEMO.WAX lúc đó mới có thể chạy được trong Procomm Plus.

Tuy nhiên, như đã giới thiệu ở trên, môđun điều khiển robot TG-45 chỉ có các lệnh đơn giản để điều khiển các mô-tơ dẫn động các khớp quay. Nếu chỉ điều khiển robot bằng các lệnh đơn thì không thể mở rộng khả năng làm việc của robot được, hơn nữa việc lập trình cũng mất nhiều công sức vì khó xác định được các tọa độ mà ta yêu cầu bàn tay robot phải đạt tới. Do đó việc lập trình điều khiển robot phải tạo ra các chức năng khác khi điều khiển robot như :

- 1) Chương trình có thể giúp người sử dụng dạy robot học mà robot có thể lặp lại các chuyển động đã được dạy-học một cách chính xác.
- 2) Thiết kế điều khiển động học thuận : nghĩa là chương trình cho phép người sử dụng điều khiển robot theo giá trị các góc quay của khớp (tính bằng độ) khi xác định trước một cấu hình nào đó của robot.
- 3) Thiết kế điều khiển động học ngược : nghĩa là người sử dụng chương trình có thể điều khiển robot theo các tọa độ vị trí và hướng của bàn

tay đã được xác định trước. Khi ta nhập các giá trị về toạ độ và hướng của bàn tay thì chương trình tự động tính toán các góc quay của các khớp để robot chuyển động đến vị trí yêu cầu với hướng đã được xác định.

4) Thiết kế các tiện ích khác như : điều khiển theo đường, tạo các meta keys, tạo ra các trợ giúp cho người sử dụng...

Các nội dung 2 và 3 cần phải thiết lập hệ phương trình động học của robot TERGAN-45 và giải hệ phương trình động học đó. Phần tính toán có thể viết bằng ngôn ngữ Pascal hoặc C++ mà nó được gọi từ chương trình điều khiển (dùng lệnh RUN), chương trình điều khiển xử lý kết quả tính toán qua các file trung gian dạng text.

Phần mềm Procomm cung cấp nhiều tiện ích để ta có thể thiết kế chương trình kiểu trực quan (Visual), giúp cho việc viết chương trình và thao tác trong quá trình sử dụng được dễ dàng, thuận tiện hơn.

5.6. Kết luận :

Trong chương này chỉ giới thiệu một cách tổng quát về các phương pháp lập trình điều khiển robot. Khó có thể đi sâu, cụ thể vào một ngôn ngữ nào vì nó phụ thuộc rất nhiều vào loại robot được sử dụng. Phần ngôn ngữ ASPECT trong phần mềm Procomm được nghiên cứu ở trên là một ví dụ về ứng dụng các phần mềm dùng cho nhiều mục đích để điều khiển robot. Trong thực tế, tùy nhiệm vụ cụ thể của mỗi robot, phối hợp với hoạt động chung của các máy móc thiết bị khác mà ta thiết kế các chương trình cụ thể để robot hoạt động theo những mục đích mong muốn.

Bài tập chương V :

Bài 1 : Hãy viết một Function của hàm $\arctg_2(y,x)$ bằng ngôn ngữ Turbo Pascal.

Bài 2 : Viết một chương trình (ngôn ngữ tùy ý) để nhập các thông số DH và tự động xác lập các ma trận A_n (Biểu hiện kết quả trên màn hình và ghi vào một file dạng text).

Bài 3 : Viết một chương trình bằng Turbo Pascal để tính toán động học ngược (Xác định các góc quay) của robot TERGAN-45. Dữ liệu nhập từ bàn phím. Ghi kết quả vào một file dạng text.

Bài 4 : Viết một chương trình điều khiển robot TERGAN-45 bằng ngôn ngữ ASPECT để robot có cấu hình như sau :

$$\theta_1 = +30^0; \theta_2 = -10^0; \theta_3 = -30^0; \theta_4 = -25^0.$$

Bàn tay robot nắm lại sau khi di chuyển đến vị trí yêu cầu.

Bài 5 : Viết một chương trình bằng ngôn ngữ ASPECT, gọi chương trình tính động học ngược viết bằng Turbo Pascal (như bài 3), xử lý kết quả tính toán để điều khiển robot TERGAN-45 theo tọa độ vị trí và hướng của bàn tay.

CHƯƠNG 6

MÔ PHỎNG ROBOT TRÊN MÁY TÍNH

(Robot Simulation)

(Phân thực hành trên máy tính)

6.1. Kỹ thuật mô phỏng robot :

Mô phỏng là một kỹ thuật hiện đại, được áp dụng trong nhiều lĩnh vực nghiên cứu và sản xuất.

Khi nghiên cứu về điều khiển robot, ta có thể thực hiện điều khiển trực tiếp robot hoặc điều khiển mô phỏng. Điều khiển mô phỏng là dùng các mô hình tính toán động học và động lực học của robot kết hợp với các phương pháp đồ họa trên máy vi tính để mô tả về kết cấu và hoạt động của cánh tay robot.

Nghiên cứu về mô phỏng hoạt động của robot trên máy tính giúp cho các nhà thiết kế nhanh chóng lựa chọn được phương án hình - động học của robot, có thể kiểm tra khả năng hoạt động của robot trên mẫu hình, kiểm tra sự phối hợp của robot với các thiết bị khác trong dây chuyền. Điều này rất có ý nghĩa trong quá trình thiết kế chế tạo robot mới hoặc bố trí dây chuyền sản xuất.

Qua mô phỏng người thiết kế có thể đánh giá tương đối đầy đủ khả năng làm việc của phương án thiết kế mà không cần chế thử. Nó cũng được xem là phương tiện đối thoại, hiệu chỉnh thiết kế theo yêu cầu đa dạng của người sử dụng.

Phương pháp lập trình mô phỏng cũng giúp người thiết kế chọn được quỹ đạo công nghệ hợp lý của robot trong quá trình làm việc với một đối tượng cụ thể hay phối hợp với các thiết bị khác trong một công đoạn sản xuất được tự động hoá.

Hiện nay có nhiều phần mềm công nghiệp và các phần mềm nghiên cứu khác nhau để mô phỏng robot, phạm vi ứng dụng và giá thành của chúng cũng khác nhau. ở đây chúng ta nghiên cứu phương pháp mô phỏng robot dùng phần mềm EASY-ROB.

6.2. Giới thiệu phần mềm EASY-ROB :

EASY-ROB là công cụ mô phỏng robot sử dụng đồ họa trong không gian 3 chiều (3D) và các hình ảnh có thể hoạt động được. Một hệ thống 3D-CAD đơn giản cho phép tạo ra các khối hình học cơ bản như khối trụ, khối cầu, khối chữ nhật, khối tam giác ... để vẽ kết cấu của robot. Trong EASY-ROB chúng ta có thể dùng chuột để quay hoặc tịnh tiến robot đến một tọa độ tùy ý. EASY-ROB cũng có các chức năng phóng to, thu nhỏ đối tượng vẽ như nhiều phần mềm thiết kế khác... Chương trình cho phép thiết kế các robot đến 12 bậc tự do. Chuyển động của Robot có thể được điều khiển theo các biến khớp hoặc các tọa độ Đề-cát. Chúng ta cũng có thể mô tả động học của robot theo kiểu DH hoặc trong hệ tọa độ toạ cục (Universal Coordinates). Easy-Rob đã có sẵn các trình điều khiển động học thuận và ngược của các cấu hình robot thông dụng, khi thiết kế ta chỉ cần khai báo kiểu động học thích hợp. Trong trường hợp robot có kết cấu đặc biệt hoặc có các khâu bị động gắn với các chuyển động của các khớp thì cần phải giải bài toán động học ngược hoặc xác định hàm toán học mô tả sự phụ thuộc của khâu bị động đối với khớp quay, viết chương trình xác định sự phụ thuộc đó bằng ngôn ngữ C và sau đó dùng tập tin MAKE.EXE trong C để dịch thành tập tin thư viện liên kết động er_kin.dll (Easy-Rob kinematic Dynamic link library), khi chạy chương trình, EASY-ROB sẽ liên kết với tập tin này và thực hiện kiểu động học đã được khai báo trong chương trình điều khiển.

Easy-ROB có một số các lệnh điều khiển riêng, Chương trình được viết theo kiểu xử lý tuần tự, tập tin dạng Text, có thể soạn thảo chương

trình trong bất kỳ trình soạn thảo nào. Các công cụ gắn trên khâu chấp hành cuối có thể thay đổi được. Chúng ta có thể viết một chương trình chuyển động cho một robot theo một quỹ đạo mong muốn, có thể kiểm tra khả năng vươn tới của cánh tay, xác định vùng làm việc của robot . . . Robot mô phỏng có thể cầm nắm hoặc thả các đối tượng làm việc. Các chuyển động của robot có thể ghi vào một tập tin và có thể thực hiện lại.

Phần mềm cho phép ta xem được các hệ tọa độ đã gắn trên các khâu của robot, xem được quỹ đạo chuyển động của điểm cuối công cụ gắn trên khâu chấp hành cuối. Phần mềm còn có nhiều tiện ích khác như : cho phép ta lập trình điều khiển robot bằng phương pháp dạy học, thiết kế các đối tượng làm việc của robot, có các cửa sổ về tọa độ và giá trị góc quay của các khớp tại từng thời điểm khi robot hoạt động...

Việc sử dụng phần mềm EASY-ROB để mô phỏng robot giúp chúng ta hai khả năng nghiên cứu :

a/ Mô phỏng lại một robot đã có và các đối tượng làm việc của nó. Đánh giá khả năng làm việc và mức độ linh hoạt của robot, xác định các thông số điều khiển, quỹ đạo chuyển động để dùng trong điều khiển thực.

b/ Nghiên cứu thiết kế động học, các kích thước và kết cấu của robot trên máy tính để có thể chọn được phương án động học tốt nhất, đảm bảo cho robot hoàn thành các nhiệm vụ yêu cầu.

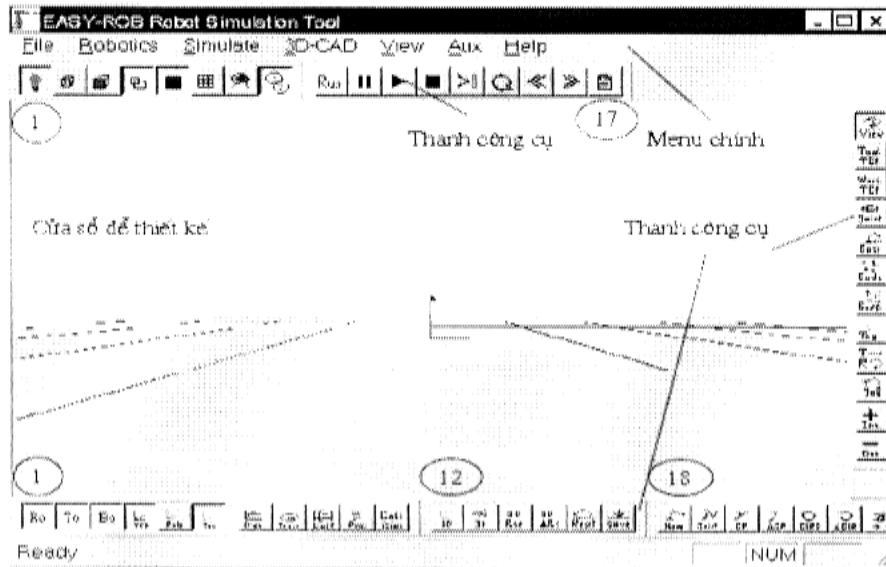
6.3. Tìm hiểu màn hình EASY-ROB :

a- Menu chính :

Menu chính của phần mềm EASY-ROB cung cấp các nội dung hoạt động khác nhau của phần mềm. Bước đầu làm quen, ta cần quan tâm các Menu sau :

Menu FILE : Xử lý các tác vụ trên File. Trong Easy-Rob có nhiều loại file được qui định bởi phần mở rộng (đuôi của File), ví dụ :

File có dạng *.Cel : (Cellfile) để mô tả kết cấu Robot, công cụ làm việc và đối tượng làm việc của robot. Đây là một File tổng hợp, bao gồm cả chương trình dùng để điều khiển robot.



Hình 6.1 : Màn hình EASY-ROB.

File có dạng *.Rob : (Robotfile) để mô tả riêng kết cấu của một robot.

File có dạng *.Bod : (Bodyfile) để mô tả các đối tượng làm việc của robot.

File có dạng *.Tol : (Toolfile) để mô tả công cụ gắn trên khâu chấp hành cuối của robot.

File có dạng *.Vie : (Viewfile) để xác định góc nhìn trong không gian.

File có dạng *.igp : (Igrid Partfile) lưu trữ một bộ phận kết cấu.

File có dạng *.Prg : (Programm) Chương trình điều khiển.

v.v....

Menu Robotics : Dùng để nhập các thông số DH, xác định vị trí của dụng cụ, xác định vị trí robot và các thông số khác.

Menu 3D-CAD : Cung cấp các công cụ để vẽ kết cấu robot trong không gian 3 chiều (3D) cũng như để thiết kế các công cụ, các đối tượng làm việc. Để vẽ được kết cấu của robot, dựa vào các khối hình học đơn giản ta có thể lắp ghép chúng lại để tạo nên các hình dáng khác nhau của robot.

.....

b- Các thanh công cụ :

Các nút trên thanh công cụ dùng để thực hiện các thao tác như của menu chính (mở không cần vào menu). Sử dụng các nút trên thanh công cụ cho phép ta thao tác nhanh hơn là phải vào menu chính. Chức năng của các nút chính trên thanh công cụ như sau :

Thanh công cụ nằm ngang phía trên, tính từ trái sang phải :

1. Bật tắt chế độ chiếu sáng các đối tượng vẽ.
2. Chuyển tất cả các đối tượng sang dạng lưới.
3. Chuyển đối tượng dạng trụ / khối phức tạp.
4. Thể hiện/không thể hiện sôn.
5. Thể hiện sôn ở dạng lưới.
6. Reset vị trí robot trên màn hình.
7. Chuyển đổi cửa sổ khi mở Cellfile hoặc igip partfile (kết hợp với nút 7).
8. Chạy chương trình.
9. Tạm dừng chương trình.
10. Tiếp tục chạy chương trình.
11. Kết thúc chương trình.
12. Chạy chương trình theo từng bước.
13. Lặp lại chương trình sau khi kết thúc.
14. 15. Giảm và tăng tốc độ điều khiển.
16. Đánh giá sai số và xem các giá trị động học.

Thanh công cụ nằm ngang phía dưới, tính từ trái sang phải :

1. Thấy hoặc không thấy kết cấu robot.
 2. Thấy hoặc không thấy dụng cụ.
 3. Thấy hoặc không thấy các đối tượng làm việc.
 4. Thể hiện/không thể hiện hệ tọa độ gắn với dụng cụ .
 5. Thể hiện/không thể hiện hệ tọa độ gắn trên các khâu của robot.
 6. Thể hiện vị trí điều khiển.
 7. Mô phỏng động lực học.
 8. Thể hiện quỹ đạo chuyển động.
 9. Sử dụng các giới hạn của khớp.
 10. Soạn thảo chương trình và dạy học.
 12. Thể hiện hoặc không thể hiện Hệ tọa độ gắn trên đối tượng hiện thời.
 13. Chuyển đến đối tượng tiếp theo (khi thiết kế).
 14. Xác định vị trí tuyệt đối của đối tượng hiện tại.
 15. Xác định vị trí tương đối của đối tượng hiện tại.
 16. Reset vị trí của đối tượng hiện tại.
 17. Ghi lại vị trí của đối tượng sau khi điều chỉnh.
 18. Đ-a robot về vị trí dừng (Home position).
 19. Điều khiển robot theo khớp quay.
- Thanh công cụ thả đứng (Thao tác bằng chuột)[☞], tính từ trên xuống :
1. Dùng chuột để view, zoom và Pan.
 - 2.3. Điều khiển hướng của khâu chấp hành cuối bằng chuột.
 4. Điều khiển các khớp 1,2,3 (Dùng các phím chuột).
 5. Di chuyển thân robot. (hệ tọa độ cơ sở)
 6. Di chuyển các đối tượng (body) bằng chuột.
 7. Di chuyển tất cả các đối tượng bằng chuột.

9. Chuyển đổi chuyển động là quay hoặc tịnh tiến (Dùng khi hiệu chỉnh đối tượng vẽ).

11.12. Tăng giảm tốc độ điều khiển bằng chuột.

6.4. Thao tác chuột :

Easy-Rob cho phép dùng chuột với nhiều chức năng như :

Khi nút lệnh số 1 của thanh công cụ thẳng đứng được chọn :

zoom (Phóng to, thu nhỏ) : ấn nút chuột phải, rê chuột lên xuống theo phương thẳng đứng của màn hình.

Pan (thay đổi vị trí của đối tượng so với khung màn hình) : ấn đồng thời hai nút chuột phải và trái, rê chuột trên màn hình.

Rotate (quay robot để nhìn ở các góc độ khác nhau) : ấn chuột trái, rê chuột.

Khi nút lệnh số 4 của thanh công cụ thẳng đứng được chọn :

Quay khớp 1: ấn nút chuột phải, rê chuột (nếu là khớp tịnh tiến sẽ làm khâu chuyển động tịnh tiến).

Quay khớp 2: ấn đồng thời 2 nút chuột phải và trái, rê chuột.

Quay khớp 3: ấn nút chuột trái, rê chuột.

6.5. Gắn hệ tọa độ :

Muốn xác định hệ tọa độ của robot trước hết phải thực hiện bằng tay các công việc sau:

Vẽ sơ đồ động robot ở vị trí dừng, gắn hệ tọa độ của các khâu lên hình vẽ trên giấy, xác định các thông số DH.

Các bước tiếp theo :

1- Bật nút lệnh số 5 trên menu ngang, dưới.

2- Vào menu chính : FILE -> LOAD -> ROBOTFILE chọn DHTempl -> OPEN.

3- Vào menu chính : ROBOTICS -> ROBOTMOTION + KINEMATICS -> KINEMATICS DATA.

4- Chọn Active Joint -> Ok -> Active Joint (1) RZ (hoặc chọn TZ nếu là khớp tịnh tiến) -> Ok -> Nhập các thông số DH của khâu thứ nhất.

5- Chọn Quit -> Ok.

Vào lại bước 4 -> Number Active Joint(1) -> Ok -> ấn đúp chuột vào vệt xanh hoặc đ-a con trỏ vào phần nhập dữ liệu (text box) ấn 2 (Bây giờ số khâu động là 2), nhập các thông số DH cho khâu số 2 ...

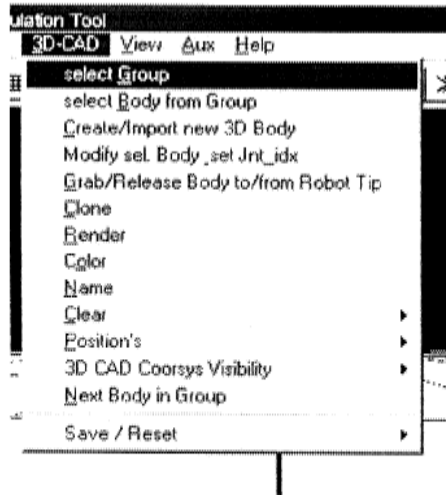
Làm tương tự cho đến khi đủ số khớp yêu cầu.

Ta có thể kiểm tra các số liệu đã nhập bằng cách kích chuột vào menu : ROBOTICS -> ROBOTMOTION + KINEMATICS -> KINEMATICS DATA-> KINEMATIC INFORMATION để xem lại số khâu, khớp và các thông số DH. Nếu vào dữ liệu sai ta có thể hiệu chỉnh lại.

Để thể hiện hệ toạ độ của robot trên màn hình (Hệ toạ độ màu vàng), nhỡ kích chuột vào nút số 5 của thanh công cụ nằm ngang phía dưới.

6.6. Vẽ hình dáng robot :

Sau khi hoàn thành việc gán hệ toạ độ của robot, bước tiếp theo là vẽ hình dáng của nó. Hình dáng của robot có thể được mô phỏng giống như robot thực nhờ công cụ 3D CAD của EasyiRob. Menu 3D-CAD cho phép tạo ra các khối hình học cơ bản như khối trụ, khối cầu, khối chữ nhật, khối tam giác ... Sự phối hợp hợp lý về kích thước và vị trí của các khối hình học này cho phép thể hiện được các kết cấu khác nhau của robot.



Hình 6.2 : Menu 3D-CAD

Các Menu kéo xuống của Menu 3D-CAD như hình 6.2, một số các chức năng chính như sau :

+ Select group : Chọn nhóm đối tượng để thiết kế : 1/Robot group, 2/Tool group hay 3/ Body group.

+ Select body from group : Chọn các bộ phận của robot đã vẽ (theo tên đặt trước) của nhóm chọn hiện hành.

+ Create/Import new 3D body : Tạo mới hoặc nhập một bộ phận đã có sẵn. Cần nhập các thông số cần thiết để tạo ra đối tượng mong muốn.

+ Modify sel. Body _set Jnt_idx : Hiệu chỉnh các thuộc tính của bộ phận hiện hành.

+ Clone : Copy bộ phận đang vẽ thành nhiều hình

+ Render : Biểu hiện đối tượng ở dạng lưới, dạng hộp, . . .

+ Color : Thay đổi màu sắc. + Name : Thay đổi tên bộ phận đang vẽ.

+ Clear : Xoá đối tượng (bộ phận) hiện hành.

+ Position's : Thay đổi vị trí của đối tượng (bộ phận) hiện hành.

+ 3D CAD Coorsys Visibility : Cho hiện hoặc ẩn hệ tọa độ của đối tượng vẽ.

+ Next Body in group : Chọn đối tượng vẽ tiếp theo.

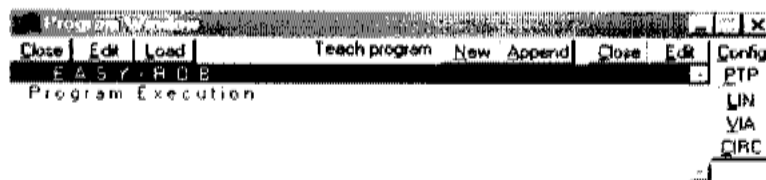
Dùng menu 3D CAD ta lần lượt vẽ tất cả các khâu của robot, có thể dùng các màu sắc khác nhau để thể hiện hình dáng của robot. Lưu ý trong quá trình vẽ, nếu vẽ sai phải dùng mục CLEAR để xóa đi hoặc dùng mục MODIFY CEL để hiệu chỉnh. Mỗi đối tượng vẽ phải gắn với một khâu nhất định, được khai báo trong mục SET JOINT INDEX.

Có thể dùng thanh công cụ thẳng đứng phía phải để thay đổi vị trí của các đối tượng vẽ cho thích hợp.

6.7. Lập trình điều khiển robot đã mô phỏng :

Để lập trình điều khiển robot đã mô phỏng ta dùng phương pháp lập trình kiểu dạy học. Sau khi đã thiết kế hình dáng robot, công cụ gắn trên khâu chấp hành cuối, các đối tượng làm việc khác . . . ta có thể lập trình để điều khiển robot đã mô phỏng. Việc lập trình thực hiện theo trình tự sau đây :

Nhấp chuột vào nút lệnh số 10 (Show program window) để kích hoạt cửa sổ lập trình như hình 6.3 :



Hình 6.3 : Cửa sổ lập trình.

Chọn New để đặt tên cho File chương trình.

Chọn Append nếu muốn bổ sung một chương trình đã có trên đĩa.

Xác định vị trí các điểm mở dụng cụ phải đi qua (dùng chuột để điều khiển các khớp, dùng menu đứng). Cứ sau mỗi lần xác định được một vị trí thì ấn nút PTP (điều khiển điểm) hoặc LIN (điều khiển đường) hoặc VIA

(điểm trung gian dẫn hướng khi điều khiển đường cong), CIRC (điều khiển theo đường cong). Làm liên tục cho tất cả các điểm để có một chương trình hoàn thiện.

Sau khi kết thúc việc dạy robot học, ấn nút Close trên Program Window để kết thúc. Để hiệu chỉnh và bổ sung các lệnh điều khiển khác vào chương trình, ấn chuột vào nút EDIT, Dùng các lệnh của EasyRob như dưới đây để hoàn thiện chương trình.

ERPL - EASY-ROB-Program Language

Ghi chú :

- Đơn vị chiều dài là Mét [m], Góc là độ [deg] hoặc [%]

- Đơn vị của tốc độ là [m/s]

- Vị trí và hướng của hệ tọa độ gắn trên khâu chấp hành cuối được xác định gồm : X, Y và Z : chỉ tọa độ vị trí, A, B và C chỉ góc hướng.

Hướng của khâu chấp hành cuối xác định theo các góc ABC là:

$Rot(A,B,C) = Rot(X,A) * Rot(Y,B) * Rot(Z,C)$

Cấu trúc chung của chương trình, Mô tả cú pháp một số lệnh hay dùng :

PROGRAMFILE : Bắt đầu chương trình

CALL fct_name : Gọi một hàm có tên fct_name(), đã được định nghĩa trong chương trình.

CALL FILE filename : Gọi một File chương trình có tên filename, File phải có cùng cấu trúc như chương trình chính.

FCT fct_name() : Bắt đầu Định nghĩa một hàm có tên fct_name().

ENDFCT : Kết thúc định nghĩa một function.

! Các ghi chú trong chương trình.

TOOL X Y Z A B C [m,deg] : Định tọa độ điểm cuối của dụng cụ so với khâu chấp hành cuối.

PTP X Y Z A B C [m,deg] : Di chuyển robot đến điểm mới (tọa độ tuyệt đối). Điều khiển điểm.

PTP_REL dX dY dZ dA dB dC [m,deg] : Di chuyển robot đến điểm mới (tọa độ tương đối). Điều khiển điểm.

LIN X Y Z A B C [m,deg] : Di chuyển robot đến điểm mới (tọa độ tuyệt đối). Điều khiển đường.

LIN_REL dX dY dZ dA dB dC [m,deg] : Di chuyển robot đến điểm mới (tọa độ tương đối). Điều khiển đường.

CIRC X Y Z A B C [X2 Y2 Z2] [m,deg] : Di chuyển robot đến điểm mới (tọa độ tuyệt đối). Điều khiển đường cong.

[X2 Y2 Z2] - Điểm trung gian (3 điểm để xác định một cung tròn).

CIRC_REL dX dY dZ dA dB dC [dX2 dY2 dZ2] [m,deg] : Di chuyển robot đến điểm mới (tọa độ tương đối). Điều khiển đường cong.

WAIT x [sec] : Robot dừng hoạt động trong x giây.

ERC TRACK ON,OFF : Thể hiện hoặc không thể hiện quỹ đạo chuyển động.

ERC LOAD TOOL filename : Gọi một Tool file (*.tol)

ERC LOAD VIEW filename : Gọi một View file (*.vie)

ERC LOAD ROBOT filename Loads a Robot file (*.rob)

ERC LOAD BODY filename Loads a Body file (*.bod)

ERC LOAD TAGS filename Loads a Tag file (*.tag)

ERC GRAB BODY 'bodyname' : Dụng cụ cầm lấy một vật thể (body) có tên Bodyname.

ERC GRAB BODY_GRP : Dụng cụ cầm lấy một nhóm vật thể (Body_Grp).

ERC RELEASE BODY 'bodyname' : Dụng cụ thả (buông) một vật thể (body) có tên Bodyname.

ERC RELEASE BODY_GRP Dụng cụ thả (buông) một nhóm vật thể (Body_Grp).

ERC ROBOT_BASE XYZ ABC [m,deg] : Di chuyển gốc tọa độ cơ bản của robot đến vị trí mới.

v.v...

Còn rất nhiều các lệnh khác của Easy-Rob, có thể tham khảo trên Website:

<http://www.easy-rob.com>.

CHƯƠNG VII ĐỘNG LỰC HỌC ROBOT (Dynamic of Robot)

7.1. Nhiệm vụ và phương pháp phân tích động lực học robot

Nghiên cứu động lực học robot là công việc cần thiết khi phân tích cũng như tổng hợp quá trình điều khiển chuyển động. Việc nghiên cứu động lực học robot thường giải quyết hai nhiệm vụ sau đây :

1/ Xác định momen và lực động xuất hiện trong quá trình chuyển động. Khi đó qui luật biến đổi của biến khớp $q_i(t)$ coi như đã biết.

Việc tính toán lực trong cơ cấu tay máy là rất cần thiết để chọn công suất động cơ, kiểm tra độ bền, độ cứng vững, đảm bảo độ tin cậy của robot.

2/ Xác định các sai số động tức là sai lệch so với qui luật chuyển động theo chương trình. Lúc này cần khảo sát Phương trình chuyển động của robot có tính đến đặc tính động lực của động cơ và các khâu.

Có nhiều phương pháp nghiên cứu động lực học robot, nhưng thường gặp hơn cả là

phương pháp cơ học Lagrange, cụ thể là dùng phương trình Lagrange - Euler. Đối với các

khâu khớp của robot, với các nguồn động lực và kênh điều khiển riêng biệt, không thể bỏ qua các hiệu ứng trọng trường (gravity effect), quán tính (initial), tương hỗ (Coriolis), ly tâm (centripetal)... mà những khía cạnh này chưa được xét đầy đủ trong cơ học cổ điển; Cơ học Lagrange nghiên cứu các vấn đề nêu trên như một hệ thống khép kín nên đây là nguyên lý cơ học thích hợp đối với các bài toán động lực học robot.

7.2. Cơ học Lagrange với các vấn đề động lực của robot.

Hàm Lagrange của một hệ thống năng lượng được định nghĩa :

$$L=K-P \quad (7.1)$$

Trong đó : K là tổng động năng của hệ thống

P là tổng thế năng

K và P đều là những đại lượng vô hướng nên có thể chọn bất cứ hệ tọa độ thích hợp nào để bài toán được đơn giản. Đối với một robot có n khâu, ta có:

$$K = \sum_{i=1}^n K_i \quad \text{và} \quad P = \sum_{i=1}^n P_i$$

Ở đây, K_i và P_i là động năng và thế năng của khâu thứ i xét trong hệ tọa độ chọn. Ta biết mỗi đại lượng K_i và P_i là một hàm số phụ thuộc nhiều biến số:

$$K_i = K(q_i, \dot{q}_i) \quad \text{và} \quad P_i = P(q_i, q_j)$$

Với q_i là tọa độ suy rộng của khớp thứ i. Nếu khớp thứ i là khớp quay thì q_i là góc quay θ_i , nếu là khớp tịnh tiến thì q_i là độ dài tịnh tiến d_i .

Ta định nghĩa : Lực tác dụng lên khâu thứ i ($i=1, 2, \dots, n$) với quan

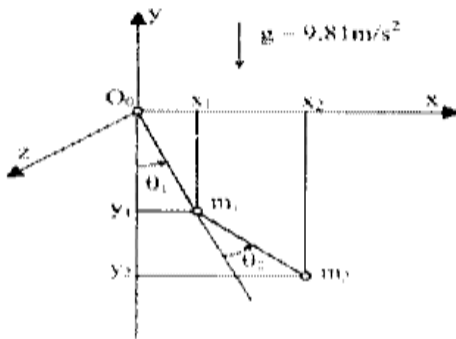
niệm là lực tổng quát (Generalized forces), nó có thể là một lực hoặc một momen (phụ thuộc vào biến khớp q_i là tịnh tiến hoặc quay), được xác định bởi:

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}$$

Phương trình này được gọi là phương trình Lagrange-Euler, hay thường được gọi tắt là phương trình Lagrange.

7.3 Ví dụ áp dụng:

Xét một robot có hai khâu như hình vẽ, Các khâu có chiều dài là d_1 và d_2 với các khối lượng tương ứng m_1 và m_2 qui đổi về đầu mút của khâu. Robot được đặt thẳng đứng chịu gia tốc trọng trường g . Các khớp chuyển động quay với các biến khớp θ_1 và θ_2 . Tính lực tổng quát.



Qua ví dụ này, chỉ với một mối liên kết hai khâu, các vấn đề đặt ra đều đã có mặt trong quá trình nghiên cứu động lực và do đó, ví dụ nêu trên có thể mở rộng để áp dụng trong những trường hợp phức tạp hơn.

Đối với khâu 1:

$$K_1 = \frac{1}{2} m_1 v_1^2 = \frac{1}{2} m_1 d_1^2 \dot{\theta}_1^2$$

$$P_1 = -m_1 g d_1 \cos \theta_1$$

Đối với khâu 2 :

Về tọa độ :

$$x_2 = d_1 \sin \theta_1 + d_2 \sin(\theta_1 + \theta_2)$$

$$y_2 = -d_1 \cos \theta_1 - d_2 \cos(\theta_1 + \theta_2)$$

Chiều cao thế năng:

$$h = d_1 \cos \theta_1 + d_2 \cos(\theta_1 + \theta_2)$$

Về vận tốc:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

Với

$$= \frac{d}{dt} x_2 = d_1 \cos(\theta_1) \dot{\theta}_1 + d_2 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)$$

$$= \frac{d}{dt} y_2 = d_1 \sin(\theta_1) \dot{\theta}_1 + d_2 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)$$

$$= [d_1^2 \dot{\theta}_1^2 + d_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) + 2d_1 d_2 \cos(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)]$$

Động năng và thế năng sẽ là :

$$= \frac{1}{2} m_2 v_2^2 = \frac{1}{2} m_2 [d_1^2 \dot{\theta}_1^2 + d_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) + 2d_1 d_2 \cos(\theta_2) (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2)]$$

Giá trị $= -m_2 g [d_1 \cos(\theta_1) + d_2 \cos(\theta_1 + \theta_2)]$

7.4. Hàm Lagrange và lực tổng quát :

Áp dụng hàm Lagrange cho ví dụ trên, ta có :

$$L = (K1 + K2) - (P1 + P2)$$

$$L = \frac{1}{2} (m_1 + m_2) d_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 d_2^2 (\dot{\theta}_1^2 + 2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) + m_2 d_1 d_2 \cos \theta_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) + (m_1 + m_2) g d_1 \cos \theta_1 + m_2 g d_2 \cos(\theta_1 + \theta_2)$$

Khi tính lực tổng quát, các biến của hệ : q1 = θ1 và q2 = θ2.

Đối với khâu 1 :

$$\frac{\partial L}{\partial q_1} = \frac{\partial L}{\partial \theta_1} = (m_1 + m_2) d_1^2 \dot{\theta}_1 + m_2 d_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + 2m_2 d_1 d_2 \cos \theta_2 \dot{\theta}_1 + m_2 d_1 d_2 \cos \theta_2 \dot{\theta}_2$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) d_1^2 \ddot{\theta}_1 + m_2 d_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) - 2m_2 d_1 d_2 \sin \theta_2 \dot{\theta}_2 \dot{\theta}_1 + 2m_2 d_1 d_2 \cos \theta_2 \ddot{\theta}_1 - m_2 d_1 d_2 \sin \theta_2 \dot{\theta}_2^2 + m_2 d_1 d_2 \cos \theta_2 \ddot{\theta}_2$$

$$\frac{\partial L}{\partial q_1} = \frac{\partial L}{\partial \theta_1} = -(m_1 + m_2) g d_1 \sin \theta_1 - m_2 g d_2 \sin(\theta_1 + \theta_2)$$

Vậy:

$$F_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = [(m_1 + m_2) d_1^2 + m_2 d_2^2 + 2m_2 d_1 d_2 \cos \theta_2] \ddot{\theta}_1 + [m_2 d_2^2 + m_2 d_1 d_2 \cos \theta_2] \ddot{\theta}_2 - 2m_2 d_1 d_2 \sin \theta_2 \dot{\theta}_2 \dot{\theta}_1 - m_2 d_1 d_2 \sin \theta_2 \dot{\theta}_2^2 + (m_1 + m_2) g d_1 \sin \theta_1 + m_2 g d_2 \sin(\theta_1 + \theta_2)$$

Muốn cho khâu 1 quay được một góc θ1 thì động cơ phải tạo ra một lực tổng quát ≥ F1. Lực tổng quát này có đặc tính phi tuyến, là hợp tác dụng của nhiều yếu tố (non linear and coupling).

Tương tự, để tính lực tổng quát của khâu thứ hai, ta có :

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 \dot{\theta}_1 + m_2 d_2^2 \dot{\theta}_2 + m_2 d_1 d_2 \cos \theta_2 \dot{\theta}_1$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 \ddot{\theta}_1 + m_2 d_2^2 \ddot{\theta}_2 + m_2 d_1 d_2 \cos \theta_2 \ddot{\theta}_1 - m_2 d_1 d_2 \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2$$

$$\text{và } \frac{\partial L}{\partial \theta_2} = -m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1^2 - m_2 g d_2 \sin(\theta_1 + \theta_2)$$

Vậy:

$$F_2 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = [m_2 d_2^2 + m_2 d_1 d_2 \cos \theta_2] \ddot{\theta}_1 + m_2 d_2^2 \ddot{\theta}_2$$

$$+ m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1^2 + m_2 g d_2 \sin(\theta_1 + \theta_2)$$

Để phân tích ý nghĩa các thành phần trong biểu thức tính lực tổng quát, ta viết lại các biểu thức F_1, F_2 như sau :

$$\begin{aligned}
 F_1 &= D_{11}\ddot{\theta}_1 + D_{12}\ddot{\theta}_2 + D_{111}\dot{\theta}_1^2 + D_{122}\dot{\theta}_2^2 + D_{112}\dot{\theta}_1\dot{\theta}_2 + D_{121}\dot{\theta}_1\dot{\theta}_2 + D_1 \\
 F_2 &= D_{12}\ddot{\theta}_1 + D_{22}\ddot{\theta}_2 + D_{211}\dot{\theta}_1^2 + D_{222}\dot{\theta}_2^2 + D_{212}\dot{\theta}_1\dot{\theta}_2 + D_{221}\dot{\theta}_1\dot{\theta}_2 + D_2
 \end{aligned}$$

Hiệu ứng quán tính Hiệu ứng ly tâm Hiệu ứng tương hỗ Hiệu ứng trọng trường
 Effective inertias Centripetal effect Coriolis effect Gravity

(Trong đó : $D_{111} = 0; D_{222} = 0; D_{112} = D_{121} = D_{212} = D_{221} = -m_2d_1d_2\sin\theta_2 \dots$)

Trong các biểu thức trên, các hệ số dạng D_{ij} hoặc D_{ij} thể hiện hiệu ứng quán tính tại khớp i hoặc j gây ra bởi gia tốc tại khớp i hoặc j . Các số hạng có dạng $D_{ij}\dot{\theta}_j^2$ là lực ly tâm tác động lên khớp i gây ra bởi vận tốc tại khớp j . Số hạng dạng $D_{ijk}\dot{\theta}_j\dot{\theta}_k + D_{ikj}\dot{\theta}_k\dot{\theta}_j$ là lực Coriolis tác động lên khớp thứ i gây ra do vận tốc tại khớp j và k . Số hạng có dạng D_i là lực trọng trường tác động lên khớp i .

7.5. Phương trình động lực học robot :

Xét khâu thứ i của một robot có n khâu. Tính lực tổng quát F_i của khâu thứ i với khối lượng vi phân của nó là dm . Lực tổng quát F_i đóng vai trò rất quan trọng khi xây dựng sơ đồ khối để thiết lập hàm điều khiển cho robot có n bậc tự do.

7.5.1. Vận tốc của một điểm trên robot :

Một điểm trên khâu thứ i được mô tả trong hệ tọa độ cơ bản là :

$$R = T_i + r$$

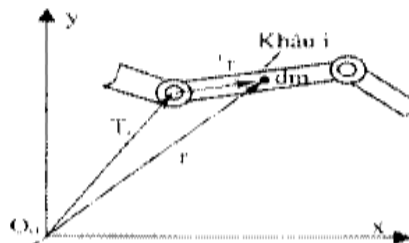
Trong đó: ir là tọa độ của điểm xét đối với khâu thứ I , ir không thay đổi theo thời gian. T_i là ma trận chuyển đổi từ khâu thứ I về hệ tọa độ gốc: $T_i = A_1.A_2 \dots A_i$. Như vậy r là một hàm của thời gian t .

Tốc độ của vi khối lượng dm được tính bởi công thức:

$$\dot{r} = \frac{dr}{dt} = \frac{d}{dt} T_i^{-1} r - \left(\sum_{j=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j \right)^{-1} r$$

Khi tính bình phương của vận tốc này ta có :

$$\dot{r} \cdot \dot{r} = \sum r^2 (\dot{x}_n, \dot{y}_n, \dot{z}_n) = Tr(\dot{r} \dot{r}^T)$$



Với r^T là chuyển vị vectow và Tr là viết tắt của Tracee (vết của ma trận):

$$\text{Trace} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n1} & a_{nn} \end{bmatrix} = \sum_{i=1}^n a_{ii}$$

Hay :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot [x \ y \ z] = \begin{bmatrix} x^2 & & \\ & y^2 & \\ & & z^2 \end{bmatrix}$$

Do vậy:

$$\begin{aligned} \dot{r}^2 &= \text{Tr}(\dot{r} \cdot \dot{r}^T) = \text{Tr}\left(\frac{d}{dt} T_i^{-1} r, \frac{d}{dt} T_i^{T-1} r^T\right) \\ &= \text{Tr}\left[\sum_{j=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j^{-1} r, \sum_{k=1}^i \frac{\partial T_i^{-1}}{\partial q_k} \dot{q}_k^{-1} r^T\right] \\ &= \text{Tr}\left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} r^T r^T \cdot \frac{\partial T_i^{-1}}{\partial q_k} \dot{q}_j \dot{q}_k\right] \end{aligned}$$

7. 5. 2. Tính động năng của vi khối lượng dm.

Ký hiệu K_i là động năng của khâu thứ i . dK_i là động năng của vi khối lượng dm đặt tại vị trí r trên khâu thứ i .

$$\begin{aligned} dK_i &= \frac{1}{2} \cdot \text{Tr}\left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} r^T r^T \cdot \frac{\partial T_i^{-1}}{\partial q_k} \dot{q}_j \dot{q}_k\right] dm \\ &= \frac{1}{2} \text{Tr}\left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} (r \cdot dm \cdot r^T) \cdot \frac{\partial T_i^{-1}}{\partial q_k} \dot{q}_j \dot{q}_k\right] \end{aligned}$$

Và do đó động năng của khâu thứ i sẽ là:

$$K_i = \int_{khâu i} dK = \frac{1}{2} Tr \left[\sum_{j=1}^3 \sum_{k=1}^3 \frac{\partial T_i}{\partial \dot{q}_j} \left(\int_{khâu i} r_i^j r_i^k dm \right) \cdot \frac{\partial T_i}{\partial \dot{q}_i} \dot{q}_j \dot{q}_k \right]$$

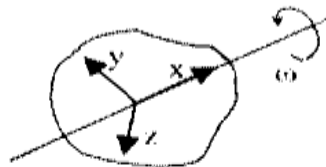
Đặt $J_i = \int_{khâu i} r_i^j r_i^k dm$ gọi là ma trận quán tính (Pseudo inertia matrix).

Ý nghĩa "giả quán tính" được sử dụng vì khi thiết lập đầy đủ các phần tử của ma trận J_i ta có thể liên hệ với các khái niệm "mômen quán tính độc cực" và trình bày các phần tử của J_i giống như các phần tử của mômen quán tính độc cực. Ta xét mối quan hệ này như sau :

Theo định nghĩa ta có :

$$J_i = \int_{khâu i} r_i^j r_i^k dm = J_i = \begin{bmatrix} \int x^2 dm & \int x^j y dm & \int x^j z dm & \int x dm \\ \int x^j y dm & \int y^2 dm & \int y^j z dm & \int y dm \\ \int x^j z dm & \int y^j z dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix}$$

Bây giờ ta nhắc lại mômen quán tính độc cực của một vật thể bất kỳ như hình vẽ.



Hình 7.2 : Mômen quán tính độc cực

Theo định nghĩa ta có:

$$I_{xx} = \int (y^2 + z^2) dm$$

$$I_{yy} = \int (x^2 + z^2) dm$$

$$I_{zz} = \int (x^2 + y^2) dm$$

Vũ vỹ : $x^2 = -\frac{1}{2}(y^2 + z^2) + \frac{1}{2}(x^2 + z^2) + \frac{1}{2}(x^2 + y^2)$

Vậy : $\int x^2 dm = (-I_{xx} + I_{yy} + I_{zz}) / 2 ; \dots$

Ngoài ra ta còn có:

$$I_{xy} = \int xy dm ; I_{yz} = \int yz dm ; I_{xz} = \int xz dm$$

Giáo trình Rc $m_x = \int x dm ; m_y = \int y dm ; m_z = \int z dm$ Trang 115

Đối chiếu với ma trận giả quán tính J_i , ta có thể trình bày J_i như sau :

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & mx \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & my \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & mz \\ mx & my & mz & m \end{bmatrix}$$

Như vậy ý nghĩa biểu trưng của J_i đã rõ.

Vậy ta có:

$$K_i = \frac{1}{2} Tr \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial \dot{q}_j} J_i \frac{\partial T_i}{\partial \dot{q}_k} \dot{q}_j \dot{q}_k \right]$$

Cuối cùng động năng của một robot có n khâu được tính:

$$K = \sum_{i=1}^n K_i$$

7. 5. 3. Tính thế năng của robot :

Thế năng của khâu i có khối lượng m_i , trọng tâm được xác định bởi vector r_i (vector

biểu diễn trọng tâm của khâu i trong hệ tọa độ cơ bản) là :

$$P_i = -m_i \cdot g \cdot r_i = -m_i \cdot g \cdot T_i r_i$$

Trong đó, vector gia tốc trọng trường g được biểu diễn dưới dạng một ma trận cột :

$$g = \begin{bmatrix} g \\ g \\ g \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -9,8 \\ 0 \end{bmatrix}$$

Thế năng của toàn cơ cấu robot n khâu động sẽ là :

$$P = - \sum_{i=1}^n m_i g T_i^T r_i$$

7. 5.4. Hàm Lagrange :

Sau khi xác định động năng và thế năng của toàn cơ cấu, ta có hàm Lagrange của robot có n bậc tự do :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=0}^i \sum_{k=1}^j Tr\left(\frac{\partial T_i}{\partial q_j} J_i \frac{\partial T_i'}{\partial q_k} \right) \dot{q}_j \dot{q}_k + \frac{1}{2} \sum_{i=1}^n m_i g T_i' r_i$$

Chúng ta chú ý rằng, trong hàm Lagrange vẫn chưa đề cập đến ảnh hưởng của nguồn truyền động (gồm các phần tĩnh (stator) và phần động (Rotor) của động cơ điện).

7. 5. 5. Phương trình động lực học robot :

Ta đã biết lực tổng quát đặt lên khâu thứ i của robot có n khâu (Phương trình Lagrange - Euler) :

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}$$

Sau khi thiết lập hàm Lagrange, với p = 1... n, ta tính được: (p là chỉ số lần lượt lấy theo j và k)

$$\frac{\partial L}{\partial \dot{q}_p} = \frac{1}{2} \sum_{i=1}^n \sum_{k=1}^i Tr \left(\frac{\partial T_i}{\partial \dot{q}_p} J_i \frac{\partial T_i'}{\partial \dot{q}_k} \right) \dot{q}_k + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i Tr \left(\frac{\partial T_i}{\partial \dot{q}_j} J_i \frac{\partial T_i'}{\partial \dot{q}_p} \right) \dot{q}_j$$

Thay đổi chỉ số giả j thành k trong số hạng thứ 2 và để ý rằng:

$$\left(\frac{\partial T_i}{\partial \dot{q}_j} J_i \frac{\partial T_i'}{\partial \dot{q}_p} \right) = Tr \left(\frac{\partial T_i}{\partial \dot{q}_j} J_i \frac{\partial T_i'}{\partial \dot{q}_p} \right)^T = Tr \left(\frac{\partial T_i}{\partial \dot{q}_p} J_i \frac{\partial T_i'}{\partial \dot{q}_j} \right)$$

Ta có:

$$\frac{\partial L}{\partial \dot{q}_p} = \sum_{i=1}^n \sum_{k=1}^i Tr \left(\frac{\partial T_i}{\partial \dot{q}_k} J_i \frac{\partial T_i'}{\partial \dot{q}_p} \right) \dot{q}_k$$

Cũng để ý rằng : trong $T_i(q_1, q_2, \dots, q_i)$, với q_i là các biến khớp của i khớp đầu tiên. Do vậy, nếu $i < p$ thì

$$\frac{\partial T_i}{\partial \dot{q}_p} = 0$$

Cuối cùng ta có:

$$\frac{\partial L}{\partial \dot{q}_p} = \sum_{i=1}^n \sum_{k=1}^i Tr \left(\frac{\partial T_i}{\partial \dot{q}_k} J_i \frac{\partial T_i'}{\partial \dot{q}_p} \right) \dot{q}_k$$

Lấy vi phân theo thời gian t của phương trình trên :

$$\begin{aligned}
\underline{Tru} \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} &= \frac{d}{dt} \sum_{i=p}^n \sum_{k=1}^i Tr \left(\frac{\partial T_i}{\partial q_k} J_i \frac{\partial T_i'}{\partial q_p} \right) \dot{q}_k \\
&= \sum_{i=p}^n \sum_{k=1}^i Tr \left(\frac{\partial T_i}{\partial q_k} J_i \frac{\partial T_i'}{\partial q_p} \right) \ddot{q}_k + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i Tr \left[\frac{\partial^2 T_i}{\partial q_k \partial q_m} J_i \frac{\partial T_i'}{\partial q_p} \right] \dot{q}_k \dot{q}_m \\
&\quad + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i Tr \left[\frac{\partial^2 T_i}{\partial q_p \partial q_m} J_i \frac{\partial T_i'}{\partial q_k} \right] \dot{q}_k \dot{q}_m
\end{aligned}$$

Số hạng cuối của phương trình Lagrange Euler là :

$$\begin{aligned}
&\sum_{i=p}^n \sum_{j=1}^i \sum_{k=1}^i Tr \left(\frac{\partial^2 T_i}{\partial q_j \partial q_p} J_i \frac{\partial T_i'}{\partial q_k} \right) \dot{q}_j \dot{q}_k + \\
&\quad + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i Tr \left(\frac{\partial^2 T_i}{\partial q_k \partial q_p} J_i \frac{\partial T_i'}{\partial q_j} \right) \dot{q}_j \dot{q}_k + \sum_{i=p}^n m_i g \frac{\partial T_i'}{\partial q_p} r_i
\end{aligned}$$

Cuối cùng ta có lực tổng quát của khâu p:

$$F_p = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} - \frac{\partial L}{\partial q_p}$$

Thay thế các chỉ số p và i thành i và j, ta sẽ có :

$$F_i = \sum_{j=i}^n \sum_{k=1}^j Tr \left(\frac{\partial T_j}{\partial q_k} J_j \frac{\partial T_j'}{\partial q_i} \right) \ddot{q}_k + \sum_{j=i}^n \sum_{k=1}^j \sum_{m=1}^j Tr \left[\frac{\partial^2 T_j}{\partial q_k \partial q_m} J_j \frac{\partial T_j'}{\partial q_i} \right] \dot{q}_k \dot{q}_m - \sum_{i=1}^n m_i g r_i$$

Với một robot có n bậc tự do thì :

$$q = [q_1, q_2, \dots, q_n]^T$$

$$\dot{q} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T$$

Và $F = F[F_1, F_2, \dots, F_n]^T$

Để cho gọn ta biểu diễn:

$$F = J(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)$$

Trong đó :

J thể hiện tác dụng của quán tính, là một ma trận đối xứng ($n \times n$);

C thể hiện tác dụng của lực ly tâm và Coriolis, là một vector ($n \times 1$);

G thể hiện tác dụng của lực trọng trường, cũng là một vector ($n \times 1$).

Đây là phương trình động lực học của robot.

Nếu thêm vào phương trình trên các tác dụng khác như : FEX đặc trưng cho các ngoại lực tác dụng lên trục, V đặc trưng cho hiệu ứng ma sát, ta có :

$$F = J(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + V(\dot{q}) + F_{LV}$$

CHƯƠNG VIII

THIẾT KẾ QUỸ ĐẠO ROBOT

(Trajectory Planing)

Trong các ứng dụng công nghiệp của robot, ta thường gặp hai trường hợp sau :

Trường hợp 1 : Khâu chấp hành cuối của robot chỉ cần đạt được vị trí và hướng tại các điểm nút (điểm tựa : Knot point). Đây chính là phương pháp điều khiển điểm (PTP). Tại đó, bàn tay robot thực hiện các thao tác cầm nắm đối tượng hoặc buông thả đối tượng. Đây là trường hợp của các robot thực hiện công việc vận chuyển và trao đổi phôi liệu trong một hệ thống tự động linh hoạt robot hoá. Bàn tay robot không trực tiếp tham gia vào các nguyên công công nghệ như hàn, cắt kim loại ... Các điểm nút là mục tiêu quan trọng nhất, còn dạng đường đi tới các điểm nút là vấn đề thứ yếu. Trong trường hợp này Robot thường được lập trình bằng phương pháp dạy học (Teach and playback mode). Trong trường hợp này không cần tính toán phương trình động học hoặc động học ngược robot, chuyển động mong muốn được ghi lại như một tập hợp các góc khớp (thực tế là tập hợp các giá trị mã hoá của biến khớp) để robot thực hiện lại (Playback) khi làm việc.

Trường hợp 2 : Khâu chấp hành cuối của robot phải xác định đường đi qua các điểm nút theo thời gian thực. Đó là trường hợp các tay máy trực tiếp thực hiện các nguyên công công nghệ như sơn, hàn, cắt kim loại ... Vấn đề thiết kế quỹ đạo cho các robot trong trường hợp này là rất quan trọng. Nó quyết định trực tiếp chất lượng thực hiện các nguyên công công nghệ mà robot đảm nhận. Trong chương này, chúng ta đề cập đến bài toán thiết kế quỹ đạo với một số quỹ đạo điển hình. Các quỹ đạo này không chỉ có ý nghĩa trong trường hợp ứng dụng thứ hai mà nó bao hàm một ý nghĩa chung cho mọi robot, vì ngay cả trường hợp đơn giản như các robot thuộc

ứng dụng thứ nhất cũng thực hiện những chuyển động quỹ đạo cơ bản mà chúng ta sẽ nghiên cứu dưới đây.

8.1. Các khái niệm về quỹ đạo robot :

Để xác định được đường đi mong muốn của robot theo thời gian, quỹ đạo có thể được tính toán thiết kế trong một hệ tọa độ truyền thống Oxyz (Cartesian Space) hoặc thiết kế trong không gian biến khớp (không gian trường vectơ các tọa độ suy rộng của robot), chẳng hạn với robot 6 bậc tự do thì $X = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$. Thiết kế quỹ đạo ở đây được hiểu là xác định qui luật chuyển động của các biến khớp để điều khiển chuyển động của từng khớp và tổng hợp thành chuyển động chung của robot theo một quỹ đạo đã được xác định.

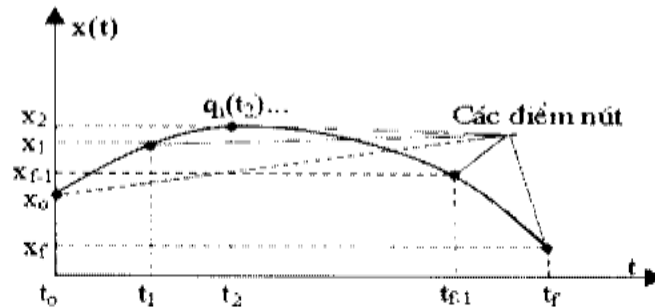
Quỹ đạo cần thiết kế nhất thiết phải đi qua một số điểm nút cho trước (ít nhất là điểm đầu và điểm cuối). Ngoài các điểm nút chính, ta còn có thể chọn thêm các điểm nút phụ gọi là điểm dẫn hướng (via point) để tránh các chướng ngại vật.

Khi thiết kế quỹ đạo trong không gian biến khớp, tại mỗi điểm nút phải xác định giá trị của các biến khớp bằng phương pháp tính toán động học ngược. Thời gian yêu cầu của mỗi đoạn quỹ đạo (giữa 2 điểm nút) là giống nhau cho tất cả các khớp vì vậy yêu cầu tất cả các khớp phải đạt đến điểm nút đồng thời. Ngoài việc yêu cầu thời gian phải giống nhau cho các khớp, việc xác định các hàm quỹ đạo của mỗi biến khớp không phụ thuộc vào các hàm của các khớp khác. Vì vậy việc thiết kế quỹ đạo trong không gian biến khớp đơn giản và dễ tính toán hơn khi mô tả trong hệ tọa độ Đề-các.

Quỹ đạo thiết kế phải đảm bảo các điều kiện liên tục (continuous conditions) bao gồm :

- + Liên tục về vị trí (Position)
- + Liên tục về tốc độ (Velocity)

+ Liên tục về gia tốc (Acceleration)



Hình 8.1. Tính liên tục của quỹ đạo robot.

Để thiết kế quỹ đạo robot, người ta thường dùng phương pháp xấp xỉ các đa thức bậc n, các quỹ đạo thường gặp là :

- + Quỹ đạo CS (Cubic Segment) : Tổng đồng đa thức bậc 3;
- + Quỹ đạo LS (linear Segment) : Tổng đồng đa thức bậc 1;
- + Quỹ đạo LSPB (Linear Segment with Parabolic Blend) : Phối hợp đa thức bậc 2 với đa thức bậc 1.



Hình 8.2 : Quỹ đạo LSPB

+ Quỹ đạo BBPB (Bang Bang Parabolic Blend) : là trường hợp đặc biệt của quỹ đạo LSPB khi đoạn tuyến tính thu về bằng 0 và xuất hiện điểm uốn.



Hình 8.2 : Quỹ đạo BBPB

Nếu cho trước nhiều điểm nút, ta có thể áp dụng nhiều dạng quỹ đạo cơ bản khác nhau cho một biến khớp.

8.2. Quỹ đạo đa thức bậc 3 :

Khi thiết kế quỹ đạo robot theo đa thức bậc 3 qua các điểm nút, mỗi đoạn quỹ đạo giữa hai điểm nút sẽ được biểu diễn bằng một phương trình bậc 3 riêng biệt. Quỹ đạo đa thức bậc 3 đảm bảo sự liên tục của đạo hàm bậc nhất và bậc hai tại các điểm nút.

Tại thời điểm $t_k \leq t \leq t_{k+1}$, quỹ đạo xấp xỉ đa thức bậc 3 của biến khớp thứ i là $q_i(t)$ có dạng :

$$q_i(t) = a_i + b_i(t - t_k) + c_i(t - t_k)^2 + d_i(t - t_k)^3 \tag{8.1}$$

Với các ràng buộc :

$$\begin{aligned} q_i(t_k) &= q_k & \text{và} & \quad \dot{q}_i(t_k) = \dot{q}_k \\ q_i(t_{k+1}) &= q_{k+1} & \text{và} & \quad \dot{q}_i(t_{k+1}) = \dot{q}_{k+1} \end{aligned}$$

Từ (8.1) ta thấy : $t = t_k \rightarrow a_i = q_k$
 Lấy đạo hàm của (8.1) theo t , ta có :

$$\dot{q}_i(t) = b_i + 2c_i(t - t_k) + 3d_i(t - t_k)^2$$

Tại : $t = t_k \rightarrow b_i = \dot{q}_k$ (8.3)

Tại $t = t_{k+1}$ ta có hai tham số :

$$c_i = \frac{3(q_{k+1} - q_k) - (2\dot{q}_k + \dot{q}_{k+1}) \delta t_k}{\delta t_k^2} \tag{8.4}$$

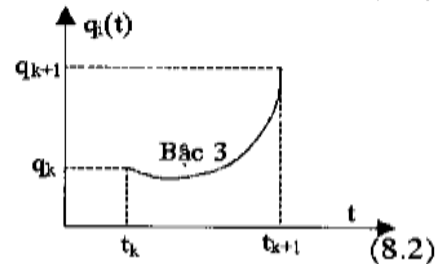
$$d_i = \frac{(\dot{q}_{k+1} + \dot{q}_k) \delta t_k - 2(q_{k+1} - q_k)}{\delta t_k^3} \tag{8.5}$$

Trong đó : $\delta t_k = t_{k+1} - t_k$

Các phương trình (8.4) và (8.5) nhận được khi giải (8.1) ... (8.3).

Tính liên tục của vận tốc là sự đảm bảo cho quỹ đạo không gặp khúc, giật cục, gây sốc trong quá trình hoạt động của robot. Vận tốc và gia tốc tại điểm cuối của một đoạn đường cong bậc 3 chính bằng vận tốc và gia tốc của đoạn cong bậc 3 tiếp theo.

Cần chú ý rằng khi thiết kế quỹ đạo trong không gian Đề cát, để điều khiển được robot, ở mỗi thời điểm đều phải tìm được nghiệm của bài toán



động học ngược. Vì vậy yêu cầu "não bộ" của robot (máy tính) phải thực hiện

một khối lượng các phép tính khổng lồ trong một khoảng thời gian rất ngắn (vài chục microgiây) để đảm bảo thời gian thực khi robot hoạt động. Nếu ta không tìm cách cải biến thiết kế quỹ đạo thì rất khó đảm bảo yêu cầu này.

* Ví dụ về thiết kế quỹ đạo CS:

Thiết kế quỹ đạo CS (Path with Cubic segment) của khớp thứ i đi qua hai điểm nút có giá trị q_0 và q_f . Với các rặng buộc $\dot{q}_0 = 0$, $\dot{q}_f = 0$

Từ các công thức (8.2) . . . (8.5) ta xác định các hệ số của đa thức bậc 3 như sau :

$$a_i = q_0 ; b_i = 0;$$

$$c_i = \frac{3(q_f - q_0)}{(t_f - t_0)^2} \quad \text{và} \quad d_i = \frac{-2(q_f - q_0)}{(t_f - t_0)^3}$$

Do vậy quỹ đạo $q_i(t)$ có dạng như sau :

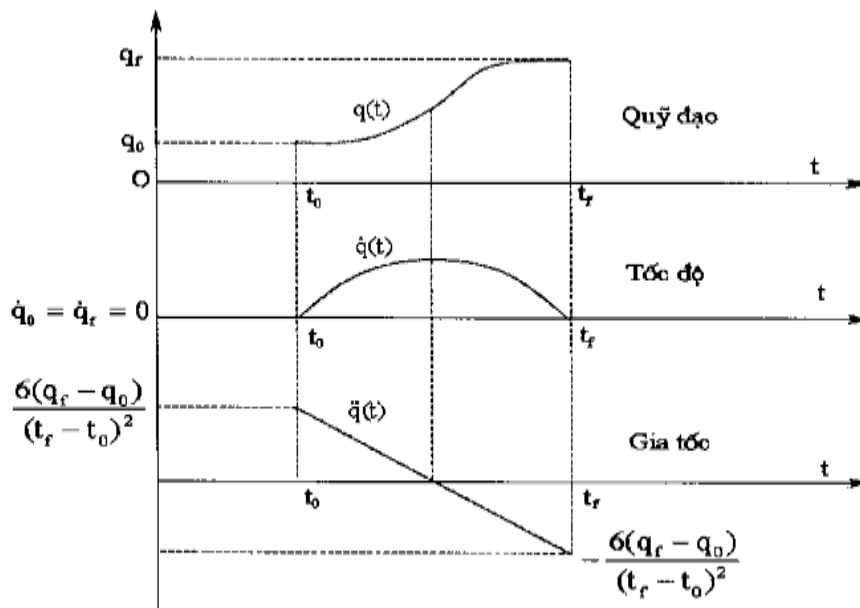
$$q_i(t) = q_0 + \frac{3(q_f - q_0)}{(t_f - t_0)^2} (t - t_0)^2 - \frac{2(q_f - q_0)}{(t_f - t_0)^3} (t - t_0)^3$$

$$\text{Vận tốc là : } \dot{q}_i(t) = \frac{6(q_f - q_0)}{(t_f - t_0)^2} (t - t_0) - \frac{6(q_f - q_0)}{(t_f - t_0)^3} (t - t_0)^2$$

$$\text{Và gia tốc là : } \ddot{q}_i(t) = \frac{6(q_f - q_0)}{(t_f - t_0)^2} - \frac{12(q_f - q_0)}{(t_f - t_0)^3} (t - t_0)$$

Trong ví dụ trên, giả sử thời gian $t_0 = 0$ và $t_f = 1$ giây, thì :

$$q_i(t) = q_0 + 3(q_f - q_0) t^2 - 2(q_f - q_0) t^3$$

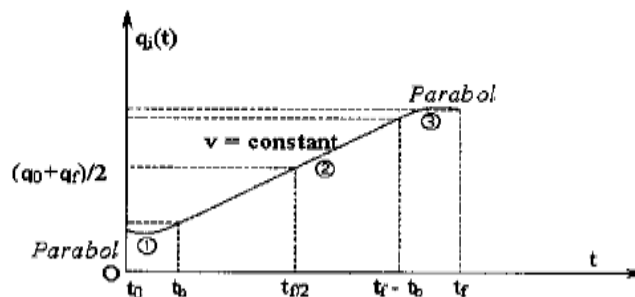


Hình 8.3. Thiết kế quỹ đạo CS

Từ các phương trình quỹ đạo, phương trình vận tốc và phương trình gia tốc ta xây dựng được các biểu đồ đặc tính chuyển động của khớp thứ i trên đoạn quỹ đạo thiết kế.

8.3. Quỹ đạo tuyến tính với cung ở hai đầu là parabol (LSPB) :

Khi yêu cầu công cụ gắn trên khâu chấp hành cuối của robot chuyển động với vận tốc đều đặn, ta dùng quỹ đạo LSPB.



Hình 8.3. Quỹ đạo LSPB.

Các điều kiện liên tục của quỹ đạo này thể hiện ở :

$$q(t_0) = q_0 ; q(t_f) = q_f, \text{ và } \dot{q}(t_0) = \dot{q}(t_f) = 0$$

và điều kiện công nghệ là $v = \text{constant}$.

Quỹ đạo được chia làm 3 đoạn :

a/ Trong đoạn 1 : $0 \leq t \leq t_b$ quỹ đạo Parabol có dạng :

$$q_i(t) = \alpha + \beta t + \gamma t^2 \quad (8.6)$$

$$\text{Khi } t = 0 \text{ thì } \alpha = q(t_0) = q_0 \quad (8.7)$$

$$\text{Lấy đạo hàm (8.6) : } \dot{q}(t) = \beta + 2\gamma t \quad (8.8)$$

$$\text{Khi } t = 0 \text{ thì } \beta = \dot{q}(t_0) = 0$$

Tại thời điểm t_b ta cần có vận tốc bằng hằng số vận tốc cho trước v :

$$\text{Nên khi } t = t_b \quad \gamma = v/2t_b$$

Đặt $v/t_b = a \Rightarrow \gamma = a/2$ và quỹ đạo có dạng :

$$q_i(t) = q_0 + at^2/2 \quad (0 \leq t \leq t_b) \quad (8.9)$$

b/ Trong đoạn 2 : $[t_b, (t_f - t_b)]$ quỹ đạo tuyến tính có dạng :

$$q_i(t) = \alpha_0 + vt$$

$$\text{Do tính đối xứng : } q\left(\frac{t_f}{2}\right) = \frac{(q_0 + q_f)}{2}$$

$$\text{Suy ra: } \frac{(q_0 + q_f)}{2} = \alpha_0 +$$

$$\text{Vậy: } \alpha_0 = \frac{(q_0 + q_f - vt_f)}{2}$$

Phương trình quỹ đạo tuyến tính sẽ là :

$$q_i(t) = \frac{q_0 + q_f - vt_f}{2} + vt \quad (8.10)$$

Từ điều kiện liên tục về vị trí tại thời điểm t_b ta có:

$$q_o + \frac{at_b^2}{2} = \frac{q_o + q_f - vt_f}{2} + vt_b$$

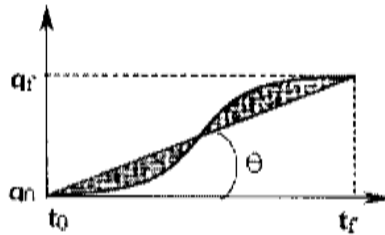
Rút ra:

$$t_b = \frac{q_o - q_f + vt_f}{v}$$

Với điều kiện tồn tại: $0 < t_b \leq \frac{t_f}{2}$, dẫn đến:

$$\frac{q_f - q_o}{v} < t_f \leq \frac{2(q_f - q_o)}{v}$$

Điều này xác định vận tốc phải nằm giữa các giới hạn trên, nếu không chuyển động sẽ không thực hiện được.



Về mặt vật lý:

Nếu $t_f > \frac{(q_f - q_o)}{v}$ và $t_f \leq 2 \frac{(q_f - q_o)}{v}$

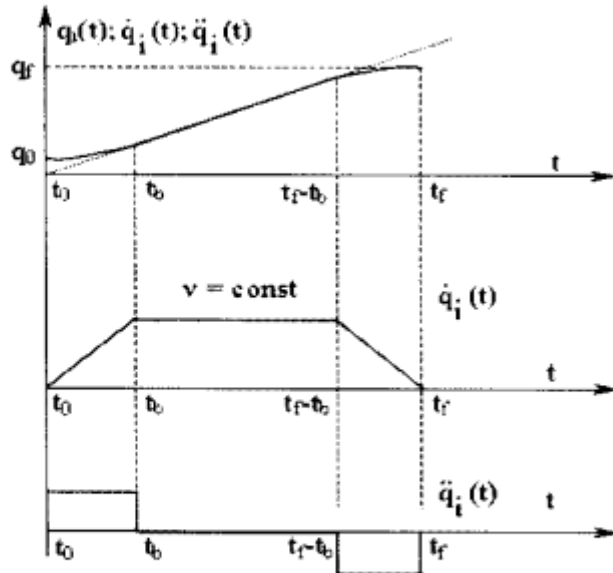
thì $v > \frac{(q_f - q_o)}{t_f}$ và $v \leq 2 \frac{(q_f - q_o)}{t_f}$

Nghĩa là: $tg\theta < v \leq tg2\theta$

c/ Trong đoạn 3: $(t_f - t_b) \leq t \leq t_f$ quỹ đạo Parabole có dạng:

$$q_i(t) = q_f - \frac{at_f^2}{2} + at_f t - \frac{a}{2} t^2 \tag{8.11}$$

Từ các phương trình (8.9)...(8.11) ta xây dựng đặc tính chuyển động theo quỹ đạo LSPB của khớp q_i như sau:



Hình 8.4 : Đặc tính quỹ đạo LSPB

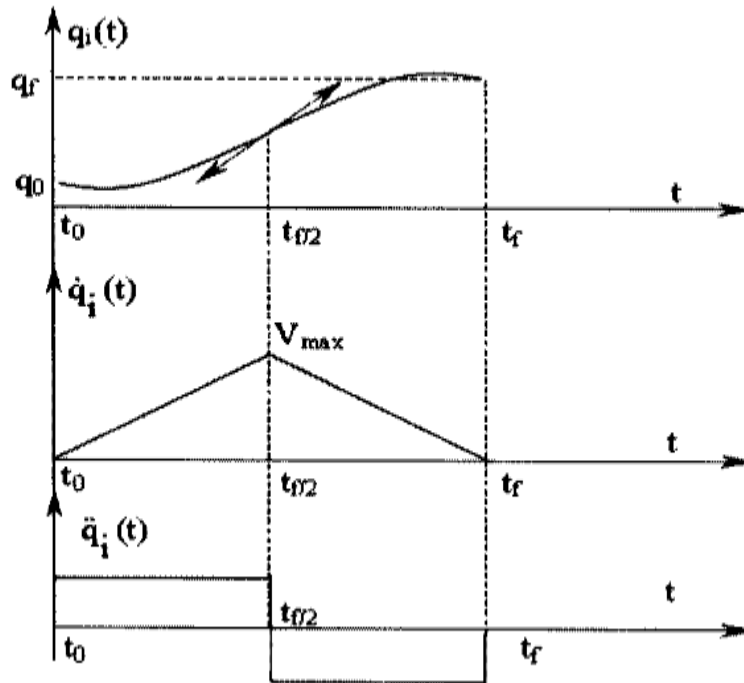
8.4. Quỹ đạo Bang Bang Parabolic blend (BBPB) :

Như đã trình bày ở trên, đây là trường hợp đặc biệt của quỹ đạo LSPB khi đoạn tuyến tính thu về 0.

Với : $0 \leq t \leq \frac{t_f}{2}$ $q(t) = q_0 + \frac{at^2}{2}$

và với $\frac{t_f}{2} \leq t \leq t_f$ $q(t) = 2q_0 - q_f + 2a \sqrt{\frac{q_f - q_0}{a} t - \frac{at^2}{2}}$

Đồ thị đặc tính quỹ đạo này như sau:



Hình 8.5. Đặc tính quỹ đạo BBPB

CHƯƠNG 9

TRUYỀN ĐỘNG VÀ ĐIỀU KHIỂN ROBOT

9.1. Truyền động điện trong robot:

Truyền động điện được dùng khá nhiều trong kỹ thuật robot, vì có nhiều ưu điểm như là điều khiển đơn giản không phải dùng các bộ biến đổi phụ, không gây bẩn môi trường, các loại động cơ điện hiện đại có thể lắp trực tiếp trên các khớp quay...

Tuy nhiên so với truyền động thuỷ lực hoặc thuỷ khí thì truyền động điện có công suất thấp và thông thường phải cần thêm hộp giảm tốc vì thường các khâu của robot chuyển động với tốc độ thấp.

Trong kỹ thuật robot, về nguyên tắc có thể dùng động cơ điện các loại khác nhau, nhưng trong thực tế chỉ có hai loại được dùng nhiều hơn cả. Đó là động cơ điện một chiều và động cơ bước.

Ngày nay, do những thành tựu mới trong nghiên cứu điều khiển động cơ điện xoay chiều, nên cũng có xu hướng chuyển sang sử dụng động cơ điện xoay chiều để tránh phải trang bị thêm bộ nguồn điện một chiều. Ngoài ra, loại động cơ điện một chiều không chổi góp (DC brushless motor) cũng bắt đầu được ứng dụng vào kỹ thuật robot.

9.1.1. Động cơ điện một chiều :

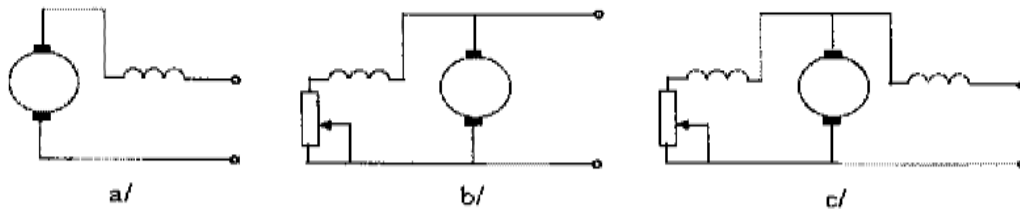
Động cơ điện một chiều gồm có hai phần :

+ Stato cố định với các cuộn dây có dòng điện cảm hoặc dùng nam châm vĩnh cửu. Phần này còn được gọi là phần cảm. Phần cảm tạo nên từ thông trong khe hở không khí.

+ Roto với các thanh dẫn. Khi có dòng điện một chiều chạy qua và với dòng từ thông xác định, roto sẽ quay. Phần này gọi là phần ứng.

Tùy cách đấu dây giữa phần cảm so với phần ứng, ta có những loại động cơ điện một chiều khác nhau :

- + Động cơ kích từ nối tiếp (Hình 9.1.a);
- + Động cơ kích từ song song (Hình 9.1.b);
- + Động cơ kích từ hỗn hợp (Hình 9.1.c).



Hình 9.1. Các loại động cơ điện một chiều.

Các thông số chủ yếu quyết định tính năng làm việc của động cơ điện một chiều là :

- U : Điện áp cung cấp cho phần ứng;
- I : Cường độ dòng điện của phần ứng;
- r : Điện trở trong của phần ứng;
- Φ : Từ thông;
- E : Sức phản điện động phần ứng.

Các quan hệ cơ bản của động cơ điện một chiều là :

$$E = U - rI = kn\Phi$$

k là hệ số phụ thuộc vào đặc tính của dây cuộn và số thanh dẫn của phần ứng.

$$\text{Số vòng quay của động cơ điện một chiều : } n = \frac{U - Ir}{k\Phi}$$

Mômen động C xác định từ phương trình cân bằng công suất :

$$EI = 2\pi nC$$

$$\text{Hay : } C = 2 \frac{k\Phi I}{\pi}$$

Muốn điều chỉnh tốc độ động cơ điện một chiều có thể thực hiện bằng cách :

- Thay đổi từ thông Φ , thông qua việc điều chỉnh điện áp dòng kích từ. Trong trường hợp giữ nguyên điện áp phản ứng U, tăng tốc độ từ 0 đến tốc độ định mức, thì công suất không đổi còn momen giảm theo tốc độ.

- Điều chỉnh điện áp phản ứng. Trong trường hợp từ thông không đổi, khi tăng tốc độ từ 0 đến tốc độ định mức thì momen sẽ không đổi, còn công suất tăng theo tốc độ.

Muốn đảo chiều quay của động cơ điện một chiều cần thay đổi hoặc chiều của từ thông (tức chiều của dòng điện kích từ) hoặc thay đổi chiều dòng điện phản ứng.

9.1.2. Động cơ bước :

Nguyên tắc hoạt động :

Trên hình 9.2 là sơ đồ động cơ bước loại đơn giản nhất dùng nam châm vĩnh cửu gồm stato có 4 cực và roto có 2 cực.



Hình 9.2 : Sơ đồ nguyên lý hoạt động của động cơ bước.

Nếu cấp điện cho cuộn dây $\alpha\alpha'$ thì roto sẽ dừng ở vị trí mà dòng từ qua cuộn dây là lớn nhất. Nếu cấp điện cho cuộn dây $\beta\beta'$ thì roto sẽ quay đi $\pm 90^0$ (Phụ thuộc chiều dòng điện cấp vào). Khi đồng thời cấp điện cho cả 2 cuộn dây α và β thì roto sẽ dừng ở vị trí giữa 0^0 và 90^0 , và nếu dòng điện vào 2 cuộn dây hoàn toàn như nhau thì roto sẽ dừng ở vị trí 45^0 .

Như vậy vị trí của roto phụ thuộc vào số cực được cấp điện trên stato và chiều của dòng điện cấp vào.

Trên đây là sơ đồ nguyên lý của động cơ bước loại có ít cực và dùng nam châm vĩnh cửu. Trên cơ sở đó ta có thể tìm hiểu các loại động cơ có nhiều cực và dùng nam châm điện có từ tính thay đổi.

Như vậy tùy theo cách cấp điện cho các cuộn dây trên stato ta có thể điều khiển các vị trí dừng của roto. Việc cấp điện cho các cuộn dây có thể số hoá, cho nên có thể nói động cơ bước là loại động cơ điện chuyển các tín hiệu số đầu vào thành chuyển động cơ học từng nấc ở đầu ra.

Ưu nhược điểm :

+ Khi dùng động cơ bước không cần mạch phản hồi cho cả điều khiển vị trí và vận tốc.

+ Thích hợp với các thiết bị điều khiển số. Với khả năng điều khiển số trực tiếp, động cơ bước trở thành thông dụng trong các thiết bị cơ điện tử hiện đại.

Tuy nhiên phạm vi ứng dụng động cơ bước là ở vùng công suất nhỏ và trung bình. Việc nghiên cứu nâng cao công suất động cơ bước đang là vấn đề rất được quan tâm hiện nay. Ngoài ra, nói chung hiệu suất của động cơ bước thấp hơn các loại động cơ khác.

Các thông số chủ yếu của động cơ bước :

Góc quay :

Động cơ bước quay một góc xác định ứng với mỗi xung kích thích. Góc bước θ cũng nhỏ thì độ phân giải vị trí cũng cao. Số bước s là một thông số quan trọng :

$$s = \frac{360^\circ}{\theta}$$

Tốc độ quay và tần số xung :

Tốc độ quay của động cơ bước phụ thuộc vào số bước trong một giây. Đối với hầu hết các động cơ bước, số xung cấp cho động cơ bằng số bước (tính theo phút) nên tốc độ có thể tính theo tần số xung f . Tốc độ quay của động cơ bước tính theo công thức sau :

$$n = \frac{60f}{s} \quad (f : \text{bước/phút}) / (s : \text{bước /vòng})$$

Trong đó: n - tốc độ quay (vòng/phút)

f- tần số xung (Hz)

s- số bước trong 1 vòng quay.

Ngoài ra còn các thông số quan trọng khác như độ chính xác vị trí, momen và quán tính của động cơ...

Các loại động cơ bước :

Tuỳ theo kiểu của roto, động cơ bước được chia thành các loại sau :

+ Động cơ bước kiểu từ trở biến đổi (VR : Variable Resistance)

+ Động cơ bước nam châm vĩnh cửu (PM : Permanent Magnet)

+ Động cơ bước kiểu lai (Hybrid)

Tuỳ theo số cuộn dây độc lập trên stato động cơ bước được chia thành các loại : 2 pha, 3 pha hoặc 4 pha.

Roto động cơ bước có nhiều cực (còn gọi là răng). Số cực của roto phối hợp với số cực của stato xác định giá trị góc bước θ . Góc bước lớn nhất là 90^0 ứng với động cơ có số bước $s = 4$ bước/vòng. Phần lớn những động cơ bước hiện nay có số bước $s = 200$, nên $\theta = 1,8^0$.

Số bước cũng lớn độ phân giải cũng cao và định vị cũng chính xác. Nhưng trong thực tế, không thể tăng số bước lên quá cao. Tuy nhiên có thể dùng công nghệ tạo bước nhỏ để chia bước thành 2 nửa bước (như hình b/ 9.2) hoặc từ 10 đến 125 bước nhỏ. Công nghệ tạo bước nhỏ còn gọi là tạo vi bước, chỉ đơn giản là mở rộng phương pháp nói trên cho nhiều vị trí trung gian bằng cách cung cấp những giá trị dòng khác nhau cho mỗi cuộn dây. Động cơ được tạo bước nhỏ có độ phân giải tinh hơn nhiều. Ví dụ, nếu phân 125 bước nhỏ trong một bước đầy, với 200 bước/vòng thì độ phân giải của động cơ là $125 \times 200 = 25.000$ bước nhỏ/ vòng.

9.2. Truyền động khí nén và thủy lực :

Ngoài truyền động điện, trong kỹ thuật robot còn thường dùng các loại truyền động khí nén hoặc thủy lực.

9.2.1. Truyền dẫn động khí nén :

Dùng khí nén trong hệ truyền động robot nhiều thuận lợi như : Do các phân xưởng công nghiệp thường có mạng lưới khí nén chung, nên đơn giản hoá được phần thiết bị nguồn động lực cho robot. Hệ truyền dẫn khí nén gọn nhẹ, dễ sử dụng, dễ bảo trì, ... Tuy nhiên hệ truyền dẫn khí nén cũng có nhiều nhược điểm như : do tính nén được của chất khí nên chuyển động thường kèm theo dao động, dừng không chính xác, ngoài ra còn cần trang bị thêm các thiết bị phun dầu bôi trơn, lọc bụi, giảm tiếng ồn ...

9.2.2. Truyền dẫn động thuỷ lực :

Hệ truyền dẫn thuỷ lực có những ưu điểm như : Tải trọng lớn, quán tính bé, dễ thay đổi chuyển động, dễ điều khiển tự động.

Tuy nhiên chúng cũng có những nhược điểm như : Hệ thuỷ lực luôn đòi hỏi bộ nguồn, bao gồm thùng dầu, bơm thuỷ lực, thiết bị lọc, bình tích dầu, các loại van điều chỉnh, đường ống ... làm hệ truyền động cho robot khá cồng kềnh so với truyền động khí nén và truyền động điện.

Nhìn chung, hệ truyền dẫn thuỷ lực vẫn được sử dụng khá phổ biến trong robot, nhất là trong trường hợp tải nặng.

Các phần tử trong hệ truyền động bằng khí nén và thuỷ lực đã được tiêu chuẩn hoá.

Các tính toán thiết kế hệ truyền dẫn khí nén và thuỷ lực đã được nghiên cứu trong các giáo trình riêng.

9.3. Các phương pháp điều khiển Robot :

Nhiệm vụ quan trọng đầu tiên của việc điều khiển robot là bảo đảm cho điểm tác động cuối E (End-effector) của tay máy dịch chuyển bám theo một quỹ đạo định trước. Không những thế, hệ tọa độ gắn trên khâu chấp hành cuối còn phải đảm bảo hướng trong quá trình di chuyển. Giải bài toán ngược phương trình động học ta có thể giải quyết về mặt động học yêu cầu trên. Đó cũng là nội dung cơ bản để xây dựng chương trình điều khiển vị trí cho robot.

Tuy nhiên việc giải bài toán này chưa xét tới điều kiện thực tế khi robot làm việc, như là các tác động của momen lực, ma sát ... Tuỳ theo yêu cầu nâng cao chất lượng điều khiển (độ chính xác) mà ta cần tính đến ảnh hưởng của các yếu tố trên, và theo đó, phương pháp điều khiển cũng trở nên đa dạng và phong phú hơn.

9.3.1. Điều khiển tỉ lệ sai lệch (PE : *Propotional Error*):

Nguyên tắc cơ bản của phương pháp này rất dễ hiểu; đó là làm cho hệ thống thay đổi theo chiều hướng có sai lệch nhỏ nhất. Hàm sai lệch có thể là $\varepsilon = \theta_d - \theta(t)$, ở đây θ_d là góc quay mong muốn và $\theta(t)$ là giá trị quay thực tế của biến khớp, ta sẽ gọi θ_d là "góc đặt". Khi $\varepsilon = 0$ thì khớp đạt được vị trí mong muốn. Nếu $\varepsilon < 0$, thì khớp đã di chuyển quá mức và cần chuyển động ngược lại. Như vậy, kiểu điều khiển chuyển động này là luôn có chiều hướng làm cho sai lệch ε xấp xỉ zero.

Bên cạnh đó, chúng ta cũng cần quan tâm đến phần độ lớn, nghĩa là, chúng ta không những cần biết "làm cho động cơ chuyển động bằng cách nào?" mà còn cần biết "cần cung cấp cho động cơ một năng lượng (mômen động) là bao nhiêu?". Để trả lời câu hỏi này một lần nữa, chúng ta có thể dùng tín hiệu sai số $\varepsilon = \theta_d - \theta$. Chúng ta hãy áp dụng một tín hiệu điều khiển mà nó tỉ lệ với ε :

$$F = K_p(\theta_d - \theta(t))$$

Qui luật này xác định một hệ điều khiển phản hồi và được gọi là hệ điều khiển tỉ lệ sai lệch.

9.3.2. Điều khiển tỉ lệ - đạo hàm (PD : *Propotional Derivative*):

Phương pháp điều khiển tỉ lệ sai lệch còn nhiều nhược điểm như : Hệ dao động lớn khi ma sát nhỏ (tình trạng vượt quá) và ở trạng thái tĩnh, khi $\varepsilon \rightarrow 0$ thì momen cũng gần bằng không, nên không giữ được vị trí dưới tác dụng của tải.

Để khắc phục điều trên, có thể chọn phương pháp điều khiển tỉ lệ - đạo hàm (PD), với lực tổng quát :

$$F = K_p \varepsilon + K_d \dot{\theta}(t) \tag{9.2}$$

Trong đó : ε - sai số vị trí của khớp động. $\varepsilon = \theta_d - \theta(t)$.
 $\dot{\theta}(t)$ - Thành phần đạo hàm - vận tốc góc.
 K_p - Hệ số tỉ lệ sai lệch vị trí.
 K_d - Hệ số tỉ lệ vận tốc.

9.3.3. Điều khiển tỉ lệ - tích phân - đạo hàm

(PID : Propotional Integral Derivative):

Hệ thống với cấu trúc luật điều khiển PD vẫn còn một số nhược điểm, không phù hợp với một số loại robot. Một hệ thống điều khiển khác có bổ sung thêm tín hiệu tốc độ đặt $\dot{\theta}$ và sai lệch tốc độ $\dot{\varepsilon} = \dot{\theta}_d - \dot{\theta}(t)$ tác động vào khâu khuếch đại K_d . Phương trình lực tác động lên khớp động có dạng :

$$F = K_p \varepsilon + K_d \dot{\varepsilon} + K_i \int_0^t \varepsilon(t) dt \tag{9.3}$$

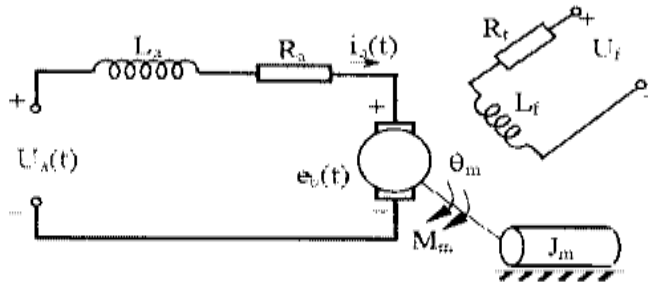
Với $\dot{\varepsilon}$ - sai số tốc độ. $\dot{\varepsilon} = \dot{\theta}_d - \dot{\theta}(t)$.

Như vậy, tùy theo cấu trúc đã lựa chọn của bộ điều khiển, ta đem đối chiếu các phương trình(9.1), (9.2) hoặc (9.3) với phương trình Lagrange - Euler, Từ đó nhận được các phương trình của hệ điều khiển t-ong ứng. Từ các phương trình này của hệ điều khiển, cần xác định các hệ số tỉ lệ K_p , K_d , K_i để hệ hoạt động ổn định.

9.3.4. Hàm truyền chuyển động của mỗi khớp động :

Nội dung phần này trình bày phương pháp xây dựng hàm truyền đối với trường hợp chuyển động một bậc tự do, mỗi khớp thường được điều khiển bằng một hệ truyền động riêng. Phổ biến hơn cả là động cơ điện một chiều.

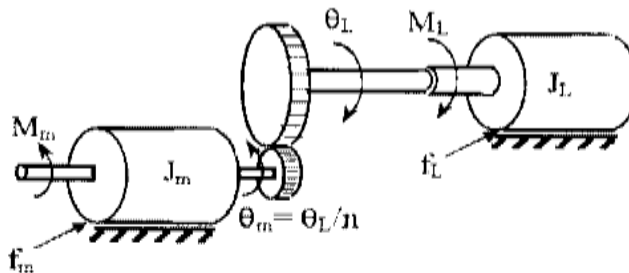
Xét sơ đồ truyền động của động cơ điện một chiều với tín hiệu vào là điện áp U_a đặt vào phần ứng, tín hiệu ra là góc quay θ_m của trục động cơ; động cơ kiểu kích từ độc lập.



Hình 9.3. Sơ đồ động cơ điện một chiều.

Trong thực tế, trục động cơ được nối với hộp giảm tốc rồi tới trục phụ tải như hình 9.4. Gọi n là tỉ số truyền, θ_L là góc quay của trục phụ tải, ta có :

$$\begin{aligned} \theta_L(t) &= n \theta_m(t) \\ \dot{\theta}_L(t) &= n \dot{\theta}_m(t) \\ \ddot{\theta}_L(t) &= n \ddot{\theta}_m(t) \end{aligned} \tag{9.4}$$



Hình 9.4. Sơ đồ động cơ điện cùng phụ tải.

Mômen trên trục động cơ bằng tổng momen cần để động cơ quay, cộng với mômen phụ tải quy về trục động cơ.

$$M(t) = M_m(t) + M_L^*(t) \tag{9.5}$$

Ký hiệu : J_m : Mômen quán tính của động cơ.
 J_L : Momen quán tính phụ tải.

Ta có :

$$M_m(t) = J_m \ddot{\theta}_m(t) + f_m \dot{\theta}_m(t) \tag{9.6}$$

$$M_L(t) = J_L \ddot{\theta}_L(t) + f_L \dot{\theta}_L(t) \tag{9.7}$$

Trong đó f_m và f_L là hệ số cản của động cơ và của phụ tải.

Theo định luật bảo tồn năng lượng, công do phụ tải sinh ra, tính trên trục phụ tải là $M_L \theta_L$ phải bằng công quy về trục động cơ $M^* L \theta_L$. Từ đó ta có :

$$M_L^*(t) = \frac{M_L(t)\theta_L(t)}{\theta_m(t)} = nM_L(t) \quad (9.8)$$

Thay (9.1) và (9.4) vào công thức trên :

$$M_L^*(t) = n^2 [J_L \ddot{\theta}_m(t) + f_L \dot{\theta}_m(t)] \quad (9.9)$$

Thay (9.3) và (9.6) vào (9.2) ta có :

$$M(t) = (J_m + n^2 J_L) \ddot{\theta}_m(t) + f_m n^2 f_L \dot{\theta}_m(t)$$

Hay :
$$M(t) = J \ddot{\theta}_m(t) + f \dot{\theta}_m(t) \quad (9.10)$$

Với :

$$\begin{aligned} J &= J_m + n^2 J_L && : \text{Mômen quán tính tổng hiệu dụng.} \\ f &= f_m + n^2 f_L && : \text{Hệ số ma sát tổng hiệu dụng.} \end{aligned}$$

Mômen trên trục động cơ phụ thuộc tuyến tính với cường độ dòng điện phản ứng và không phụ thuộc vào góc quay và vận tốc góc, ta có :

$$M(t) = K_a i_a(t) \quad (9.11)$$

Với i_a : Cường độ dòng điện phản ứng.

K_a : Hệ số tỉ lệ mômen.

Áp dụng định luật Kirchhoff cho mạch điện phản ứng :

$$U_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \quad (9.12)$$

Với R_a, L_a : điện trở và điện cảm phản ứng.

e_b : sức phản điện động của động cơ.

$$e_b(t) = K_b \dot{\theta}_m(t) \quad (9.13)$$

K_b : hệ số tỉ lệ của sức phản điện động.

Sử dụng phép biến đổi Laplace, từ (9.12) ta có :

$$I_a(s) = \frac{U_a(s) - sK_b \theta_m(s)}{R_a + sL_a} \quad (9.14)$$

Từ (9.10) và (9.11) ta có :

$$M(s) = s^2 J \theta_m(s) + s f \theta_m(s) = K_a I_a(s)$$

$$\Rightarrow \theta_m(s) = \frac{K_a I_a(s)}{s^2 J + s f} \quad (9.15)$$

Thay (9.14) vào (9.15) :

$$\theta_m(s) = K_a \left[\frac{U_a(s) - sK_b\theta_m(s)}{(s^2J + sf)(R_a + sL_a)} \right]$$

$$\Rightarrow \frac{U_a(s) - sK_b\theta_m(s)}{\theta_m(s)} = \frac{(s^2J + sf)(R_a + sL_a)}{K_a}$$

$$\frac{U_a(s)}{\theta_m(s)} = \frac{(s^2J + sf)(R_a + sL_a) + sK_aK_b}{K_a}$$

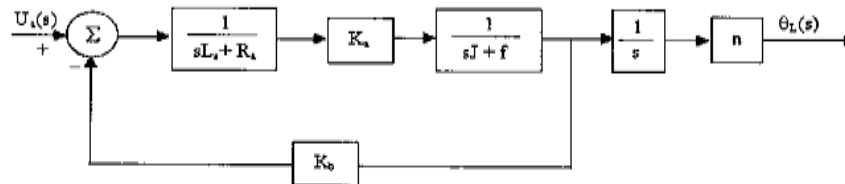
Hay :

$$\frac{\theta_m(s)}{U_a(s)} = \frac{K_a}{s[(sJ + f)(R_a + sL_a) + K_aK_b]} \tag{9.16}$$

Đây là hàm truyền cần xác định, nó là tỉ số giữa tín hiệu ra (góc quay θ_m) và tín hiệu vào của hệ thống (điện áp U_a). Vì hệ thống gồm có động cơ và phụ tải nên tín hiệu ra thực tế là góc quay của trục phụ tải vào, do đó hàm truyền chuyển động 1 bậc tự do của tay máy là :

$$\frac{\theta_L(s)}{U_a(s)} = \frac{nK_a}{s[(R_a + sL_a)(sJ + f) + K_aK_b]} \tag{9.17}$$

Và ta có sơ đồ khối tương ứng với hàm truyền trên là :



Hình 9.5 : Sơ đồ khối hàm truyền chuyển động một bậc tự do.

Trong công thức (9.17) có thể bỏ qua thành phần điện cảm phản ứng L_a , vì nó thường quá nhỏ so với các nhân tố ảnh hưởng cơ khí khác. Nên :

$$\frac{\theta_L(s)}{U_a(s)} = \frac{nK_a}{s(sR_aJ + R_a f + K_a K_b)} \quad (9.18)$$

9.3.5. Điều khiển vị trí mỗi khớp động :

Mục đích của điều khiển vị trí là làm sao cho động cơ chuyển dịch khớp động đi một góc bằng góc quay đã tính toán để đảm bảo quỹ đạo đã chọn trước(chương 8). Việc điều khiển được thực hiện như sau : Theo tín hiệu sai lệch giữa giá trị thực tế và giá trị tính toán của vị trí góc mà điều chỉnh điện áp $U_a(t)$ đặt vào động cơ. Nói cách khác, để điều khiển động cơ theo quỹ đạo mong muốn phải đặt vào động cơ một điện áp tỉ lệ thuận với độ sai lệch góc quay của khớp động.

$$U_a(t) = \frac{K_p e(t)}{n} = \frac{K_p (\tilde{\theta}_L(t) - \theta_L(t))}{n} \quad (9.19)$$

Trong đó K_p : hệ số truyền tín hiệu phản hồi vị trí.

$$e(t) = \tilde{\theta}_L(t) - \theta_L(t) : \text{độ sai lệch góc quay.}$$

Giá trị góc quay tức thời : $(t) \sim L$ được đo bằng cảm biến quang học hoặc chiết áp. Biến đổi Laplace phương trình (9.18) :

$$U_a(s) = \frac{K_p (\tilde{\theta}_L(s) - \theta_L(s))}{n} = \frac{K_p E(s)}{n} \quad (9.20)$$

Thay (9.20) vào phương trình (9.18) :

$$\frac{\theta_L(s)}{E(s)} = \frac{K_a K_p}{s(sR_aJ + R_a f + K_a K_b)} = G(s) \quad (9.21)$$

Sau khi biến đổi đại số ta có hàm truyền :

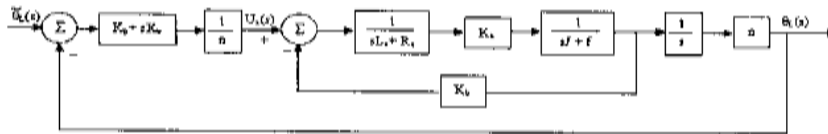
$$\frac{\theta_L(s)}{\tilde{\theta}_L(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_a K_p}{s^2 R_a J + s(R_a f + K_a K_b) + K_a K_b} = \frac{K_a K_p / R_a J}{s^2 + \frac{(R_a f + K_a K_b)}{R_a J} s + \frac{K_a K_b}{R_a J}} \quad (9.22)$$

Phương trình (9.22) cho thấy rằng hệ điều khiển tỉ lệ của một khớp động là một hệ bậc hai, nó sẽ luôn ổn định nếu các hệ số của của phương trình bậc hai là những số dương. Để nâng cao đặc tính động lực học và giảm sai số trạng thái ổn định của hệ người ta có thể tăng hệ số phản hồi vị trí K_p và kết hợp làm giảm dao động trong hệ bằng cách thêm vào thành phần đạo hàm của sai số vị trí. Với việc thêm phản hồi này, điện áp đặt lên động cơ sẽ tỉ lệ tuyến tính với sai số vị trí và đạo hàm của nó :

$$U_a(t) = \frac{K_p(\tilde{\theta}_L(t) - \theta_L(t)) + K_v(\dot{\tilde{\theta}}_L(t) - \dot{\theta}_L(t))}{n} = \frac{K_p e(t) + K_v \dot{e}(t)}{n} \quad (9.23)$$

Trong đó K_v là hệ số phản hồi của sai số về vận tốc.

Với phản hồi nêu trên, hệ thống trở thành khép kín và có hàm truyền như thể hiện trên sơ đồ khối hình (9.6). Đây là phương pháp điều khiển tỉ lệ - Đạo hàm.



Hình 9.6 : Sơ đồ khối điều khiển chuyển dịch một khớp động có liên hệ phản hồi

Biến đổi Laplace phương trình (9.23) và thay $U_a(s)$ vào (9.21) ta có:

$$\frac{\theta_L(s)}{E(s)} = \frac{K_a(K_p + sK_v)}{s(sR_a J + R_a f + K_a K_b)} = \frac{K_a K_v s + K_a K_p}{s(sR_a J + R_a f + K_a K_b)} = G(s) \quad (9.24)$$

Từ đó ta có:

$$\frac{\theta_L(s)}{\tilde{\theta}_L(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_a(K_v s + K_p)}{s^2 R_a J + s(R_a f + K_a K_b + K_a K_v) + K_a K_p} \quad (9.25)$$