บทที่ 4

วิธีการเขียนโพสต์โปรเซสเซอร์ของหุ่นยนต์ Mitsubishi RV-M1

4.1) ลักษณะของไฟล์ข้อมูลเอาต์พุตที่ได้จากโปรแกรม CATIA

หลังจากที่ได้วางแผนการทำงาน และจำลองสภาพการทำงานของหุ่นยนต์บนโปรแกรม CATIA แล้ว เราจะใช้ฟังก์ชัน CATASK UTILITY ซึ่งเป็นยูทิลิตี้ที่ใช้ในการบันทึกลักษณะการเคลื่อนที่ต่างๆลงในไฟล์เอาต์พุต ซึ่งจากไฟล์ที่ได้ดังกล่าวนั้นเมื่อนำไปแปลงโดยใช้โพสต์โปรเซสเซอร์ที่เขียนขึ้น ก็จะได้เป็นโปรแกรมที่ใช้ในการควบคุมการเคลื่อนที่สำหรับหุ่นยนต์ Mitsubishi RV-M1 (ไฟล์นามสกุล .MRL) และไฟล์ของตำแหน่งต่างๆ ที่หุ่นยนต์เคลื่อนที่ (ไฟล์นามสกุล .POS) จากไฟล์ดังกล่าวเราสามารถส่งโปรแกรมและตำแหน่งต่างๆ ผ่านการส่งข้อมูลแบบอนุกรม (serial communication) เพื่อควบคุมหุ่นยนต์ให้ทำงานตามลักษณะการจำลองการทำงานต่อไป

ไฟล์เอาต์พุตที่ได้จากโปรแกรม CATIA นั้น นอกจากจะบันทึกลักษณะการเคลื่อนที่ต่างๆ แล้ว ยังบันทึกข้อมูลลักษณะทางกายภาพของหุ่นยนต์ การทำงานภายในเซลการทำงาน รวมไปถึงอุปกรณ์จับสัญญาณที่ใช้อีกด้วย โดยมีรายละเอียดต่างๆ ของไฟล์ดังต่อไปนี้

ไฟล์เอาต์พุตที่ ได้จะแบ่งออกเป็น 2 ประเภท คือ

- ไฟล์ระดับพื้นฐาน (Basic level) จะเป็นไฟล์ที่บันทึกลักษณะการเคลื่อนที่ และค่าของมุมของข้อต่อ หรือ ตำแหน่งการเคลื่อนที่ในพิกัด Cartesian ของหุ่นยนต์ (โดยจะบันทึกตำแหน่งในลักษณะของ ZXZ Euler angles ดังจะได้กล่าวในส่วนต่อไป) โดยไม่บันทึก คำสั่งต่างๆ ที่เป็นเงื่อนไขที่ใช้ในการเคลื่อนที่ เช่น WAIT, IF...ELSE หรือ CASE OF ..... เป็นต้น

- ไฟล์ระดับสูง (Advance level) จะเป็นไฟล์ที่บันทึกคำสั่งต่างๆ ที่เป็นเงื่อนไข แต่จะไม่บันทึกลักษณะการทำงานของหุ่นยนต์ ไฟล์ประเภทนี้จะบรรจุโปรแกรมที่เขียนขึ้นด้วยภาษาระดับสูง โดยรวมคำสั่งต่างๆ ที่สามารถโปรแกรมได้โดยใช้ CATIA

ดังนั้น จากลักษณะงานวิจัยนี้ ซึ่งหุ่นยนต์มีการเคลื่อนที่ภายในเซลการทำงานโดยมีการทำงานที่สัมพันธ์กันของอุปกรณ์จับสัญญาณ เมื่อต้องการโปรแกรมที่ใช้ในการควบคุมหุ่นยนต์จึงต้องใช้ไฟล์ทั้ง 2 ประเภท โดยไฟล์ระดับพื้นฐาน จะใช้ในการหาตำแหน่งการเคลื่อนที่ของหุ่นยนต์ ในขณะที่ไฟล์ระดับสูงจะใช้ในการหาลำดับขั้นการทำงานของอุปกรณ์จับสัญญาณที่สัมพันธ์กันภายในเซลการทำงาน

ข้อมูลในไฟล์เอาต์พุตที่ได้จะแบ่งออกเป็น 3 ถึง 4 ส่วน ขึ้นอยู่กับลักษณะไฟล์ โดย 2 ส่วนแรกจะเป็นข้อมูลทั่วไปที่อธิบายถึงลักษณะการเคลื่อนที่กล่าวคือ

- คอลัมน์ที่ 1 ถึง คอลัมน์ที่ 4 จะเป็นข้อมูลตัวแรกที่ใช้อธิบายถึงคำสั่งแต่ละคำสั่ง เช่น คำสั่งที่ใช้ในการเคลื่อนที่จะได้แก่

CARA Absolute Cartesion motion

CARR Relative Cartesian motion

CARC Curve - following motion

CARF Reconfiguration motion

HOME Return to home motion

JNTA Absolute joint motion

JNTR Relative joint motion

- คอลัมน์ที่ 6 ถึง คอลัมน์ที่ 9 จะเป็นข้อมูลส่วนที่สอง ซึ่งจะให้ข้อมูลเพิ่มเติมของคำสั่งที่ใช้ (เช่น ลักษณะคำนวณการเคลื่อนที่แบบ joint (JNTI) หรือ linear (LINI) interpolation ค่าของความเร็วที่ใช้ (KIN) ซึ่งจะแสดงเป็นเปอร์เซ็นต์ของความเร็วจริง หรือ อธิบายถึงประเภทของข้อมูลที่จะอยู่ในคอลัมน์ถัดไป

- คอลัมน์ที่ 11 ถึงคอลัมน์ที่ 80 จะเป็นข้อมูลในส่วนที่ 3 หรือ 4 โดยจะอธิบายถึงลักษณะของข้อมูลที่ใช้ในการสร้างในโปรแกรม CATIA (เช่น เส้นตรง หรือแกน) หรือ ค่าของตำแหน่งของข้อต่อของหุ่นยนต์

รายละเอียดของข้อมูลต่างๆ จะอธิบายได้ดังต่อไปนี้

4.1.1) ข้อมูลทั่วไปที่มีอยู่ในทั้งไฟล์ระดับพื้นฐานและระดับสูง

จะเป็นข้อมูลที่มีอยู่ในไฟล์ทั้ง 2 ระดับที่ได้กล่าวมาแล้ว คือ

- ลักษณะของหุ่นยนต์ (robot characteristic)

จะประกอบด้วยข้อมูลดังต่อไปนี้

- ชื่อของหุ่นยนต์
- จำนวนข้อต่อ (NJNT)
- จำนวนของดีกรีความเป็นอิสระ (NDOF)
- กลุ่มของเซตทางเรขาคณิตที่เกี่ยวข้องกับการกำหนดหุ่นยนต์
- ระบบแกนอ้างอิงของหุ่นยนต์ (AXIS)
- ประเภทของข้อต่อและขอบเขตการทำงาน (REVL : revolute : PRIS : prismatic ; CPRV : coupling revolute ; SPEC, specialist)

- จลน์ศาสตร์ของปลายแขน (KINR) ซึ่งจะแสดงในรูปของความเร่ง (ACC) หรือ เวลา (TIME)
- จลน์ศาสตร์ของแต่ละข้อต่อ (KINJ) ซึ่งจะแสดงในรูปของความเร็ว (VEL) หรือเวลา (TIME)
- ตำแหน่งเริ่มต้นของหุ่นยนต์ในรูปของ Joint position (JPOS)
- ตำแหน่งเริ่มต้นของปลายแขนของหุ่นยนต์ในรูปของ cartesian position (CPOS)
- ค่าคอนฟิกูเรชั่น (configuration) ที่แสดงในรูปของ cartesian และ joint (CCON, JCON)
- สถานะของการจับชิ้นงาน (GRASP)

- ลักษณะของการวางระบบแกน (description of the positioning axis systems)
- ลักษณะของอุปกรณ์จับสัญญาณ (description of the sensors)

จะประกอบด้วยข้อมูล ของอุปกรณ์จับสัญญาณที่ใช้โดยอาจจะเป็นประเภท logical ที่มีค่าเป็น 0 หรือ 1 หรืออาจจะเป็นประเภท numerical ที่มีค่าเป็นจำนวนจริงใดๆ

SENS	NUM	Identifier N1	Initial value of N1
		Identifier N2	Initial value of N2
		and so forth	
	LOG	Identifier of L1	Initial value of L1
		Identifier of L2	Initial value of L2

รูปที่ 4.1 แสดงรูปแบบของข้อมูล sensor

- ลักษณะของคอนฟิกูเรชั่น (description of configurations)

จะประกอบด้วย คอนฟิกูเรชันที่แสดงในรูป cartesian ซึ่งเป็นการกำหนดคอนฟิกูเรชั่นของหุ่นยนต์ เมื่อหุ่นยนต์เคลื่อนที่ไปถึงจุดทำงานในสภาวะแวดล้อม และคอนฟิกูเรชันที่แสดงในรูป joint ซึ่งเป็นค่าที่แสดงจำนวนรอบของการหมุนของแต่ละข้อต่อ

	CCON	Robot class or	Catesian configuration
		Reference axis system	
	JCON	Robot identifier	Joint numbers(1 to 20)
			Number of revolutions
			(-9 to 9)

รูปที่ 4.2 แสดงรูปแบบของข้อมูล configuration

- เงื่อนไขทางจลน์ศาสตร์ (kinematic condition)

จะประกอบด้วยข้อมูลดังต่อไปนี้

SPD	ACC or TIME	Relevant set or robot	6 numerical values (columns 21 to 80)

STOP			

GRA		Grasping set	1st grasped set	2nd grasped set

REL		Release set	

WAIT	TIME or COND	Duration or Logical expression (70 Characters allowed)

รูปที่ 4.3 แสดงรูปแบบของข้อมูลทางจลน์ศาสตร์

โดยค่าตัวเลขของข้อมูล SPD จะแสดงถึง ความเร็วสัมพัทธ์ ความเร่งและความหน่วง หรือ ความเร็วสมบูรณ์ ความเร็วและความหน่วง

- คุณสมบัติ (attributes)

4.1.2) ข้อมูลสำหรับไฟล์ระดับพื้นฐาน

ข้อมูลในไฟล์ระดับพื้นฐานจะแบ่งไฟล์ออกเป็น 2 ประเภทคือ ไฟล์ประเภท JNTPOS ซึ่งเป็นไฟล์เอาต์พุตที่แสดงตำแหน่งของข้อต่อ และ ไฟล์ประเภท CARTPOS ซึ่งเป็นไฟล์เอาต์พุตที่แสดงตำแหน่งในพิกัด cartesian

a) ไฟล์ประเภท JNTPOS (joint position) จะแสดงค่าของแต่ละข้อต่อของหุ่นยนต์ในการเคลื่อนที่ โดยมี 3 คอลัมน์ คอลัมน์แรกจะเป็นลักษณะการเคลื่อนที่ประเภทต่างๆ ที่ได้แสดงในข้างต้น คอลัมน์ที่ 2 จะเป็นการอธิบายลักษณะเพิ่มเติมของการเคลื่อนที่และคอลัมน์ที่ 3 จะอธิบายถึงค่าของมุมในแต่ละข้อต่อของหุ่นยนต์ดังแสดงในรูปที่ 4.4

ROB		Robot identifier(truncated to 8 characters)
		initial joint value
CARA	LINI	joint values after the motion is executed
	KIN	Relative speed value
CARR	JNTI	joint values
	ATR	Character string
HOME	LINI	joint value equal to zero
JNTA	JNTI	joint value
JNTR	LINI	joint value
	KIN	Relative speed value

รูปที่ 4.4 ตัวอย่างข้อมูลของไฟล์ประเภท joint position

b) ไฟล์ประเภท CARTPOS (cartesian position) จะแสดงค่าของตำแหน่งในพิกัด cartesian ของหุ่นยนต์ โดยมีทั้งหมด 4 คอลัมน์ คือ คอลัมน์แรกจะเป็นลักษณะการเคลื่อนที่ประเภทต่างๆ คอลัมน์ที่ 2 จะเป็นการอธิบายลักษณะเพิ่มเติม คอลัมน์ที่ 3 จะอธิบายถึงลักษณะของ configuration ของหุ่นยนต์ และคอลัมน์ที่ 4 จะอธิบายถึงตำแหน่งที่แสดงในพิกัด Eulerian ดังแสดงในรูปที่ 4.5

ROB		Robot identifier	
CARA	LINI	Ident AS2	Position of IAS2 with reference to AS2
			Position of AS1 with reference to AS0
		Set identifier	Position of IAS2 with reference to AS0
			Position of AS3 with reference to AS0
	KIN	Relative speed	
CARR	JNTI		Position of AS1 with reference to AS0
			Position of AS2 with reference to AS0
			Position of AS3 with reference to AS0
JNTA	JNTI		Position of AS1 with reference to AS0
			Position of AS2 with reference to AS0
			Position of AS3 with reference to AS0

รูปที่ 4.5 ตัวอย่างข้อมูลของไฟล์ประเภท Cartesian Position

4.1.3) ข้อมูลสำหรับไฟล์ระดับสูง

สำหรับข้อมูลในไฟล์เอาต์พุตประเภทนี้ นอกจากข้อมูลที่แสดงถึงการเคลื่อนที่โดยทั่วๆ ไปแล้ว ยังจะมีข้อมูลที่เกี่ยวข้องกับ เงื่อนไข และลำดับการทำงานต่างๆ ของหุ่นยนต์ภายในเซลการทำงาน รวมไปถึงข้อมูลของตัวแปร (variable) ที่กำหนดขึ้นเพื่อนำมาใช้ในลักษณะงานใดงานหนึ่ง ดังรายละเอียดต่อไปนี้

- ข้อมูลที่แสดงถึงการกำหนดตัวแปร (Declaration of Variable) โดยจะเป็นการกำหนดประเภทของตัวแปร

- ข้อมูลที่แสดงถึงการทำงานทาง arithmetic ของอุปกรณ์จับสัญญาณหรือตัวแปร (Arithmetical Instruction)

- ข้อมูลที่แสดงถึงค่าชั่งทางด้านเงื่อนไข (Logical Instruction) ซึ่งมีรูปแบบดังแสดงในรูปที่ 4.6

CAS		Logical expression	
DO			
	(Inst'ns)		
COND		Logical expression	
DO			
	(Inst'ns)		
COND		Logical expression	
DO			
	(Inst'ns)		
....			
ANY			
	(Inst'ns)		
ENDC			

IF		Logical expression	
THEN			
	(Inst'ns)		
ELSE			
	(Inst'ns)		
ENDI			

WHL DO ENDW	(Inst'ns)	Logical expression	

FOR RPT NEXT	(Instructions)	Identifier of the variable	1st value of identifiers of the variable	2nd value of identifiers of the variable

BRAN		Label identifier

LAB		Label identifier

รูปที่ 4.6 แสดงรูปแบบของคำสั่งทางด้านเงื่อนไขในไฟล์ระดับสูง

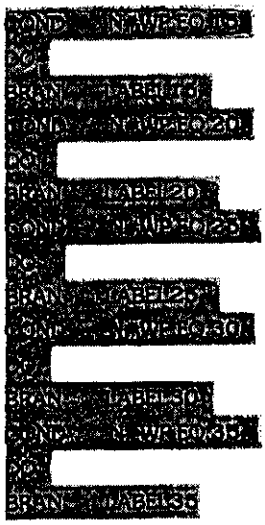
จากการจำลองการทำงานภายในระบบผลิตอัตโนมัติที่ใช้ในการวิจัยนี้ จะพบว่าลักษณะของไฟล์เอาต์พุตที่ได้จะเป็นไฟล์ระดับสูง เนื่องจากว่ามีการส่งข้อมูลของอุปกรณ์จับสัญญาณระหว่างสถานีต่างๆ โดยในขั้นตอนจำลองการทำงานจะทำการกำหนดให้สถานีต่างๆ มีคุณสมบัติเป็นหุ่นยนต์ตัวหนึ่ง (ดังนั้นภายในระบบจะมีหุ่นยนต์อยู่ 4 ตัว คือ สถานีสายพานลำเลียง สถานีปฏิบัติการ หุ่นยนต์ MITSUBISHI และสถานีแยกชิ้นงาน) จากนั้นจึงทำการกำหนดค่าของอุปกรณ์จับสัญญาณที่ใช้ โดยอุปกรณ์จับสัญญาณที่ใช้จะมีอยู่ 2 ประเภท คือ อุปกรณ์จับสัญญาณที่ใช้ในการรับ หรือ ส่งระหว่างสถานี จะเป็นอุปกรณ์จับสัญญาณประเภทลอจิก (logic) ที่มีค่าเป็น 0 หรือ 1 (L#1, L#2, L#3, L#4) และอุปกรณ์จับสัญญาณที่ใช้ในการแยกประเภทของชิ้นงานซึ่งเป็นอุปกรณ์จับสัญญาณประเภทตัวเลข (numerical) (N#1,N#WP) หลังจากเสร็จสิ้นการจำลองการทำงาน ไฟล์เอาต์พุตที่ได้จากโปรแกรม CATIA ของหุ่นยนต์แต่ละตัวจะแสดงถึงเงื่อนไขการทำงานของอุปกรณ์จับสัญญาณระหว่างสถานี โดยจะบอกถึงประเภทและสถานะของอุปกรณ์จับสัญญาณที่สัมพันธ์กันดังแสดงในรูปที่ 4.7, 4.8, 4.9, 4.10 ซึ่งแสดงตัวอย่างไฟล์เอาต์พุตที่ได้จากสถานีสายพานลำเลียง สถานีปฏิบัติการ หุ่นยนต์ MITSUBISHI และ สถานีแยกชิ้นงาน จากไฟล์ที่ได้ดังกล่าวนั้น ไฟล์เอาต์พุตของสถานีหุ่นยนต์สามารถนำไปแปลงเป็นคำสั่งที่ใช้ควบคุมหุ่นยนต์ได้โดยใช้โพสต์โปรเซสเซอร์ที่มีอยู่ได้ ในขณะ

ที่ไฟล์เอาต์พุตของสถานีที่เหลือจะต้องทำการพัฒนาโพสต์โปรเซสเซอร์เพิ่มเติมเพื่อแปลงไฟล์ที่ได้ไปเป็นภาษาควบคุม PLC ซึ่งจะทำให้สามารถควบคุมระบบทั้งหมดได้ด้วยวิธี off-line programming

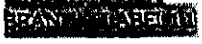
```
DATE : MONDAY JUNE 16, 1997        14.14.07
FILE : /home/catadm/db
MODEL : MY THESIS : FMS-SYSTEM
TASK : TASK-STATION3/4
CONTROLLER : *CTL3
TYPE : ADVANCED
REL      PIECES1
[illegible]
[illegible]
HOME JNTI STATION3
   KIN     100.000
[illegible]
GRA      ST3/4-2  PIECES2
JNTA JNTI STATION3    270.000
   KIN     100.000
REL      PIECES2
[illegible]
[illegible]
HOME JNTI STATION3
   KIN     100.000
[illegible]
GRA      ST3/4-2  PIECES3
JNTA JNTI STATION3    405.000
   KIN     100.000
```

รูปที่ 4.7 แสดงไฟล์เอาต์พุตที่ได้จากการจำลองการทำงานของสถานีสายพานลำเลียง

```
DATE : MONDAY JUNE 16, 1997        14.14.59
FILE : /home/catadm/db
MODEL : MY THESIS : FMS-SYSTEM
TASK : TASK-STATION5
CONTROLLER : *CTL4
TYPE : ADVANCED
LOC  INT  COUNT
DATA INT  COUNT        0
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```



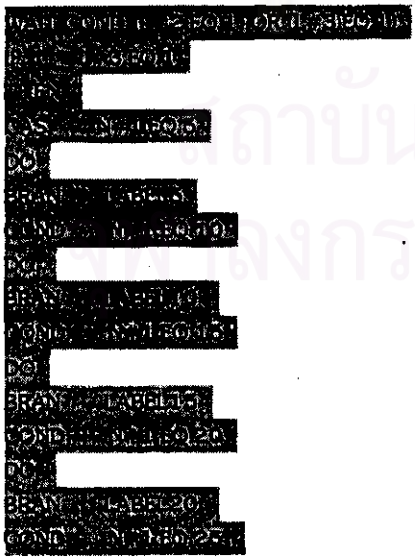
ENDC
LAB LABEL5
ATR WORKPIECE-1
GRA ST5-2 PIECES1
JNTR JNT1 STATION5 90.000
KIN 100.000



LAB LABEL10

รูปที่ 4.8 แสดงไฟล์เอาต์พุตที่ได้จากการจำลองการทำงานของสถานีปฏิบัติการ

DATE : MONDAY JUNE 16, 1997 16.06.24
FILE : /home/catadm/db_s
MODEL : MY THESIS : FMS-SYSTEM
TASK : TASK-MISUBISHI
CONTROLLER : *CTL1
TYPE : ADVANCED



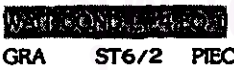
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
ENDI
ENDI
LAB     LABEL5
   ATR  MAGAZINE1:SCRAP-PART
CARA JNTI TGRIP     339.100  198.400  282.500  -50.315  103.087  -90.000
      TENV      324.500  302.400  215.300    .000   90.000  -90.000
   KIN    100.000
CARA JNTI TGRIP     339.100  198.400  282.500
   TSL  TENV      324.500  302.400  175.200
   KIN    100.000
CARA JNTI TGRIP     339.100  198.400  282.500  -50.315  103.087  -90.000
      TENV     -176.000  -23.700  235.400   90.000   90.000  -90.000
   CCON          ELBOW    BELOW
   KIN    100.000
REL     PIECES1
```

```
BRAN    LABEL-A
```

รูปที่ 4.9 แสดงไฟล์เอาต์พุตที่ได้จากการจำลองการทำงานของสถานีหุ่นยนต์

```
DATE : MONDAY  JUNE 16, 1997       14.15.34
FILE : /home/catadm/db
MODEL : MY THESIS : FMS-SYSTEM
TASK  : TASK-STATION6
CONTROLLER : *CTL5
TYPE  : ADVANCED
CMT
VERS    V4R1.6
ACEL    *AXS1
RPOS    ST6/2
      *AXS285    191.300  282.800  137.700   90.000    .000    .000
CMT
```



```
GRA     ST6/2   PIECES5
JNTA JNTI STATION6   579.500
   KIN    100.000
REL     PIECES5
HOME JNTI STATION6
   KIN    100.000
END
```

รูปที่ 4.10 แสดงไฟล์เอาต์พุตที่ได้จากการจำลองการทำงานของสถานีแยกชิ้นงาน

4.2) ความหมายของค่า parameter ต่าง ๆ (Characteristic of motion)

4.2.1) การควบคุมการเคลื่อนที่ (Motion Control)

สามารถแบ่งลักษณะการเคลื่อนที่ออกได้เป็น 2 ประเภท คือ

4.2.1.1) การควบคุมแบบจุดต่อจุด (Point to Point Control : PTP)

ซึ่งหมายถึง การควบคุมทางเดินให้ผ่านจุดต่างๆ ที่กำหนดไว้

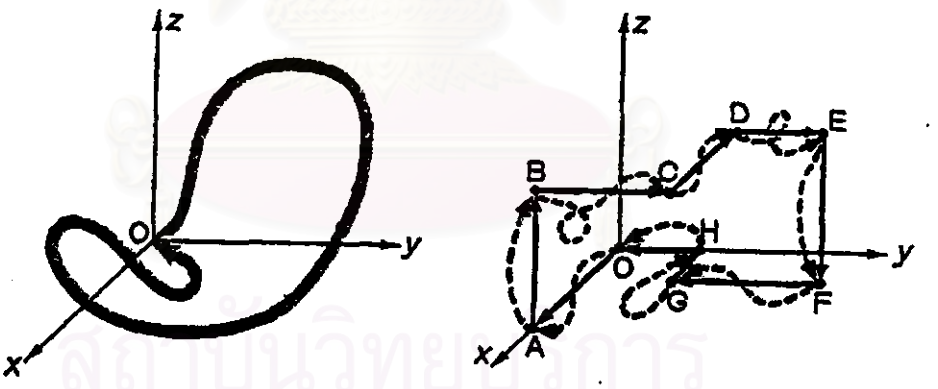
4.2.1.2) การควบคุมทางเดินต่อเนื่อง (Continuous Path Control : CP)

(ข้อมูล CARC ในคอลัมน์ที่ 1)

ซึ่งหมายถึง การควบคุมทางเดินตลอดการเคลื่อนที่ที่กำหนด

ข้อแตกต่างระหว่างการควบคุมทั้ง 2 แบบจะอยู่ที่ ในการควบคุมแบบ CP จะต้องมีการกำหนดจุดต่างๆต่อเนื่องกันเป็นทางเดิน นอกจากนั้นยังกำหนดเวลาที่ผ่านจุดต่างๆ เหล่านี้ด้วย ในขณะที่การควบคุมแบบ PTP จะกำหนดเฉพาะจุดที่ต้องผ่าน แต่ไม่ได้กำหนดทางเดินระหว่างจุด และเวลา

รูปที่ 4.11 จะอธิบายความแตกต่างดังกล่าว สำหรับการเคลื่อนที่แบบ PTP จะไม่มีการกำหนดทางเดินระหว่างจุด ถ้าวิ่งตามเส้นทึบที่แสดงในรูปได้จะดีที่สุด แต่ในทางปฏิบัติอาจทำไม่ได้ เพราะความอิสระ (degree of freedom) ในแกนอื่น อาจมีการเคลื่อนที่ ทำให้ทางเดินไม่ตรงดังในรูปก็ได้

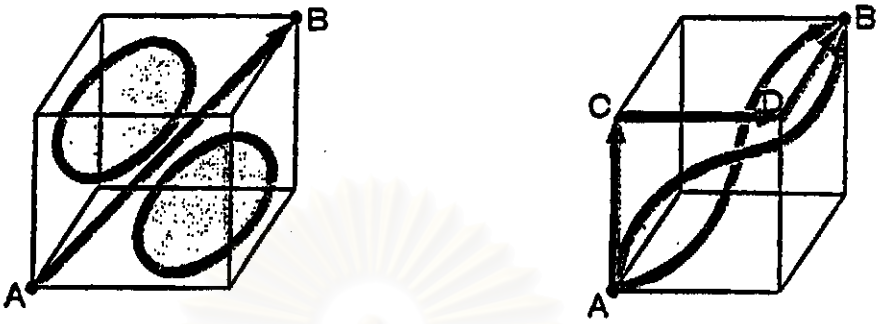


ก) การควบคุม CP ข) การควบคุม PTP

รูปที่ 4.11 เปรียบเทียบการควบคุมแบบ CP กับการควบคุม PTP

ในการเคลื่อนไหวแขนของหุ่นยนต์จากจุดหนึ่งไปสู่อีกจุดหนึ่งนั้น ระหว่างกลาง อาจมีสิ่งกีดขวาง หุ่นยนต์จะต้องหลบหลีกสิ่งกีดขวางนี้ ดังในรูปที่ 4.12 ต้องการเคลื่อนที่จากจุด A ไปจุด B โดยระหว่างกลางมีสิ่งกีดขวาง การควบคุมแบบ CP จะทำให้แขนของหุ่นยนต์หลบ

หลีกสิ่งกีดขวางได้ง่าย แต่การควบคุมแบบ PTP ก็สามารถหลบหลีกได้เช่นเดียวกัน โดยการเพิ่มจุดในระหว่างทางเดิน ซึ่งอาจจะทำให้การเคลื่อนที่ไปสู่เป้าหมายเสียเวลามากขึ้น



ก) เมื่อมีสิ่งกีดขวาง การควบคุม CP จะควบคุมผ่านสิ่งกีดขวางเป็นเส้นตรง

ข) การหลบสิ่งกีดขวางทำได้โดยการเพิ่มจุดผ่าน A - C - D - B

รูปที่ 4.12 แสดงข้อแตกต่างของการเลือกใช้งานการควบคุมแบบ CP กับการควบคุมแบบ PTP

ถ้าเราเพิ่มจุดที่จะต้องผ่านในทางเดินให้กับการควบคุม PTP ให้มากขึ้น ให้ระยะห่างระหว่างจุดน้อยที่สุด การควบคุมแบบ PTP จะเข้าใกล้การควบคุมแบบ CP อย่างไรก็ตาม ผลของการควบคุมก็ยังไม่เหมือนการควบคุมแบบ CP อยู่ดี เนื่องจากความแม่นยำในการกำหนดตำแหน่ง ยังแตกต่างกันมาก นอกจากนี้ถ้าการเคลื่อนที่ระหว่างจุดมีการหน่วงเวลา เนื่องจากความเฉื่อย ก็อาจส่งผลทำให้ระบบเกิดการสั่นขึ้นได้

ประเภทของการเคลื่อนที่แบบจุดต่อจุด (Point to Point Motion)

- Absolute or relative Cartesian motion (ข้อมูล CARA หรือ CARR ในคอลัมน์ที่ 1)

ซึ่งจะแสดงค่าของตำแหน่งสุดท้ายในการเคลื่อนที่ในระบบพิกัด cartesian

- Absolute Cartesian motion

เป็นการเคลื่อนที่ซึ่งตำแหน่งสุดท้ายในการเคลื่อนที่ สามารถกำหนดโดยให้ปลายของอุปกรณ์จับของหุ่นยนต์ทับกับตำแหน่งดังกล่าวในสภาวะแวดล้อม

- Relative Cartesion motion

เป็นการเคลื่อนที่ซึ่งตำแหน่งสุดท้ายในการเคลื่อนที่สามารถกำหนดโดยระยะทาง (displaccment) ของปลายของอุปกรณ์จับหุ่นยนต์ เมื่อเทียบกับตำแหน่งปัจจุบัน

- Absolute or relative Joint motion (ข้อมูล JNTA หรือ JNTR ในคอลัมน์ที่ 1)

ซึ่งจะแสดงค่าของตำแหน่งสุดท้ายโดยอธิบายถึง ค่าของแต่ละข้อต่อของหุ่นยนต์

- Absolute Joint motion

เป็นการเคลื่อนที่ ซึ่งเทียบกับตำแหน่งอ้างอิง (reference home position) ซึ่งมีค่าของมุมของข้อต่อทั้งหมดมีค่าเป็น 0

- Relative Joint motion

เป็นการเคลื่อนที่ ซึ่งเทียบกับตำแหน่งปัจจุบัน

4.2.2) การคำนวณทางเดินโดยการ Interpolate (Path Interpolation Mode)

ในการเคลื่อนที่แบบจุดต่อจุด (PTP) นั้น จะมีการคำนวณเพื่อหาว่าหุ่นยนต์จะเคลื่อนที่ไปจุดต่างๆได้อย่างไร โดยแบ่งออกเป็น 2 ประเภทคือ

4.2.2.1) Joint interpolation mode (ข้อมูล JNTI ในคอลัมน์ที่ 2)

เป็นการคำนวณหาทางเดินการเคลื่อนที่ โดยใช้ลักษณะโครงสร้างของหุ่นยนต์ และการเคลื่อนที่ไปพร้อมๆ กันของข้อต่อที่ควบคุมได้ (controllable joints) พารามิเตอร์ที่ใช้ในการเคลื่อนที่ จะเป็นความเร็ว ความเร่ง ของข้อต่อ

4.2.2.2) Linear interpolation mode (ข้อมูล LINI ในคอลัมน์ที่ 2)

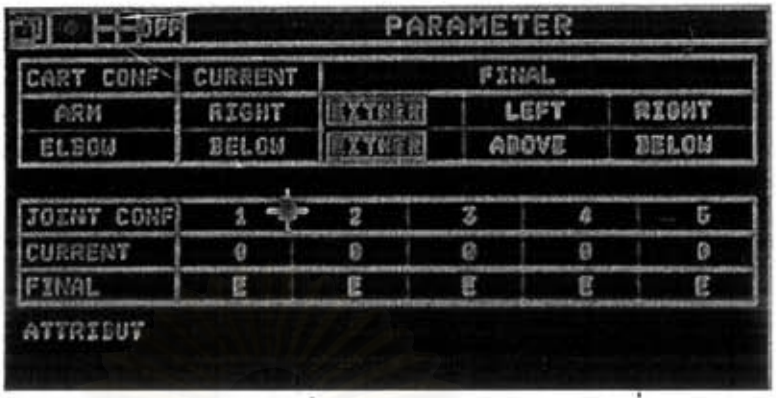
เป็นการคำนวณหาทางเดิน โดยจะได้ลักษณะของทางเดินเป็นส่วนของเส้นตรงต่อเนื่องกันระหว่างตำแหน่งปัจจุบันและตำแหน่งสุดท้ายของจุดปลายอุปกรณ์จับของหุ่นยนต์ (piloted point) พารามิเตอร์ที่ใช้ในการเคลื่อนที่ จะเป็นความเร็ว ความเร่ง ของหน้าแปลน (flange)

4.2.3) ลักษณะของหุ่นยนต์ (Robot Configuration) (ข้อมูล CCON หรือ JCON ในคอลัมน์ที่ 2)

เป็นการกำหนดลักษณะการเคลื่อนที่ของหุ่นยนต์ ณ จุดใดจุดหนึ่ง เนื่องจากว่า ณ จุดหนึ่งๆ นั้นหุ่นยนต์จะสามารถเคลื่อนที่ได้หลายลักษณะขึ้นอยู่กับการกำหนดการเคลื่อนที่ของแต่ละข้อต่อ ดังนั้น เพื่อบอกให้หุ่นยนต์ทราบถึงลักษณะการเคลื่อนที่ของแต่ละลิงค์ รวมไปถึงลักษณะการทำงานเมื่อหุ่นยนต์เคลื่อนที่จากจุด singular point ไปยังจุดทำงานใดๆ จึงต้องมีการกำหนดค่าคอนฟิกูเรชั่น ดังกล่าว

ค่าคอนฟิกูเรชั่นจะแบ่งออกเป็น 2 ประเภท คือ

4.2.3.1) ค่าคอนฟิกูเรชั่นที่แสดงในรูป cartesian ซึ่งมีการกำหนดค่าต่างๆ ดังแสดงในรูปที่ 4.13



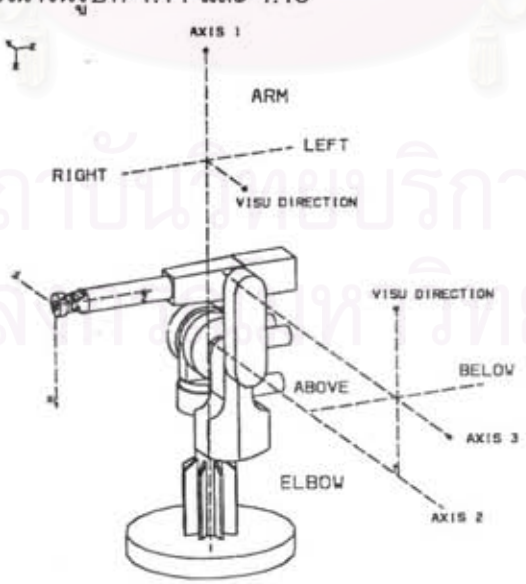
รูปที่ 4.13 แสดงค่า parameter ในการกำหนดค่าคอนฟิกูเรชั่นของหุ่นยนต์

โดยจากรูปนั้น เราสามารถกำหนดค่าคอนฟิกูเรชั่นของแขน (arm) ข้อศอก (elbow) และข้อมือ (wrist) ของหุ่นยนต์ได้ดังต่อไปนี้

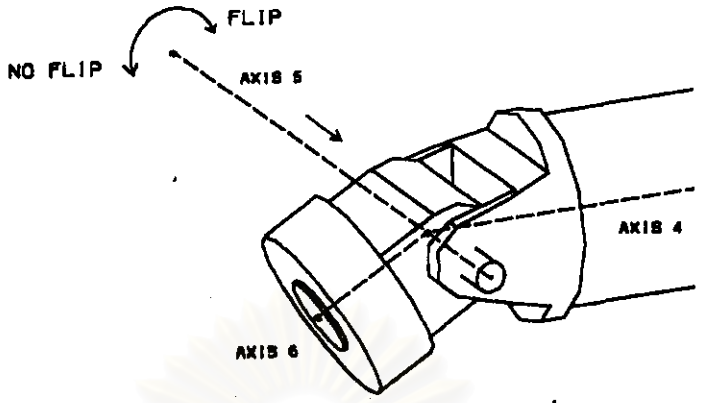
- arm configuration โดยการพิจารณาตำแหน่งของการมอง และทิศทางที่ใช้มอง จากนั้นจึงทำการตรวจสอบจุดตัดของแกนของข้อมือของหุ่นยนต์ เมื่อเทียบกับทิศทางที่ใช้มองว่าอยู่ทางซ้ายหรือขวา (LEFT OR RIGHT)

- elbow configuration โดยการพิจารณาในทำนองเดียวกัน และตรวจสอบตำแหน่งของจุดตัดว่าอยู่ด้านบนหรือด้านล่าง (ABOVE OR BELOW) (ข้อต่อที่พิจารณาอยู่เหนือหรือต่ำกว่าจุดตัดที่ปลายของหุ่นยนต์)

- wrist configuration โดยจะเป็นการพิจารณาทิศทางการหมุนของแกนข้อมือ โดยถ้าการหมุนเป็นทิศทางบวกจะเรียกว่า NO FLIP และถ้าเป็นทิศทางลบจะเรียกว่า FLIP ดังแสดงในตัวอย่างการพิจารณาในรูปที่ 4.14 และ 4.15



รูปที่ 4.14 แสดงวิธีการหาค่า cartesian คอนฟิกูเรชั่นของแขน (arm) และไหล่ (elbow)



รูปที่ 4.15 แสดงวิธีการหาค่า cartesian คอนฟิกูเรชั่นของข้อมือ (wrist)

4.2.3.2) ค่าคอนฟิกูเรชั่นที่แสดงในรูป joint ซึ่งจะกำหนดเป็นจำนวนรอบของการเคลื่อนที่ของแต่ละ joint ของหุ่นยนต์โดยมีค่าตั้งแต่ -9 ถึง 9

4.3) ลักษณะของเอาต์พุตไฟล์ ที่ได้จากการแปลงโดยใช้โพสต์โปรเซสเซอร์

4.3.1) ขั้นตอนการทำงานของหุ่นยนต์ภายในระบบ MPS

ดังที่ได้กล่าวมาแล้วว่า ภายในระบบ MPS นี้หุ่นยนต์จะทำหน้าที่เป็นอุปกรณ์ลำเลียงชิ้นงาน เพื่อจัดเก็บ ดังนั้นภายในระบบ MPS จึงต้องมีอุปกรณ์ตรวจสอบที่ใช้ตรวจสอบประเภท ขนาด สี และลักษณะที่ต้องการของชิ้นงาน และข้อมูลดังกล่าวจะถูกส่งผ่านระบบ bus ไปยังหุ่นยนต์โดยผ่าน I/O พอร์ต ข้อมูลที่ส่งดังกล่าวจะเป็นข้อมูลขนาด 5 บิต ซึ่งจะแสดงถึงลักษณะการทำงานต่างๆ ของหุ่นยนต์ โดยจะแสดงในรูปแบบฐาน 10 รายละเอียดต่อไปนี้

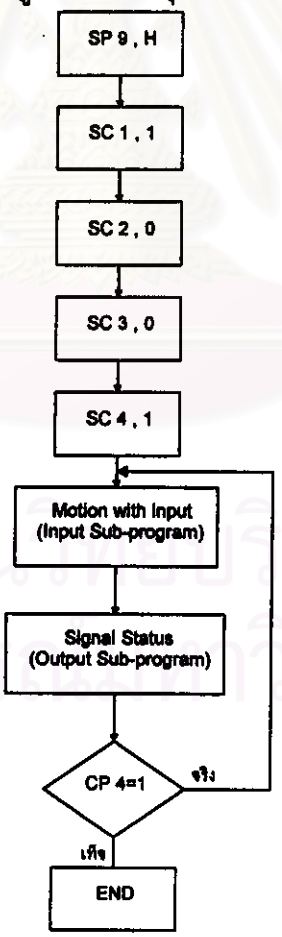
0 ไม่เกิดการทำงาน
1 เคลื่อนที่ไปยังตำแหน่งเริ่มการทำงาน (nest position)
2 เคลื่อนที่ไปยังตำแหน่งเริ่มต้น (initial position)
3 หยิบชิ้นงานจากสถานีพักชิ้นงาน
4 หยิบชิ้นงานจากสถานีปฏิบัติการ
5 วางชิ้นงานที่สถานีปฏิบัติการ
6 วางชิ้นงานที่สถานี เรียงชิ้นงาน
7 วางชิ้นงานที่แมกกาซีน 1
8 วางชิ้นงานที่แมกกาซีน 2
9 วางชิ้นงานที่แมกกาซีน 3
10 วางชิ้นงานที่แมกกาซีน 4
11 สภาวะของสัญญาณ

ในขณะเดียวกัน เมื่อหุ่นยนต์กำลังทำงาน ก็จะส่งสัญญาณ ซึ่งแสดงสภาวะการทำงานต่างๆ ออกมา โดยข้อมูลดังกล่าวจะเป็นข้อมูลขนาด 4 บิต โดยจะแสดงในรูปเลขฐาน 10 ดังต่อไปนี้

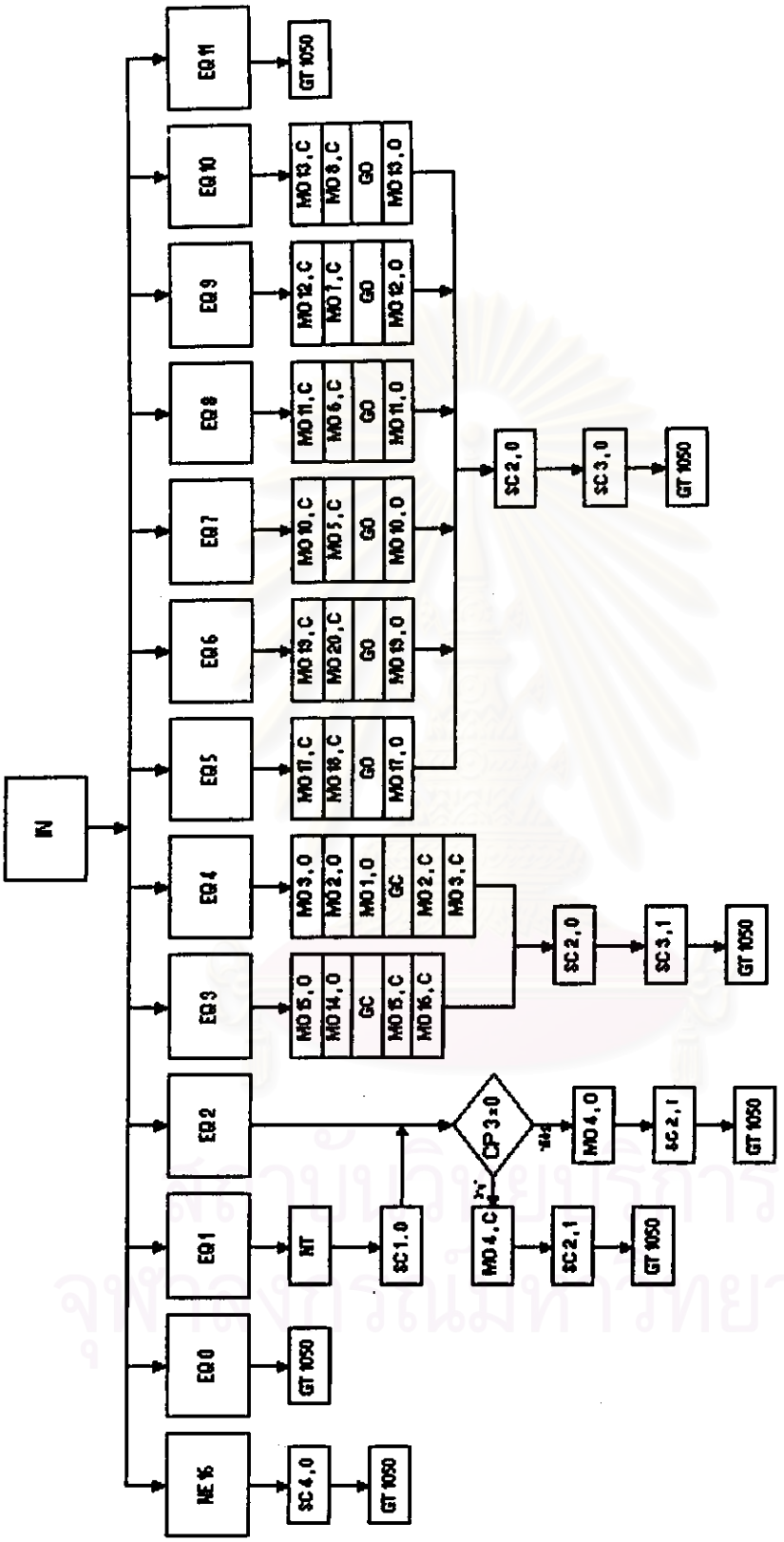
1 ต้องการการเคลื่อนที่ไปยังตำแหน่งเริ่มการทำงาน
2 อยู่ตำแหน่งเริ่มต้น ไม่ได้หยิบชิ้นงาน
3 อยู่ตำแหน่งเริ่มต้น หยิบชิ้นงาน
4 ไม่อยู่ในตำแหน่งเริ่มต้น ไม่ได้หยิบชิ้นงาน
5 ไม่อยู่ในตำแหน่งเริ่มต้น หยิบชิ้นงาน

ขั้นตอนการทำงานของหุ่นยนต์ จะสามารถแสดงเป็น flow chart ได้ ดังแสดงในรูปที่ 4.16 , 4.17 , 4.18 โดยจะแบ่ง flowchart เป็น 3 ส่วนคือ

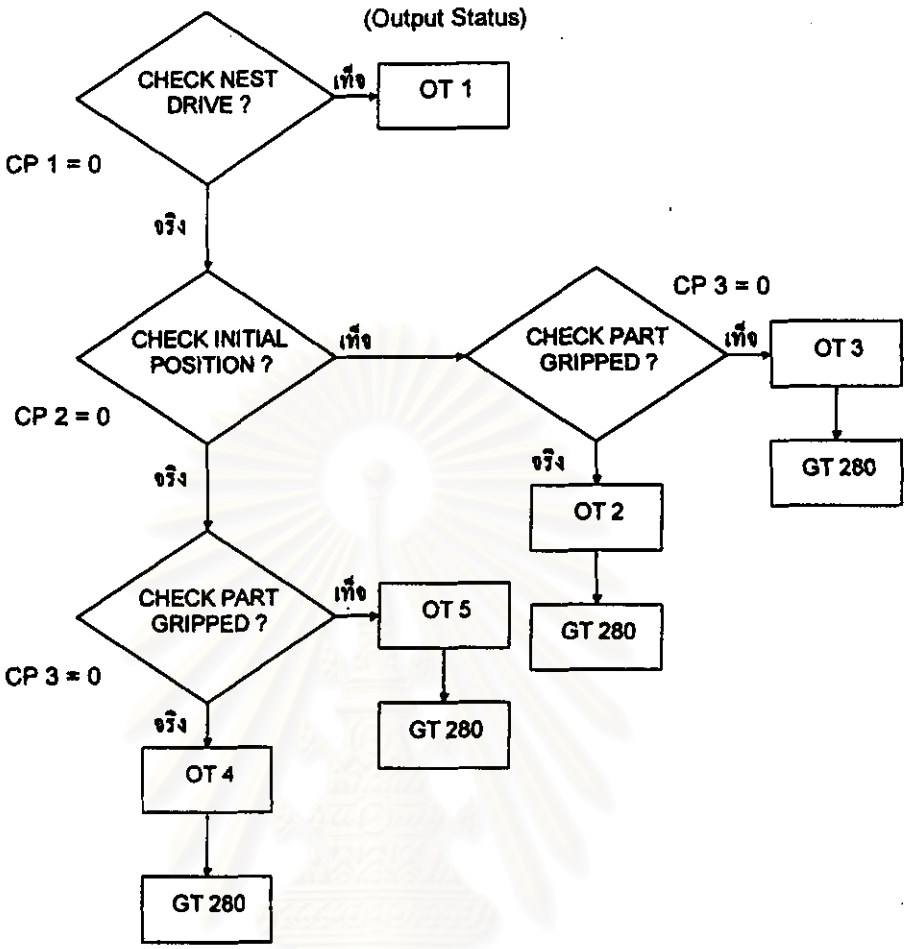
- ส่วนโปรแกรมหลัก (main program)
- ส่วนโปรแกรมข้อมูลของอินพุต (input program)
- ส่วนโปรแกรมข้อมูลของเอาต์พุต (output program)



รูปที่ 4.16 แสดงแผนผังของส่วนโปรแกรมหลัก (main program)



รูปที่ 4.17 แสดงแผนผังของส่วนโปรแกรมข้อมูลของอินพุต (input program)



รูปที่ 4.18 แสดงแผนผังของส่วนโปรแกรมข้อมูลของเอาต์พุต (output program)

จากแผนผังที่ 1 (main program) ซึ่งแสดงถึง ขั้นตอนการทำงานของโปรแกรมหลัก จะเริ่มจากการกำหนดความเร็วและความเร่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ให้มีค่าสูงสุด (SP 9,H) จากนั้นจึงทำการกำหนดค่าเริ่มต้นให้กับเคาน์เตอร์ของตัวควบคุมหุ่นยนต์ เพื่อใช้ในการตรวจสอบสถานะการทำงานของหุ่นยนต์ โดยจะทำการ load ค่าจำนวนจริงในเลขฐานสิบ คือ ค่า 1,0,0,1 ให้กับเคาน์เตอร์ที่ 1,2,3,4 จากนั้นจึงเข้าสู่ลูปการทำงานของหุ่นยนต์ เมื่อมีข้อมูลอินพุต (motion with input) โดยมีการทำงานดังจะได้กล่าวในแผงผังที่ 2 เมื่อเสร็จสิ้นการทำงาน จึงเข้าสู่ลูบการส่งข้อมูลเอาต์พุตเพื่อแสดงสภาวะการทำงานของหุ่นยนต์ และสุดท้ายจึงทำการตรวจสอบการสิ้นสุดการทำงานของโปรแกรม

จากแผนผังที่ 2 (input program) ซึ่งจะแสดงการทำงานของหุ่นยนต์เมื่อมีข้อมูลอินพุตจากอุปกรณ์ภายนอก จะเริ่มขั้นตอนการทำงานด้วยการรับสัญญาณอินพุตจากโปรฟิบัสผ่าน I/O พอร์ต เข้ามาเก็บไว้ในรีจิสเตอร์ (register) และนำค่าที่เก็บดังกล่าวไปเปรียบเทียบกับค่า

เคานน์เตอร์ตามเงื่อนไขที่กำหนดไว้ และเมื่อค่าในรีจิสเตอร์ตรงกับเงื่อนไขใด โปรแกรมก็จะข้ามไปทำงานยังบรรทัดนั้นๆ เช่น ถ้าข้อมูลในรีจิสเตอร์มีค่าเท่ากับ 5 (EQ 5) โปรแกรมจะข้ามไปทำงานซึ่งเป็นการหยิบชิ้นงานจากสถานีพักชิ้นงานไปงานลงบนสถานีปฏิบัติการ จากแผนผังจะสังเกตเห็นสภาวะการทำงานเป็น 6 ส่วนคือ ในส่วนแรก เมื่อเปรียบเทียบค่าในรีจิสเตอร์มีค่าเป็น 0 จะแสดงว่าไม่เกิดการทำงาน ในส่วนที่ 2 เมื่อเปรียบเทียบมีค่าเป็น 1 จะเป็นการสั่งให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งเริ่มการทำงาน (nest position) และทำการเปลี่ยนค่าของเคาน์เตอร์ที่ 1 ให้มีค่าเป็น 0 ซึ่งเป็นการแสดงว่าได้เคลื่อนที่ไปยังตำแหน่งเริ่มการทำงานแล้ว จากนั้นจึงเคลื่อนที่ไปยังตำแหน่งเริ่มต้น (initial position) โดยก่อนหน้านี้จะทำการตรวจสอบเคาน์เตอร์ที่ 3 ซึ่งแสดงถึงสภาวะการหยิบหรือไม่หยิบชิ้นงาน ในส่วนที่ 3 เมื่อเปรียบเทียบมีค่าเป็น 2 ก็จะเป็นการเคลื่อนที่ไปยังตำแหน่งเริ่มต้น (initial position) และทำการตรวจสภาวะการหยิบชิ้นงานดังที่ได้กล่าวในส่วนที่ 2 ในส่วนที่ 4 เมื่อเปรียบเทียบมีค่าเป็น 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 จะเป็นการเคลื่อนที่ของหุ่นยนต์ในการหยิบและวางชิ้นงานในสถานีต่างๆ โดยเมื่อมีการเคลื่อนที่และเกิดการหยิบชิ้นงานขึ้น จะมีการเปลี่ยนค่าของเคาน์เตอร์ที่เกี่ยวข้อง โดยเมื่อมีการเคลื่อนที่หยิบชิ้นงานจากสถานีพักชิ้นงาน และสถานีปฏิบัติงาน จะต้องเปลี่ยนค่าเคานเตอร์ที่ 2 ให้มีค่าเป็น 0 และเคาน์เตอร์ที่ 3 ให้มีค่าเป็น 1 เพื่อแสดงสภาวะของหุ่นยนต์ว่าไม่อยู่ในตำแหน่งเริ่มต้น และอยู่ในสภาวะการหยิบชิ้นงาน ในขณะที่เมื่อมีการวางชิ้นงานที่แมกกาซีนหรือ สถานีแยกชิ้นงาน ก็จะต้องมีการเปลี่ยนค่าเคาน์เตอร์ที่ 2 ให้มีค่าเป็น 0 และเคาน์เตอร์ที่ 3 ให้มีค่าเป็น 0 เพื่อแสดงสภาวะของหุ่นยนต์ว่าไม่อยู่ในตำแหน่งเริ่มต้นและไม่มีการหยิบชิ้นงาน ในส่วนที่ 5 จะเป็นการเปรียบเทียบค่าในรีจิสเตอร์กับค่า 11 ซึ่งเป็นการแสดงสถานะของหุ่นยนต์ ดังจะได้กล่าวในส่วนของโปรแกรมเอาต์พุต (output program) ต่อไป ในส่วนที่6 จะเป็นการตรวจสอบการสิ้นสุดการทำงานของโปรแกรม โดยเมื่อเปรียบเทียบค่าในรีจิสเตอร์มีค่าเป็น 16 โปรแกรมจะไปทำการเปลี่ยนค่าของเคาน์เตอร์ที่ 4 ให้มีค่าเป็น 0 ซึ่งแสดงถึงการสิ้นสุดการทำงานในของโปรแกรม

จากแผนผังที่ 3 (output program) ซึ่งจะแสดงถึงค่าเอาต์พุตที่เกิดขึ้นในแต่ละสภาวะการทำงาน จะเริ่มต้นด้วยการเปรียบเทียบค่าเคาน์เตอร์ที่ 1 (CP 1=0) เพื่อตรวจสอบดูสภาวะการเคลื่อนที่ไปยังตำแหน่งเริ่มการทำงาน จากนั้นจึงทำการเปรียบเทียบค่าเคาน์เตอร์ที่ 2 (CP 2=0) เพื่อตรวจสอบดูสภาวะของตำแหน่งเริ่มต้นของหุ่นยนต์ว่าอยู่ในตำแหน่งเริ่มต้นหรือไม่ ซึ่งจากสภาวะของหุ่นยนต์ที่อยู่หรือไม่อยู่ในตำแหน่งเริ่มต้น ก็จะทำการเปรียบเทียบค่าเคาน์เตอร์ที่ 3 (CP 3=0) เพื่อทำการตรวจสอบสภาวะการหยิบหรือไม่หยิบชิ้นงาน ดังนั้นเราจึงได้ค่าเอาต์พุตที่แสดงถึงสภาวะการทำงานของหุ่นยนต์ในขณะใดขณะหนึ่ง

4.3.2) ขั้นตอนการเขียนโพสต์โปรเซสเซอร์

จากที่ได้กล่าวในหัวข้อที่ 4.1 ว่าไฟล์เอาต์พุตที่ได้จากโปรแกรม CATIA จะแบ่งออกเป็น 2 ประเภท คือ ไฟล์ระดับพื้นฐาน และไฟล์ระดับสูง ซึ่งแตกต่างกันในเรื่องของคำสั่งเงื่อนไข โดยในระบบ MPS ที่ใช้ในการวิจัยนี้ ไฟล์ที่ได้จะเป็นไฟล์ระดับสูง คือเป็นระบบที่หุ่นยนต์จะทำงานสัมพันธ์กับอุปกรณ์ประเภทอื่น โดยผ่านการส่งค่าของอุปกรณ์จับสัญญาณและการทำงานตามเงื่อนไข อย่างไรก็ตามตำแหน่งต่างๆ ที่ใช้ในการเคลื่อนที่จะมีอยู่ในไฟล์ระดับพื้นฐานเท่านั้น ดังนั้นจึงต้องเขียนโพสต์โปรเซสเซอร์เพื่อแปลงไฟล์ทั้งคู่เพื่อนำไปใช้งาน ดังมีขั้นตอนต่างๆ ดังต่อไปนี้

4.3.2.1) ขั้นตอนการออกแบบโพสต์โปรเซสเซอร์สำหรับไฟล์ระดับพื้นฐาน

การออกแบบโปรแกรม สำหรับไฟล์เอาต์พุตของการเคลื่อนที่ (.MRL) มีขั้นตอนโดยสรุปดังนี้

1) เปิดแฟ้มข้อมูล TASK FILE (CART)

2) อ่านแฟ้มข้อมูลทีละบรรทัด โดยจะเก็บข้อมูลตัวแรกเข้าไปในตัวแปร เพื่อใช้ในการเปรียบเทียบ

3) ทำการเปรียบเทียบคำสั่งที่ได้ว่าเป็นคำสั่งใด แล้วทำการแปลงเป็นภาษา MRL

4) บันทึกโปรแกรมภาษา MRL ที่ได้ลงในแฟ้มข้อมูลปลายทาง

5) ตรวจสอบการสิ้นสุดของแฟ้มข้อมุลที่เปิดอ่าน ถ้าสิ้นสุดให้ปิดแฟ้มข้อมูลปลายทางแล้วสิ้นสุดการทำงาน ถ้ายังไม่สิ้นสุดให้กลับไปทำตามข้อ 2 ถึง 4

การออกแบบโปรแกรมสำหรับไฟล์เอาต์พุตของตำแหน่ง(.POS) มีขั้นตอนดังนี้

1) เปิดแฟ้มข้อมูล CART และ JOINT

2) ทำการอ่านแฟ้มข้อมูลทั้งสองทีละบรรทัด โดยจะเก็บข้อมูลส่วนแรก เข้าไปในตัวแปร เพื่อใช้ในการเปรียบเทียบเช่นเดียวกับไฟล์ MRL

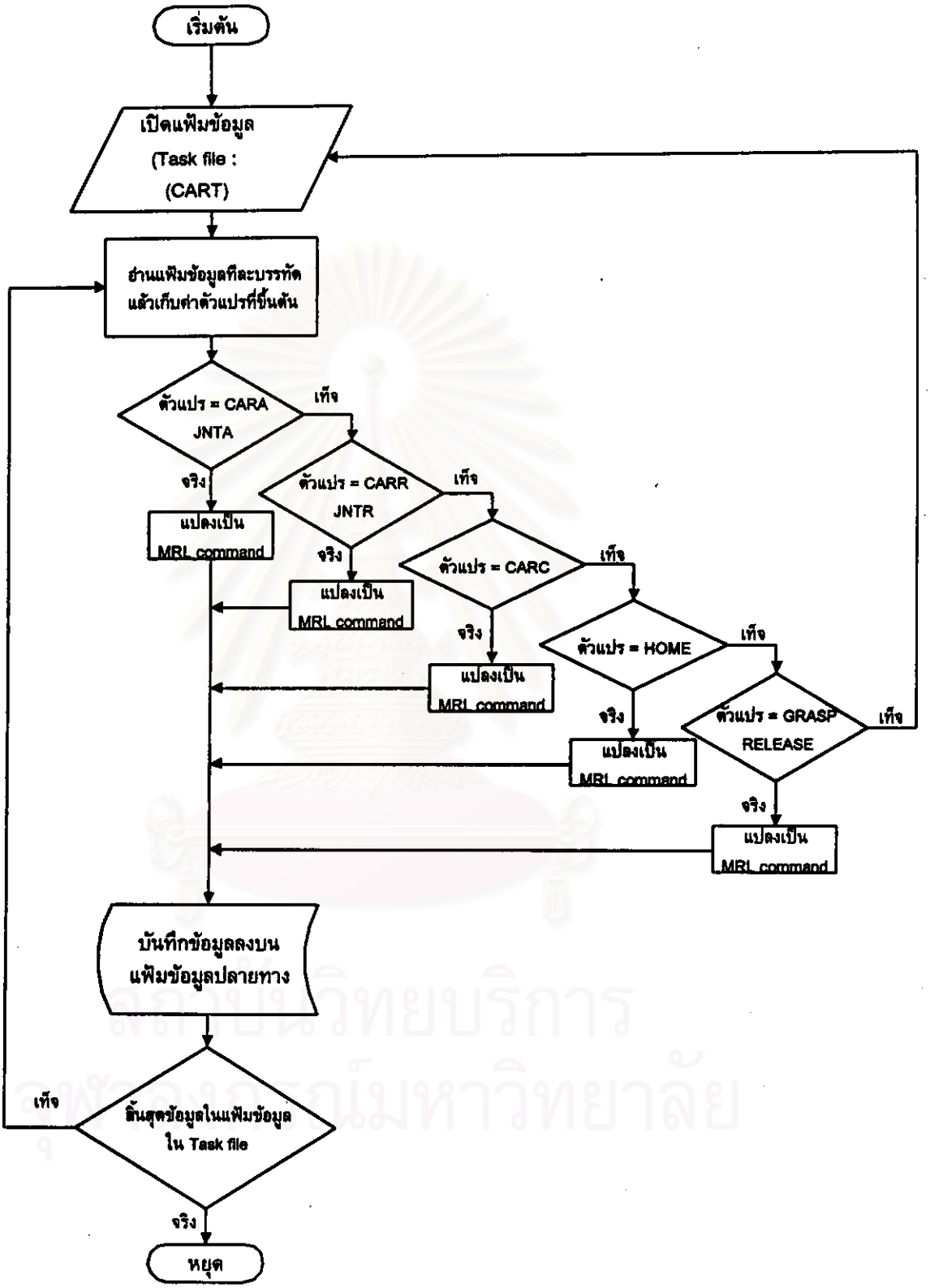
3) ทำการเปรียบเทียบคำสั่ง เพื่อพิจารณาลำดับในการทำงาน

4) จากไฟล์ CART ทำการอ่านข้อมูลของตำแหน่งในส่วนที่สอง คือค่า x , y , z , phi, teta , psi ส่วนไฟล์ joint จะอ่านข้อมูลตำแหน่งของมุม roll angle

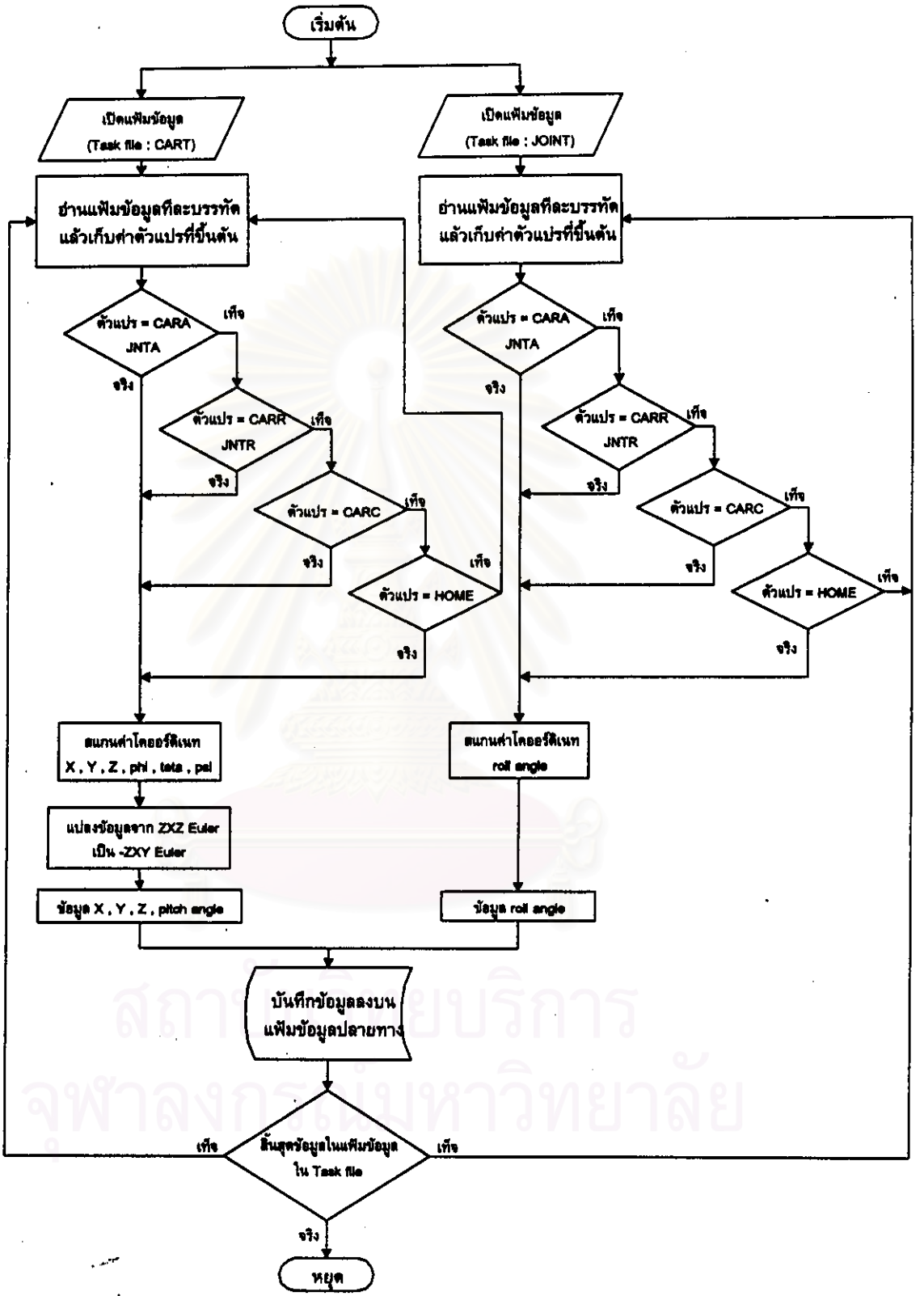
5) แปลงค่าตำแหน่งที่ได้จาก จาก ZXZ Euler angle เป็น -ZXY Euler angle

6) บันทึกตำแหน่งที่ได้จากการแปลงไฟล์ CART และตำแหน่งของมุม roll จากไฟล์ JOINT

7) ตรวจสอบการสิ้นสุดของแฟ้มข้อมุลที่เปิดอ่าน ถ้าสิ้นสุดให้ปิดแฟ้มข้อมูลปลายทางแล้วสิ้นสุดการทำงาน ถ้ายังไม่สิ้นสุดให้กลับไปทำตามข้อ 2 ถึง 4



รูปที่ 4.19 แสดงแผนผังของการเขียนไฟล์เอาต์พุตของการเคลื่อนที่ (.MRL)



รูปที่ 4.20 แสดงแผนผังของการเขียนไฟล์เอาต์พุตของตำแหน่ง (.POS)

4.3.2.2) ขั้นตอนการออกแบบโพสโปรเซสเซอร์สำหรับไฟล์ระดับสูง

การออกแบบโปรแกรมมีขั้นตอนโดยสรุปดังนี้คือ

1) เปิดแฟ้มข้อมูล Advance file

2) อ่านแฟ้มข้อมูลทีละบรรทัด โดยจะเก็บข้อมูลในส่วนแรก เพื่อใช้ในการเปรียบเทียบ

3) ทำการเปรียบเทียบคำสั่งว่าเป็นคำสั่งใด แล้วทำตามขั้นตอนการทำงานคือ

3.1) คำสั่ง IF

3.1.1) เช็คสถานะของเอาต์พุตที่ได้ในขณะทำงาน โดยทำการเปรียบเทียบกับค่าตัวแปรในส่วนที่สอง (L # init และ L # status)

3.1.2) ทำการตรวจสอบเงื่อนไขการทำงาน โดยทำการเปรียบเทียบกับค่าตัวแปรในส่วนที่สอง (L # 3, L # 2 และ N # 1)

3.2) คำสั่ง LAB

เพื่อทำการตรวจสอบลักษณะของการทำงานของหุ่นยนต์ โดยทำการเปรียบเทียบกับค่าตัวแปรในส่วนที่สอง (LAB)

3.3) คำสั่ง WAIT

เพื่อทำการตรวจสอบลักษณะการเคลื่อนที่กลับไปยังตำแหน่งเริ่มต้นของหุ่นยนต์

4) บันทึกโปรแกรมภาษา MRL ที่ได้ลงในแฟ้มข้อมูลปลายทาง

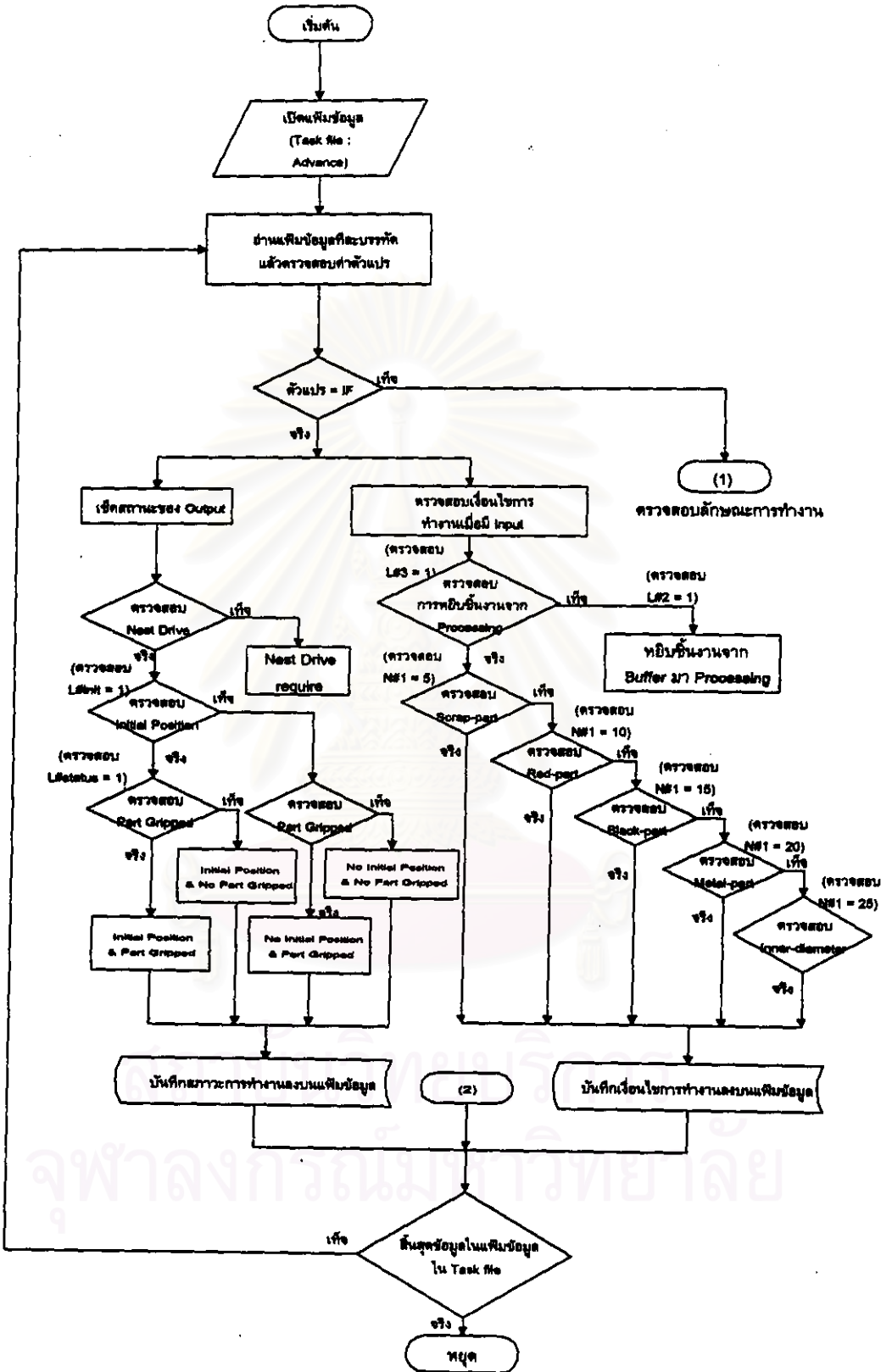
5) ตรวจสอบการสิ้นสุดของข้อมูลที่เปิดอ่าน ถ้าสิ้นสุดให้ทำการปิดแฟ้มข้อมูลปลายทางแล้วสิ้นสุดการทำงานของข้อมูล ถ้ายังไม่สิ้นสุดให้กลับไปทำตามข้อ 2 ถึง 4

4.3.3) ความสัมพันธ์ระหว่างตำแหน่งที่ได้จาก CATIA และตำแหน่งที่ใช้งานสำหรับหุ่นยนต์ MITSUBISHI

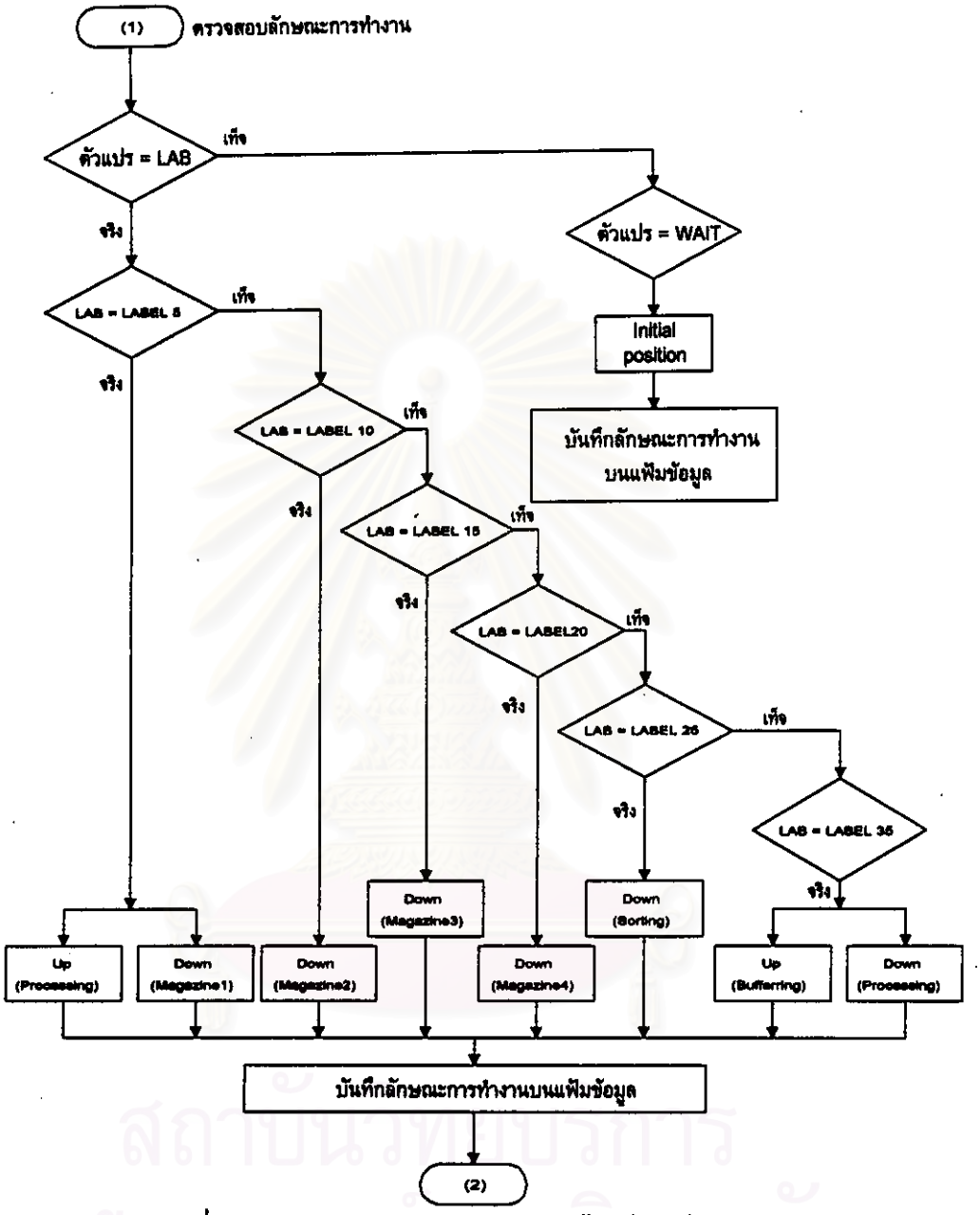
จากลักษณะของเอาต์พุตที่ได้จากไฟล์ประเภทพื้นฐาน ชนิด cartesian จะพบว่าตำแหน่งที่ใช้ในการเคลื่อนที่ได้จะแสดงอยู่ในระบบการวัดมุมแบบ ZXZ Euler angle โดยการวัดเทียบระหว่างแกนที่ปลายแขนของหุ่นยนต์เทียบกับแกนอ้างอิงที่ฐาน ซึ่งค่าตำแหน่งที่ได้จะประกอบไปด้วย

- ค่า X , Y , Z ซึ่งเป็นค่าแสดงโคออร์ดิเนต ของจุดเริ่มต้นของแกนที่ปลาย เทียบกับแกนอ้างอิงที่ฐาน

- PSI (ψ) เป็นค่าของมุมที่หมุนรอบแกน Z



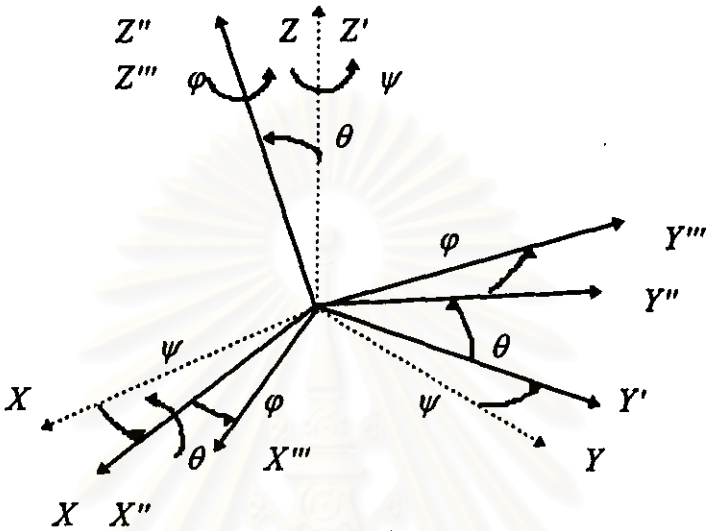
รูปที่ 4.21 แสดงแผนผังของการเขียนไฟล์เอาต์พุตระดับสูง



รูปที่ 4.22 แสดงแผนผังของการเขียนไฟล์เอาต์พุตระดับสูง (ต่อ)

- THETA (θ) เป็นค่าของมุมที่หมุนรอบแกน X′ ซึ่งเป็นแกน X ใหม่ที่ได้จากการหมุนครั้งแรก

- PHI (φ) เป็นค่าของมุมที่หมุนรอบแกน Z″ ซึ่งเป็นแกน Z ใหม่ที่ได้จากการหมุนครั้งที่สอง ดังแสดงในรูปที่



รูปที่ 4.23 แสดง ZXZ Eulerian Angles Convention

ในขณะที่ลักษณะของตำแหน่งต่างๆ ที่ใช้สำหรับตัวควบคุมของหุ่นยนต์ MITSUBISHI RV-M1 นั้นจะแสดงอยู่ในรูปของค่า x , y , z , roll , pitch โดยค่า x , y , z เป็นค่าที่แสดงโดออร์ดิเนตเช่นเดียวกัน แต่การแสดง Orientation จะแสดงค่าของมุม roll ซึ่งเป็นมุมของแกนที่ปลายวัดเทียบกับระนาบ xy และมุม pitch ซึ่งเป็นมุมที่ใช้หมุนข้อต่อสุดท้าย (ข้อต่อที่ 5) ดังนั้นสำหรับโพสต์โปรเซสเซอร์ที่เขียนขึ้น จึงต้องทำการแปลงลักษณะของตำแหน่งจากระบบ ZXZ Euler angle เป็นระบบตำแหน่งสำหรับหุ่นยนต์ MITSUBISHI โดยมีวิธีการดังต่อไปนี้

4.3.3.1) ZXZ Euler angle

จากรูปที่ 4.23 จะได้ Transformation Matrix ของแกนที่ปลายแขนของหุ่นยนต์เทียบกับแกนอ้างอิงที่ฐาน คือ

$$
{}^{A}_{B}R_{ZXZ} = R_z(\alpha).R_x(\beta).R_z(\gamma)
$$

$$
= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$= \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

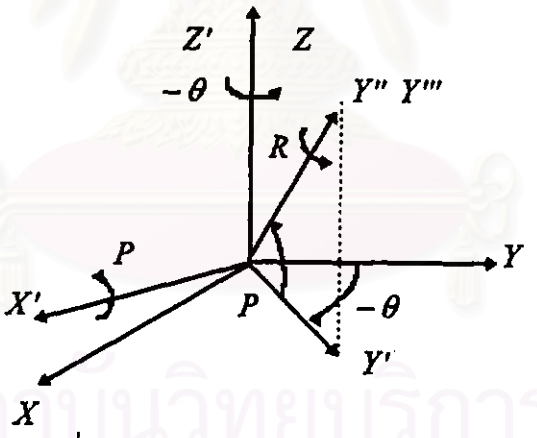
โดยที่

$$A_{11} = \cos(\alpha).\cos(\gamma) - \sin(\alpha).\cos(\beta).\sin(\gamma)$$
$$A_{12} = -\cos(\alpha).\sin(\gamma) - \sin(\alpha).\cos(\beta).\cos(\gamma)$$
$$A_{13} = \sin(\alpha).\sin(\beta)$$

$$A_{21} = \sin(\alpha).\cos(\gamma) + \cos(\alpha).\cos(\beta).\sin(\gamma)$$
$$A_{22} = -\sin(\alpha).\sin(\gamma) + \cos(\alpha).\cos(\beta).\cos(\gamma)$$
$$A_{23} = -\cos(\alpha).\sin(\beta)$$

$$A_{31} = \sin(\beta).\sin(\gamma)$$
$$A_{32} = \sin(\beta).\cos(\gamma)$$
$$A_{33} = \cos(\beta)$$

4.3.3.2) -ZXY Euler angle



รูปที่ 4.24 แสดง -ZXY Eulerian Angles

จากรูปที่ 4.24 จะได้ Transformation Matrix ของแกนที่ปลายแขนของหุ่นยนต์เทียบกับแกนอ้างอิงที่ฐาน คือ

$$\overset{A}{B}R_{-ZXY} = R_z(-\theta).R_X(P).R_y(R)$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(P) & -\sin(P) \\ 0 & \sin(P) & \cos(P) \end{bmatrix} . \begin{bmatrix} \cos(R) & 0 & \sin(R) \\ 0 & 1 & 0 \\ -\sin(R) & 0 & \cos(R) \end{bmatrix}$$

$$= \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ A_{31} & B_{32} & B_{33} \end{bmatrix}$$

โดยที่

$$B_{11} = \cos(\theta).\cos(R) + \sin(\theta).\sin(P).\sin(R)$$
$$B_{12} = \sin(\theta).\cos(P)$$
$$B_{13} = \cos(\theta).\sin(R) - \sin(\theta).\sin(P).\cos(R)$$

$$B_{21} = -\sin(\theta).\cos(R) + \cos(\theta).\sin(P).\sin(R)$$
$$B_{22} = \cos(\theta).\cos(P)$$
$$B_{23} = -\sin(\theta).\sin(R) - \cos(\theta).\sin(P).\cos(R)$$

$$B_{31} = -\cos(P).\sin(R)$$
$$B_{32} = \sin(P)$$
$$B_{33} = \cos(P).\cos(R)$$

จากสมการของ Transformation Matrix ทั้งสอง เมื่อเราทราบค่าของ Transformation Matrix ของ ZXZ Euler angle จาก CATIA ก็จะสามารถหาค่าของตัวแปร (parameter) ของมุม Q , P และ R ของระบบ -ZXY Euler angle ได้โดยใช้สมการต่อไปนี้

$$P(pitch) = a\tan 2(A_{32}, \sqrt{(A_{12})^2 + (A_{22})^2}) \quad \text{.....(3)}$$

$$\theta(theta) = a\tan 2(A_{12}, A_{22}) \quad \text{.....(4)}$$

$$R(roll) = a\tan 2(-A_{31}, A_{33}) \quad \text{.....(5)}$$

โดย atan 2 (y,x) มีค่าเท่ากับ tan (y/x) โดยมีค่าของมุมใน quardant ตามเครื่องหมายของ x และ y ดังนั้นเมื่อพิจารณาตามเครื่องหมายแล้ว เราจะสามารถหาค่าของมุม P ได้ดังเงื่อนไขต่อไปนี้

ถ้า $A_{32} \geq 0$ และ $A_{22} \geq 0$ จะได้ค่าของมุม P อยู่ใน quardant ที่ 1 (P ที่ได้ = P ที่คำนวณ)

ถ้า $A_{32} \geq 0$ และ $A_{22} < 0$ จะได้ค่าของมุม P อยู่ใน quardant ที่ 2 (P ที่ได้ = 180 - P ที่คำนวณ)

ถ้า $A_{32} < 0$ และ $A_{22} < 0$ จะได้ค่าของมุม P อยู่ใน quardant ที่ 3 (P ที่ได้ = -180 - P ที่คำนวณ)

ถ้า $A_{32} < 0$ และ $A_{22} \geq 0$ จะได้ค่าของมุม P อยู่ใน quardant ที่ 4 (P ที่ได้ = P ที่คำนวณ)

ซึ่งค่าของมุมที่ได้จะนำไปใช้ สำหรับบอกตำแหน่งที่ตัวควบคุมหุ่นยนต์เข้าใจต่อไป

4.4) ลักษณะการใช้งานโปรแกรมโพสต์โปรเซสเซอร์

โปรแกรมโพสต์โปรเซสเซอร์ที่เขียนขึ้นในงานวิจัยนี้ มีชื่อว่า POST.C และเมื่อทำการคอมไพล์แล้ว จะได้โปรแกรม POST.exe ก่อนที่จะทำการเรียกใช้ ผู้ใช้จะต้องมีไฟล์การเคลื่อนที่ (ไฟล์ CART, JOINT หรือ ADVANCE) ซึ่งเป็นไฟล์เอาต์พุตที่ได้จากการจำลองการทำงานบนโปรแกรม CATIA เก็บไว้ในที่อยู่ (directory) เดียวกับโปรแกรม เมื่อทำการเรียกใช้ ผู้ใช้จะต้องเลือกประเภทของไฟล์เอาต์พุต (basic หรือ advance) ก่อนที่จะใส่ชื่อของไฟล์อินพุตและชื่อของไฟล์เอาต์พุตที่ต้องการ ดังแสดงในรูปที่ 4.25 และ 4.26

```
==========================================
The output file from CATIA has 2 types
==========================================
     1) Basic level file : <<no logical statement>>
     2) Advance level file : <<logical statement>>
What type of file which you select ==>1,2:  1
 Enter name of input file
   (Cartesian Position) : cart6
 Enter name of input file
     (Joint Position)   : joint6
 ============================================================
 PROGRAM ASSIGNMENT FOR MITSUBISHI Model RV-M1 ROBOT
 HAS 2 PARTS: 1) Position lists (file with extension:.pos)
          2) Program list (file with extension:.mrl)
 ============================================================
 ----------------------------
 Enter name of output file(Position lists<*.POS>): pong6
out1------>pong6.pos
 ----------------------------
```

```
 Enter name of output file(Program lists)<*.MRL>: pong6
out2------>pong6.mrl
 Please open file to 'CHECK'==><<pong6.pos>> and <<pong6.mrl>>
```

รูปที่ 4.25 แสดงการใส่ค่าเมื่อไฟล์เอาต์พุตเป็นประเภท basic

```
==============================================
 POST-PROCESSOR FOR MITSUBISHI model RV-M1 ROBOT
 ==============================================
==========================================
The output file from CATIA has 2 types
==========================================
     1) Basic level file : <<no logical statement>>
     2) Advance level file : <<logical statement>>
What type of file which you select ==>1,2: 2
 Enter name of input file
     (Advance Position)  : advance4
 Enter name of output file(Program lists<*.MRL>): adv4
out3------>adv4.mrl
```

รูปที่ 4.26 แสดงการใส่ค่าเมื่อไฟล์เอาต์พุตเป็นประเภท advance

ต่อไปนี้จะเป็นการแสดงตัวอย่างของไฟล์ที่ได้เมื่อผ่านโพสต์โปรเซสเซอร์

4.4.1) เมื่อประเภทไฟล์เอาต์พุตเป็นประเภท basic

```
DATE : THURSDAY APRIL 24, 1997     22.53.42
 FILE : /home/catadm/db
 MODEL : MY THESIS : MITSUBISHI RV-M1: CONFIG
 TASK : TASK COMMAND
 CONTROLLER : *CTL1
 TYPE : BASIC CARTESIAN POSITIONS
CMT
VERS   V4R1.6
CMT
CMT
ACEL   *AXS1
ROB    GUNDAM
CARC   *AXS2       .000    .000    .000    .000    .000    .000
      EN      264.097  215.885  100.000  129.264   94.295 -180.000
              235.185  253.586  100.000  137.156   93.004 -180.000
              202.215  287.311  100.000  144.861   91.515 -180.000
```

```
                166.307  315.830  100.000  152.230   89.993 -180.000
                124.481  340.226  100.000  159.904   88.539 -180.000
                 81.368  356.619  100.000  167.147   87.585 -180.000
                 40.263  364.557  100.000  173.698   87.315 -180.000
                 -2.002  365.161  100.000 -179.686   87.753 -180.000
                -44.538  358.108  100.000 -172.911   88.924 -180.000
                -87.597  343.069  100.000 -165.677   90.771 -180.000
               -128.488  321.238  100.000 -158.200   92.970 -180.000
               -166.830  293.784  100.000 -150.409   95.180 -180.000
               -202.320  261.926  100.000 -142.316   97.051 -180.000
               -234.705  226.836  100.000 -134.023   98.293 -180.000
               -263.751  189.552  100.000 -125.704   98.731 -180.000
               -289.221  150.889  100.000 -117.551   98.345 -180.000
               -310.845  111.348  100.000 -109.708   97.264 -180.000
               -329.565   67.582  100.000 -101.589   95.567 -180.000
               -342.840   22.357  100.000  -93.731   93.625 -180.000
               -349.975  -25.674  100.000  -85.804   91.629 -180.000
               -350.551  -70.367  100.000  -78.650   89.827 -180.000
   CCON *AXS2     ARM   RIGHT  ELBOW  BELOW
   JCON         1  2  3  4  5
                0  0  0  0  0

CARA JNTI EN        480.000    .000  300.000  -90.000    .000    .000
   CCON *AXS2     ARM   RIGHT  ELBOW  BELOW  WRIST  EITHER
   JCON         1  2  3  4  5
                0  0  0  0  0
GRA     GRIP    SPHERE
HOME JNTI           -.002  589.000  300.000    .000    .000    .000
   CCON *AXS2     ARM   RIGHT  ELBOW  EITHER
REL     MDLO1_MY
JNTA JNTI         -589.000   -.002  300.000   90.000    .000    .000
HOME JNTI           -.002  589.000  300.000    .000    .000    .000
GRA     GRIP    SPHERE
CARA JNTI EN        480.000    .000  300.000  -90.000    .000    .000
   CCON *AXS2     ARM   RIGHT  ELBOW  BELOW  WRIST  EITHER
REL     MDLO1_MY
HOME JNTI           -.002  589.000  300.000    .000    .000    .000
   CCON *AXS2     ARM   RIGHT  ELBOW  EITHER
       END
```

รูปที่ 4.27 แสดงตัวอย่างไฟล์อินพุตที่ได้จากการจำลองการทำงานของแขนกลโดยเป็นไฟล์ที่แสดง CARTESIAN POSITION

```
DATE : THURSDAY APRIL 24, 1997      22.54.24
FILE : /home/catadm/db
```

```
MODEL : MY THESIS : MITSUBISHI RV-M1: CONFIG
TASK  : TASK COMMAND
CONTROLLER : *CTL1
TYPE  : BASIC JOINT POSITIONS
CMT
VERS     V4R1.6
CMT
CMT
ACEL     *AXS1
ROB      GUNDAM
CARC      50.736   19.876  -61.607  -52.565    .000
          42.844   19.779  -61.195  -51.588    .000
          35.139   19.646  -60.716  -50.445    .000
          27.770   19.489  -60.223  -49.259    .000
          20.097   19.318  -59.747  -48.110    .000
          12.853   19.195  -59.432  -47.347    .000
           6.303   19.158  -59.343  -47.131    .000
            -.314  19.217  -59.488  -47.483    .000
          -7.089   19.365  -59.874  -48.416    .000
         -14.323   19.573  -60.476  -49.867    .000
         -21.800   19.777  -61.185  -51.561    .000
         -29.591   19.934  -61.889  -53.225    .000
         -37.683   20.031  -62.478  -54.604    .000
         -45.977   20.076  -62.866  -55.503    .000
         -54.296   20.089  -63.002  -55.818    .000
         -62.448   20.078  -62.882  -55.541    .000
         -70.292   20.040  -62.545  -54.759    .000
         -78.411   19.957  -62.012  -53.513    .000
         -86.269   19.828  -61.396  -52.058    .000
         -94.196   19.658  -60.755  -50.532    .000
        -101.350   19.471  -60.171  -49.128    .000
CARA JNTI    90.000   32.093  -88.208   56.115    .000
GRA      GRIP    SPHERE
HOME JNTI     .000     .000     .000     .000     .000
REL      MDLO1_MY
JNTA JNTI   -90.000     .000     .000     .000     .000
HOME JNTI     .000     .000     .000     .000     .000
GRA      GRIP    SPHERE
CARA JNTI    90.000   32.093  -88.208   56.115    .000
REL      MDLO1_MY
HOME JNTI     .000     .000     .000     .000     .000
END
```

รูปที่ 4.28 แสดงตัวอย่างไฟล์อินพุตที่ได้จากการจำลองการทำงานของแขนกลโดยเป็นไฟล์ที่แสดง JOINT POSITION

```
PD  1,  264.0, 215.8, 100.0, -94.3,   0.0
PD  2,  235.1, 253.5, 100.0, -93.1,   0.0
PD  3,  202.2, 287.3, 100.0, -91.6,   0.0
PD  4,  166.3, 315.8, 100.0, -89.9,   0.0
PD  5,  124.4, 340.2, 100.0, -88.5,   0.0
PD  6,   81.3, 356.6, 100.0, -87.5,   0.0
PD  7,   40.2, 364.5, 100.0, -87.3,   0.0
PD  8,   -2.0, 365.1, 100.0, -87.7,   0.0
PD  9,  -44.5, 358.1, 100.0, -88.9,   0.0
PD 10,  -87.5, 343.0, 100.0, -90.8,   0.0
PD 11,-128.4, 321.2, 100.0, -93.0,   0.0
PD 12,-166.8, 293.7, 100.0, -95.2,   0.0
PD 13,-202.3, 261.9, 100.0, -97.1,   0.0
PD 14,-234.7, 226.8, 100.0, -98.3,   0.0
PD 15,-263.7, 189.5, 100.0, -98.8,   0.0
PD 16,-289.2, 150.8, 100.0, -98.4,   0.0
PD 17,-310.8, 111.3, 100.0, -97.3,   0.0
PD 18, -329.5,  67.5, 100.0, -95.6,   0.0
PD 19, -342.8,  22.3, 100.0, -93.7,   0.0
PD 20, -349.9, -25.6, 100.0, -88.3,   0.0
PD 21, -350.5, -70.3, 100.0, -89.8,   0.0
PD 22,  480.0,    0.0, 300.0,   0.0,   0.0
PD 23,      0.0, 589.0, 300.0,   0.0,   0.0
PD 24,- 589.0,    0.0, 300.0,   0.0 ,  0.0
PD 25,   0.0,   589.0, 300.0,   0.0,   0.0
PD 26, 480.0,     0.0, 300.0,   0.0,   0.0
```

รูปที่ 4.29 แสดงตัวอย่างไฟล์เอาต์พุตของตำแหน่ง (*.POS) ที่ได้เมื่อแปลงไฟล์ในรูปที่ 4.27 และ 4.28 ด้วยโพสต์โปรเซสเซอร์ที่เขียนขึ้น

```
 10 MC   1,21  * CARTESIAN CURVE MOTION *
 20 MO  22, O  * ABSOLUTE CATESIAN MOTION *
 30 GC         * GRIP CLOSED *
 40 MO 23, C
 50 GO         * GRIP OPEN *
 60 MO  24, O
 70 MO  25, O
 80 GC
 90 MO 26, C
100 GO
110 ED
RN
```

รูปที่ 4.30 แสดงตัวอย่างไฟล์เอาต์พุตของคำสั่งการเคลื่อนที่ (*.MRL) ที่ได้เมื่อแปลงไฟล์ในรูป 4.27 และ 4.28 ด้วยโพสต์โปรเซสเซอร์ที่เขียนขึ้น

4.4.2) เมื่อประเภทไฟล์เอาต์พุตเป็นประเภท advance

```
290   IN
300   EQ 0, 1050 * No job
310   EQ 1, 450   * Do Nest drive
320   EQ 2, 480   * Go to initial position
330   EQ 3, 550   * Pick up from buffer station
340   EQ 4, 620   * Pick up from processing station
350   EQ 5, 720   * Put down at processing station
360   EQ 6, 770   * Put down at sorting station
370   EQ 7, 820   * Put down at magazine 1
380   EQ 8, 870   * Put down at magazine 2
390   EQ 9, 920   * Put down at magazine 3
400   EQ 10, 970  * Put down at magazine 4
410   EQ 11, 1050 * Signal status
420   NE 16, 440
430   SC 4,0      * finish program
440   GT 1050
450   NT          * Do Nest drive
460   SC 1,0
470   GT 480      * Go to initial position
480   CP 3        * Part gripped ?
490   EQ 0, 520
500   MO 4,C      * Go to initial position
510   GT 530
520   MO 4,O      * Go to initial position
530   SC 2,1
540   GT 1050
550   MO 15,O     * Pick up from buffer station
570   MO 14,O
580   GC
590   MO 15,C
600   MO 16,C
610   GT 690
620   MO 3,O      * Pick up from processing station
630   MO 2,O
640   MO 1,O
650   GC
660   MO 2,C
670   MO 3,C
680   GT 690
```

```
690   SC 2,0     * Not in initial position
700   SC 3,1     * Part gripped
710   GT 1050
720   MO 17,C    * Put down at processing station
730   MO 18,C
740   GO
750   MO 17,O
760   GT 1020
770   MO 19,C    * Put down at sorting station
780   MO 20,C
790   GO
800   MO 19,O
810   GT 1020
820   MO 10,C    * Put down at magazine 1
830   MO 5,C
840   GO
850   MO 10,O
860   GT 1020
870   MO 11,C    * Put down at magazine 2
880   MO 6,C
890   GO
900   MO 11,O
910   GT 1020
920   MO 12,C    * Put down at magazine 3
930   MO 7,C
940   GO
950   MO 12,O
960   GT 1020
970   MO 13,C    * Put down at magazine 4
980   MO 8,C
990   GO
1000  MO 13,O
1010  GT 1020
1020  SC 2,0     * Not in initial position
1030  SC 3,0     * No part gripped
1040  GT 1050
1050  RT
```

รูปที่ 4.31 แสดงตัวอย่างไฟล์เอาต์พุตของคำสั่งการเคลื่อนที่ (*.MRL) ที่ได้จากการแปลงโดยโพสต์โปรเซสเซอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย