สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบและพัฒนาเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์

Design and Development of Computerized Numerical Controller

หนังสืออ้างอิง
ห้ามนำออกนอกห้องสมุด

นายประภาษ อุคคกิมาพันธุ์
Mr. PRAPART UKAKIMAPARN

T021202

อาจารย์ที่ปรึกษา
รองศาสตราจารย์ กิตติ ตีรเศรษฐ
ADVISOR
Assc. Prof. KITTI TIRASESTH M.Eng

วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิศวกรรมไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

ISBN 974-621-140-4

เลขหมู่
เลขทะเบียน 21202
วัน, เดือน, ปี - 3 ส.ค. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ การออกแบบและพัฒนาเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์
ผู้วิจัย นายประภาษ อุดคกิมาพันธุ์
อาจารย์ผู้ควบคุมวิทยานิพนธ์ รองศาสตราจารย์ กิตติ ตีรเศรษฐ
ระดับการศึกษา วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
ปีการศึกษา 2536

บทคัดย่อ

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ เข้ามามีบทบาทอย่างมากในการควบคุมการทำงานของเครื่องจักรอุตสาหกรรม ข้อดีของเครื่องควบคุมแบบนี้คือ สามารถทำงานได้รวดเร็ว มีความเที่ยงตรง และประสิทธิภาพการทำงานสูง เพราะมีความยืดหยุ่นในการเปลี่ยนแปลงแก้ไขการทำงานของเครื่องจักรด้วยโปรแกรม แต่ในปัจจุบันเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ยังมีราคาสูงเพราะต้องนำเข้าจากต่างประเทศ

วิทยานิพนธ์ฉบับนี้ เสนอการออกแบบ และพัฒนาสร้างเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่สามารถจัดหาอุปกรณ์ได้ในประเทศ และใช้เครื่องกลึงเป็นกรณีศึกษา ผลการศึกษาวิจัยพบว่าเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ออกแบบ และสร้างขึ้น ความสามารถในการควบคุมได้ในระดับความละเอียดดีกว่า 100 ไมโครเมตร ซึ่งเพียงพอสำหรับงานที่มีความละเอียดปานกลาง สามารถเปลี่ยนแปลงการทำงานของเครื่องจักรด้วยโปรแกรม โดยมีขีดความสามารถในการปฏิบัติตามคำสั่งพื้นฐาน และยังสามารถเชื่อมต่อกับอุปกรณ์ภายนอกได้ทัดเทียมกับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ใช้ในอุตสาหกรรม จากการที่สามารถพัฒนาซอฟท์แวร์ได้เอง ทำให้อุปกรณ์ที่สร้างขึ้นสามารถบรรลุวัตถุประสงค์ของผู้ใช้ได้ทุกประการ ในขณะที่เครื่องนำเข้าไม่สามารถดัดแปลงซอฟท์แวร์ใดๆ ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Design and Development of Computerized Numerical Controller
Author	Mr. Prapart Ukakimaparn
Thesis Advisor	Assc. Prof. Kitti Tirasesth
Graduate Level	Master of Engineering in Electrical Engineering
Academic Year	1993

Abstract

Computerized Numerical Controller (CNC) is now widely use and play important roles in industrial machine control system. The high speed operation, high accuracy and high performance are some of the advantages of the CNC. Since the execution of CNC is done through the instruction program which easily changed by the user, then the operation of the controlled machine can be simple modified. However, most of the CNC available is still expensive because it must be imported from abroad.

This thesis presents the design and development of CNC machine using components available in Thailand. The application to the lathe machine is introduce as a case study. The accuracy of the designed and developed CNC is better than 100 micron, which is suitable for moderate precision of the work that not requires such high accuracy. In addition, the execution of CNC under the basic instruction program is achieved. The operation of the lathe machine can be changed in accordance with the change of control program of CNC. And as it should be, the designed and developed CNC can be interfaced with the another peripheral equipment as well as the industrial type. Since, the software can be modified by the user easily, therefore the developed CNC can be satisfied by the user, while the imported CNC have a limitation on software development.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
สารบัญ	ค
บทที่ 1 ประวัติความเป็นมาของเครื่องควบคุมเชิงเลข	1
บทที่ 2 คณิตศาสตร์สำหรับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์	3
2.1 ตรีโกณมิติฟังก์ชัน	3
2.2 สามเหลี่ยมออฟบิท	5
2.3 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์	6
2.4 ยูนิทเวคเตอร์และทิศทางโคซายน์	11
บทที่ 3 เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์	19
3.1 หน่วยคอมพิวเตอร์กลาง	20
3.2 หน่วยอินพุทเอาท์พุทพื้นฐาน	53
3.3 หน่วยคอนโซล	58
3.4 หน่วยควบคุมตำแหน่งและซิงโคร	58
3.5 หน่วยขับกำลัง	63
บทที่ 4 การโปรแกรมทางเดินของคมตัด	70
4.1 การโปรแกรมแบบค่าสัมบูรณ์	70
4.2 การโปรแกรมแบบค่าเพิ่ม	70
4.3 การโปรแกรมแบบจุดกำเนิด	70
4.4 การเคลื่อนที่แบบต่างๆ ของ CNC	71
4.5 การคำนวณระยะของเส้นกึ่งกลางของแนวคมตัด	71
บทที่ 5 แระกระทำที่เกิดขึ้นจากคมตัดขณะเครื่องจักรทำงาน	77
5.1 การเจาะ	77
5.2 งานที่เกี่ยวกับการกลึง	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
5.3 การตัดชิ้นงานด้วยคมตัดโดยวิธีการกัด	83
บทที่ 6 การออกแบบส่วนประกอบของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์	87
6.1 หน่วยคอมพิวเตอร์กลาง	87
6.2 หน่วยควบคุมตำแหน่ง	104
6.3 หน่วยแสดงผล	113
บทที่ 7 การทดสอบการทำงานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์	119
7.1 ขั้นตอนเบื้องต้นก่อนทดสอบการกลึง	119
7.2 การทดสอบการกลึง	121
สรุปผลและวิจารณ์	125
ข้อเสนอแนะ	126
กิตติกรรมประกาศ	127
เอกสารอ้างอิง	128
ภาคผนวก ก	
ภาคผนวก ข	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ประวัติความเป็นมาของเครื่องควบคุมเชิงเลข

ในศตวรรษที่ 18 John Wilkinson ได้สร้างเครื่องตัดและคว้านโลหะ ซึ่งเป็นจุดเริ่มต้นของการนำเครื่องจักรอัตโนมัติมาใช้ในงานอุตสาหกรรมผลิตชิ้นส่วนเครื่องจักรกล ต่อมาในปลายศตวรรษที่ 19 ความต้องการที่จะนำเครื่องจักรมาใช้ในอุตสาหกรรมเครื่องจักรกลได้ขยายตัวมากขึ้น ความซับซ้อนของการแมชชีน (Machining) มีมากขึ้นตามลำดับ โดยที่การแมชชีนหมายถึงการสร้างชิ้นงาน เช่น การกัดผิวงาน การปอกผิวด้วยวิธีการกลึง ความเร็วในการทำงานตลอดจนความถูกต้องเที่ยงตรงเริ่มถูกนำมาพิจารณาด้วย ขณะนั้นอุปกรณ์ไฮดรอลิค (Hydraulic) แทรกเซอร์คอนโทรล (TracerControl) ได้ถูกพัฒนาเพื่อใช้กับเครื่องจักรประเภทเครื่องกลึง และเครื่องกัด ซึ่งอาจกล่าวได้ว่าเป็นจุดเริ่มต้นของการพัฒนาออกแบบระบบควบคุมเพื่อนำมาใช้กับเครื่องจักรกลอย่างจริงจัง ในเวลาต่อมาเครื่องจักรอัตโนมัติได้ถูกพัฒนาเพื่อใช้ในอุตสาหกรรมเคมี อุตสาหกรรมผลิตอุปกรณ์ไฟฟ้าภายในบ้านในศตวรรษที่20 ได้เกิดเหตุการณ์ที่เป็นปัจจัยของการพัฒนาระบบควบคุมสำหรับเครื่องจักรกลคือ

- อุตสาหกรรมการผลิตรถยนต์ ได้เริ่มมีการพัฒนาเปลี่ยนแปลงเป็นสายการผลิต
- กระบวนการผลิตทางเคมีและผลิตอาหาร ได้พัฒนาเข้าสู่อุตสาหกรรม
- เริ่มมีการประมวลผลข้อมูลเกี่ยวกับเงินเดือน และการแก้ไขวิเคราะห์ข้อมูล ช่วงสงครามโลกครั้งที่2 กองทัพอากาศของประเทศสหรัฐอเมริกามีโครงการสร้างเครื่องตัดโลหะชนิดลอกเลียนแบบ (Turning Pointer) เพื่อจุดประสงค์ในการผลิตอาวุธ และชิ้นส่วนจักรกลสำหรับอากาศยาน ในปี ค.ศ.1947 กองทัพอากาศของประเทศสหรัฐอเมริกาไม่สามารถผลิตชิ้นส่วนจักรกลดังกล่าวได้โดยลำพัง จึงได้ร่วมมือกับบริษัทพาสันคอร์เปอเรชัน พัฒนาอุปกรณ์เฉพาะขึ้นใช้ในการผลิตชิ้นส่วนจักรกล คืออุปกรณ์การตัดเหล็กแบบกึ่งอัตโนมัติ โดยมีจุดประสงค์เพื่อให้ได้ประสิทธิภาพในการผลิตสูงสุด การออกแบบเน้นความถูกต้องตามรายละเอียดของชิ้นงาน มีความแม่นยำ และมีความยืดหยุ่นในการทำงานสูง โดยไม่ต้องมีการดัดแปลงเครื่องมือกล (Tool) ตลอดจนอุปกรณ์บริด (Bridge) ใด ๆ ต่อมาบริษัทพาสันคอร์เปอเรชัน และสถาบันเทคโนโลยี แห่ง แมสซาจูเสท (MIT) ได้ร่วมมือกันพัฒนาต้นแบบของเครื่องควบคุมขึ้นในปี ค.ศ.1951 ระบบควบคุมที่สร้างขึ้นสามารถประยุกต์ใช้กับเครื่องจักรได้อย่างกว้างขวาง โดยวิธีการขับเคลื่อนเกลียวนำ (Lead Screw) ที่ติดตั้งอยู่กับโต๊ะพาชิ้นงานให้เลื่อนไปมาบนรางเลื่อน (Slide way) โดยการส่งงานผ่านระบบเชื่อมต่อเอาท์พุท และอินพุทของคอมพิวเตอร์ และในปี ค.ศ.1952 บริษัทซินซินเนติ ไฮโดรเทล มิลลิงแมชชีนอิควิปเมนท์ ได้พัฒนาเครื่องควบคุมขึ้นอีกโดยใช้เทคโนโลยีที่ใหม่กว่า เรียกว่าเครื่องควบคุมเชิงเลข (Numerical Control : NC) โดยรับสัญญาณอินพุทผ่านแถบเจาะรู ซึ่งถือได้ว่าเป็นเครื่องควบคุมเชิงเลขเครื่องแรกของโลก เครื่องควบคุมเชิงเลขเครื่องแรกนี้ประกอบขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบขึ้นด้วยหลอดสูญญากาศ อุปกรณ์รีเลย์ และอุปกรณ์เชื่อมต่อ (Interface) ที่ยุ่งยาก ต่อมาเครื่องควบคุมเชิงเลขได้ถูกพัฒนาปรับปรุงให้ใช้หลอดสูญญากาศที่มีขนาดเล็กลง ตลอดจนการนำอุปกรณ์โซลิดสเตท (Solid State) มาใช้ การเปลี่ยนแปลงในระยะต่อมามีการพัฒนาที่รวดเร็วมาก ปัจจุบันเครื่องควบคุมเชิงเลขได้ถูกนำไปประยุกต์ใช้กับควบคุมเครื่องจักรประเภทต่าง ๆมากมาย เช่น เครื่องกลึง เครื่องเจียรนัย เครื่องกัด ตลอดจนเครื่อง EDM (Electrical Discharge Machine) และได้ถูกพัฒนาให้มีความคล่องตัวในการทำงานดียิ่งขึ้น เช่นสามารถจำลองการทำงานของโปรแกรมเพื่อให้ผู้ปฏิบัติงานทราบ การคำนวณหาค่าเบื้องต้นต่าง ๆสำหรับโปรแกรม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

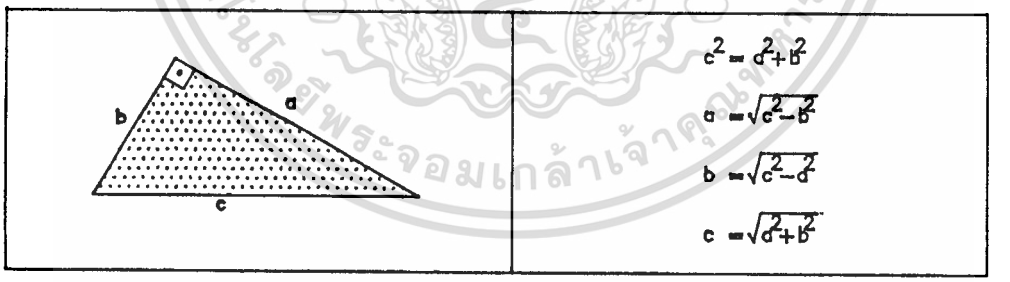
บทที่ 2
คณิตศาสตร์ที่ใช้สำหรับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ (Computerized Numerical Controller : CNC) มีลักษณะการทำงานในรูปการควบคุมการเคลื่อนที่ของแกนเครื่องจักร ซึ่งติดตั้งคมตัดสำหรับตัดชิ้นงานที่อาจอยู่ในรูปของการกลึงหรือการกัด หรือเรียกว่าการ"แมชชีน" โดยรูปทรงการแมชชีนดังกล่าวเป็นรูปทรงทางเรขาคณิตที่ประกอบขึ้นด้วยเส้นสองประเภทคือ เส้นโค้งและเส้นตรง ดังนั้นคณิตศาสตร์ที่ใช้ในการหาค่าเกี่ยวกับระยะพิกัดบนรูปทรงเรขาคณิตจึงอยู่บนพื้นฐานของตรีโกณมิติฟังก์ชัน ซึ่งเป็นเรื่องของความสัมพันธ์ระหว่างด้าน 3 ด้าน และมุม 3 มุม

2.1 ตรีโกณมิติฟังก์ชัน

2.1.1 Pythagorian Theorem

เป็นทฤษฎีว่าด้วยความสัมพันธ์ของผลรวมพื้นที่ที่เกิดขึ้นของสามเหลี่ยมมุมฉาก และแสดงดังรูปที่ 2.1

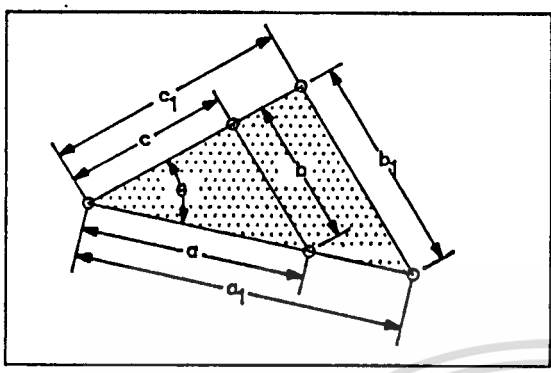


รูปที่ 2.1 Pythagorian Theorem

2.1.2 ความคล้ายคลึงของสามเหลี่ยม(Similar Triangles)

ถ้าส่วนของสามเหลี่ยมถูกตัดแบ่งออกด้วยเส้นตรงที่ขนานกับด้านใดด้านหนึ่งแล้ว สามเหลี่ยมที่เกิดขึ้นใหม่จะมีความคล้ายคลึงกับสามเหลี่ยมเดิมนั้น และมีอัตราส่วนของด้านของสามเหลี่ยมที่เกิดขึ้นใหม่เป็นอัตราส่วนเดียวกันทุกด้าน ดังแสดงในรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



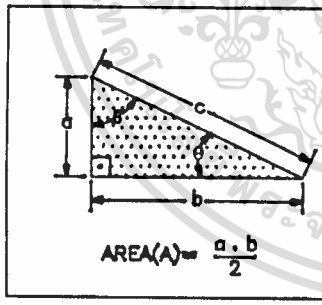
$$\frac{c'}{c} = \frac{a'}{a} \; ; \; \frac{a'}{a} = \frac{b'}{b} \; ; \; \frac{b'}{b} = \frac{c'}{c}$$

$$a' = \frac{a' \cdot c}{a} \; ; \; a' = \frac{a \cdot b'}{b} \; ; \; b = \frac{b \cdot a'}{c}$$

$$c = \frac{a \cdot c'}{a'} \; ; \; a = \frac{a' \cdot b}{b'} \; ; \; b = \frac{b' \cdot c}{a'}$$

รูปที่ 2.2 ความคล้ายคลึงของสามเหลี่ยม

2.1.3 ซายน์และโคซายน์ฟังก์ชัน(Sine and Cosine Function) แสดงถึงความสัมพันธ์ของด้านของสามเหลี่ยมมุมฉาก ดังในรูปที่ 2.3



$$\text{SINE } \theta = \frac{\text{OPPOSITE SIDE}}{\text{HYPOTENUSE}} = \text{COS } \theta$$

$$\sin\theta = \frac{a}{c} \; ; \; a = c \cdot \sin\theta \; ; = c = \frac{a}{\sin\theta}$$

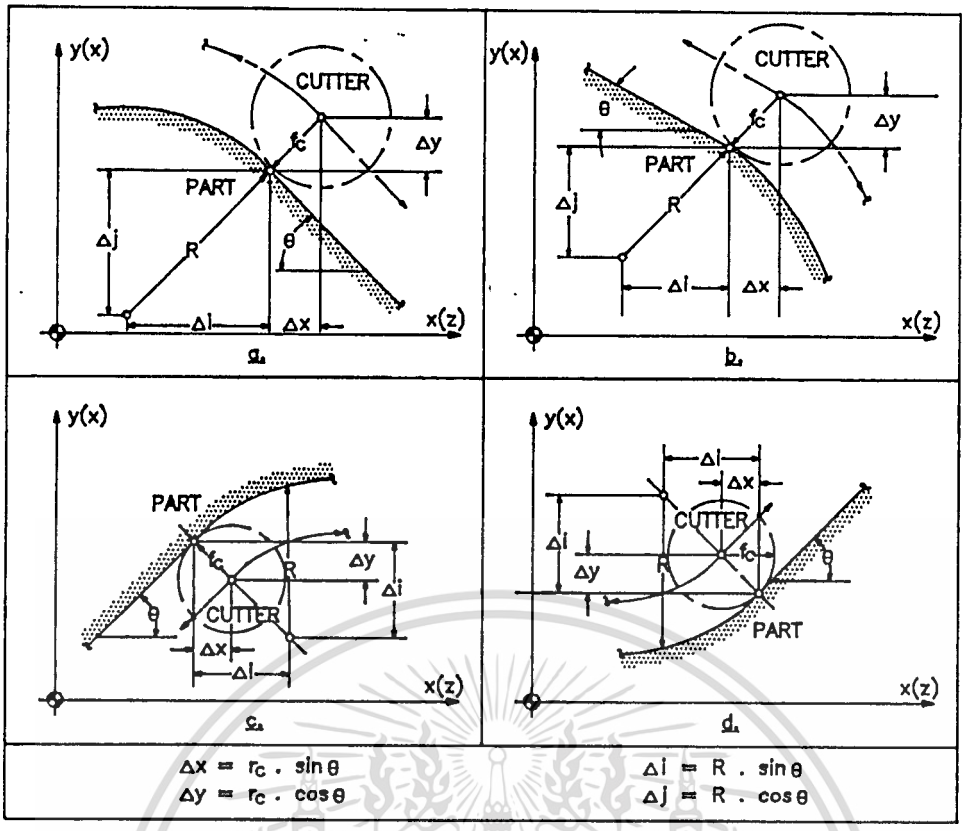
$$\cos\theta = \frac{b}{c} \; ; \; b = c \cdot \cos\theta; = c = \frac{b}{\cos\theta}$$

รูปที่ 2.3 ซายน์และโคซายน์

2.1.4 แทนเจนท์และโคแทนเจนท์ฟังก์ชัน(Tangent and Cotangent Function)

เป็นการแสดงความสัมพันธ์ของด้านสองด้านที่ตั้งฉากกัน ซึ่งบ่อยครั้งผู้ปฏิบัติงานใช้ความสัมพันธ์นี้ในการหาขนาดของมุมและความชัน และแสดงไว้ในรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แทนเจนท์และโคแทนเจนท์

2.1.5 ความสัมพันธ์ของมุมในตรีโกณมิติฟังก์ชัน(Angular Relation)

ถ้า $a = c \cdot \sin\theta$ และ $b = c \cdot \cos\theta$ สามารถแทนค่าได้

$$\tan\theta = \sin\theta/\cos\theta$$

ในทำนองเดียวกัน $\tan\theta = a/b$

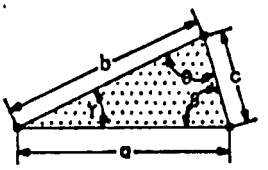
และ $b = a \cdot \cot\theta$

โดยการแทนค่าจะได้ $\tan\theta = 1/\cot\theta$

2.2 สามเหลี่ยมออฟบิท(Oblique Triangles)

ในบางครั้งผู้ปฏิบัติงานสามารถคำนวณหาด้านและมุมของสามเหลี่ยมที่ไม่ใช่สามเหลี่ยมมุมฉาก โดยพิจารณาจากความสัมพันธ์ของด้านและมุม ดังแสดงในรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



	$AREA\ (A) = \frac{a \cdot b \cdot \sin\gamma}{2}$	
	$\sin\theta = \frac{a \cdot \sin\beta}{b}$	
	$\sin\gamma = \frac{b \cdot \sin\theta}{a} = \frac{b \cdot \sin\gamma}{a}$	
$\gamma = 180 - (\theta+\beta)$	$\sin\gamma = \frac{c \cdot \sin\gamma}{a}$	
$a = \frac{b \cdot \sin\theta}{\sin\beta}$	$b = \frac{a \cdot \sin\beta}{\sin\theta}$	$c = \frac{b \cdot \sin\gamma}{\sin\beta} = \frac{a \cdot \sin\gamma}{\sin\theta}$

รูปที่ 2.5 สามเหลี่ยมออฟบิท

2.3 การวิเคราะห์ทางตรีโกณมิติ

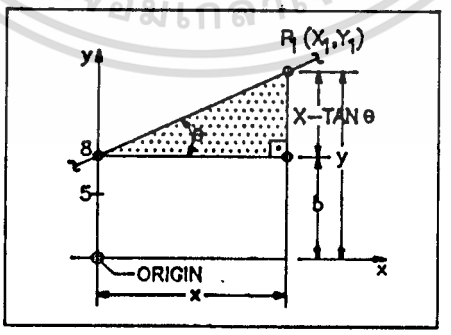
2.3.1 สมการเส้นตรง(Equation of Straight Line)

เส้นตรงสามารถกำหนดให้มีการผ่านแกนyเรียกว่า จุดตัดแกน y(y-Intercept) และมีความสัมพันธ์ที่ทำให้เกิดความชันเนื่องจาก x ที่มีค่าเป็นบวก ดังรูปที่ 2.6 ค่าความชันที่เกิดจากเส้นตรงที่ลากผ่านแกน y สามารถแสดงได้ดังสมการต่อไปนี้

$$y = m \cdot x + b$$

โดยที่ $m = \tan\theta$

$b = 8$



รูปที่ 2.6 จุดตัดแกน y

จากสมการสามารถหาขนาดของ y โดยสมมติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$x = 6$ และ $x = 8$

และถ้า $= 30^{o}$ และ $b = 8$

$m = \tan 30^{o} = 0.57735$

$y = 0.57735 \cdot x + 8$

เมื่อ $x = 6$; $y = 0.57735 \cdot 6 + 8$

$= 11.4641$

เมื่อ $x = 8$; $y = 0.57735 \cdot 8 + 8$

$= 12.6188$

กรณีมีการกำหนดจุดบนเส้นตรง (x_1, y_1) ณ จุด P_1 และความชันของเส้นสามารถหาความสัมพันธ์ในรูปของ y ดังนี้

$$y = y_1 + m \cdot (x - x_1)$$

จุดตัดบนแกน y ในกรณีที่ $x_1 = 6$ และ $y_1 = 11.4641$ สามารถแทนค่าได้ดังนี้

$\tan 30^{o} = 0.57735 = m$

$y = 11.4641 + 0.57735 \cdot (x - 6)$

$= 0.57735 \cdot x + 8$

2.3.2 สมการวงกลม(Equation of Circle)

ถ้าจุดกึ่งกลางของวงกลมอยู่ที่จุดกำเนิดของระบบโคออดิเนท(Coordinate) สมการจะสามารถเขียนได้ดังนี้

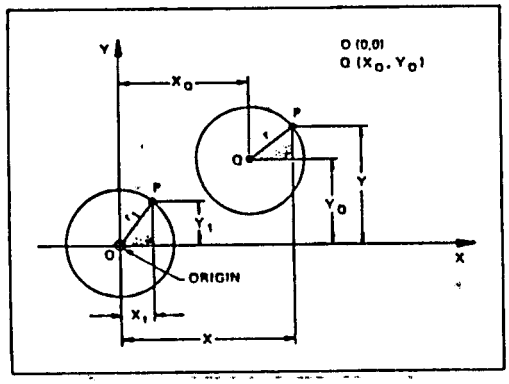
$$r^2 = x^2 + y^2$$

ในกรณีจุดกึ่งกลางของวงกลมไม่ได้อยู่ที่จุดกำเนิดของระบบโคออดิเนท แต่อยู่บนจุดใดๆ คือ $Q(x_q, y_q)$ สามารถแสดงสมการวงกลมได้ดังนี้

$$r^2 = (x - x_q)^2 + (y - y_q)^2$$

สมการนี้ใช้เป็นพื้นฐานในการหาจุดตัดในการกำหนดการเคลื่อนที่ของ CNC โดยผู้ปฏิบัติงาน ซึ่งจุดตัดดังกล่าวอาจจะอยู่ในรูปของ เส้นตรงสัมผัสเส้นโค้ง เส้นโค้งสัมผัสเส้นโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงสมการวงกลมที่มีจุดกึ่งกลางอยู่ที่จุดกำเนิดและที่จุดใดๆ

2.3.3 จุดตัดของเส้นตรงสองเส้น

สมมติเส้นตรงสองเส้นมีสมการเป็น

Line LN1 $y = 0.5 \cdot x + 3.25$

LN2 $y = -2.3 \cdot x + 7$

ถ้า m_1, m_2 คือค่าความชันของเส้นตรงสองเส้น และ b_1, b_2 ตามลำดับ จุดตัดของเส้นตรงสองเส้นอยู่ที่จุด P คือ x_P, y_P สามารถแสดงการหาจุดตัดของเส้นตรงทั้งสองได้ดังนี้

$$x_P = (b_2 - b_1)/(m_1 - m_2)$$

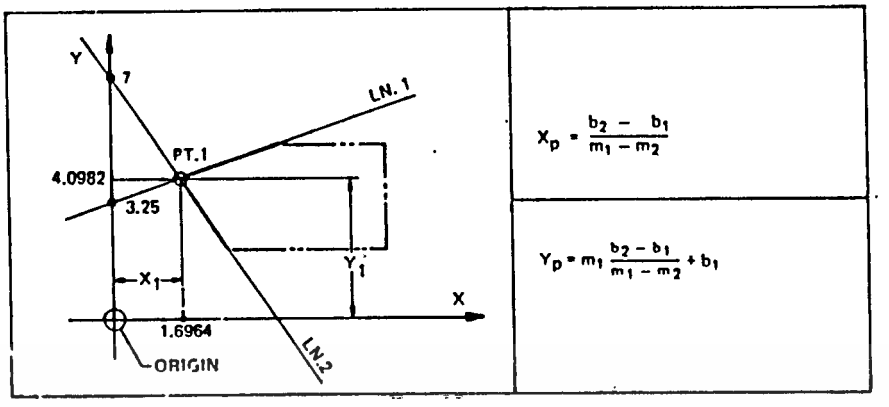
$$y_P = m_1 \cdot (b_2 - b_1) + b_1/(m_1 - m_2)$$

แทนค่า m_1, m_2, b_1, b_2 เพื่อหาค่า x_P, y_P ได้ดังนี้

$$x_P = (b_2 - b_1/(m_1 - m_2)$$
$$= 1.6964$$

$$y_P = m_1 \cdot (b_2 - b_1) + b_1/(m_1 - m_2)$$
$$= 4.0982$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงจุดตัดของเส้นตรงสองเส้น

2.3.4 จุดตัดของเส้นตรงและวงกลม

โคออดิเนทของจุดทั้งสองระหว่างเส้นตรงและวงกลม สามารถแสดงดังรูปที่ 2.9 สมการของเส้นตรง และสมการวงกลมที่มีจุดกำเนิดที่จุด x_q, y_q ใด ๆ จะสามารถแสดงได้ดังนี้

สมการเส้นตรง $y = m \cdot x + b$

และสมการวงกลม $y = y_q \pm \sqrt{(r^2 - (x - x_q)^2)}$

ตำแหน่งของจุดตัดระหว่างเส้นตรงและวงกลมที่จุด $P_1(x_1, y_1)$ และจุด $P_2(x_2, y_2)$ จะหาได้จากสมการทั้งสอง ตัวอย่างเช่น

LNI $y = x + 2$

Circle CIR $y = 4 \pm \sqrt{(4 - (x - 4)^2)}$

$x + 2 = 4 \pm \sqrt{(4 - (x - 4)^2)}$

$(x - 2)^2 = 4 - (x^2 - 8x + 16)$

$(x - 2)^2 = -x^2 + 8x - 12$

$x = [6 \pm \sqrt{(36 - 32)}]/2$

นั่นคือ

$x_1 = 4$

$x_2 = 2$

$y_1 = 6$

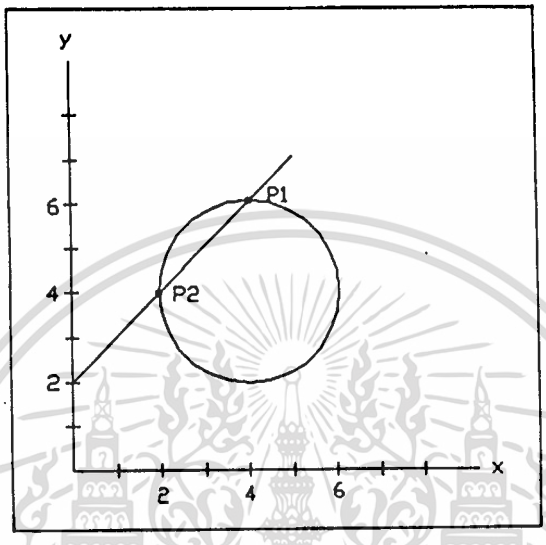
$y_2 = 4$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจุดตัดทั้งสองจะอยู่ที่

$P_1 = (4,6)$

$P_2 = (2,4)$



รูปที่ 2.9 จุดตัดของเส้นตรงและวงกลม

2.3.5 จุดตัดที่เกิดจากวงกลมสองวง

จุดตัดของวงกลมสองวงสามารถหาได้ดังนี้

วงกลมวงที่ 1 คือ $y = \sqrt{(r^2 - x^2)}$; $r = 4$

และวงกลมวงที่ 2 $y = y_q \pm \sqrt{(r^2 - (x - x_q)^2)}$; $x_q = y_q = 3$, $r = 3$

จุดตัดของวงกลมทั้งสองคือ

$$(r^2 - x^2) = y_q \pm \sqrt{(r^2 - (x - x_q)^2)}$$

แทนค่าสัมประสิทธิ์ที่กำหนดให้ (r,x_q,y_q)

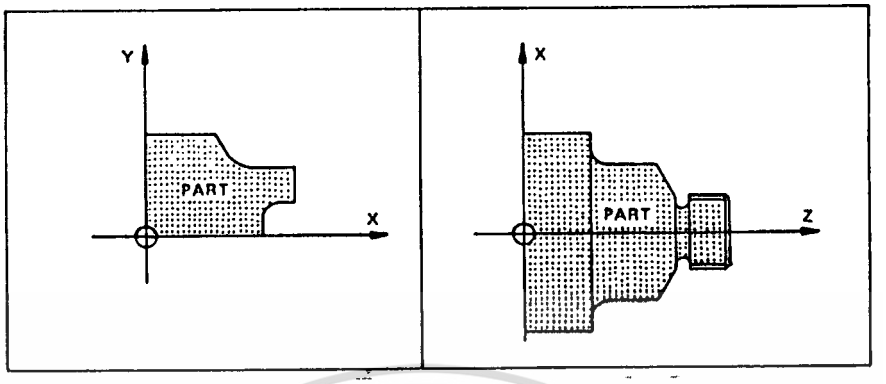
$$(16 - x^2) = 3 \pm \sqrt{(9 - (x - 3)^2)}$$

จะได้ว่า

$$x = [4.16 \pm \sqrt{(17.3056 - 2.72)}]/2$$

$$= 2.08 \pm 1.91$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงจุดตัดของวงกลมทั้งสอง

ดังนั้นค่าพิกัดบนโคออดิเนทจะเป็น

$x_1 = 3.99$

$y_1 = 0.28$

$x_2 = 0.17$

$y_2 = 3.99$

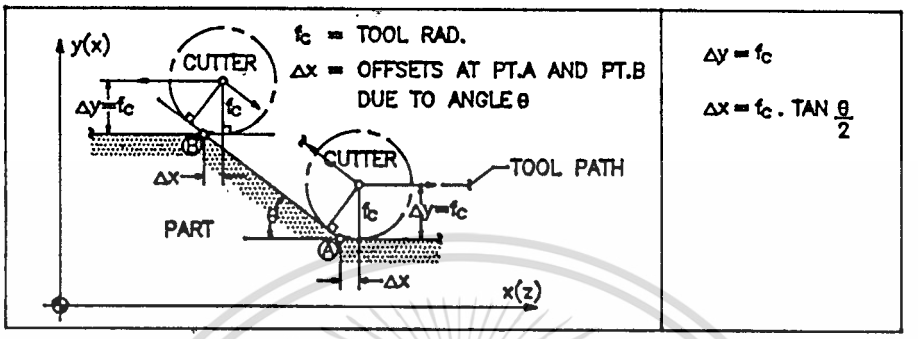
นันคือ $P_1 = (3.99, 0.28)$

$P_2 = (0.17, 3.99)$

2.4 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์(Cutter)

จากพื้นฐานตรีโกณมิติที่กล่าวแล้วข้างต้น จะมีประโยชน์อย่างมากในการคำนวณหาขนาดเผื่อระยะของทางเดินของคมตัดสำหรับงานตัดด้วยวิธีการกลึงและการกัด ซึ่งโคออดิเนทแสดงได้ดังรูปที่ 2.11

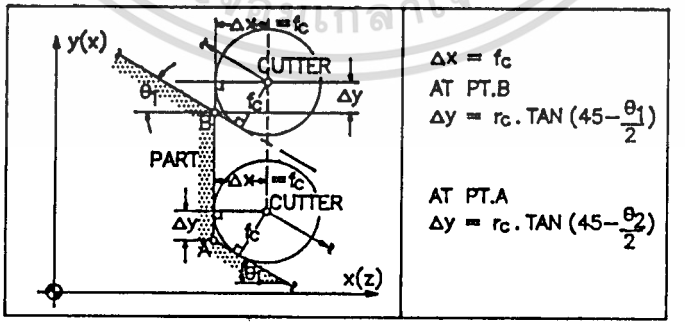
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ขนาดเผื่อระยะของทางเดินของคมตัด

2.4.1 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ที่ขนานกับแกน x โดยการวัดมุมเทียบกับแกน y สามารถหาระยะ x ซึ่งเป็นระยะพิกัดที่คัทเตอร์สัมผัสชิ้นงานพอดีโดยตรีโกนมิติ สรุปเป็นสูตรสำเร็จได้ดังรูปที่ 2.12

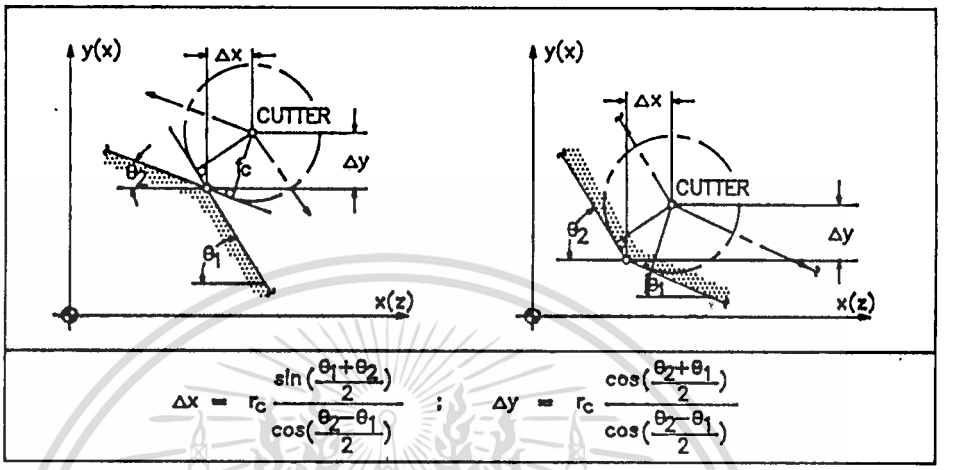


รูปที่ 2.12 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ที่ขนานกับแกน y

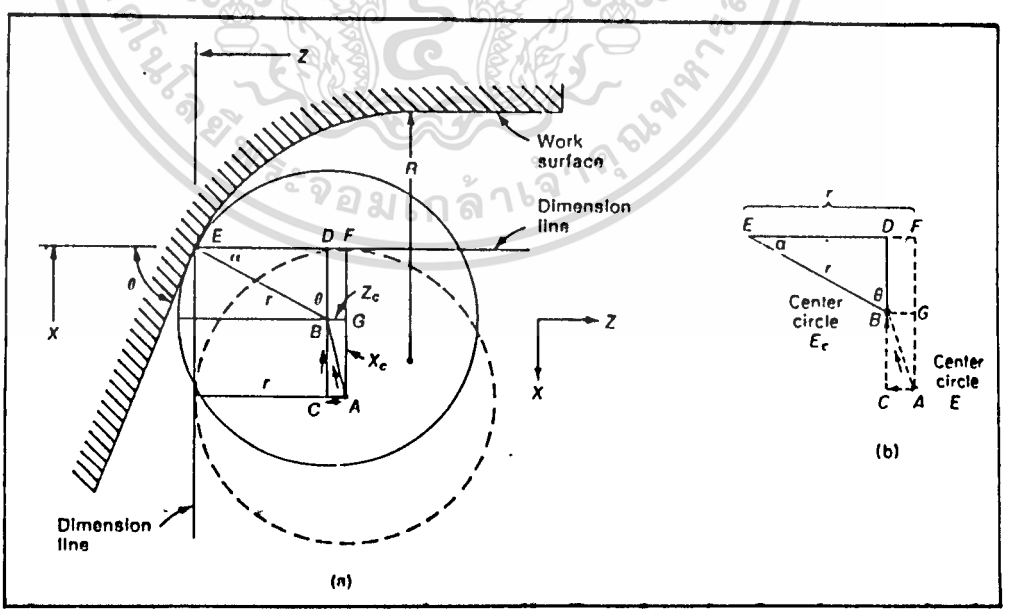
โดยการวัดมุมเทียบกับแกน x และใช้ฟังก์ชันตรีโกณมิติ สามารถสรุปเป็นสูตรสำเร็จได้ดังรูปที่ 2.13



รูปที่ 2.13 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

2.4.3 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ที่เกิดจากแนวการเคลื่อนที่สองแนว ที่ไม่ขนานกับแกน x และ y

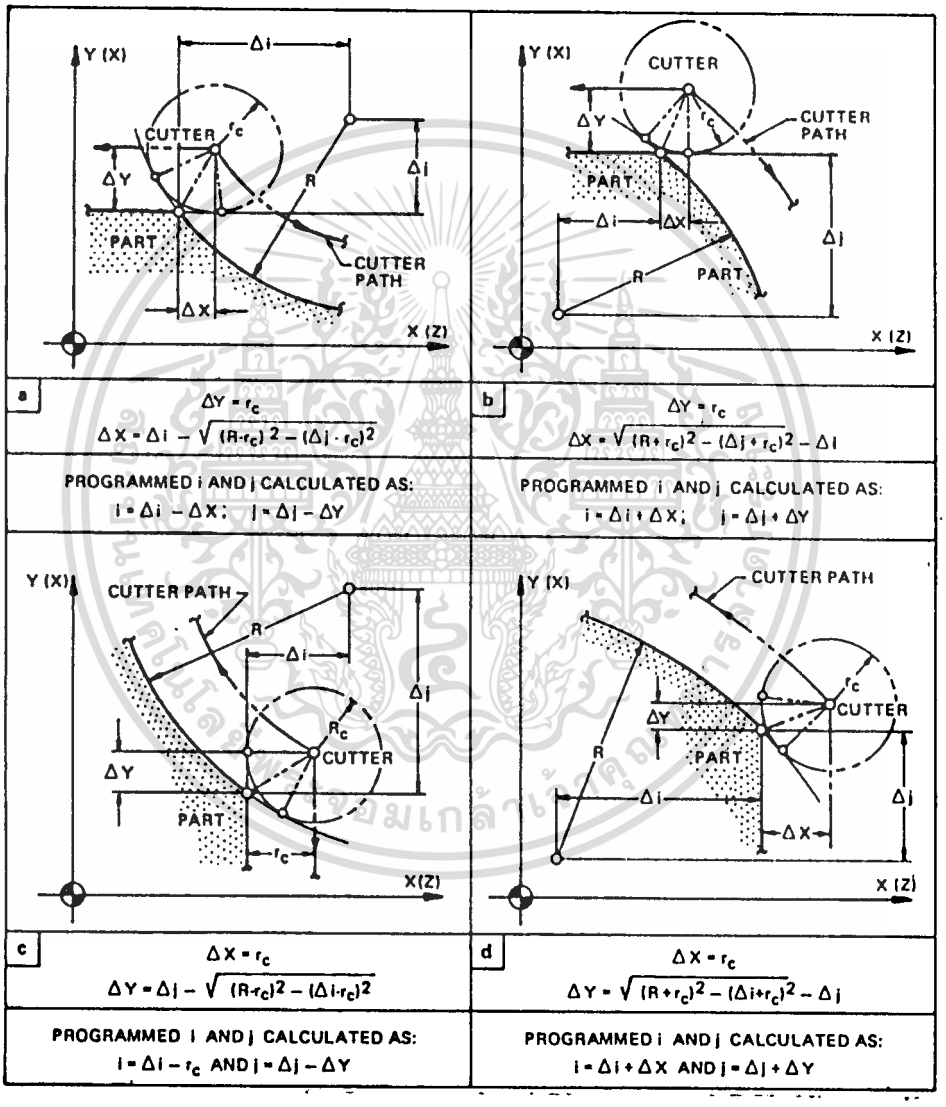
สามารถสรุปเป็นสูตรสำเร็จได้ดังรูปที่ 2.14



รูปที่ 2.14 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

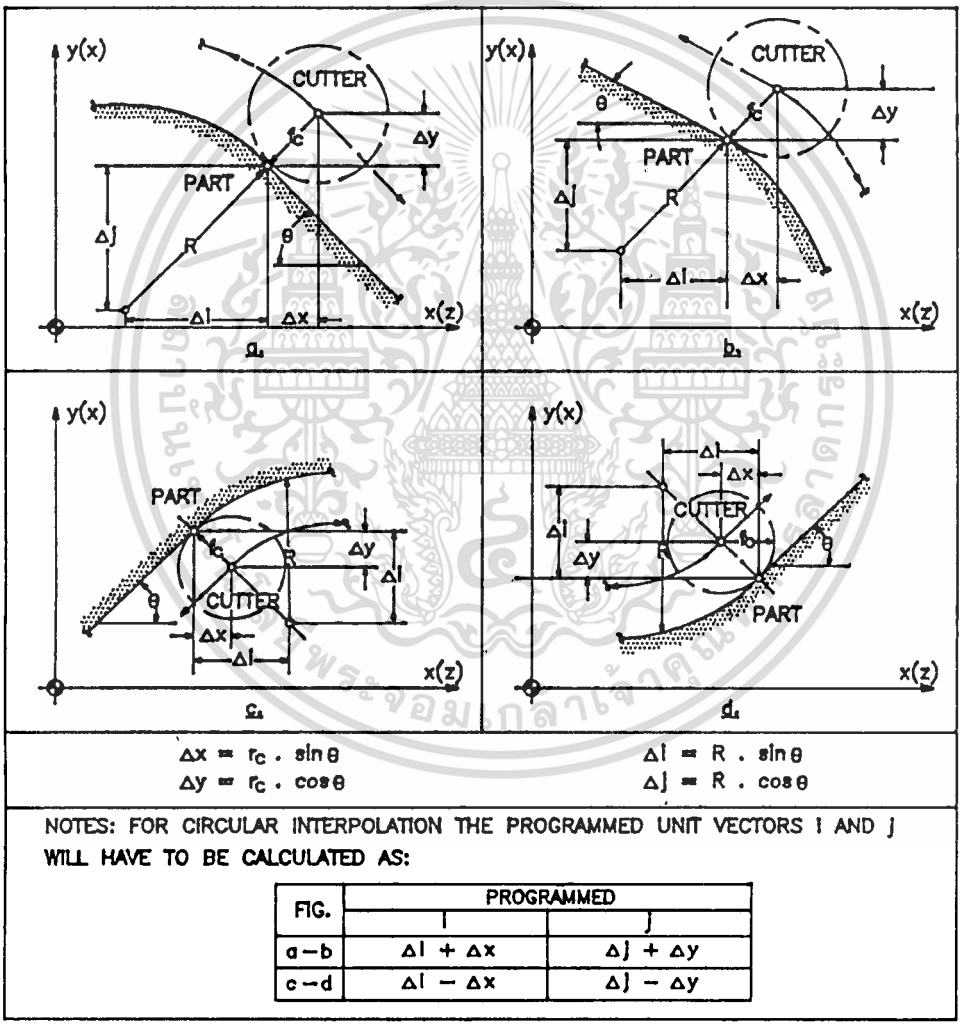
2.4.4 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ ที่เกิดจากแนวการเคลื่อนที่ของเส้นตรงและเส้นโค้ง เส้นตรงสัมผัสกับส่วนโค้ง แต่ไม่ขนานกับแกน x และแกน y ของโคออดิเนท สามารถสรุปเป็นสูตรสำเร็จได้ดังรูปที่ 2.15



รูปที่ 2.15 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ ที่เกิดจากแนวการเคลื่อนคัทเตอร์วิถีโค้งและวิถีตรง โดยที่แนวการเคลื่อนที่วิถีตรงขนานกับแกน x หรือแกน y ของโคออดิเนท สามารถสรุปเป็นสูตรสำเร็จได้ดังรูปที่ 2.16



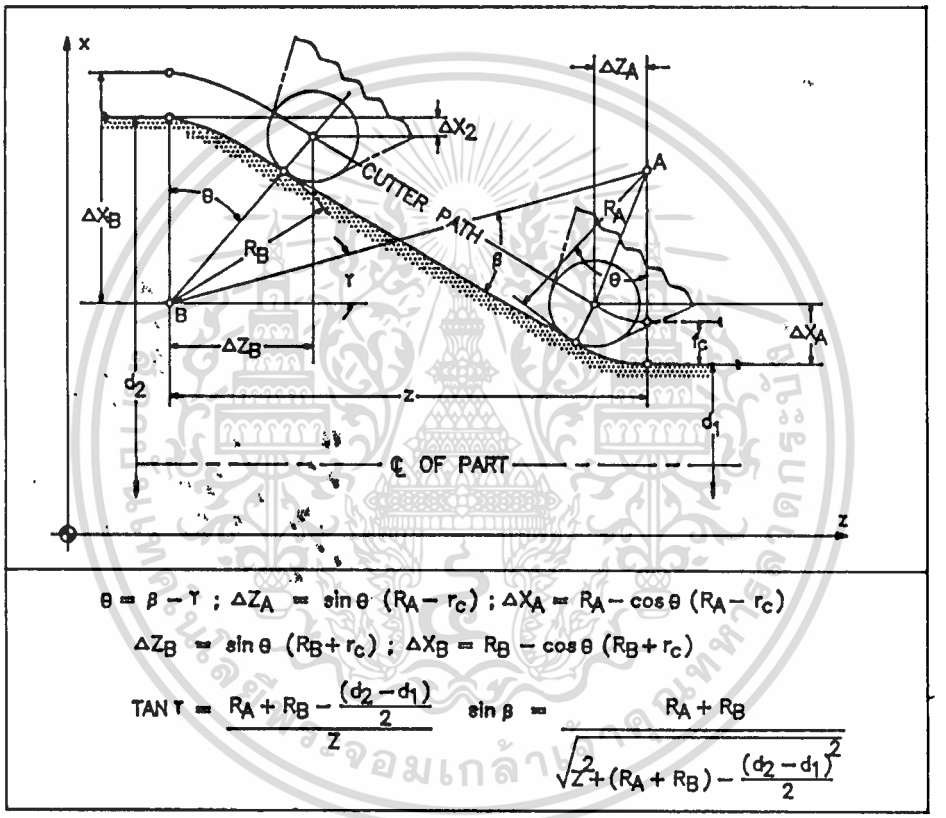
NOTES: FOR CIRCULAR INTERPOLATION THE PROGRAMMED UNIT VECTORS I AND J WILL HAVE TO BE CALCULATED AS:

FIG.	PROGRAMMED	
	I	J
a–b	Δl + Δx	Δj + Δy
c–d	Δl − Δx	Δj − Δy

รูปที่ 2.16 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์ที่เกิดจากวิถีการเคลื่อนที่เป็นส่วนโค้ง ในกรณีที่วิถีการเคลื่อนที่เป็นแนวตรงสัมผัสส่วนโค้งทั้งสอง ซึ่งจะพบบ่อยในงานกลึงและงานกัดเนื่องจากชิ้นงานโดยทั่วไปจะประกอบด้วยวิถีการเคลื่อนที่เป็นแนวตรงและโค้งสัมผัสไปมา เพื่อไม่ให้เกิดคมและเพื่อความแข็งแรงของชิ้นงาน สามารถสรุปเป็นสูตรได้ดังรูปที่ 2.17



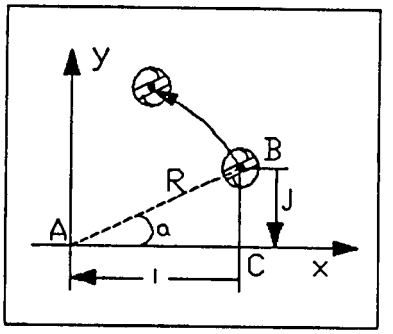
รูปที่ 2.17 จุดเปลี่ยนแนวการเคลื่อนที่ของคัทเตอร์

2.5 ยูนิทเวคเตอร์และทิศทางโคซายน์ (Unit Vector and Direction Cosine)

การบอกลักษณะหรือพฤติกรรมการเคลื่อนที่ของศมตัดโดยวิธีทางคณิตศาสตร์ ในทางเรขาคณิตยูนิทเวคเตอร์จะบอกถึงทิศทางการเคลื่อนที่ และแสดงจุดตำแหน่งของขนาด หรืออาจจะอยู่ในรูปของตำแหน่งหรือความเร็วก็ตาม และจะถูกแยกเป็นขนาดของตำแหน่งหรือความเร็ว ให้อยู่ในระนาบสองระนาบ ในกรณีโคออดิเนทประกอบขึ้นด้วยแกน x และ y ขนาดที่ถูกแยกให้อยู่ระนาบใดระนาบหนึ่งที่เกิดจากการแยกขนาดโดยรวมเรียกว่า การทาบหรือการฉาย (Projection) ลงมาในระนาบใดระนาบหนึ่ง ดังแสดงในรูปที่ 2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



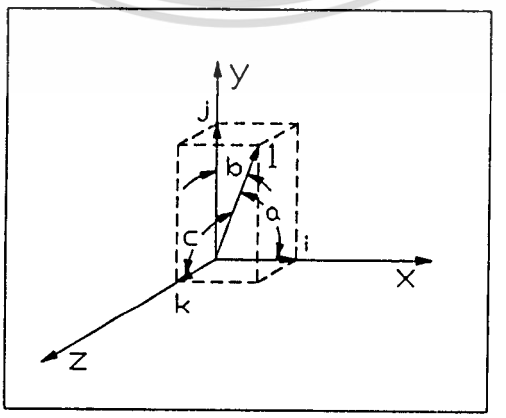
รูปที่ 2.18 การทาบหรือการฉาย

จากรูป แสดงถึงกรณีวิถีการเคลื่อนที่ของคมตัดที่เป็นเส้นโค้ง (Circular) ที่อยู่บนโคออดิเนท x และ y มิดิของ I,J บอกถึงตำแหน่งก่อนการเคลื่อนที่ หรืออาจเรียกได้ว่าจุดเริ่มต้นการเคลื่อนที่ของคมตัดเป็นส่วนโค้งซึ่งวัดจากจุดศูนย์กลางของความโค้ง สามารถใช้ทฤษฎีพิทากอรัส (Pythagoras) ในสามเหลี่ยม ABC รัศมีความโค้งสามารถหาได้ดังนี้

$$R^2 = I^2 + J^2$$

ทิศทางการเคลื่อนที่ของคมตัดแสดงโดยลูกศรบนส่วนโค้งและขนาด สามารถหาได้จากรัศมีความโค้ง ค่าของ I และ J ได้จากการทาบหรือการฉายขนาดลงมายังแกน x และ y ตามลำดับ สมมติว่าผลลัพธ์ของมิติของรัศมีความโค้งมีขนาดเท่ากับหนึ่งหน่วย ก็จะแสดงว่า i และ j คือ ยูนิทเวคเตอร์ และทิศทางโคซายน์ คือ I และ J

ขนาดและทิศทางของการเคลื่อนที่แบบวิถีโค้งที่มีรัศมีความโค้ง R จะหมายถึงการทาบหรือการฉายI,J และมียูนิทเวคเตอร์เป็นตัวบอกทิศทาง จากรูปแสดงถึงยูนิทเวคเตอร์ที่ทาบลงบนแกนสามแกนได้แกน x,y และ z สิ่งที่ได้จากการทาบได้แก่ I,J,k เรียกว่าทิศทางโคซายน์



รูปที่ 2.19 ทิศทางโคซายน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป แสดงยูนิทเวคเตอร์ และสามารถคำนวณหาค่าของ i,j และ k ดังนี้

$$1 = i^2 + j^2 + k^2$$

$$\cos a = i/\sqrt{(i^2 + j^2 + k^2)}$$

$$= i$$

$$\cos b = j/\sqrt{(i^2 + j^2 + k^2)}$$

$$= j$$

$$\cos q\, c = k/\sqrt{(i^2 + j^2 + k^2)}$$

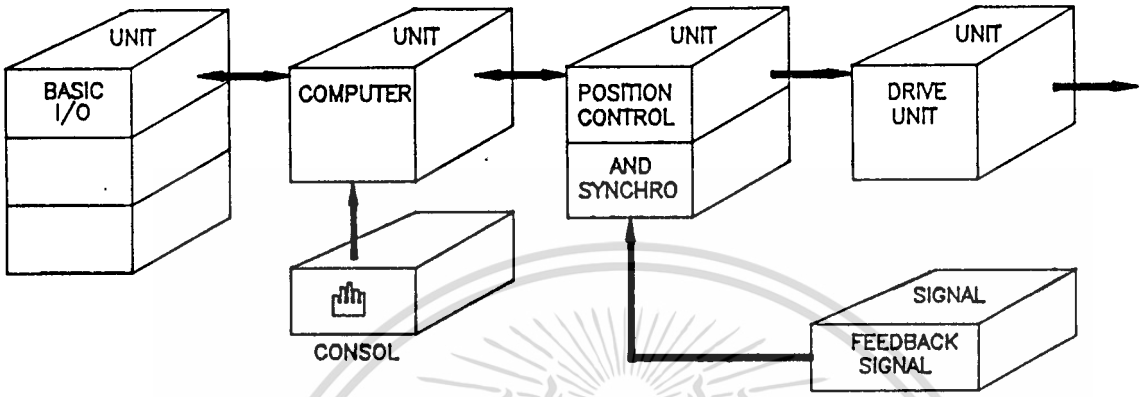
$$= k$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ หมายถึง การประยุกต์นำไมโครคอมพิวเตอร์มาใช้จำลองการทำงานของเครื่องควบคุมเชิงเลขซึ่งแต่เดิมวิทยาการทางด้านอิเล็คทรอนิคส์ไม่เจริญก้าวหน้า ดังนั้นเครื่องควบคุมเชิงเลขจึงประกอบด้วยฮาร์ดแวร์ที่มีความยืดหยุ่นทางซอฟท์แวร์น้อยมาก ฟังก์ชันหรือการคำนวณต่าง ๆ จะกระทำโดยฮาร์ดแวร์ทั้งสิ้น การใช้คอมพิวเตอร์เอนกประสงค์ในยุคเริ่มแรกของเครื่องควบคุมเชิงเลขไม่สามารถทำได้ในเชิงพาณิชย์ เนื่องจากมีขนาดใหญ่และมีมูลค่ามากเกินกว่าที่จะลงทุน ประกอบกับคอมพิวเตอร์ในยุคนั้นมีความเร็วในการปฏิบัติคำสั่งไม่มาก ต่อมาภายหลังวิทยาการทางด้านอิเล็คทรอนิคส์ได้เจริญก้าวหน้ามาจนถึงยุคที่ผลิตวงจรประเภทตรรก ที่มีความสลับซับซ้อนได้สำเร็จ จึงเกิดการพัฒนาไมโครโปรเซสเซอร์ที่มีความสามารถในการปฏิบัติคำสั่ง สามารถจำลองการคำนวณทางฮาร์ดแวร์ของเครื่องควบคุมเชิงเลขเดิมได้ทันต่อผลตอบสนองของการควบคุมการเคลื่อนที่ของโคออดิเนทเครื่องจักรโดยไม่เกิดข้อผิดพลาด อย่างไรก็ตามการพัฒนาเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ในยุคแรกจะมีมูลค่าสูงแต่ก็จะมีข้อดีในด้านการออกแบบทางฮาร์ดแวร์ที่มีรูปแบบที่แน่นอน แต่มีความยืดหยุ่นในทางซอฟท์แวร์มาก ซึ่งเป็นผลให้ความยืดหยุ่นของการทำงานสูงตามไปด้วย ความยืดหยุ่นในการทำงาน หมายถึง ความสามารถทางฮาร์ดแวร์ที่สามารถเปลี่ยนรูปแบบการทำงานได้หลายแบบโดยการเปลี่ยนแปลงซอฟท์แวร์หรือการเพิ่มหรือลดอุปกรณ์ทางฮาร์ดแวร์โดยไม่มีการแก้ไขระบบฮาร์ดแวร์เดิม ปัจจุบันไมโครโปรเซสเซอร์ได้ถูกพัฒนาปรับปรุงให้มีขนาดเล็กลง จนสามารถบรรจุลงบนวงจรรวมเพียงหน่วยเดียว มีความยืดหยุ่นทางซอฟท์แวร์และฮาร์ดแวร์สูง ความเร็วในการปฏิบัติคำสั่งในการคำนวณช้ากว่าระบบไมโครโปรเซสเซอร์เดิมที่ใช้โครงสร้างแบบ Slice แต่สามารถแบ่งการทำงานออกเป็นส่วนย่อยเพื่อใช้กับไมโครโปรเซสเซอร์หลายหน่วยที่รับผิดชอบการทำงานในส่วนย่อยนั้น ซึ่งจะกล่าวในลำดับต่อไป การใช้คอมพิวเตอร์หลายหน่วยในการทำงานร่วมกันเพื่อจุดประสงค์ของงานหนึ่งระบบเรียกว่า มัลติโปรเซสเซอร์(Multi-processor) ระบบโดยทั่วไปของเครื่องควบคุมเชิงเลขสามารถแสดงได้ดังบล็อกไดอะแกรมดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.1 บล็อกไดอะแกรมแสดงระบบของเครื่องควบคุมเชิงเลข

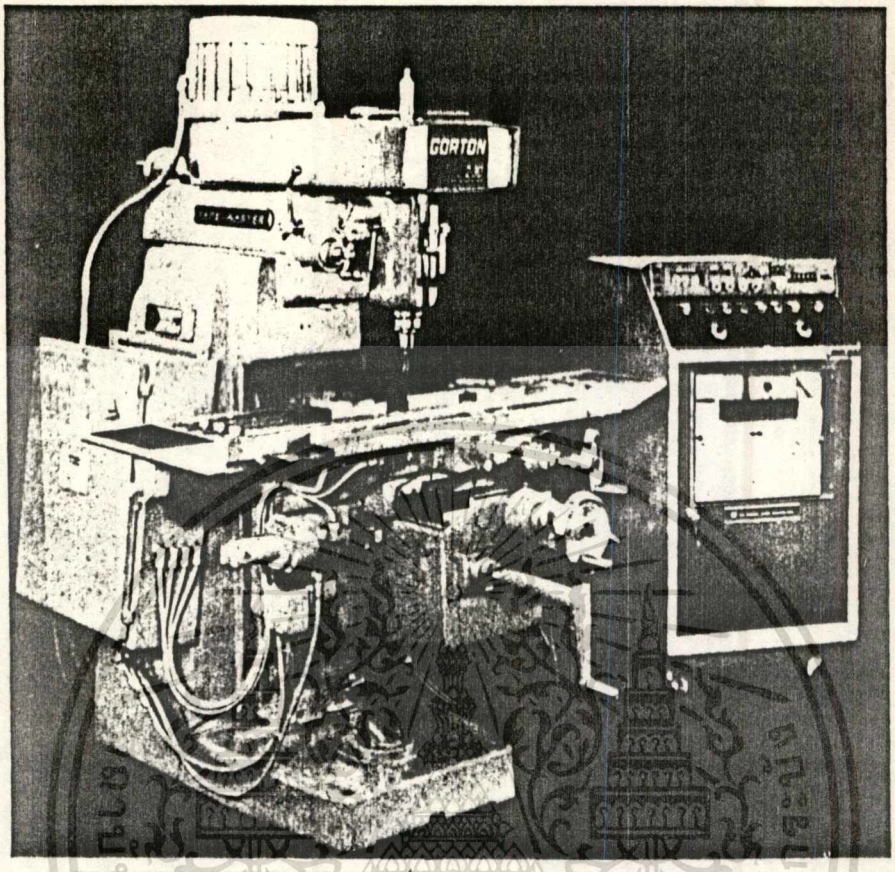
ระบบควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ สามารถแบ่งเป็นหน่วยย่อยได้ 6 หน่วยคือ

1. หน่วยคอมพิวเตอร์กลาง(Central Computer Unit)
2. หน่วยอินพุทเอาท์พุทพื้นฐาน(Basic Input/Output Unit)
3. หน่วยคอนโซล(Console Unit)
4. หน่วยควบคุมตำแหน่งและซิงโครไนซ(Position and Synchronization Control Unit)
5. หน่วยขับกำลัง(Power Drive Unit)
6. หน่วยต้นกำลัง

3.1 หน่วยคอมพิวเตอร์กลาง

หน่วยคอมพิวเตอร์กลางทำหน้าที่ คำนวณ กำหนดตำแหน่งการเคลื่อนที่ของโคออดิเนท บริหารระบบให้เป็นไปตามเงื่อนไขการทำงานของเครื่องจักรแต่ละชนิด ระบบโคออดิเนทของเครื่องจักรโดยทั่วไปเป็นคาร์เทเซียน(Cartesian)แบบ 1 ระนาบขึ้นไป รูปที่ 3.2 แสดงให้เห็นถึงลักษณะของเครื่องที่พบเห็นในกระบวนการผลิตทั่วไปแบบ 1 ระนาบ และแบบ 2 ระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.2 เครื่องจักรกลที่ใช้ระบบควบคุมเชิงเลข

3.1.1 ระบบโคออดิเนทของเครื่องจักร

การเคลื่อนที่ส่วนประกอบของเครื่องจักร ที่ทำให้เกิดโคออดิเนทเพียงส่วนประกอบเดียวโดยอิสระเรียกว่าแกนของโคออดิเนท โคออดิเนทจะเกิดขึ้นได้ต้องประกอบขึ้นด้วยแกนของโคออดิเนทตั้งแต่ 2 แกนขึ้นไป และแกนของโคออดิเนทที่น้อยที่สุดดังกล่าวข้างต้นเป็นผลทำให้เกิดระนาบ เครื่องจักรที่แสดงในรูปที่ 3.2 จะเป็นลักษณะของเครื่องจักร 2 แกนและ 3 แกน ได้แก่เครื่องกลึงและเครื่องกัด

จุดโคออดิเนทเหล่านี้ จะถูกเปรียบเหมือนจุดที่ปลายปากกาสัมผัสกับกระดาษเมื่อปลายปากกาเคลื่อนที่ไปก็จะเกิดทางเดินของจุด ทางเดินของจุด หรือโคออดิเนทซึ่งจะถูกสร้างหรือกำเนิดจากหน่วยคอมพิวเตอร์กลางนี้แบ่งได้สองประเภทคือ

3.1.1.1 เส้นตรง หมายถึงการเชื่อมต่อจุดสองจุดด้วยเส้นตรงได้แก่จุดเริ่มต้นของการเคลื่อนที่ซึ่งอยู่ที่ตำแหน่งปัจจุบัน และจุดสิ้นสุดการเคลื่อนที่ของโคออดิเนทตามค่าที่ถูกกำหนดในโปรแกรมที่ถูกจัดทำขึ้นโดยผู้ปฏิบัติงานภาพเส้นตรงที่ปรากฏขึ้นในระนาบใด ๆ สามารถแบ่งออกได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) ภาพเส้นตรงที่เกิดขึ้นในระนาบ2 มิติ คือภาพที่เกิดจากการเคลื่อนที่ของเครื่องจักร 2 แกน เช่น เครื่องกลึง เครื่องตัดประเภทต่าง ๆ และเครื่องฉลุ เป็นต้น ภาพเส้นตรงที่เกิดขึ้นในระนาบ 2 มิตินี้ยังสามารถแบ่งออกเป็นสองลักษณะ คือ

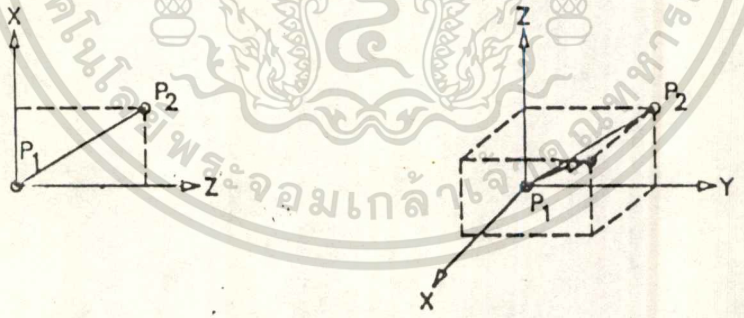
ก. ภาพที่เกิดในระนาบ2มิติที่มีการหมุนแกนใดแกนหนึ่งไปรอบๆ แกนที่เหลือ ซึ่งรูปทรงที่เกิดจากการหมุนแกนใดแกนหนึ่งไปรอบแกนที่เหลือจะเป็นรูปทรงกระบอก 3 มิติ เช่น เครื่องกลึงหรือเทอนนิงแมชชีน (Turning Machine) ซึ่งก็คือเครื่องจักรประเภทที่มีการหมุนชิ้นงานรอบตัวเอง

ข. ภาพที่เกิดในระนาบ 2 มิติ ที่ไม่มีการหมุนแกนใดแกนหนึ่งไปรอบแกนที่เหลือ ซึ่งภาพเส้นตรงที่เกิดขึ้นเปรียบเหมือนการลากเส้นตรงลงบนกระดาษ

(2) ภาพเส้นตรงที่เกิดขึ้นในระนาบ 3 มิติ คือภาพที่เกิดจากการเคลื่อนที่ของเครื่องจักรประเภท 3 แกน ได้แก่เครื่องกัด เป็นต้น ภาพเส้นตรงที่เกิดขึ้นในระนาบ 3 มิติ สามารถแบ่งออกเป็น 2 ลักษณะเช่นเดียวกับภาพเส้นตรงที่เกิดขึ้นในระนาบ 2 มิติคือ

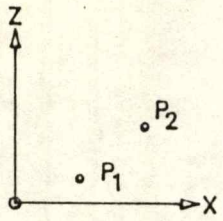
ก. ภาพเส้นตรงที่เกิดจากการหมุนของแกนสองแกนใด ๆ รอบแกนที่เหลือ ซึ่งเปรียบเหมือนการวางโคออดิเนทแบบทรงกระบอกไว้บนระนาบ 3 มิติ ดังรูปที่ 3.5 ซึ่งจะมีลักษณะคล้ายภาพที่เกิดขึ้นในระนาบ 2 มิติ ปรากฏการณ์ที่กล่าวมานี้จะเกิดขึ้นกับเครื่องจักรประเภท 4 แกน

ข. ภาพเส้นตรงที่เกิดขึ้นในระนาบ 3 มิติ ที่ไม่มีการหมุนแกนใด ๆ รอบแกนที่เหลือคือการลากเส้นตรงบนระนาบ 3 มิติ ที่พบเห็นโดยทั่วไป

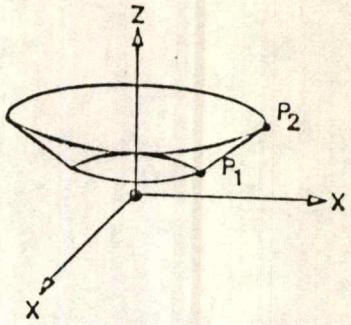


รูปที่ 3.3 แสดงโคออดิเนทของเครื่องจักร 2 แกน และ 3 แกน

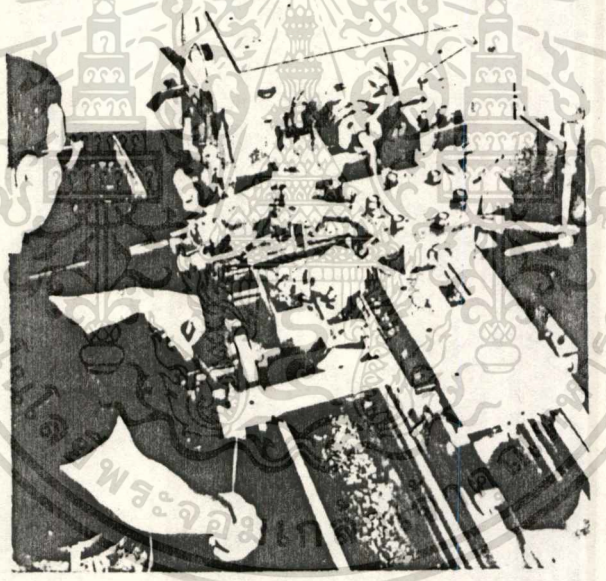
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปสองมิติที่ไม่มีการหมุนแกนใดแกนหนึ่ง-
รอบแกนที่เหลือ

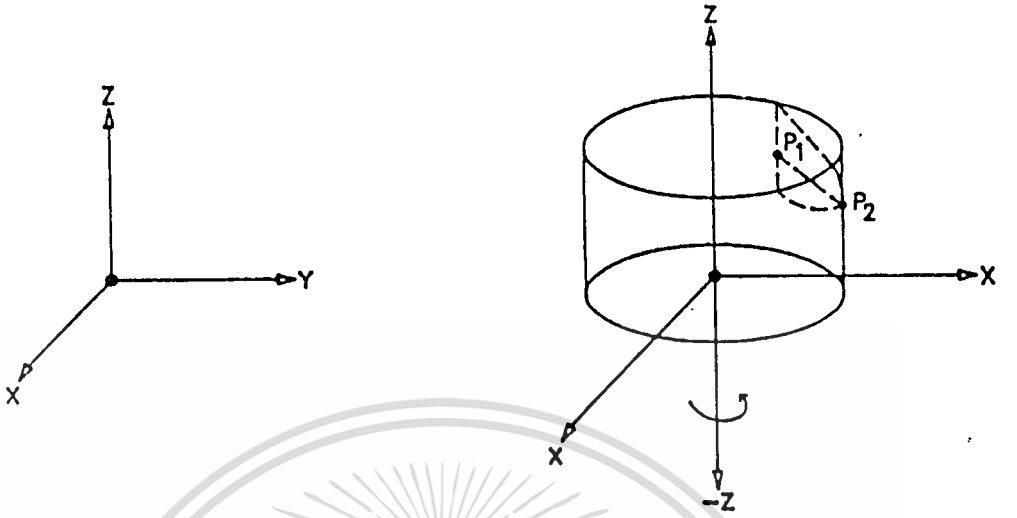


รูปสองมิติที่มีการหมุนแกน X
ไปรอบ ๆ แกน Z



รูปแสดงเครื่องกลึง
รูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



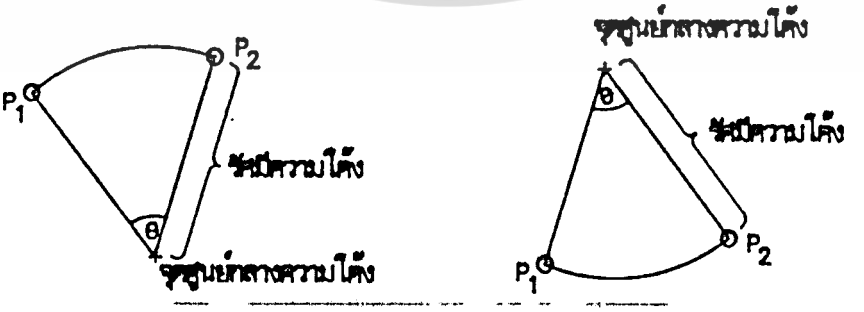
รูปที่ 3.5 ภาพเส้นตรงที่เกิดจากการหมุนของแกนสองแกนใด ๆ รอบแกนที่เหลือ

3.1.1.2 เส้นโค้ง หมายถึงการเชื่อมจุดสองจุด ซึ่งได้แก่จุดเริ่มต้นของการเคลื่อนที่ และจุดสุดท้ายหรือจุดสิ้นสุดของการเคลื่อนที่ โดยจุดทั้งสองนี้จะถูกเชื่อมต่อด้วยการเคลื่อนที่ในลักษณะส่วนโค้งของวงกลมบนระนาบ 2 มิติ การเคลื่อนที่ในลักษณะส่วนโค้งจะมีตัวแปรที่สำคัญสองตัวแปรคือ

(1) ค่ารัศมีหรือเส้นผ่าศูนย์กลางของส่วนโค้งที่ประกอบกันเป็นวงกลมสมบูรณ์

(2) มุมของจุดสัมผัสส่วนโค้ง หรือมุมของเส้นรัศมีที่สัมผัสกับจุดสองจุด

รูปที่ 3.6 แสดงถึงตัวแปรทั้งสองนี้



รูปที่ 3.6 ค่ารัศมีหรือเส้นผ่าศูนย์กลางและมุมของจุดสัมผัสของส่วนโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปทรงที่ได้จากการกัดหรือกลึงชิ้นงานทุกชนิดจะเป็นรูปทรงทางเรขาคณิตกล่าวคือจะประกอบด้วยทางเดินของจุด 2 แบบคือ เส้นตรงและเส้นโค้ง ดังนั้นลักษณะของคำสั่งของเครื่องควบคุมเชิงเลขจะถูกแบ่งเป็นส่วนย่อย ๆ ซึ่งในส่วนย่อยของคำสั่งจะประกอบด้วยจุดเริ่มต้น ลักษณะการเคลื่อนที่ของจุดว่า เป็นเส้นตรงหรือเส้นโค้ง จุดสิ้นสุด และความเร็วในการเคลื่อนที่ของจุด (Feed Rate) ซึ่งเป็นความเร็วเชิงเส้นของจุดต่อจุดหรือความเร็ววิถีของเส้นที่กำลังเคลื่อนที่ไปยังจุดสิ้นสุดด้วยความเร็วสม่ำเสมอ

การเคลื่อนที่ข้างต้นที่กล่าวมานี้เป็นการเคลื่อนที่ของคมตัด (Tools) โดยจุดดังกล่าวจะหมายถึงปลายของคมตัดที่สัมผัสกับชิ้นงานเมื่อมีการเคลื่อนที่เกิดขึ้น คมตัดที่สัมผัสกับชิ้นงานจะตัดชิ้นงานเป็นวิถีเส้นตรงหรือวิถีเส้นโค้ง ดังนั้นรูปทรงและขนาดของชิ้นงานหมายถึงการประกอบกันของส่วนย่อย ๆ ของวิถีการเดินทางของเส้นตรงและหรือเส้นโค้ง ซึ่งรูปทรงและขนาดของชิ้นงานนี้จะขึ้นอยู่กับค่าของตัวแปรที่กำหนดโดยผู้ปฏิบัติงาน

3.1.2 โปรแกรมสำหรับผู้ปฏิบัติงาน

ส่วนย่อยของคำสั่งที่กล่าวมาข้างต้นจะเรียกว่าบล็อก (Block) ซึ่งในบล็อกหนึ่งๆ จะประกอบด้วยข้อมูลค่าหรือพิกัดเริ่มต้น พิกัดสุดท้าย และความเร็วของวิถีการเคลื่อนที่ โดยชนิดของวิถีที่เดินทางจะถูกกำหนดด้วยรหัสที่ใช้เป็นคำสั่งของเครื่องควบคุมเชิงเลข ซึ่งแบ่งออกได้เป็น 3 กลุ่มคือ

3.1.2.1 รหัส G คือรหัสที่ใช้เป็นคำสั่งเกี่ยวกับโคออดิเนท และชนิดของเดินทางแบบต่าง ๆ ตัวอย่างเช่น G00 x 100 Z 50 มีความหมายว่าให้เคลื่อนแกน x ไปในทิศทางบวก 100 ไมครอน และเคลื่อนแกน Z ออกไป 50 ไมครอน ในแนวเส้นตรงด้วยความเร็วสูงสุด

3.1.2.2 รหัส M คือรหัสที่เป็นคำสั่งเกี่ยวกับตัวเครื่องจักรเช่น M05 หมายถึงให้มอเตอร์ที่ทำหน้าที่เป็นสพินเดิล(Spindle) หยุดการทำงาน

3.1.2.3 รหัส T คือรหัสที่เป็นคำสั่งเกี่ยวกับการเลือกเครื่องมือกล ซึ่งโดยทั่วไปเครื่องจักรประเภทที่ใช้เครื่องควบคุมเชิงเลขเป็นตัวควบคุมจะมีเครื่องมือกลอยู่หลายหน่วย ซึ่งมีลักษณะการใช้งานที่แตกต่างกันไปตามขั้นตอนการแมชชีน เช่น งานกลึงจะต้องมีการกลึงผิวนอก ด้านหน้า ด้านข้าง มุมด้านหลัง การเจาะและคว้าน ซึ่งงานลักษณะนี้อาจใช้เครื่องมือกลแบบเดียวกันในการแมชชีนด้านหน้า และด้านข้าง หลังจากนั้นให้เครื่องจักรเปลี่ยนเครื่องมือกลเพื่อใช้สำหรับกลึงด้านหลังงานซึ่งมีขนาดเล็กและคมตัดอยู่ในทิศทางตรงกันข้ามกับเครื่องมือกลที่ใช้แมชชีนด้านหน้า แล้วเริ่มทำการแมชชีน หลังจากนั้นให้เครื่องจักรเปลี่ยนเป็นเครื่องมือกลที่ใช้เจาะและคว้านในตัวเดียวกันเป็นขั้นตอนไป เป็นต้น

โปรแกรมสำหรับผู้ปฏิบัติงานจะประกอบไปด้วยรหัสที่ใช้เป็นคำสั่งสำหรับเครื่องควบคุมเชิงเลขเพื่อให้เครื่องควบคุมเชิงเลขควบคุมวิถีการเคลื่อนที่ของแกนเครื่องจักร ให้เป็นไปตามวัตถุประสงค์ของรายละเอียดของการแมชชีน ซึ่งจะประกอบด้วยกลุ่มของบล็อกโปรแกรม และนอกจากวิถีที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นตรงและเส้นโค้งแล้ว ยังมีรหัสที่มีวิถีการเคลื่อนที่เป็นฟังก์ชันของการตัดเกลียว การเจาะ การคำนวณเพื่อชดเชยรัศมีของคมตัดที่มีผลต่อการเคลื่อนที่เป็นมุมหรือส่วนโค้ง ฟังก์ชันเหล่านี้ถูกออกแบบเพื่อให้บล็อกโปรแกรมมีขนาดเล็กลง ทำให้การทำงานของผู้ปฏิบัติงานมีความสะดวกมากขึ้น ยังผลให้เกิดการผิดพลาดในการทำงานต่ำ คำสั่งที่มีฟังก์ชันพิเศษเหล่านี้เรียกว่า ไซเกิล (CycleProgram) ซึ่งจะมีลักษณะการทำงานเทียบเท่ากลุ่มของบล็อกคำสั่ง ตัวอย่างเช่นการตัดเกลียว การตัดมุมเชิงลาด การเจาะที่มีการถอนดอกสว่านเพื่อคายเศษวัสดุที่เหลือจากการเจาะ

3.1.3 มาตรฐานโครงสร้างของบล็อกโปรแกรม

ผู้ปฏิบัติงานสามารถสร้างบล็อกโปรแกรม ภายนอกระบบเครื่องควบคุมเชิงเลขได้สองวิธีคือ

3.1.3.1 โดยใช้แถบกระดาษ วิธีนี้เป็นการสร้างบล็อกโปรแกรมด้วยการเจาะเป็นรหัสบนแถบกระดาษ ซึ่งใช้มาตั้งแต่ยุคเริ่มแรกของเครื่องควบคุมเชิงเลข เนื่องจากเทคโนโลยีทางด้านหน่วยความจำ ตลอดจนอุปกรณ์เชื่อมต่อข้อมูลยังไม่เจริญรุดหน้าเหมือนในปัจจุบันจึงอาศัยรหัสที่ถูกเจาะไว้บนแถบกระดาษต่อเนื่องกันเป็นบล็อกโปรแกรม แถบกระดาษที่เจาะรูเป็นรหัสนี้เปรียบได้กับหน่วยความจำแบบอ่านได้อย่างเดียวหรือ ROM (Read Only Memory)

3.1.3.2 การเก็บข้อมูลที่สร้างเป็นบล็อกโปรแกรมไว้บนระบบจัดเก็บข้อมูล ของเครื่องคอมพิวเตอร์ส่วนบุคคล และทำการถ่ายข้อมูลจากหน่วยความจำหรือระบบจัดเก็บข้อมูลของเครื่องคอมพิวเตอร์ส่วนบุคคลผ่านระบบเชื่อมต่อแบบอนุกรมชนิด RS-232 หรือ RS-358 เข้าสู่หน่วยความจำของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ วิธีนี้ได้ถูกใช้อย่างแพร่หลายในปัจจุบัน อย่างไรก็ดี วิธีทั้งสองที่กล่าวข้างต้นจำเป็นจะต้องมีมาตรฐานโครงสร้างของบล็อกโปรแกรมที่แน่นอน เพื่อให้โปรแกรมที่ถูกสร้างขึ้นสามารถเทียบเคียงหรือเข้ากันได้กับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์เครื่องอื่น ๆ มาตรฐานของรหัสของเครื่องควบคุมเชิงเลข และเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์มี 2 ชนิดคือ

(1) EIA (Electrical Industrial Association)

(2) ISO (International Standard Organization)

ลักษณะมาตรฐานของรหัสทั้งสองนี้มีส่วนคล้ายคลึงกับรหัส ASCII (American Standard Code Information Interchange) ซึ่งประกอบด้วยอักษร 26 ตัว และสัญญลักษณ์ต่าง ๆ รวม 128 รหัส ลักษณะโครงสร้างของโปรแกรมสำหรับมาตรฐานข้อมูลทั้งสอง(ISO และ EIA) ประกอบด้วยส่วนสำคัญเรียงตามลำดับ คือ

(1) Tape Start เป็นรหัสที่ใช้บอกให้ทราบถึงจุดเริ่มต้นของเส้นเทป รหัสนี้จะอยู่ตอนต้นของเส้นเทป เมื่อเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์อ่านพบรหัสนี้ครั้งแรก ก็จะถือว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นจุดเริ่มต้นของเส้นเทป และเมื่ออ่านพบในครั้งต่อไป จะถือว่าเป็นการสิ้นสุดของเส้นเทป เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์จะหยุดอ่านเส้นเทปทันที

(2) Leader Sectionand LabelSkip หมายถึงรหัสหรือข้อมูลต่าง ๆ ที่อาจเจาะรูไว้เพื่อความเข้าใจของผู้ปฏิบัติงาน ตัวอย่างเช่น การเจาะรูเป็นสัญญลักษณ์บนหรือตัวอักษรเพื่อการดู มิใช่การอ่านเพื่อเป็นรหัสของโปรแกรมหลังจากรหัส Tape Start ดังกล่าวข้างต้น หลังจากสิ้นสุดส่วนของLeaderต่าง ๆ แล้วจะต้องปิดท้ายด้วยรหัส LF (Line Feed) สำหรับมาตรฐาน ISO หรือ CR (Carriage Return) สำหรับมาตรฐาน EIA เพื่อแสดงถึงขอบเขตของ Leader Section

(3) Program Start คือสัญญลักษณ์ที่บอกให้ทราบว่าเป็นจุดเริ่มต้นของบล็อกโปรแกรมแรก

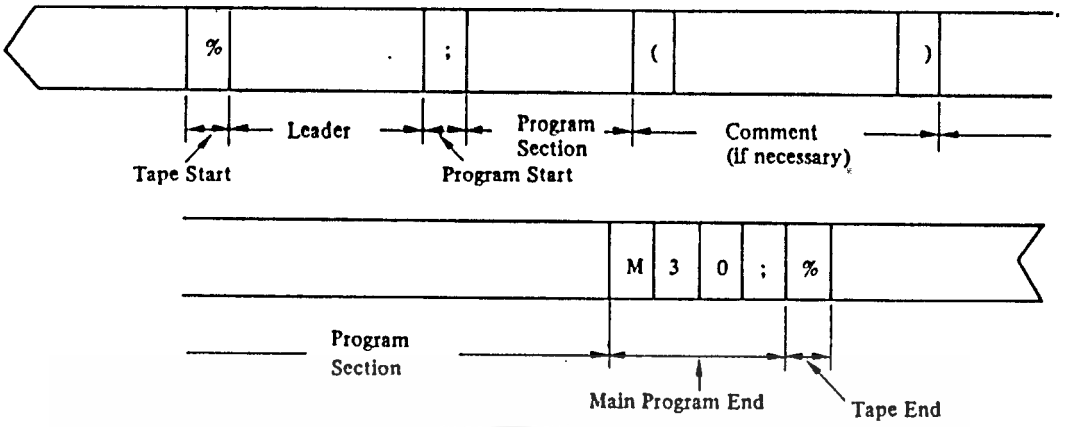
(4) Program Section หมายถึงโปรแกรมบนเส้นเทปที่อยู่ระหว่าง Program Start และ Program End ซึ่ง Program Section จะเป็นข้อมูลเกี่ยวกับการเคลื่อนที่ของแกนเครื่องจักร และการเปิดปิดเครื่องจักร ข้อมูลของโปรแกรมบนเส้นเทปส่วนนี้ถือเป็นข้อมูล ที่มีความสำคัญต่อ CNC และจะเรียกส่วนของโปรแกรมบนเส้นเทปส่วนนี้ว่า Significant Section โดยจะ แตกต่างจากส่วนที่ CNC ไม่ได้ใช้ประโยชน์ในการปฏิบัติคำสั่งหรือปฏิบัติงานต่อ CNC ซึ่งจะเรียกว่า Non-Significant ที่เป็นส่วนอำนวยความสะดวกสำหรับผู้ปฏิบัติงาน แต่จะไม่มีประโยชน์สำหรับการปฏิบัติงานหรือปฏิบัติคำสั่งของ CNC โดยตรงลักษณะโครงสร้างของ Program Section แสดงดังรูปที่ 3.7

(5) Comment หมายถึงส่วนที่มีไว้สำหรับให้ผู้ปฏิบัติงานบันทึกรายละเอียดของโปรแกรมนั้นๆ ว่าเป็นโปรแกรมอะไร ส่วนนี้จะไม่เกี่ยวข้องกับการทำงานของ CNC แต่อย่างใดและส่วนนี้จะอยู่ภายในเครื่องหมาย ( )

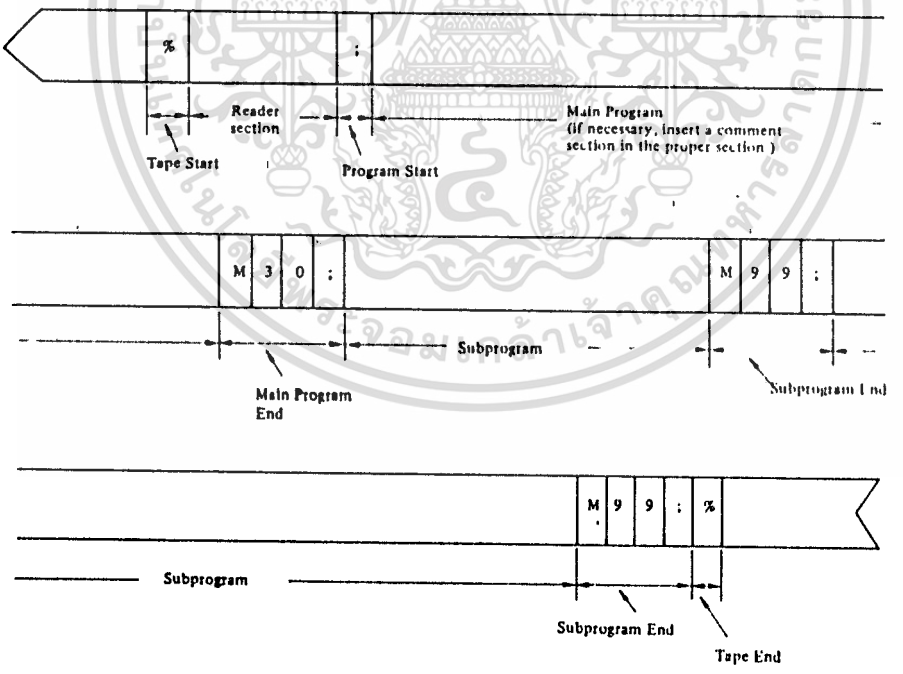
(6) Program End แสดงจุดสิ้นสุดการทำงานของโปรแกรม ใช้สำหรับโปรแกรมบนเส้นเทปที่เป็นโปรแกรมหลักและโปรแกรมย่อย โดยที่บนเส้นเทปจะมีโครงสร้างเหมือนกันยกเว้นรหัสปิดท้ายบล็อกสุดท้ายของโปรแกรม จะใช้ M30 สำหรับโปรแกรมหลัก และใช้ M99 สำหรับโปรแกรมย่อย อย่างไรก็ตามในทางปฏิบัติ โปรแกรมหลักและโปรแกรมย่อยสามารถนำมารวมกันบนเส้นเทปเส้นเดียวกันได้ ซึ่งมีโครงสร้างดังรูปที่ 3.8

(7) Tape End เป็นรหัสที่บอกให้ CNC หยุดการอ่านเส้นเทป (ดูคำอธิบายในหัวข้อ Tape Start)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 3.7 รูปแสดงโครงสร้างของโปรแกรมบนเส้นเทปในส่วนของ Program Section



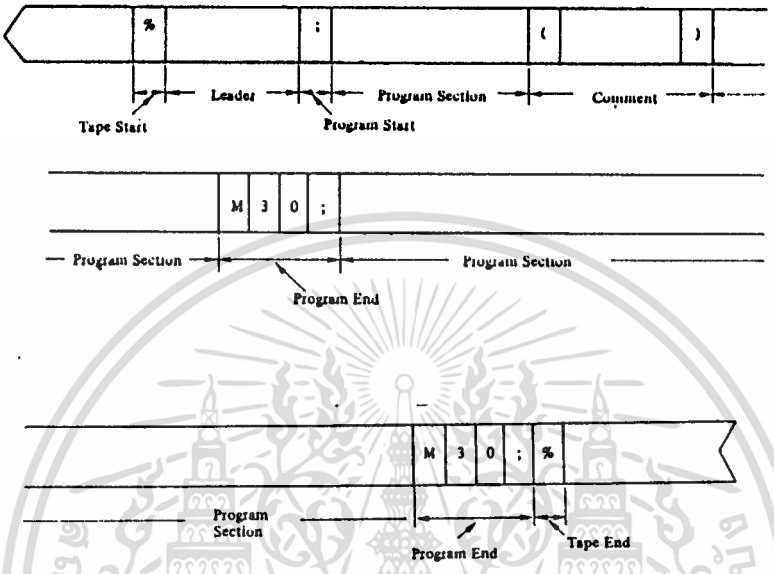
รูปที่ 3.8 รูปแสดงโครงสร้างโปรแกรมบนเส้นเทปที่มีการรวมโปรแกรมหลัก และโปรแกรมย่อยเข้าไว้ด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ชนิดของโปรแกรม

ลักษณะโปรแกรมบนเส้นเทปสามารถแบ่งออกได้เป็นสองลักษณะคือ

3.1.4.1 โปรแกรมหลัก (Main Program) บนเส้นเทป มีโครงสร้างดังรูปที่ 3.9

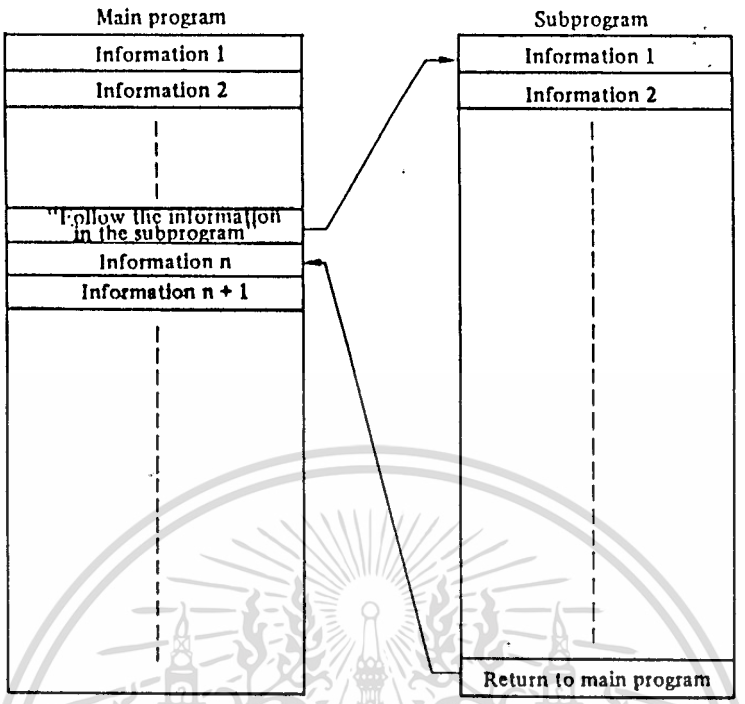


รูปที่ 3.9 โปรแกรมหลัก

จากรูป แสดงถึงโครงสร้างของโปรแกรมตามมาตรฐาน EIA และ ISO มาตรฐาน ISO จะใช้สัญญลักษณ์;และ%โดยที่สัญญลักษณ์%จะใช้คอน Tape Start เพื่อแสดงว่าเป็นจุดเริ่มต้นของเส้นเทป และสัญญลักษณ์ ; จะเป็นรหัสที่ใช้ขึ้นหน้าบล็อกคำสั่ง เพื่อแสดงถึงจุดเริ่มต้นของบล็อกโปรแกรมหรือส่วนที่ใช้เป็นบล็อกคำสั่ง ซึ่งอาจรวมถึง Comment ของโปรแกรมด้วย และเมื่อสิ้นสุดโปรแกรมคำสั่งทั้งหมดแล้วจะต้องตามด้วยสัญญลักษณ์ ; อีกครั้งหนึ่ง และเมื่อพบสัญญลักษณ์ % ซ้ำแสดงว่าเป็น Tape End สำหรับรหัส LF (LineFeed) จะใช้คั่นระหว่างบล็อกโปรแกรมแต่ละบล็อก ส่วนมาตรฐาน EIA จะใช้รหัส ER (End of Roll) สำหรับการเริ่มต้นและสิ้นสุดของเส้นเทป และใช้ ; สำหรับคั่นระหว่างบล็อกโปรแกรม

โดยทั่วไป ผู้ปฏิบัติงานจะใช้โปรแกรมหลักเป็นส่วนใหญ่ เนื่องจากเป็นส่วนที่สั่งให้เครื่องจักรทำงานตามขั้นตอนที่ต้องการ ส่วนฟังก์ชันการทำงานที่มีอยู่สามารถเรียกใช้จากโปรแกรมย่อย และเมื่อ CNC ทำงานตามโปรแกรมย่อยเรียบร้อยแล้ว จะกลับเข้าสู่การทำงานของโปรแกรมหลักโดยอัตโนมัติดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

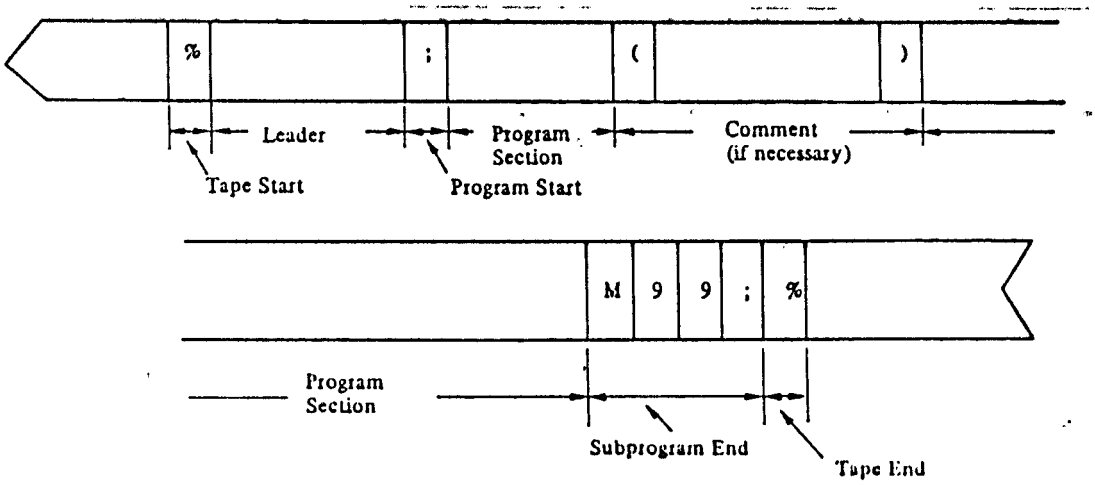


รูปที่ 3.10 แสดงการติดต่อระหว่างโปรแกรมหลักและโปรแกรมย่อย

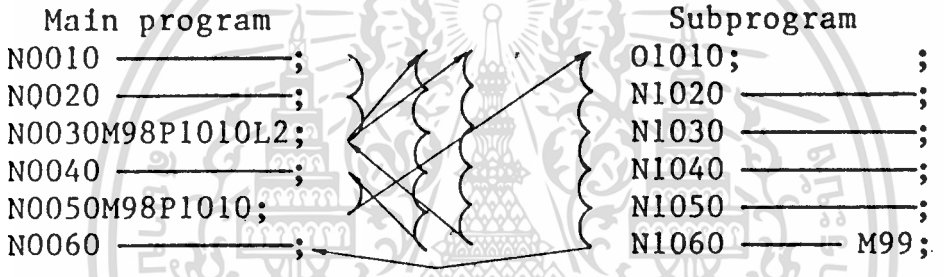
3.1.4.2 โปรแกรมย่อย (Subprogram) บนเส้นเทป มีโครงสร้างดังรูปที่ 3.11 ในกรณีที่ลักษณะของลำดับขั้นการทำงานของเครื่องจักรในการแมชชีนมีการกระทำซ้ำ ๆ กัน การที่ผู้ปฏิบัติงานจะสร้างโปรแกรมหลักที่มีลำดับการทำงานครบทุกขั้นตอนที่ซ้ำๆ กัน จะเป็นการยุ่งยากในการแก้ไขข้อมูลต่าง ๆ และเป็นการสิ้นเปลืองหน่วยความจำที่มีอยู่ของ CNC ดังนั้นลำดับการทำงานซ้ำ ๆ กันบ่อยครั้ง ซึ่งอาจมีลำดับการทำงานที่ยืดยาว จะถูกแยกเป็นโปรแกรมย่อยที่รวมอยู่ในโปรแกรมหลัก และสามารถเรียกใช้งานได้โดยโปรแกรมหลักเมื่อต้องการใช้ลำดับการทำงานนั้น ๆ ตัวอย่างของโปรแกรมย่อยเหล่านี้ได้แก่ การตัดเกลียวในงานกลึง ที่มีการตัดเกลียวหลายจุด ซึ่งโปรแกรมหลักสามารถเรียกใช้โปรแกรมย่อยได้ทันทีที่ต้องการ ดังรูปที่ 3.12

ในกรณีโปรแกรมย่อยบนเส้นเทป หรือมีการรวมโปรแกรมย่อยเข้ากับโปรแกรมหลักบนเส้นเทป โปรแกรมย่อยบนเส้นเทปเหล่านั้นจะใช้การได้หลังจากถูกใส่เข้าไปในหน่วยความจำสำหรับ Part Program ก่อนจึงจะสามารถเรียกนำขึ้นมาใช้ได้ ดังนั้นโปรแกรมย่อยจึงไม่สามารถใช้กับเครื่องควบคุมเชิงเลข (NC) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 โปรแกรมย่อย



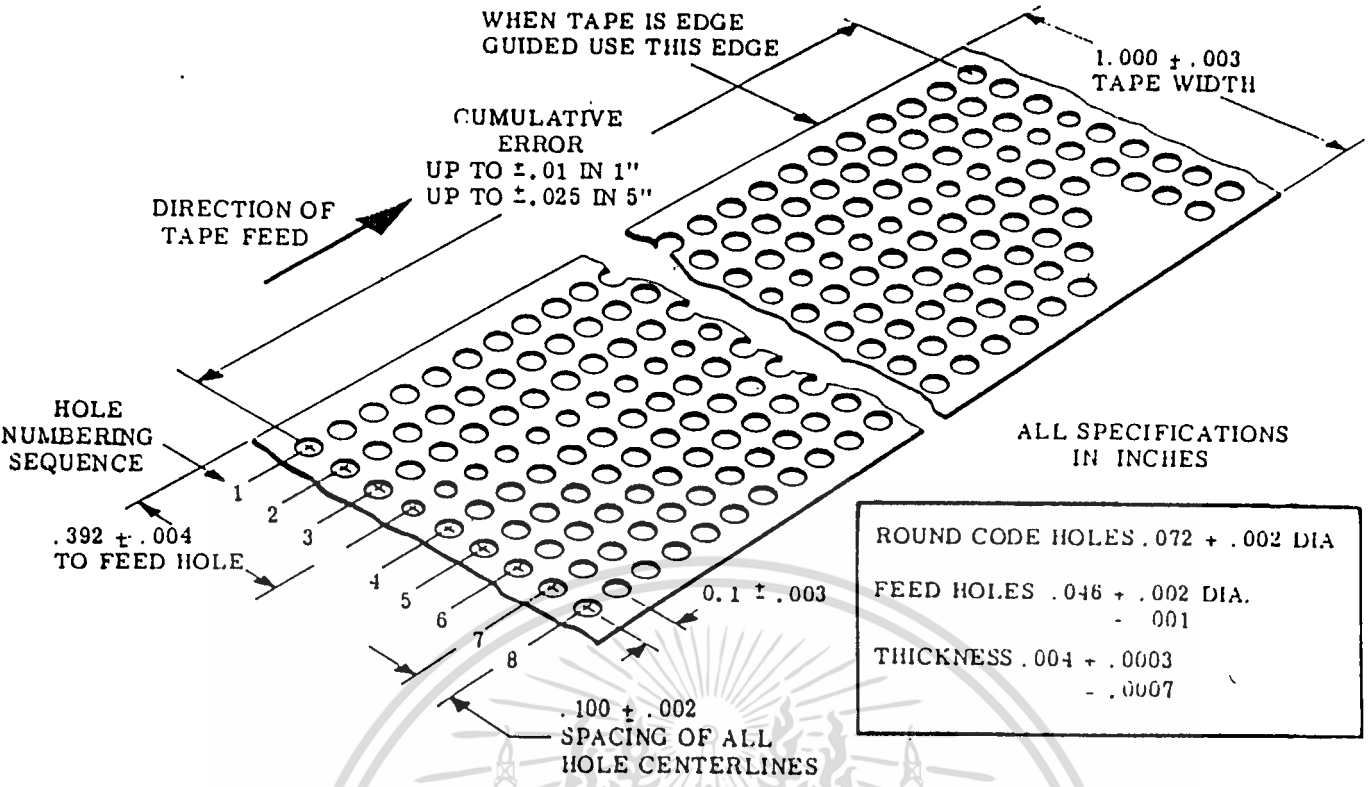
รูปที่ 3.12 ตัวอย่างข้อมูลการตัดเกลียวที่ใช้โปรแกรมหลักและโปรแกรมย่อย

ในกรณีที่ต้องการแก้ไขข้อมูลในโปรแกรมหลักหรือโปรแกรมย่อย ต้องกระทำโดยถ่ายข้อมูลจากเทปโปรแกรมไปสู่หน่วยความจำของ CNC จากนั้น CNC จะเริ่มอ่านโปรแกรมจากเส้นเทปเข้าสู่หน่วยความจำของ CNC การอ่านโปรแกรมจากเส้นเทปจะสิ้นสุดเมื่อพบรหัสสิ้นสุดโปรแกรม (%)

3.1.5 เส้นเทปและรหัสบนเส้นเทป

เส้นเทปและรหัสโปรแกรมบนเส้นเทปกระดาษที่กล่าวข้างต้น สามารถทำได้โดยการเจาะรูรหัสบนเส้นเทป รหัสบนเส้นเทปจะประกอบด้วย รหัสที่เป็นตัวเลข ตัวอักษร และสัญญลักษณ์พิเศษต่าง ๆ ซึ่งที่พบเห็นอยู่เสมอคือ RS 244 และ RS 358 รูปที่ 3.13 แสดงถึงลักษณะของเส้นเทป

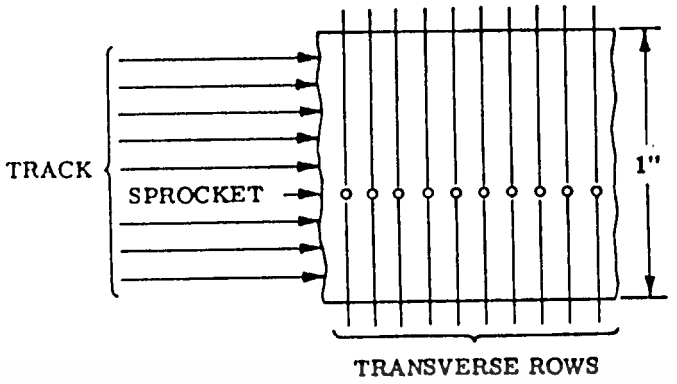
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ลักษณะของเส้นเทป

ลักษณะของเส้นเทปปกติจะมีขนาดมาตรฐานหน้าเทปวัดตามด้านหน้าประมาณ 25 มิลลิเมตร (มม.) ระยะรูเจาะรูต่อรู 2.5 มม. นิยมทำจากวัสดุประเภทกระดาษหรือไมลา (Mylar) ขึ้นอยู่กับลักษณะการใช้งานสำหรับงานที่ต้องมีการใช้เส้นเทปเดินโปรแกรมซ้ำไปซ้ำมาบ่อยครั้ง อาจใช้เทปที่ทำจากวัสดุไมลา สำหรับงานที่ใช้เทปเดินโปรแกรมเพียงชั่วครั้งชั่วคราว เช่นใช้ในการถ่ายโปรแกรมเข้าสู่หน่วยความจำของ CNC อาจใช้เพียงเทปกระดาษเพื่อลดค่าใช้จ่าย เส้นเทปจะแบ่งส่วนตามด้านหน้าได้ 9 ส่วน ซึ่ง 8 ส่วนใน 9 ส่วน ถูกออกแบบเปรียบได้เหมือนกับ แทร็ก (Track) ของเทปคาร์เส็ทส่วนที่ 9 ถูกออกแบบให้ใช้เป็นร่องสำหรับขับเคลื่อนให้เส้นเทปเลื่อนไป ขณะที่เครื่องอ่านเทปทำงานโดยหนามเตย ที่เรียกว่ารูหนามเตย (Sprocket Hold) ซึ่งจะเป็นแนวแบ่งแทร็กทั้ง 8 แทร็ก ออกไว้ 2 ด้าน ซีกหนึ่งจะมี 5 แทร็ก อีกซีกหนึ่งของแนวรูหนามเตยจะมี 3 แทร็ก ดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงแทร็กและรูหนามเตย

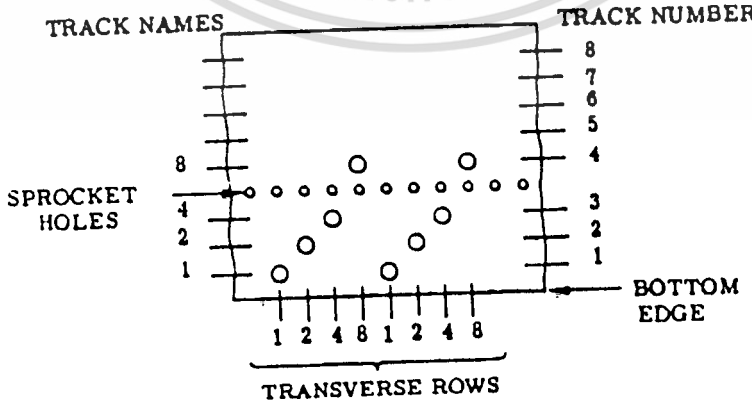
รหัสบนเส้นเทปแถวหนึ่ง (Row) จะแทนตัวอักษร (Character) ได้หนึ่งตัวอักษร ซึ่งมีความหมายดังนี้

1. ตัวอักษรใด ๆ 1 ตัวอักษร
2. ตัวเลขใด ๆ 1 หลัก (Digit)
3. เครื่องแสดงปริมาณคณิตศาสตร์ + หรือ -
4. สัญญลักษณ์พิเศษ

เมื่อนำตัวอักษรมารวมกันเป็นกลุ่ม กลุ่มของตัวอักษร เรียกว่า คำ (word)

3.1.5.1 รหัสเทปแบบ RS-244

ลักษณะของรหัสเทปแบบ RS-244 จะประกอบด้วยแทร็ก ที่ใช้เป็นตัวบอกแพรีตี้ (Parity) ซึ่งเป็นแบบ "Odd Parity" อยู่ที่แทร็ก 5 RS-244 จะใช้รหัสแบบ BCD (Binary Coded Decimal) เรียงลำดับหนึ่งถึงเก้า (1-9) มี Binary Weight 1 2 4 8 ที่แทร็ก 1 2 3 4 ตามลำดับ แสดงดังรูปที่ 3.15



รูปที่ 3.15 Binary Weight 1 2 4 8 ที่แทร็ก 1 2 3 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูในแทร็กที่สองจะแสดงเพียงหมายเลข 2 เท่านั้น

รูในแทร็กที่สามจะแสดงเพียงหมายเลข 4 เท่านั้น

รูในแทร็กที่สี่จะแสดงเพียงหมายเลข 8 เท่านั้น

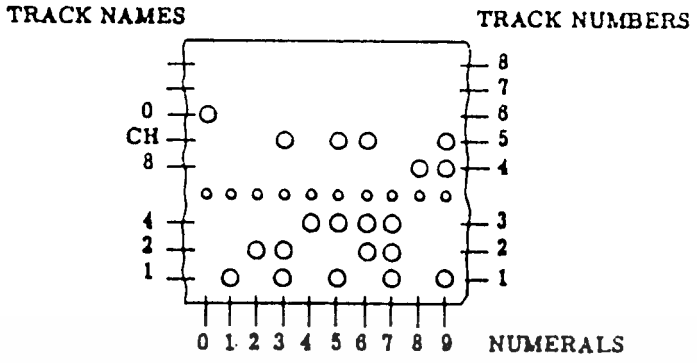
ตัวเลข 1 ถึง 9 จะได้จากการบวกกันของเลขในแต่ละแทร็ก สำหรับเลข 0 จะแสดงด้วยรหัสเจาะรูในแทร็กที่ 6 แทร็กเดียวเท่านั้น เพราะฉะนั้นรหัสตัวเลขที่ได้จากการรวมของตัวเลขของแทร็ก ต่าง ๆ แสดงได้ดังตารางที่ 3.1

Numeal	Binary Coded Docimal System	Track Punchod	Track Punchod	ParityTrack Punchod
1	1	1		No
2	2	2		No
3	1+2	2 and 1	Yes	
4	4	3		No
5	4+1	3 and 1	Yes	
6	4+2	3 and 2	Yes	
7	4+2+1	3,2 and 1	No	
8	8	4		No
9	8+1	4 and 1	Yes	
10	0	6		No

ตารางที่ 3.1 แสดงการเจาะรูรหัส

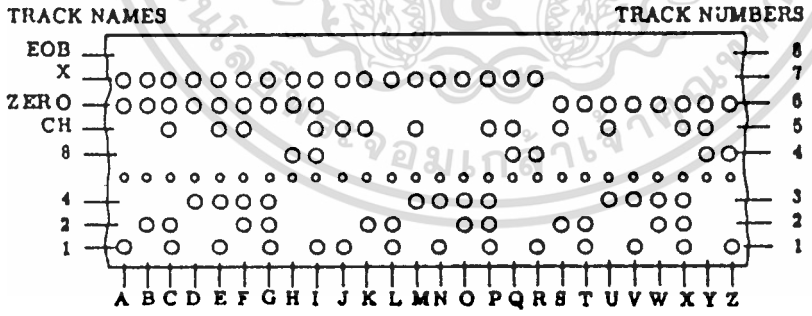
สามารถแสดงการเจาะรูเป็นลำดับตารางลงบนเส้นเทปได้ดังรูปที่ 3.16 ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงการเจาะรูเป็นตัวเลข 0-9 ลงบนเส้นเทป

แทร็กที่ 6 และ 7 ของเทปใช้เป็นคอมบิเนชัน (Combination) เพื่อใช้แสดงตัวอักษรตั้งแต่ A ถึง Z แสดงดังรูปที่ 3.17 สำหรับแทร็กที่ 8 จะเป็นแทร็กที่บอกการสิ้นสุดของบล๊อก (End of Block) หรือ Carriage Return



รูปที่ 3.17 รูปแสดงเส้นเทปเจาะรูเข้ารหัส A-Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

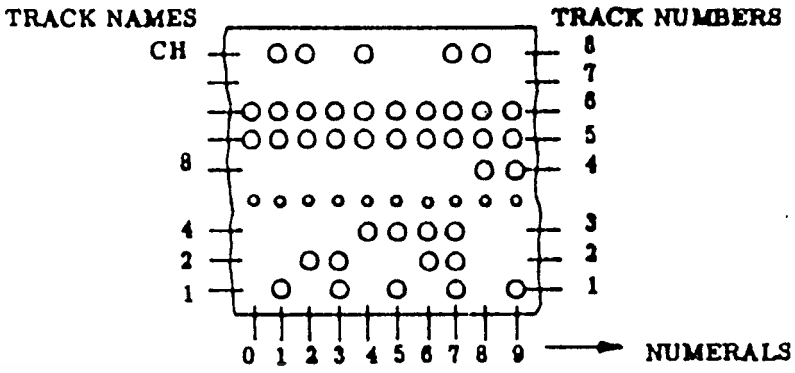
3.1.5.2 รหัสเทปแบบRS-358

รหัสเทปแบบ RS-358 จะใช้ Parity แบบคู่ (Even Parity) ซึ่งจะแสดงในแทร็กที่ 8 รหัสตัวเลข ที่ใช้ในระบบเทป RS-358จะเป็นระบบ BCD เช่นเดียวกับ RS-244 คือแสดงรหัสการเจาะรูแทร็กที่ 1 2 3 และ 4 แทน Binary Weight 1 2 4 และ 8 ตามลำดับ รหัสเทปแบบ RS-358 จะแตกต่างกับรหัสเทป RS-244 ที่แทร็ก 5 6 8 แทร็กที่ 5 และ 6 ใช้ในการสร้าง คอมบิเนชันสำหรับรหัสตัวเลข ตัวอักษร และสัญญลักษณ์พิเศษ แทร็กที่ 8 ใช้สำหรับแสดง Parity แบบคู่รหัสเทปแบบ RS-358 สามารถสร้างรหัสการเจาะรูของตัวเลข 0 ถึง 9 ได้ดังตารางที่ 3.2

Numeral	Binary Coded Decimal System	Track Punched	Parity Track Punched
0	0	5,6	No
1	1	1,5,6	Yes
2	2	2,5,6	Yes
3	2+1	1,2,5,6	No
4	4	3,5,6	Yes
5	4+1	1,3,5,6	No
6	4+2	2,3,5,6	No
7	4+2+1	1,2,3,5,6	Yes
8	8	4,5,6	Yes
9	8+1	1,4,5,6	No

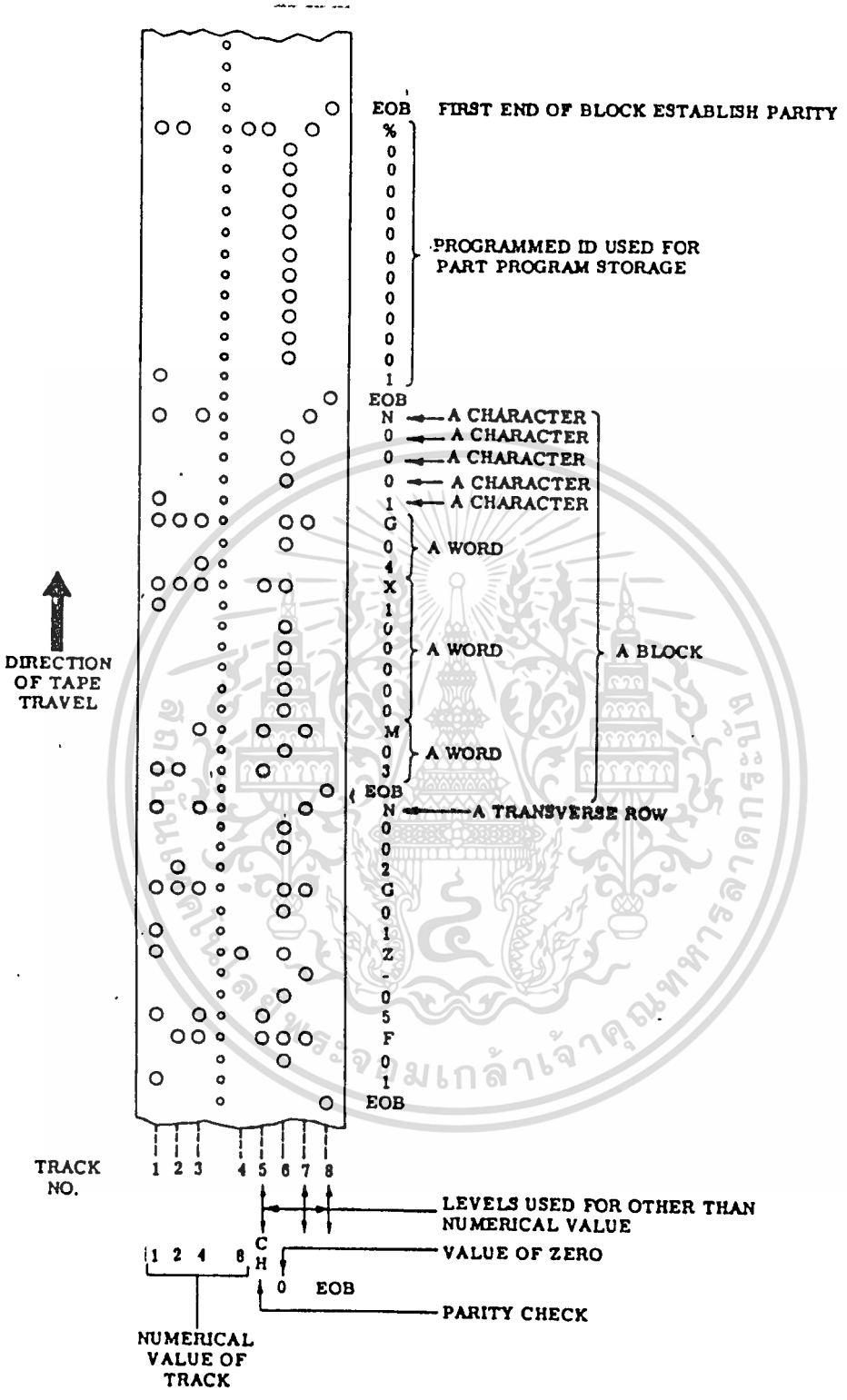
ตารางที่ 3.2 รหัสการเจาะรูของตัวเลข 0 ถึง 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



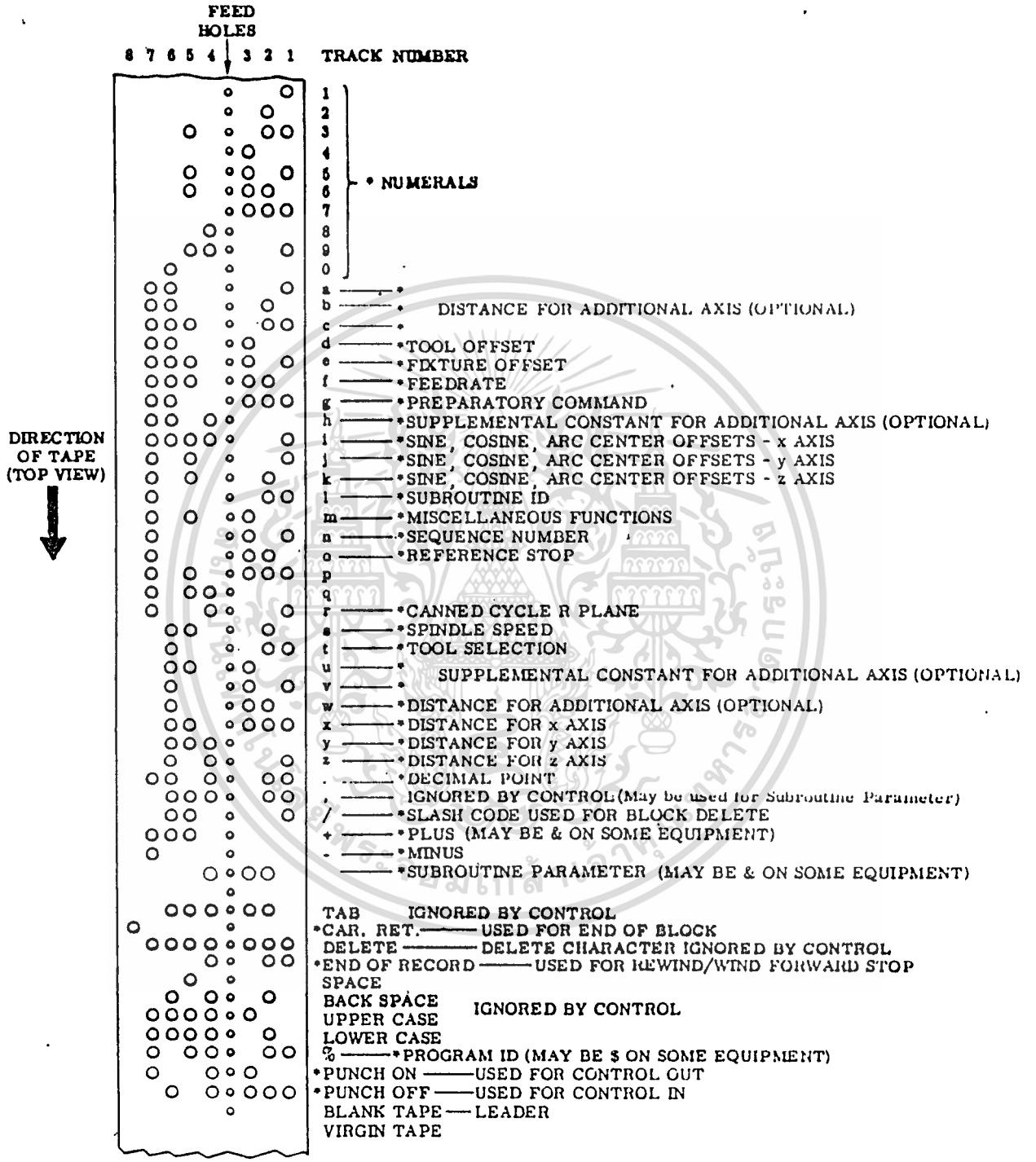
รูปที่ 3.18 รหัสการเจาะรูของตัวเลข 0 ถึง 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



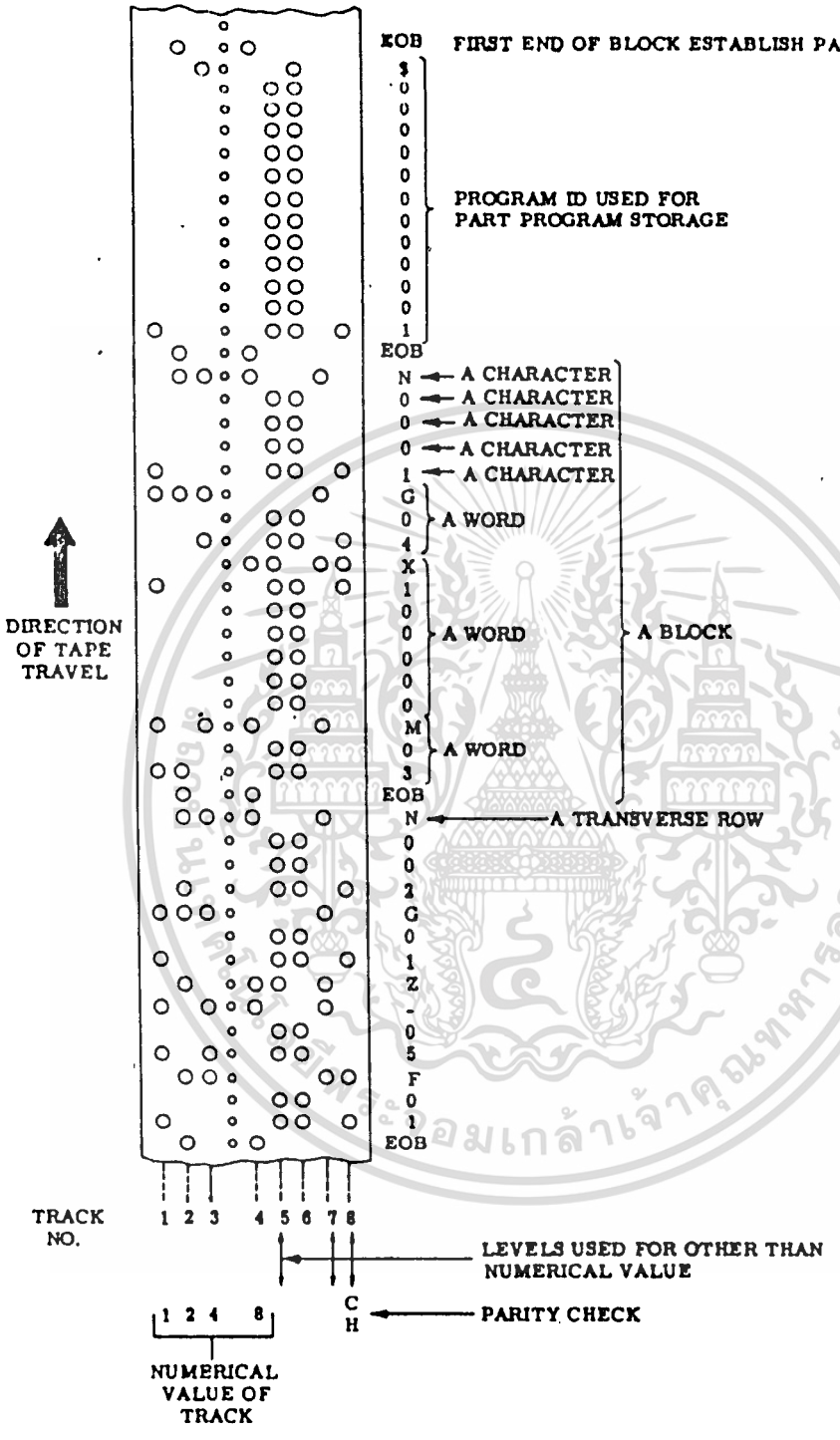
รูปที่ 3.19 แสดงการจัดบล็อกคำสั่งบนเส้นเทปแบบ RS-244

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



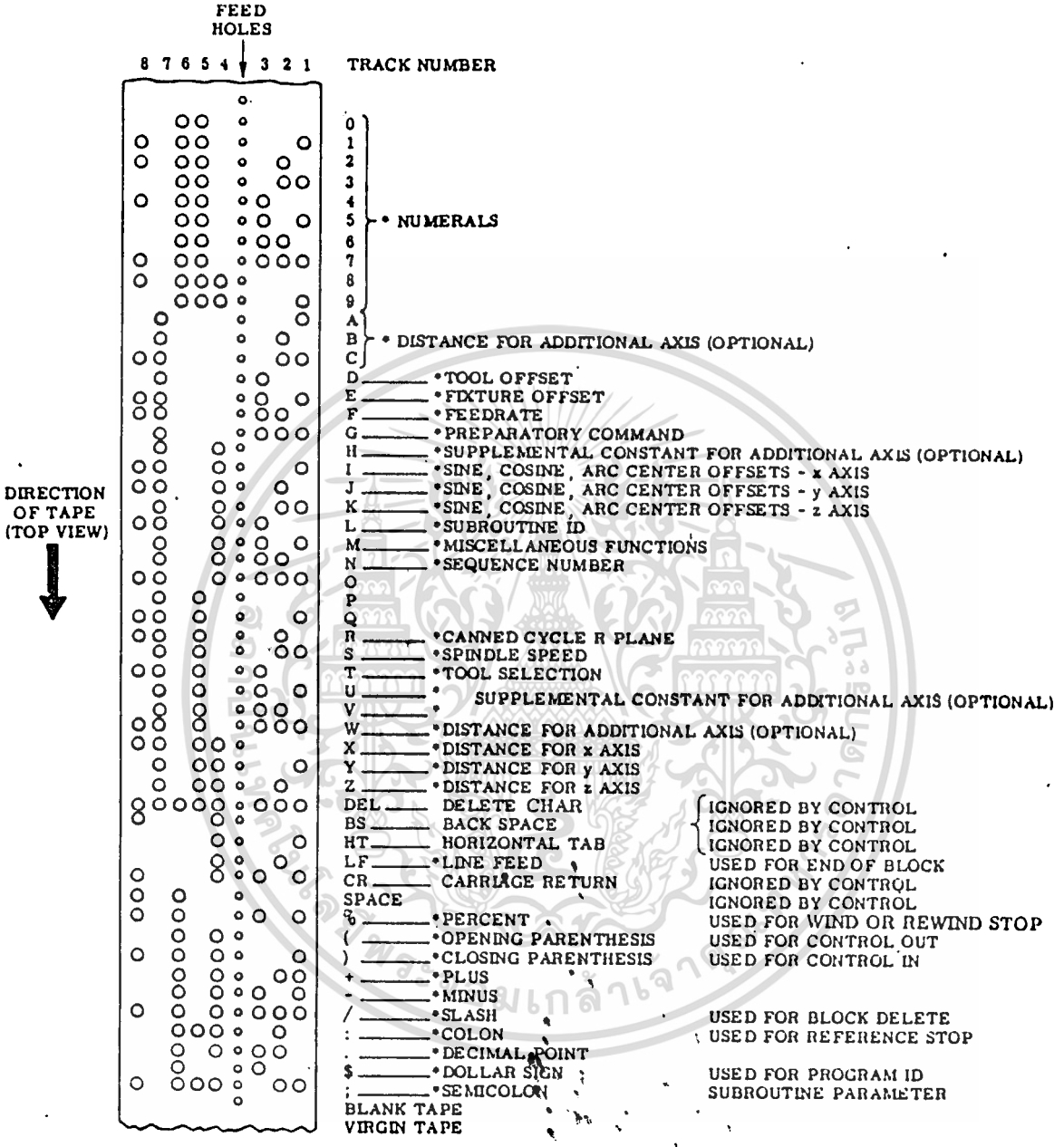
รูปที่ 3.20 แสดงตัวอักษรที่ใช้เป็นฟังก์ชันต่างๆ แบบ RS-244

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



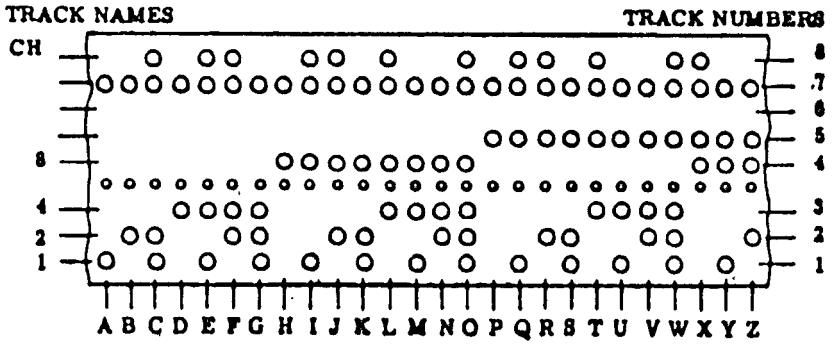
รูปที่ 3.21 แสดงการจัดบล็อกคำสั่งบนเส้นเทปแบบ RS-358

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แสดงตัวอักษรที่ใช้เป็นฟังก์ชันต่าง ๆ แบบ RS-358

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



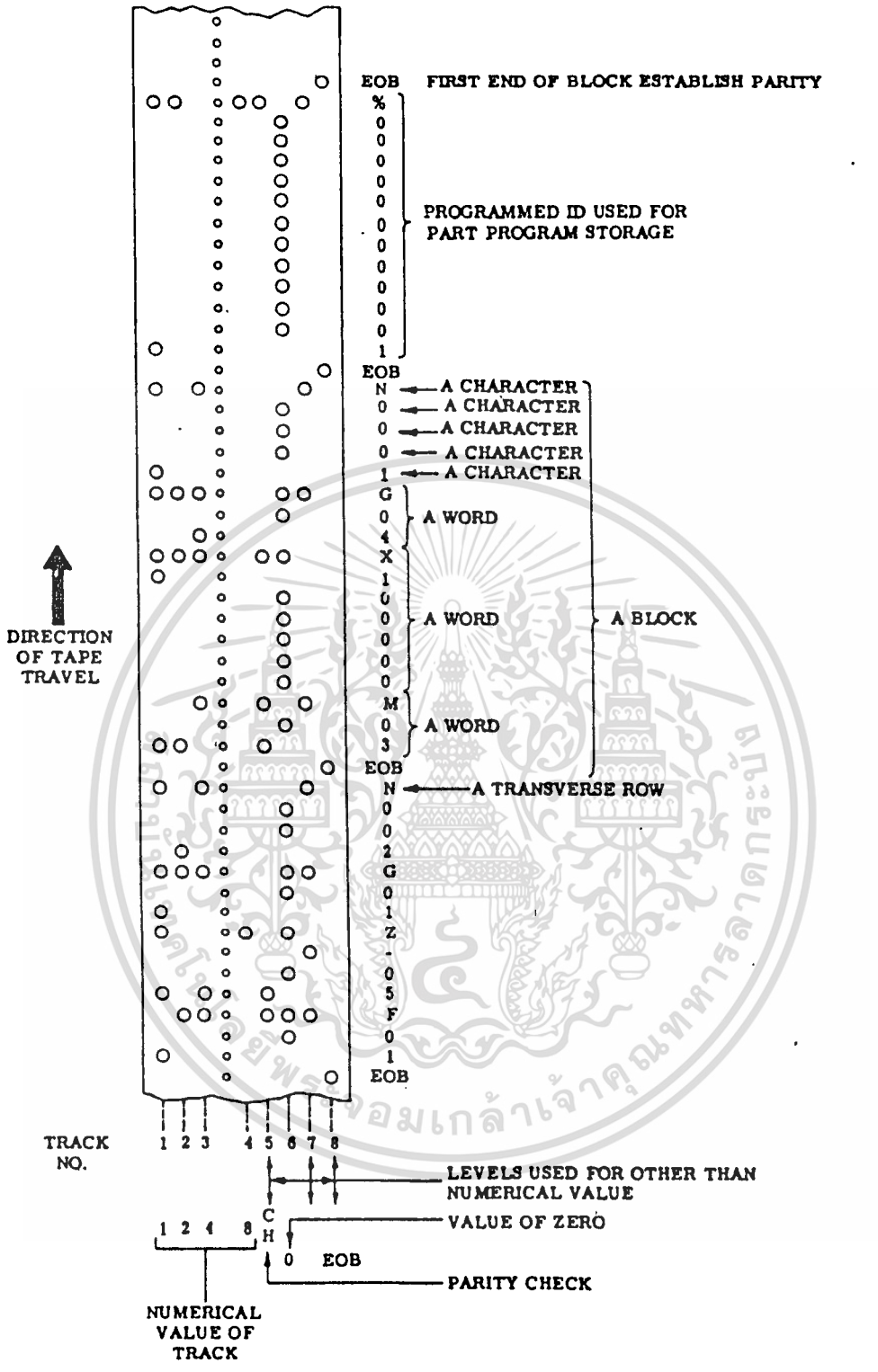
รูปที่ 3.23 แสดงการเจาะรูบนเส้นเทป

รหัสเทปแบบ RS-244 และ RS-358 เพื่อใช้แทนตัวอักษร ตัวเลข และสัญญลักษณ์พิเศษ แสดงดังรูปที่ 3.24

3.1.6 ค่าทำหนดเชิงวิศวกรรมสำหรับ CNC (Machine Setup Data)

Machine Setup Data (MSD) เป็นค่าที่ใช้กำหนดพารามิเตอร์ (Parameter) ต่าง ๆ สำหรับ CNC เพื่อกำหนดขอบเขต (Limit) การทำงานสำหรับ CNC ให้เป็นไปตามเงื่อนไขของเครื่องจักร และระบบเชื่อมต่อภายนอก Machine Setup Data จะถูกป้อนโดยผู้ผลิตเครื่องจักรเกี่ยวกับลักษณะการเคลื่อนที่ของแกน กำหนดความเร็วในการเคลื่อนที่ กำหนดค่าความละเอียดของตำแหน่งการเคลื่อนที่ค่าความผิดพลาดของอุปกรณ์ทางเครื่องจักรกล ตลอดจนค่าปรับเทียบ และโคออดิเนทของเครื่องจักรตัวอย่างของ Machine Setup Data แสดงในตารางที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 รหัสเทปแบบ RS-244 และ RS-358 เพื่อใช้แทนตัวอักษร ตัวเลข และสัญญลักษณ์พิเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1.
MACHINE SETUP DATA FOR 1050MC

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED)	SOFTWARE REVISION LEVEL
Control Setup (L11xx)	Letter Address Assignment	L1101	x _ _, y_ _	x - Selects the letter code. (See Table 2.) y - Selects the function. (See Table 3.)	See Table 3. NOTE: If it is required to change the letter address for the x, y, or z axes, see note 1, page 26.	A5
	Linear Feedback Gearing	L1102	m and axis letter(s) m_ and axis letter(s)	m = 0 - feedback units - INCH m ≠ 0 - feedback units - MM Axis letter(s) and feedback gearing ratio in inches (MM) per rev x 1000 NOTE: All linear axes must specify the same feedback ratio if they are to produce circular arcs.	All axes 0.1 inch/rev. (m0x100y100z100) If the above default is changed, see note 6, page 19. L1102 x100 y100 z100 m0	A5
	Rotary Feedback Gearing	L1103	m_ and axis letter(s)	1. Degree Rotary Axis letter(s) and feedback gearing ratio in degrees per rev x 1000 m = 0 for 360° rollover m ≠ 0 - Decimal degrees 2. Decimal Parts Rotary Axis letter(s) and zero.	No rotary axis	A5
	Format	L1104	x_ _, y_ _	x - Selects letter. (See Table 2.) y - No. of digits required. (1st digit - Metric) (2nd digit - Inch) To eliminate a selected letter for part program purposes, see note 2, page 26.	7 digits for all axes and integrands (Inch or Metric) 7 digits for r plane (Inch or Metric) 2 digits for spindle word 2 digits for tool word 5 digits for feedrate word 4 digits for sequence number word All other functions are assumed to have 2-digit words.	A5
	Display Content and Order Selection	L1105	x_ _, y_ _	x - Selects letter. (See Table 2.) y - Selects display position. (See Table 4.) To eliminate display position selected by 'y' word, see note, page 26.	See Table 4.	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED)	SOFTWARE REVISION LEVEL
Control Initialization (L12xx)	Miscellaneous Control Functions	L1201	x_, y_, m_	m = 0 - Disable function m ≠ 0 - Enable function x - 'word' y - 'bit' (See Table 5 for 'word' and 'bit'.)	Cutter Comp. arc speed change. M06 initiates tool change retract. MFO not active in traverse. L1201 x1 y4 m1 L1201 x1 y6 m1 L1201 x1 y9 m1	B5 A5
	Initial g codes-Power Up	L1202	g__	g17-xy/g18-xz/g19-yz g40-g42 - Cutter Comp. g55-59 - Canned Cycle Axis g70 - Inch/ 71 - Metric g90 - Absolute g91 - Incremental g93 - I/T; 94 - IPM; g95 - IPR	L1202 g17 L1202 g40 L1202 g57 (z axis) L1202 g70 L1202 g90 L1202 g94	A5
	Initial g codes Control Clear (Clear Pb, m02, m30)	L1203	g__	Any valid g code except g40-42 and g17-19.	L1203 g00 L1203 g80	A5
	g80 Cancels Contouring Mode	L1204	m_	m = 0 - NO m ≠ 0 - YES	No L1204 m0	A5
	Any g code Cancels Canned Cycles	L1205	m_	m = 0 - NO m ≠ 0 - YES	No L1205 m0	A5
	Number of Block Buffers	L1206	x___	Number of block buffers (3 - 127) See note 4, page 19.	8 L1206 x8	A5
	BCD Output Strobe Times	L1207	x__, y__	x - Strobe time in millisec y - Time preceding and following strobe in millisec. Refer to GEK-71607, Machine Interface.	50/70/50 millisecond L1207 x70 y50	A5
	Number of Fixture Offsets	L1208	x__	Number of offsets (1 - 96) See note 4, page 19.	16 L1208 x16	A5
	Message Translation	L1209	x_, y_,(...)	x - Message number y - Message table where 0 = Standard table 1 = Custom table (...) - Message Literal	Not applicable	B6
	Number of Tool Offsets	L1210	x__	Number of offsets (1 - 96) See note 4, page 19.	64 L1210 x64	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED)	SOFTWARE REVISION LEVEL
Control Initialization L12xx (Continued)	Number of Axes with Fixture Offsets	L1211	x_	Number of axes (1 - 3) Offsets are applied to axes No. 1 through 3 in sequence.	3 L1211 x3	A5
	EIA Contouring Arc Direction	L1212	x_, y_, z_	Non-EIA (x,y,z) = 0 EIA (x,y,z) ≠ 0 See note 7, page 19.	EIA - all planes L1212 x1 y1 z1 x - g17 plane y - g18 plane z - g19 plane	A5
	Omit Zeros Programming	L1213	m_	m = 0 - Omit leading zeros m ≠ 0 - Omit trailing zeros	Omit trailing zeros L1213 m1	A5
	Decimal Point Programming	L1214	m_	m = 0 - NO m ≠ 0 - YES Selecting decimal point overrides omit trailing zeros/ omit leading zeros on axis, feed, and reference plane words.	Yes L1214 m1	A5
	Tool Change Retract Axis(s)	L1215	Selected axis letter(s)	Axis letter(s) and a digit 0 through 3 See note 6, page 19.	z axis L1215 z0	A5
	Number of Digits in Offset Dimensions	L1216	y_ _	Number of digits for tool and fixture offsets. 1st digit - Metric 2nd digit - Inch	6 digits - Inch/Metric L1216 y66	A5
	Axes Required to be Referenced Before Cycle Start	L1217	Selected axis letter(s)_, m_	m = 0 Reference Zero not required. m ≠ 0 Reference Zero required. Axis letter and any non-zero digit. See note 4, page 19.	Reference zero required on all axes. L1217 x1 m1 L1217 y1 m1 L1217 z1 m1	A5
	Spindle Output Voltage Polarity (Used Only for D/A Spindle Option)	L1218	x_, y_	(+1 = + polarity; -1 = - polarity) x - Specifies cw polarity. y - Specifies ccw polarity.	Clockwise - plus Counterclockwise - minus L1218 x1 y-1	A5
	Spindle Creep Speeds (Used Only for D/A Spindle Option)	L1219	s_ _ _ 3 or 4 digits as specified by L1104	The first digit after the s selects the gear range. The succeeding digits select the creep speed for that gear range as a percent of maximum speed.	Creep speed is 5% of maximum for all ranges. L1219 s105 - 3 digit 's' L1219 s1050 - 4 digit 's'	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED	SOFTWARE REVISION LEVEL
Control Initialization L12xx (Continued)	Spindle Speed Override (SSO) Potentiometer maximum. USED ONLY FOR D/A SPINDLE OPTION	L1220	x_ _ _	x100 = Maximum DA voltage at 100% x120 = Maximum DA voltage at 120%	120% (BCD spindle is assumed.)	A5
	Custom Variables Single Word	L1221	x_ _, y_ _, or z_ _	x - Specifies a particular block of variables. y - Specifies a particular variable in the selected block. z - Specifies the data to be stored in the selected variable. NOTE: Refer to the description of custom features (if supplied) for detailed information on the use of custom variables.	No custom variables. These MSD codes are used for custom features only.	A5
	Custom Variables Double Word. Used for Custom Controls only. Instructions from factory when required.	L1222				
	Input Filter Complementing. Used to invert switch inputs from standard interface defined in 44C285863.	L1223	m_, x_, y_ _	m = 0 - Clear complementing bit. m ≠ 0 - Set Complementing bit. x - Word y - Bit (See reference sheets 042, 043, and 044 of 44C285863; also, see Table 7.)	See elementary Ref. Sh. 042 - Inputs See elementary Ref. Sh. 043 - Outputs NOTE: The complementing bit has to be set in order to invert the sense of the selected input or output.	A5
	Output Driver Complementing. Used to invert outputs from standard interface defined in 44C285863.	L1224				
	Number of Spindle Gear Ranges	L1225	g_	g specifies the number of required gear ranges (1-4).	USED ONLY FOR DIRECT RPM OPTION One L1225 g1	A5
	Spindle RPM per Gear Range (Normal)	L1226	g_, x_ _ _ _, y_ _ _ _ and z_ _ _ _. All 4 letters must be programmed in the same block.	g - Gear Range x - Minimum programmable RPM for this range. y - Maximum programmable RPM for this range. z - Spindle RPM when full DA voltage is output.	USED FOR DIRECT RPM OPTION ONLY This table is selected by default or through programming m41 in the part program tape. Maximum RPM is 9999 rpm.	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED	SOFTWARE REVISION LEVEL
Control Initialization L12xx (Continued)	Spindle RPM per Gear Range (Alternate)	L1227	g_, x_ _ _ _, y_ _ _ _, and z_ _ _ _ All 4 letters must be programmed in the same block.	(Same as L1226) Note: Since all L1226 data will also go into the L1227 table, the L1227 blocks must not precede L1226 blocks.	USED ONLY FOR DIRECT RPM OPTION This table is selected by m42 in the part program tape.	A5
	Punch Rate Selection for ASR port (Punch, TermiNet*)	L1228	x_	x0 - 10 CPS x1 - 15/30 CPS x2 - 60 CPS x3 - 120/240 CPS	10 CPS L1228 x0	B6
	Number of Part Programs and Subroutines	L1229	x_ _	x - Maximum total number of Part Programs and Subroutines required. See note 4, page 19.	x10 - Normal x20 - If Subroutine option is present L1229 x10 or L1229 x20	B6
	RS232 Communication Option N/C I. D. No.	L1230	(_ _ _ _ _ _ _ _) Eight Characters	Customer N/C Identification Number	GE 1050MC	B7
		L1231		Punch Press only, see GEK-25312 for details.		B6
	Spindle Growth Axis Selection	L1232	x_	x - Axis Number (0-5)	No spindle growth	B9
	Custom Machine Setup Data Code	L1233		Note: Refer to the description of custom features (if supplied) for detailed use of this Machine Setup Data Code.	No custom software. Used only for custom features	B9
	Jog Pushbutton and Direction Assignment	L1234	x_ _, y_ _	x - Pushbutton Select y - Function Select See page 29.	Used for 1050CCM Control only, see page 29.	B9
	Switchable Plane Contouring Axis Selection	L1235	x_	x1 - Axes 2, 3, and 4 x2 - Axes 1, 3, and 4 x3 - Axes 1, 2, and 4	Axes 1, 2, and 3	B9

*Trademark of General Electric Co., USA

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED	SOFTWARE REVISION LEVEL
Feed, Traverse and Jog Rates (L13xx)	Low Jog/ Incremental Feed Rate	L1301	f and any axis letter	The required rate for any axis may be selected by specifying an axis letter(s) and any non-zero digit together with f and the required rate in system feedback units. See notes 2 and 5, page 19.	10 ipm - All linear axes - .1 inch gearing L1301 f1000 x1 y1 z1 200 mmpm - All linear axes - 2 mm gearing L1301 f2000 x1 y1 z1	A5
	High Jog Rate	L1302			100 ipm - All linear axes - .1 inch gearing L1302 f10000 x1 y1 z1 2 mpm - All linear axes - 2 mm gearing L1302 f20000 x1 y1 z1	A5
	Traverse Rate	L1305		Note: x = Vector traverse rate of any or all axes, if vector traverse mode is selected by MSD code L1201.	300 ipm - All linear axes - .1 inch gearing L1305 f30000 x1 y1 z1 6 mpm - All linear axes - 2 mm gearing L1305 f60000 x1 y1 z1	A5
	Dry Run Rate (Rate applies to all axes.)	L1307	f	The rate is specified in system feedback units. (f24999 maximum for 0.1 inch and 1 mm gearing. f49999 maximum for 0.2 inch and 2 mm gearing.) See notes 2 and 5, page 19.	100 ipm - .1 inch gearing L1307 f10000 2 mpm - 2mm gearing L1307 f20000	A5
	Feed Word for Traverse Select	L1308	f	The feed word is specified in system feedback units. See notes 2 and 5, page 19. (Traverse rate obtained when f is greater than specified feed word.	f80000 or greater will force traverse rate. L1308 f80000	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED)	SOFTWARE REVISION LEVEL
Position Loop Data (L14xx)	In-Position Zone	L1401	Axis letter(s)	The distances are specified in system feedback units. See notes 2 and 5, page 19.	MAXIMUM VALUES .002 inch - all axes (.1 inch) 0.025 inch L1401 x20 y20 z20 .04 mm - all axes (2 mm) 0.5 mm L1401 x40 y40 z40	A5
	Error Limit	L1402			.4 inch - all axes (.1 inch) 3.0 inches L1402 x4000 y4000 z4000 8 mm - all axes (2 mm) 60 mm L1402 x8000 y8000 z8000	A5
	Out-of-Sync	L1403			.5 inch - all axes (.1 inch) 3.0 inches L1402 x5000 y5000 z5000 10 mm - all axes (2 mm) 60 mm L1402 x10000 y10000 z10000	A5
	In-Position Settling Time	L1408	x___	Settling time in milliseconds	100 milliseconds L1408 x100	A5
	Backlash Takeup (Unidirectional Approach)	L1409	Axis letter(s)	+ or - for final approach direction and amount of backlash in system feedback units. Functional in Traverse only. See notes 2 and 5, page 19.	-0.01 inch for x and y with 0.1 inch gearing L1409 x-100 y-100 -0.2 mm for x and y with 2 mm gearing L1409 x-200 y-200	A5
	Rotary Index Table	L1410	Axis letter(s)	The distance between index positions in degrees. See notes 2 and 5, page 19. See aux. MSD codes, page 27.	No indexing axes.	A5
Axis Reference Data (L15xx)	Reference Direction	L1501	Axis letter(s)	+ or - and any non-zero digit	Minus - All axes L1501 x-1 y-1 z-1	A5
	Reference Position	L1502	Axis letter(s)	+ or - and position value specified in the units (Inch/mm) selected for power-up.	Zero - All axes L1502 x0 y0 z0	A5
	Reference Rate	L1503	f and axis letter(s)	The required rate for any axis specified by axis letter(s) and any non-zero digit together with f and required rate in system feedback units. See notes 2 and 5, page 19.	100 ipm - All linear axes - .1 inch gearing L1503 f10000 x1 y1 z1 2 mpm - All linear axes - 2 mm gearing L1503 f20000 x1 y1 z1	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED	SOFTWARE REVISION LEVEL
Position Data (L16xx)	Negative End of Travel	L1601	Axis letter(s)	+ or - and position value in system feedback units. See notes 2 and 5, page 19.	None	A5
	Positive End of Travel	L1602			None	A5
	Primary Tool Change Retract and Emergency Retract points.	L1603	Axis letter(s)	+ or - and position value in system feedback units. See notes 2 and 5, page 19.	Zero Machine Interface for the 1050MC Control, refer to GEK-71607, for use of this function.	A5
	Secondary Tool Change Retract Point	L1604		*L1604 need not be programmed if not required, but if required, L1604 must be programmed after L1603.		
Leadscrew Error Compensation (L18xx)	Calibration Interval	L1801	Axis letter	Distance between correction points (must be the same for all axes).	None	A5
	Initialize Axis Calibration Table	L1802	Axis letter	Axis letter and zero. See note B, page 24.	None	A5
	Measuring Device Offset	L1803	Axis letter and 'K'	Axis letter - Measuring Device Offset. K = Position Preset for measuring device. See note D, page 24.	Zero	A5
	Calibration Value (Commanded Position vs. Actual Position)	L1804	Axis letter and 'K'	Axis letter = Calibration Position. K = Actual Position. See note D, page 24.	None	A5
	Calibration Value (Position Error)	L1805	Axis letter and 'K'	Axis letter = Calibration Position. K = Position Error. NOTE: Position Error must not be greater than ±0.0127 inch (0.254 mm).	None	A5
	End of Axis Calibration Table	L1806	None	L1806 is used to terminate the calibration table for each axis.	None	A5
	Cancel all Leadscrew Compensation Data	L1807	None	L1807 is used to cancel all previous leadscrew compensation data. See note C, page 24.	None	A5

(Continued)

TABLE 1. (CONT'D)

GROUP	ITEM	MSD CODE	WORDS TO BE SPECIFIED	DATA	DEFAULT CONDITION (USED BY CONTROL IF NOT SPECIFIED)	SOFTWARE REVISION LEVEL
PCI Data L2xxx) See GEK-71616 Software Section for a more complete description.	Start	L2000	None		L2000 is required.	A5
	Begin Input Buffer Word	L2001	None		Input buffer words are counted; 10 words total.	A5
	Begin Output Buffer Word	L2002	None		Output buffer words are counted; 8 words total.	A5
	Start PCI Solving	L2003	None		L2004 starts solving after loading.	A5
	Start of PCI Load	L2004	m-	m = 0 clears ladder	m0 - Clear ladder.	A5
	Message	L2005	x--	x - Message number 1 - 16 Message in parentheses (up to 32 characters)	No messages.	A5
	Strobe Sweeps	L2006	x-	x - Number of PCI solution sweeps	4 sweeps	A5
	Input Image	L21nn	xbb (hhhh)	nn - Input image word 00, 01, ... 09 bb - Bit number in image word (00 - 15) hhhh - Mask of bits in input buffer. (0000 - ffff)	bb can specify the first of several bits. For example, L2104 x00 (ffff) specifies all 16 bits in input image 4 (1F #3 U).	A5
	Output Image	L22nn	xbb (hhhh)	nn - Output image word bb - Bit number in image hhhh - Buffer bit mask		A5
	Software Relay	L23nn	xbb (hhhh)	nn - SR image word bb - Bit number hhhh - Buffer bit mask		A5

3.1.7 การวิเคราะห์ความผิดพลาด (Fault Analysis)

เป็นผลจากค่ากำหนดของ Machine Setup Data เมื่อมีข้อผิดพลาดจากค่ากำหนด หน่วยคอมพิวเตอร์กลางจะทำการตรวจสอบ โดยการแสดงหมวดของการผิดพลาดให้ผู้ปฏิบัติงานทราบถึงสาเหตุของการผิดพลาดเหล่านั้น ข้อผิดพลาดที่กล่าวถึงแบ่งออกเป็นสองสาเหตุคือ

1. ข้อผิดพลาดที่เกิดจากผู้ปฏิบัติงานได้แก่ ค่ากำหนดในโปรแกรมที่เกี่ยวกับการคำนวณ ตัวอย่าง เช่น ค่าที่กำหนดจุดเริ่มต้นและจุดสุดท้ายของส่วนโค้งไม่สัมพันธ์กับจุดศูนย์กลางความโค้ง เป็นต้น

2. ข้อผิดพลาดที่เกิดจากอุปกรณ์ภายนอกได้แก่ การคำนวณของคอมพิวเตอร์กลางผิดพลาด ซึ่งสาเหตุเกิดจากตำแหน่งของแกนเครื่องจักร ความเร็วในการเคลื่อนที่เร็วเกินไป หรือเกิดการผิดพลาดของระบบขับแกนเครื่องจักรเป็นต้น ข้อผิดพลาดในส่วนนี้คอมพิวเตอร์กลางจะรับข้อมูลแบบตรรกจากระบบเชื่อมต่อภายนอก

ข้อผิดพลาดที่เกิดจากสาเหตุดังที่กล่าวมาแล้วนี้จะมีความซับซ้อนมากหรือน้อยขึ้นอยู่กับระบบกลไกของเครื่องจักรกล่าวคือ ถ้ากลไกของเครื่องจักรมีระบบที่ซับซ้อน ระบบป้องกันหรือระบบไฟฟ้าควบคุมจะมีความยุ่งยากตามไปด้วย ดังนั้นในการหาข้อผิดพลาดที่เกิดขึ้นขณะที่เครื่องจักรทำงานในส่วนของความผิดพลาด ว่าจะเป็นส่วนที่จะหยุดอุปกรณ์ภายนอกขึ้นอยู่กับสาเหตุของข้อผิดพลาด ตัวอย่างเช่นข้อผิดพลาดที่เกิดจากการคำนวณคอมพิวเตอร์กลาง จะยุติการเคลื่อนที่ของแกนเครื่องจักรในขั้นตอนของโปรแกรมส่วนนั้น ถ้าข้อผิดพลาดเกิดจากแกนเครื่องจักรไม่สามารถควบคุมได้ คอมพิวเตอร์กลางจะตัดระบบไฟฟ้าหลักเพื่อหยุดการทำงานของเครื่องจักรนั้น พร้อมแสดงรหัสหรือเหตุของการผิดพลาดด้วยสัญญาณหรือข้อความบนหน้าปัทม์ของ CNCเป็นต้น

3.2 หน่วยอินพุทเอาท์พุทพื้นฐาน

เป็นหน่วยที่ใช้ในการเชื่อมต่อกับระบบภายนอก สามารถแบ่งออกได้เป็นสองส่วนคือ

3.2.1 หน่วยเชื่อมต่อทางข้อมูล ได้แก่หน่วยที่ใช้ในการส่งผ่านข้อมูลจากภายนอกเข้าสู่คอมพิวเตอร์กลางหรือจากส่วนของคอมพิวเตอร์กลางออกสู่ระบบภายนอกอันได้แก่ เครื่องอ่านเทปโปรแกรมคอมพิวเตอร์ส่วนบุคคลหรือ Work Station ที่ใช้ในการออกแบบรูปทรงของชิ้นงาน และส่งข้อมูลให้กับคอมพิวเตอร์กลางแบบ DNC (Direct Numerical Control) ซึ่งการเชื่อมต่อข้อมูลที่พบเห็นมักจะเป็นแบบอนุกรม เนื่องจากต้องการลดความยุ่งยากสำหรับผู้ใช้

3.2.2 หน่วยเชื่อมต่อระบบอินพุทเอาท์พุทกับไฟฟ้าควบคุม ระบบเชื่อมต่ออินพุทจะรับแรงดันไฟฟ้าจากระบบไฟฟ้าควบคุมภายนอกด้วยระดับแรงดันไฟฟ้า 24โวลท์ แบบตรรก ระบบเชื่อมต่อเอาท์พุทจะเชื่อมต่อกับส่วนอินพุทของระบบไฟฟ้าควบคุม ระบบเชื่อมต่อแบบเอาท์พุทโดยปกติจะเป็นหน้าสัมผัสซึ่งสามารถรับแรงดันได้สูง หรือเป็นทรานซิสเตอร์ เพื่อจะนำเอาท์พุทต่อกับรีเลย์ ในกรณีที่ต้องการเชื่อมต่อกับระบบไฟฟ้าควบคุมที่มีแรงดันสูง ลักษณะระบบเชื่อมต่อเอาท์พุทจะให้ระดับสัญญาณแบบตรรก ตัวอย่างการใช้งานของหน่วยอินพุทเอาท์พุทพื้นฐานที่ใช้ในเครื่องควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ต้องการเชื่อมต่อกับระบบไฟฟ้าควบคุมที่มีแรงดันสูง ลักษณะระบบเชื่อมต่อเอาท์พุทจะให้ระดับสัญญาณแบบดรรก ตัวอย่างการใช้งานของหน่วยอินพุทเอาท์พุทพื้นฐานที่ใช้ในเครื่องควบคุมเชิงเลขแสดงได้ดังตารางที่ 3.4

(Table 1. PCI Inputs for 1050MC Standard Using Suggested MSD Tape - Continued from page 12)

PCI INPUT NUMBER	1050 SIGNAL	COMMENTS
65I	OMA21 Least significant	
66I	OMA22	
67I	OMA24	
68I	OMA28	
69I	OMA11 M code BCD	
70I	OMA12	
71I	OMA14	
72I	OMA18 Most significant	
73I	OTA21 Least significant	
74I	OTA22	
75I	OTA24	
76I	OTA28	
77I	OTA11 T code BCD	
78I	OTA12	
79I	OTA14	
80I	OTA18 Most significant	
81I	OSA21 Least significant	
82I	OSA22	
83I	OSA24	
84I	OSA28	
85I	OSA11 S code BCD	
86I	OSA12	
87I	OSA14	
88I	OSA18 Most significant	
89I	OMSTRO M strobe	
90I	OSSTRO S strobe	
91I	OTSTRO T strobe	
92I	OMSTOP M stop pulse	Set for 150 ms after m00, m02, or m30 stop is initiated(after motion in same block is complete).
93I	OSPDLR Spindle reverse or stop	Set when z point is reached for g84 or g88 cycle. Cleared when z feedback to r is complete.
94I	OG84 G84 active	Set when g84 canned cycle is active.
95I	O83FED G83 feed	Set when z axis is feeding to z for g83 cycle. May be used to run customer timer which initiates g83 retract.
96I (1050MC)	OGOTOZ Go to z	Set when z axis is feeding from r to z for g80 series canned cycles.
96I (1050P)	OPTCLM Part clamps output	Set by Control when part clamps should be released

ตารางที่ 3.4 หน่วยอินพุทเอาท์พุทพื้นฐานที่ใช้ในเครื่องควบคุมเชิงเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Table 1. PCI Inputs for 1050MC Standard Using Suggested MSD Tape - Continued from page 13)

PCI INPUT NUMBER		1050 SIGNAL		COMMENTS
97I 98I 99I 100I	1050MC	IXRLS IYRLS IZRLS 14RLS	XREF LS YREF LS ZREF LS 4REF LS	Set while on a Reference Zero limit switch.
100I	1050P	IHDDWN		Head cycle set while Head Off Top in progress.
101I	1050MC	15RLS	5REF LS	
101I	1050P	IHDCPT		Head cycle clear while Head On Top complete.
102I		OPCIS1		PCI spare input 1 from executive.
103I		IRUN	RUN	Reset state indicates E-Stop or limit switch open.
104I		OLTCS	CYCLE START	Set while Cycle Start light is on.
105I		OLTFH	FEED HOLD	Set while Feed Hold light is on.
106I		O82DWL	G82 DWELL	Set when z point is reached in g82 or g89 cycle. Starts dwell.
107I 108I		OTAPE OMANUAL	TAPE MANUAL	Set when in Auto mode or Single mode. Set when in Manual mode.
109I	1050MC	O86MIL	G86 MILL	Sets on first to z cycle in g86. Clears on g00 or g80.
109I	1050P	OCYCHD	CYCLE HEAD	Sets when Head Cycle output is active.
110I	1050MC	O87ATZ	G86 at z	Set when z point is reached in g87 WEB drilling cycle.
110I	1050P	OAUTO	AUTO MODE selected	Set when in Auto mode.
111I	1050MC	OZ FEED	Z FEEDING	Set when z is feeding up or down in canned cycles.
111I	1050P	ORPCLM	REPOSITION CLAMPS	Set to apply Reposition clamps.
112I	1050MC	OCONTR	CONTOUR ACTIVE	Set when Contouring mode (g01, g02, g03) is active.
113I	1050MC	01CLMP	CLAMP X	Set state signals that axis should be clamped.
113I	1050P	OALLIP	ALL IN POSITION	Set when all axes are in position.
114I 115I 116I 117I 118I		02CLMP 03CLMP 04CLMP 05CLMP OPCIS2	CLAMP Y CLAMP Z CLAMP 4th CLAMP 5th	PCI spare input 2 from executive.
119I 120I 121I		OCUST1 OCUST2 OCUST3		PCI inputs available as custom features. (Available as output drivers as in standard controls).
122I		OCLRC	CLEAR	Set for at least 100MS when clear pushbutton or power up.
123I 124I 125I 126I		STSTM SEMRA STCRA SPU	TEST EMERG RETRACT TOOL RETRACT POWER UP	Set when in Test mode. Set during emergency retract. Set during tool change retract. Changes to set state at end of power up.
127I		SBO	BYPASS OVERRIDE	Set when MFO and spindle speeds are forced to 100%
128I		JUMPER		

ตารางที่ 3.4 หน่วยอินพุทเอาท์พุทพื้นฐานที่ใช้ในเครื่องควบคุมเชิงเลข (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Table 2. PCI Outputs for 1050MC Standard Using Suggested MSD Tape - Continued from page 17)

PCI OUTPUT NUMBER		1050 SIGNAL	COMMENTS
65PO			
66PO 67PO 68PO 69PO		OUTPUT 1-08 OUTPUT 1-09 OUTPUT 1-10 OUTPUT 1-14	Wired spares. Do not use.
70PO 71PO 72PO 73PO 74PO	1050MC		
74PO	1050P		Reserved for internal use. Do not use.
75PO 76PO 77PO		I CUST 1 I CUST 2 I CUST 3	PCI inputs for custom controls.
78PO 79PO 80PO		SOPT 1 SOPT 2 SOPT 3	Spindle creep. Spindle clockwise. Spindle counterclockwise. } For D/A spindle.
81PO 82PO 83PO 84PO 85PO 86PO	1050MC	SINHB1 MOTION INH x SINHB2 MOTION INH y SINHB3 MOTION INH z SINHB4 MOTION INH h SINHB5 MOTION INH 5 SINHB6 MOTION INH 6	Set this bit to inhibit the axis.
81PO 82PO 83PO 84PO 85PO 86PO	1050P		
87PO		SITRF TRANSFER INH	Set this bit to inhibit transfer (also an IF).
88PO		SFHL FEED HOLD LATCH	Sets feed hold; must be cleared to release feed hold. (Also an IF.)
89PO	1050MC	IDWCPT DWELL COMPLETE	Clear this bit to indicate dwell is complete in g82, g89.
89PO	1050P	ICYCTL HEAD CYCLE INITIATE	Set bit to initiate head cycle in the Single mode.
90PO	1050MC	ISREVD SPINDLE REVERSED SPINDLE STOPPED	Set this bit to indicate spindle is reversed in g84 or reversed in g88.
90PO	1050P	IHCYCI INHIBIT HEAD CYCLE	Set bit to inhibit the Head cycle.
91PO		SG83RI G83 RETRACT	Clear this bit to retract z to r in g83.
92PO		ITCRI1 TOOL CHG RET 1	Set this bit to initiate a tool change retract.
93PO		IPCIS1	PCI spare output 1 to executive.
94PO 95PO 96PO		IACYS AUX CYCLE START SPECS PCI E-STOP IPCIS2	Set this bit to initiate cycle start. Set this bit to initiate emergency stop. PCI spare output 2 to executive.

ตารางที่ 3.4 หน่วยอินพุทเอาท์พุทพื้นฐานที่ใช้ในเครื่องควบคุมเชิงเลข (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Table 2. PCI Outputs for 1050MC Standard Using Suggested MSD Tape - Continued from page 18)

PCI OUTPUT NUMBER	1050 SIGNAL	COMMENTS
97PO		
98PO	18	ODI32*1 ELEMENT NUMBER
99PO 1050MC		
99PO 1050P		NOT WIRED
100PO 101PO 102PO 103PO 104PO 105PO 106PO 107PO	20 21 22 23 24 25 26 27	ODI32 ELEMENT NUMBER
108PO 109PO 110PO		ODI32*1 28-30 are used for customer software. These are wired to 18TB and cannot be set from the PCI.
111PO 112PO	31 32	ODI32*1 ELEMENT NUMBER
113PO 114PO 115PO 116PO 117PO 118PO 119PO 120PO 121PO 122PO 123PO 124PO 125PO 126PO 127PO 128PO	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	Message 1 Set 113PO to display and queue PCI message 1, 114PO for message 2, etc. Message texts are stored by MSD. The message stays queued until the corresponding PO is cleared by the PCI. Message 16

3.3 หน่วยคอนโซล

คือหน่วยที่ใช้ในการเชื่อมต่อระหว่างหน่วยคอมพิวเตอร์กลางและผู้ปฏิบัติงาน เพื่อจะรับคำสั่งบล็อกโปรแกรมจากภายนอกเข้าสู่หน่วยความจำของหน่วยคอมพิวเตอร์กลาง หรือการอ่านค่าสภาวะภายในตลอดจนคำสั่งต่าง ๆ ที่ถูกป้อนเข้าไปเพื่อทำการตรวจสอบแก้ไขทางจอภาพ หรือหลอดไฟแสดงรหัสแบบตัวเลขและตัวอักษร หน่วยคอนโซลสามารถแบ่งหมวดย่อยได้ดังนี้

3.3.1 หมวดแสดง เป็นหมวดที่ใช้ในการแสดงค่าต่าง ๆ ภายในส่วนของหน่วยความจำสำหรับโปรแกรมและสภาวะการทำงานของ CNC ในการควบคุมการทำงานของเครื่องจักร ตัวอย่างเช่น ตำแหน่งของแกนเครื่องจักร ความเร็วของ spindle เป็นต้น

3.3.2 หมวดแก้ไขเป็นหมวดที่งานร่วมกับหมวดแสดง จะแตกต่างจากหมวดแสดงในข้อที่ว่าหมวดแก้ไขส่วนใหญ่จะต้องทำในภาวะที่ CNC ยังไม่มีการปฏิบัติคำสั่งของโปรแกรม หมวดแก้ไขจะใช้ในการแก้ไขข้อมูลทางโปรแกรมหรือเปลี่ยนแปลงค่าพารามิเตอร์ของ CNC

3.3.3 หมวด Manual เป็นหมวดที่ใช้ปฏิบัติการควบคุมแกนเครื่องจักรด้วยผู้ปฏิบัติการ

3.3.4 หมวด Auto เป็นหมวดที่ให้ CNC ปฏิบัติคำสั่งตามบล็อกโปรแกรม ซึ่งอาจจะกระทำกับหน่วยความจำหรือเทปอย่างใดอย่างหนึ่ง

3.3.5 หมวด Manual Data Input (MDI) เป็นหมวดกึ่งอัตโนมัติคือ ปฏิบัติคำสั่งครั้งละ 1 บล็อก โดยการป้อนผ่านหน่วยคอนโซลโดยผู้ปฏิบัติงาน ส่วนใหญ่หมวด MDI จะใช้ในการปรับแต่งชิ้นงานหรือเป็นงานที่เรียกว่า"Hand Make" มากกว่า

3.4 หน่วยควบคุมตำแหน่งและซิงโครไนซ

เป็นหน่วยที่รับผิดชอบในการควบคุมตำแหน่งของแกนเครื่องจักร ให้อยู่ในตำแหน่งที่ถูกต้องตามการอ้างอิงตำแหน่งของหน่วยคอมพิวเตอร์กลาง ซึ่งลักษณะการอ้างตำแหน่งของคอมพิวเตอร์กลางแบ่งออกได้เป็น

3.4.1 การอ้างตำแหน่งในหมวดที่ไม่เกี่ยวข้องกับการคำนวณ ซึ่งหมายถึงสภาวะของเครื่องจักร ที่มีแกนหยุดนิ่งในหมวดการทำงานของหน่วยคอนโซลแบบ Manual ในกรณีนี้ถ้าแกนของเครื่องจักรถูกแรงกระทำในทิศทางด้านการเคลื่อนที่ในแนวแกนเครื่อง หน่วยควบคุมตำแหน่งจะทำการแก้ไขค่าความผิดพลาดของตำแหน่งที่เกิดขึ้น

3.4.2 การอ้างตำแหน่งในหมวดที่มีการคำนวณประกอบอยู่ด้วย การอ้างตำแหน่งในลักษณะนี้จะเป็นการอ้างตำแหน่งในภาวะการเคลื่อนที่ ดังนั้นหมวดการทำงานของการอ้างตำแหน่งจากคอมพิวเตอร์กลางจะอยู่ในหมวดการทำงานแบบ MDI หรือ Auto ของหน่วยคอนโซล การเคลื่อนที่ของแกนเครื่องจักรอาจมีการเคลื่อนที่ไปพร้อม ๆ กัน โดยมีจุดเริ่มต้นและจุดปลายของการเคลื่อนที่สิ้นสุดที่เวลาเดียวกันการเคลื่อนที่ลักษณะนี้เรียกว่า Axis Simultaneous ซึ่งถ้าการเคลื่อนที่พร้อมกัน 2 แกนเรียกว่า 2 Axis Simultaneous ในขณะที่มีการเคลื่อนที่เกิดขึ้นพร้อมกันตั้งแต่สองแกนนั้น เมื่อแกนเครื่องจักรแกนใดแกนหนึ่งเกิดค่าผิดพลาด เครื่องจักรที่เหลือก็จะแก้ค่าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกนนั้น เมื่อแกนเครื่องจักรแกนใดแกนหนึ่งเกิดค่าผิดพลาด เครื่องจักรที่เหลือก็จะแก้ค่าของตำแหน่งอย่างสอดคล้อง ลักษณะการเคลื่อนที่ไปในลักษณะนี้เรียกว่า ซิงโครไนซ

3.4.3 การอ้างตำแหน่งของตำแหน่งทั้งสองที่กล่าวมาแล้วนั้น หน่วยควบคุมตำแหน่งจะส่งสัญญาณแรงดันกระแสตรงขนาด ±10 โวลท์ ให้กับหน่วยขับเคลื่อนแกนเครื่องจักร ซึ่งเป็นแรงดันมาตรฐาน เมื่อตำแหน่งของการเคลื่อนที่มีค่าผิดพลาดเกิดขึ้น หน่วยควบคุมตำแหน่งจะส่งระดับแรงดันชดเชยเพื่อให้ตำแหน่งของแกนในภาวะหยุดนิ่งหรือภาวะเคลื่อนที่ให้อยู่ในตำแหน่งที่อ้างอิงโดยคอมพิวเตอร์กลางอย่างถูกต้อง เนื่องจากการเคลื่อนที่ของตำแหน่งจะดำเนินไปอย่างต่อเนื่อง ในขณะที่ CNC ควบคุมแกนของเครื่องจักรให้ทำงานตามลำดับขั้นที่ทำโปรแกรมโดยผู้ปฏิบัติงาน ความเร็วในการเคลื่อนที่ของจุดของโคออดิเนทของเครื่องจักรจะต้องมีค่าคงที่เสมอ ซึ่งความเร็วในการเคลื่อนที่ของจุดโคออดิเนทจะถูกต้อง โดยหน่วยคอมพิวเตอร์กลางซึ่งจะแปรความหมายจากบล็อกโปรแกรม และปรับค่าความเร็วให้กับหน่วยควบคุมตำแหน่งและซิงโครไนซ

โครงสร้างของส่วนควบคุมตำแหน่ง ลักษณะของหน่วยควบคุมตำแหน่งที่กล่าวมานี้เป็นระบบควบคุมแบบลูปปิด (Closed-loop) สำหรับหน่วยควบคุมตำแหน่งของเครื่องจักรขนาดเล็กมักนิยมใช้หน่วยควบคุมตำแหน่งแบบลูปเปิด (Open-loop) ซึ่งจะมีราคาที่ประหยัดกว่า

อุปกรณ์ตรวจสอบค่าผิดพลาด หน่วยควบคุมตำแหน่งแบบลูปปิดจะต้องมีอุปกรณ์ในการวัดระยะการเคลื่อนที่ การวัดระยะสามารถทำได้สองลักษณะคือ

(1) การวัดระยะโดยการนับจำนวนของเกลียวนำ หรือการนับจำนวนรวมของเพลามอเตอร์ขับแกนเครื่องจักร ระยะที่จำนวนรอบที่ได้ จะถูกนำมาคำนวณเป็นระยะทางที่แกนของเครื่องจักรเคลื่อนที่ไป โดยระยะของการเคลื่อนที่จะมีค่าเป็น

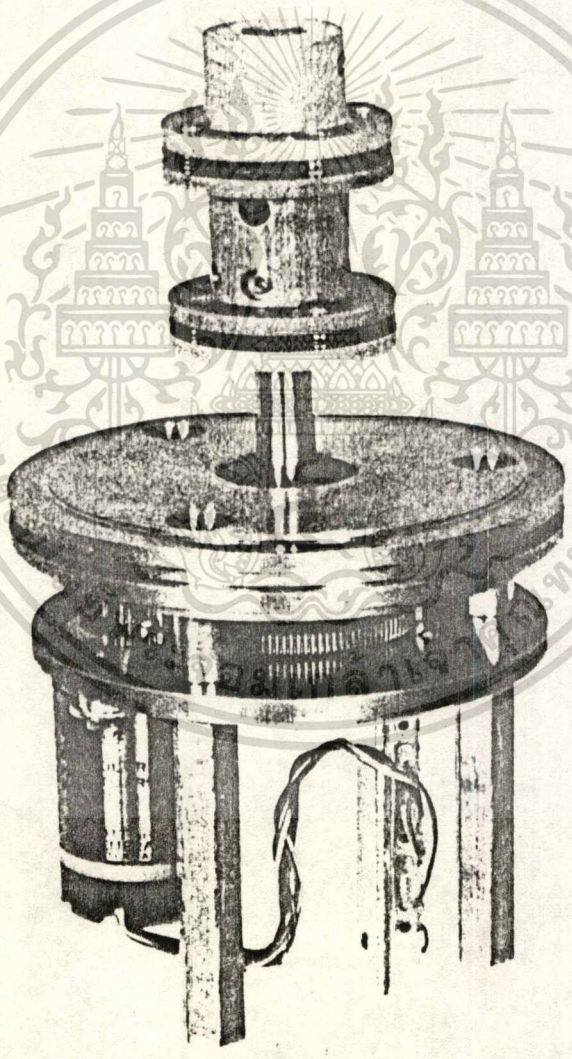
$$D = P \cdot n$$

โดยที่ D = ระยะการเคลื่อนที่
P = Pitch ของเกลียว
n = จำนวนรอบ

การวัดระยะทางแบบนี้เป็นการวัดแบบทางอ้อม

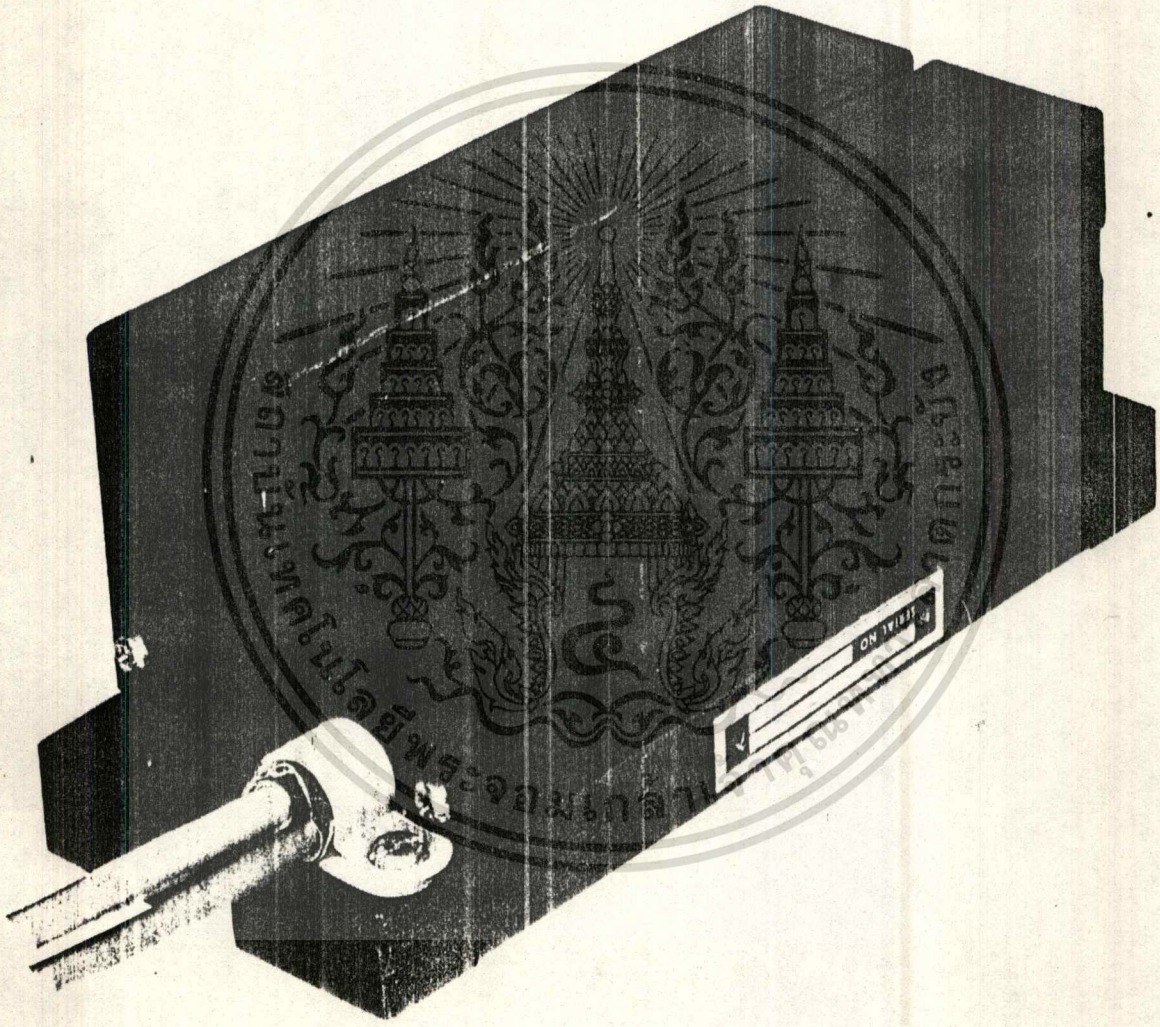
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) การวัดระยะการเคลื่อนที่แบบวัดระยะการเคลื่อนโดยสเกลบรรทัด (Linear Scale) ซึ่งอุปกรณ์การวัดระยะทางแบบนี้จะถูกติดตั้งตามแนวแกนเครื่องจักร ซึ่งประกอบด้วยหัวอ่านเรียกว่า Slider และตัวบรรทัดยาวเท่ากับระยะการเคลื่อนที่ของแกนเครื่องจักร การวัดโดยวิธีนี้จะมีความเที่ยงตรงกว่าวิธีการวัดระยะโดยการนับรอบ เนื่องจากการวัดระยะการเคลื่อนที่โดยวิธีนี้เป็นการวัดโดยวิธีตรง ค่าผิดพลาดสูงสุดของการวัดขึ้นอยู่กับความเที่ยงตรงของสเกลบรรทัด แต่ความผิดพลาดของการวัดระยะทางแบบนับรอบ จะขึ้นอยู่กับส่วนนับ รอบ คือ Encoder และ Pitch ของเกลียวนำของแกนเครื่องจักร อุปกรณ์การวัดแบบต่าง ๆ ที่ใช้ในเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์แสดงดังรูปที่ 3.25 รูปที่ 3.26 และรูปที่ 3.27 ตามลำดับ



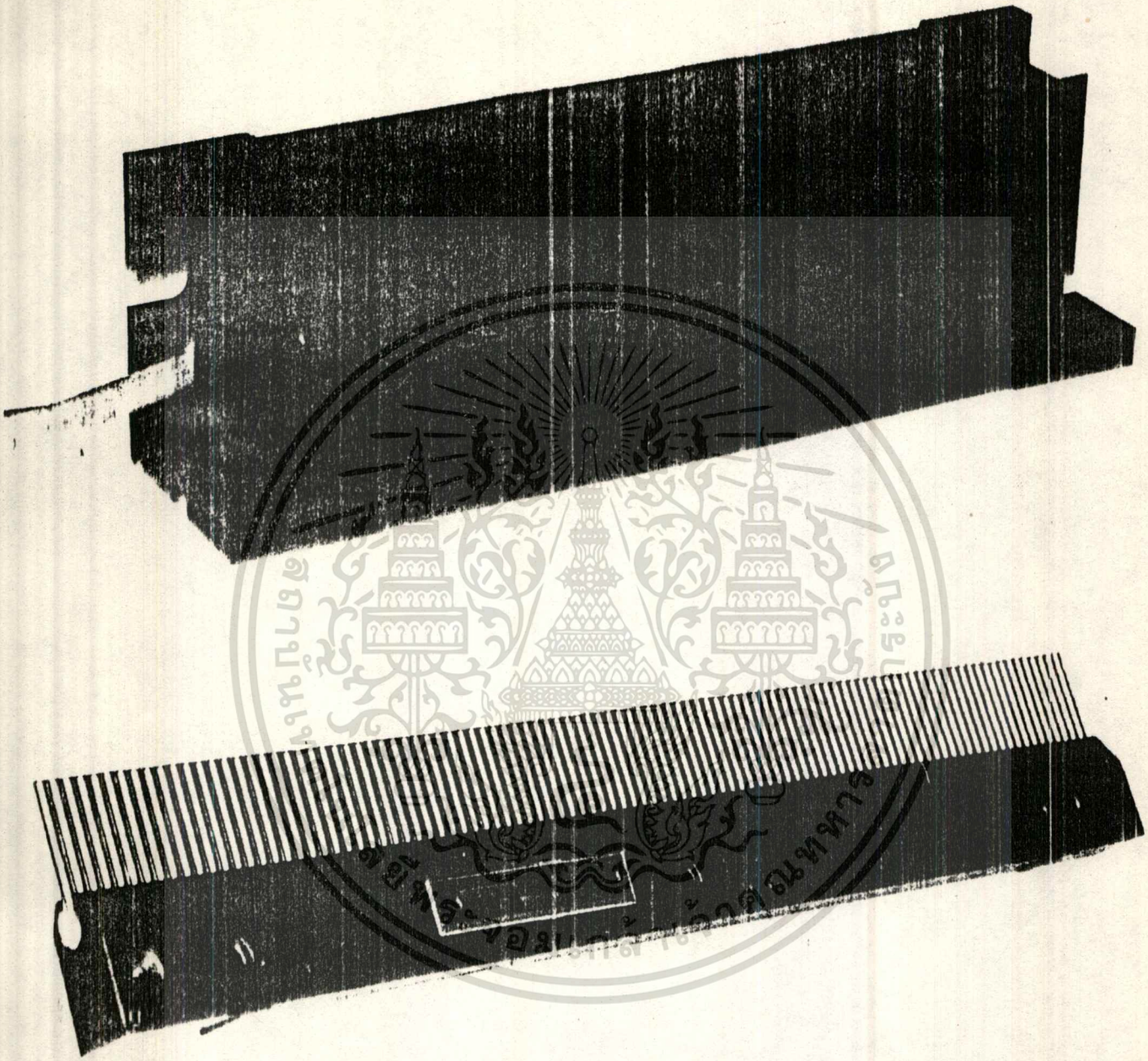
รูปที่ 3.25 Resolver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 Inductiion Linear Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 อุปกรณ์อ่านและวัดแบบ Linear Scale

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

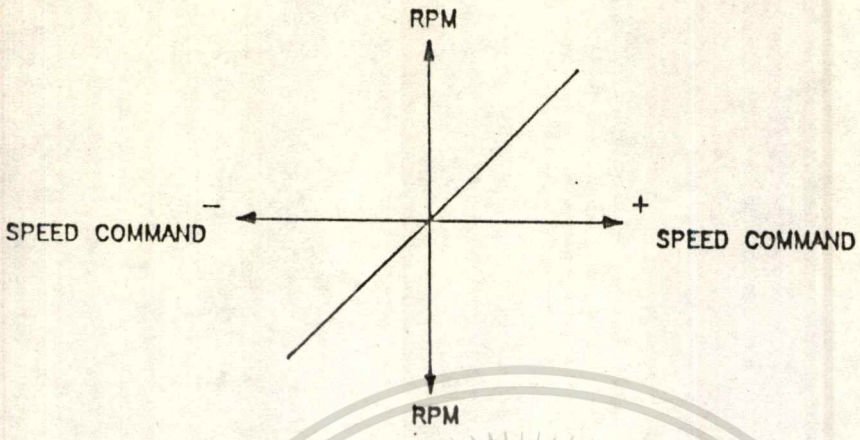
3.5 หน่วยขับกำลัง

คือหน่วยที่จ่ายกำลังงานไฟฟ้าให้กับตัวกระทำได้แก่ อุปกรณ์กลจักรไฟฟ้ามอเตอร์หรือวาล์วน้ำมันแบบเชิงเส้นสำหรับกระบอกสูบหรือมอเตอร์แบบไฮดรอลิค ซึ่งจะกล่าวในลำดับต่อไป หน่วยจ่ายกำลังที่นิยมใช้ส่วนใหญ่จะเป็นหน่วยจ่ายกำลังไฟฟ้า ซึ่งจ่ายกำลังให้กับเครื่องจักรไฟฟ้าชนิดมอเตอร์ ที่สามารถแยกตามลักษณะโครงสร้างของมอเตอร์ได้ดังนี้

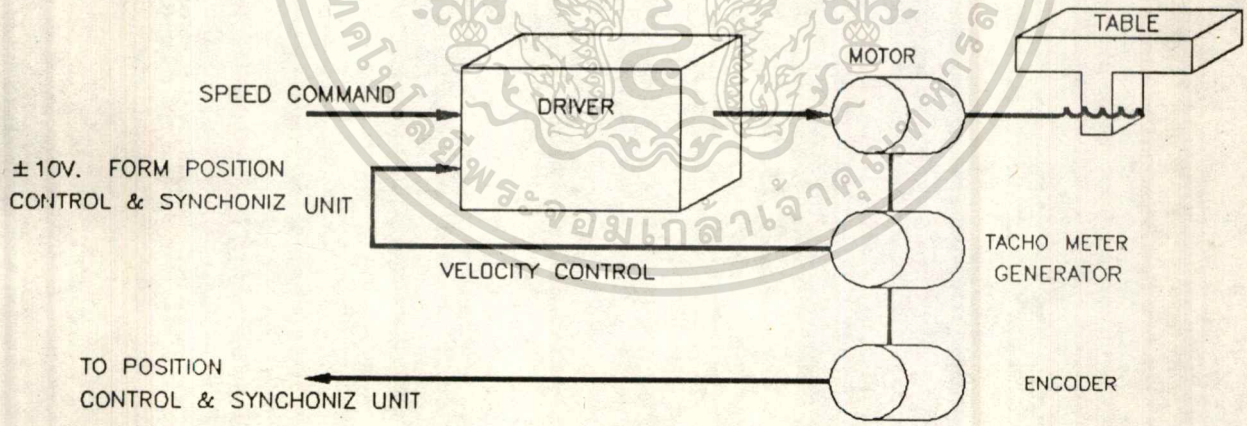
3.5.1 มอเตอร์แบบกระแสตรง

นิยมใช้อย่างแพร่หลายในเครื่องจักรยุคแรกถึงยุคปัจจุบัน เนื่องจากมอเตอร์ชนิดนี้สามารถกำเนิดแรงบิดได้สูง โดยที่โครงสร้างของมอเตอร์ไม่มีความยุ่งยาก หน่วยขับกำลังสำหรับมอเตอร์กระแสตรงเป็นระบบควบคุมความเร็วรอบแบบลูปปิด มีหมวดการควบคุมแบบพีไอดี (PID) โดยหน่วยขับกำลังไฟฟ้าสำหรับมอเตอร์กระแสตรงจะรับสัญญาณจากอุปกรณ์ตรวจจับความเร็วการหมุนของเพลามอเตอร์ที่เรียกว่า ทาโคมิเตอร์ (Tachometer) โดยอุปกรณ์ตรวจจับความเร็วจะเปลี่ยนความเร็วในการหมุนที่เพลาของมอเตอร์แปลงเป็นสัญญาณไฟฟ้าที่เป็นเชิงเส้นกับความเร็วนำมาเปรียบเทียบแบบเป็นสัดส่วนกับระดับสัญญาณควบคุมความเร็ว ความสามารถของหน่วยขับกำลังสำหรับมอเตอร์กระแสตรงบอกขนาดความสามารถในการจ่ายกำลังมีหน่วยเป็นวัตต์ (Watt) ระบบการจ่ายกำลังไฟฟ้าให้กับมอเตอร์ที่พบเห็นส่วนใหญ่จะเป็นระบบ PWM ซึ่งเชื่อว่าสามารถจ่ายกำลังให้กับมอเตอร์แบบกระแสตรงได้อย่างเป็นเชิงเส้น และลดความสูญเสียในการถ่ายทอดพลังงานจากแหล่งกำเนิดเข้าสู่มอเตอร์ต่ำกว่าวิธีอื่น ระดับแรงดันของสัญญาณควบคุมความเร็วจะมีแรงดันอยู่ในระดับ $\pm$ 10 โวลท์ เป็นเชิงเส้นกับความเร็วการหมุนของมอเตอร์ความเร็วของมอเตอร์ที่ใช้ทั่วไปสำหรับเครื่องจักรที่ใช้ CNC ในการควบคุมแกนเครื่องจักรจะมีความเร็วรอบของมอเตอร์ขับแกนสูงสุดอยู่ประมาณ 1,500-2,000 รอบต่อนาที (RPM) สำหรับเครื่องหมายซึ่งจะบอกถึงทิศทางการหมุนโดยเครื่องหมาย + (แรงดันที่มีศักดาเป็นบวก) แสดงถึงทิศทางการหมุนของเพลามอเตอร์ในทิศทางตามเข็มนาฬิกา สำหรับเครื่องหมาย- (แรงดันที่มีศักดาเป็นลบ) ก็จะหมายถึงทิศทางการหมุนของเพลามอเตอร์ในทิศทางตรงข้ามกับเครื่องหมายบวกดังรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 ความสัมพันธ์ระหว่างสัญญาณควบคุมความเร็วกับความเร็วรอบ



รูปที่ 3.29 ระบบขับเคลื่อนกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

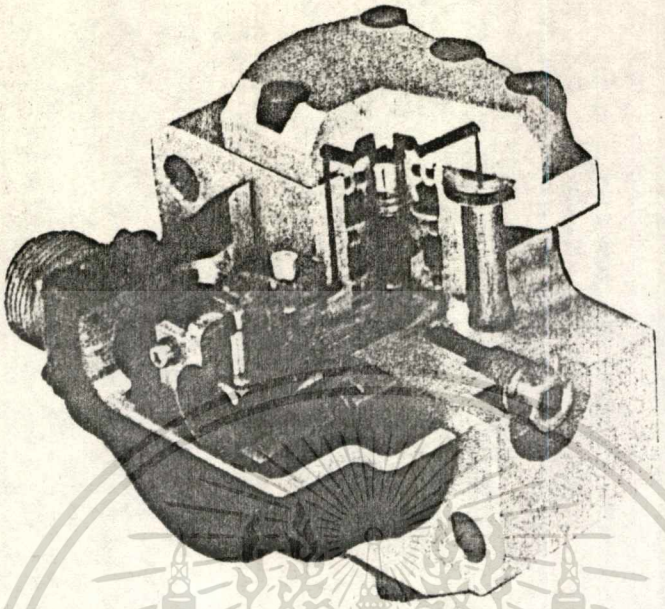
3.5.2 หน่วยขับกำลังแบบไฮดรอลิค

ในเครื่องจักรที่มีขนาดใหญ่ตัวอย่างเช่นเครื่องขนาดใหญ่มีน้ำหนักของป้อมมีด (Tool Post) มาก จำเป็นจะต้องใช้อุปกรณ์ขับเคลื่อนที่มีกำลังมากในการขับเกลียวนำ เพื่อเอาชนะแรงต้านที่เกิดจากความผิด และแรงต้านจากแรงกระทำของคมตัดกับชิ้นงาน ซึ่งถ้าใช้มอเตอร์ไฟฟ้าเป็นอุปกรณ์ต้นกำลัง จะต้องมีขนาดใหญ่ทำให้ค่าใช้จ่ายสูง ผู้ผลิตเครื่องจักรขนาดใหญ่จึงหันมาใช้อุปกรณ์ประเภทไฮดรอลิคเป็นตัวจ่ายกำลังในหน่วยขับกำลัง ทำให้ขนาดของชิ้นส่วนของระบบขับกำลังสำหรับแกนเครื่องจักรมีขนาดเล็กลง หน่วยขับกำลังแบบไฮดรอลิคนี้ประกอบด้วยส่วนสำคัญในการควบคุมความเร็วของการเคลื่อนที่ในรูปของการหมุนหรือช่วงชัก (ความเร็วตามแนวช่วงชัก) คือ 3.5.2.1 วาล์วควบคุม (Control Valve) เป็นวาล์วในระบบไฮดรอลิค แบ่งออกได้เป็นสองประเภทใหญ่ ๆ คือ

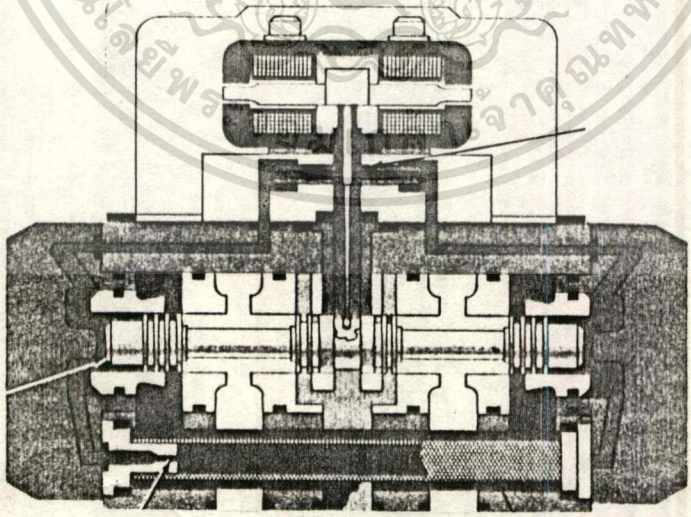
(1) วาล์วแบบตรรก ลักษณะการทำงานของวาล์วแบบนี้ จะมีสภาวะการทำงานสองสภาวะคือ สภาวะปิดกั้นการไหลของน้ำมัน สภาวะการปิดกั้นการไหลของน้ำมันนี้จะไม่ยอมให้น้ำมันมีสภาวะปิดกั้นการไหลของน้ำมัน สภาวะการปิดกั้นการไหลของน้ำมันนี้จะไม่ยอมให้น้ำมันมีการไหลผ่านวาล์วได้เลย และสภาวะที่ยอมให้ของเหลวประเภทน้ำมันไหลผ่านได้เต็มที่ตามขนาดของวาล์ว วาล์วประเภทนี้จะใช้ควบคุมการไหลของน้ำมันเพื่อจ่ายให้กับอุปกรณ์ไฮดรอลิคแบบง่าย ๆ ไม่ต้องการความเที่ยงตรงในการเคลื่อนที่ของอุปกรณ์ไฮดรอลิคมากนัก

(2) วาล์วควบคุมการไหลของน้ำมันแบบเป็นเชิงเส้น วาล์วประเภทนี้เป็นวาล์วเที่สามารถควบคุมการไหลของน้ำมันได้อย่างเป็นสัดส่วนกับแรงดันไฟฟ้า หรือแรงทางกล ขึ้นอยู่กับชนิดของวาล์ว และมีชื่อเรียกอีกอย่างหนึ่งว่าเซอร์โววาล์ว (Servo Valve) ซึ่งหมายถึงวาล์วที่สามารถควบคุมการไหลเชิงปริมาตรโดยการสั่งการอย่างไม่ผิดพลาด โครงสร้างของวาล์วแบบนี้แสดงได้ดังรูปที่ 3.30 และ รูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



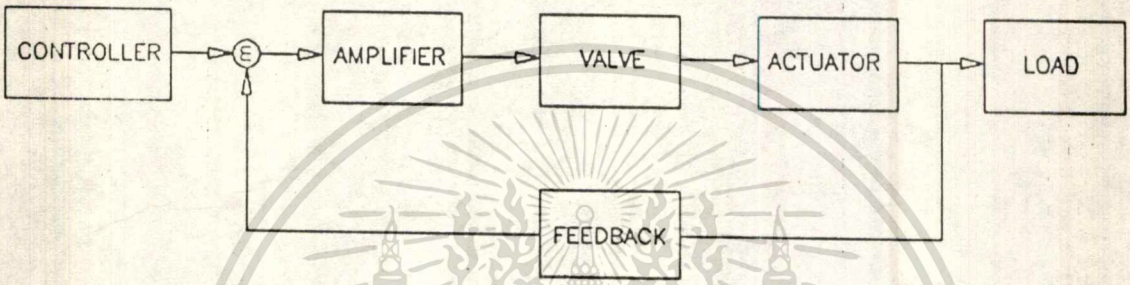
รูปที่ 3.30 วาล์วแบบเซอร์โว



รูปที่ 3.31 ภาพภายในของชิ้นส่วนวาล์วแบบเซอร์โว

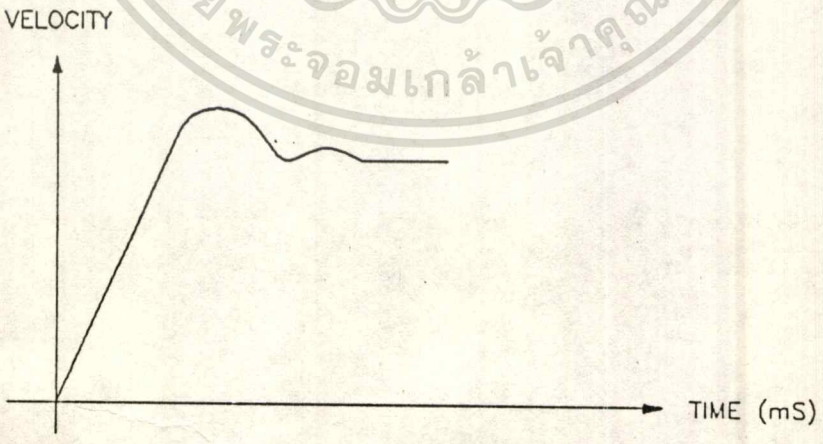
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าระบบกลไกของวาล์วเป็นระบบควบคุมแบบลูปปิด ค่าที่กำหนดจะเป็นแรงที่ได้จากการกระทำของเส้นแรงแม่เหล็ก ในรูปของกระแสไฟฟ้าที่ได้รับจากหน่วยควบคุมตำแหน่งและซิงโครไนซ การไหลของของเหลวจะถูกส่งเข้าสู่ไฮดรอลิคมอเตอร์หรือกระบอกสูบ และเปลี่ยนเป็นการเคลื่อนที่ตามแนวแกนที่ติดตั้งกระบอกสูบหรือมอเตอร์แบบไฮดรอลิคนั้น ในกรณีที่เป็นมอเตอร์แบบไฮดรอลิค จะมีอุปกรณ์ป้อนกลับความเร็วติดตั้งอยู่ที่เพลา เพื่อใช้ในการควบคุมความเร็วรอบของเพลามอเตอร์ ซึ่งบล็อกไดอะแกรมจะแสดงดังรูปที่ 3.32



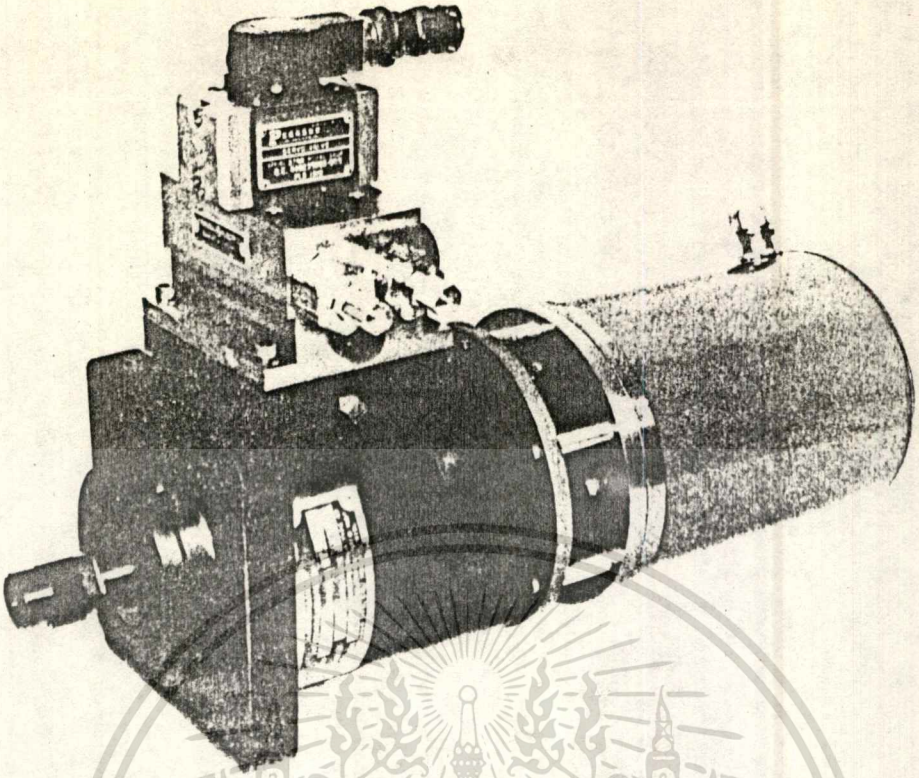
รูปที่ 3.32 ระบบป้อนกลับความเร็ว

ผลตอบสนองของวาล์วแบบเซอร์โวสามารถแสดงได้ดังรูปที่ 3.33 ซึ่งเป็นการทดสอบของ Step Response

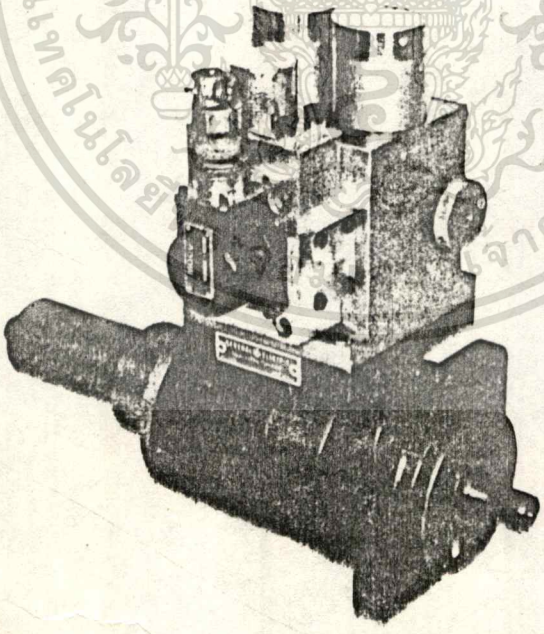


รูปที่ 3.33 ผลตอบสนองของวาล์วแบบเซอร์โว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

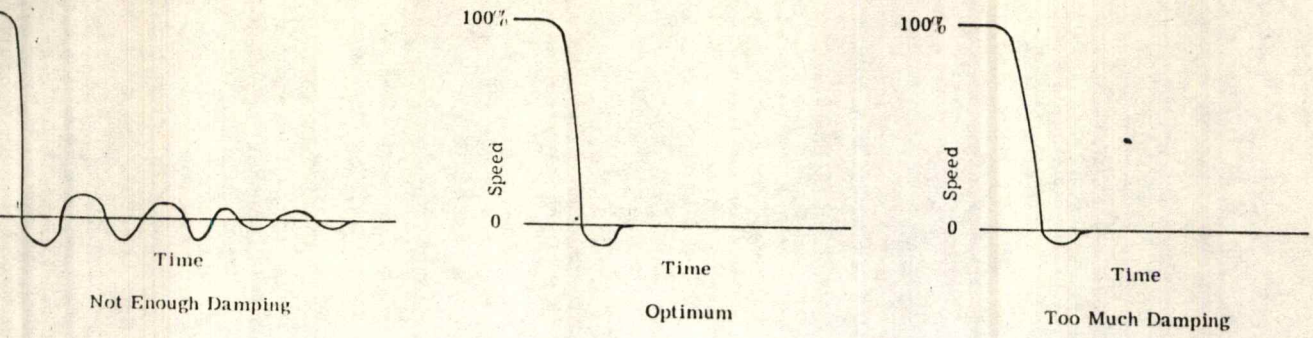


รูปที่ 3.34 ไฮดรอลิคมอเตอร์และเซอร์โววาล์ว

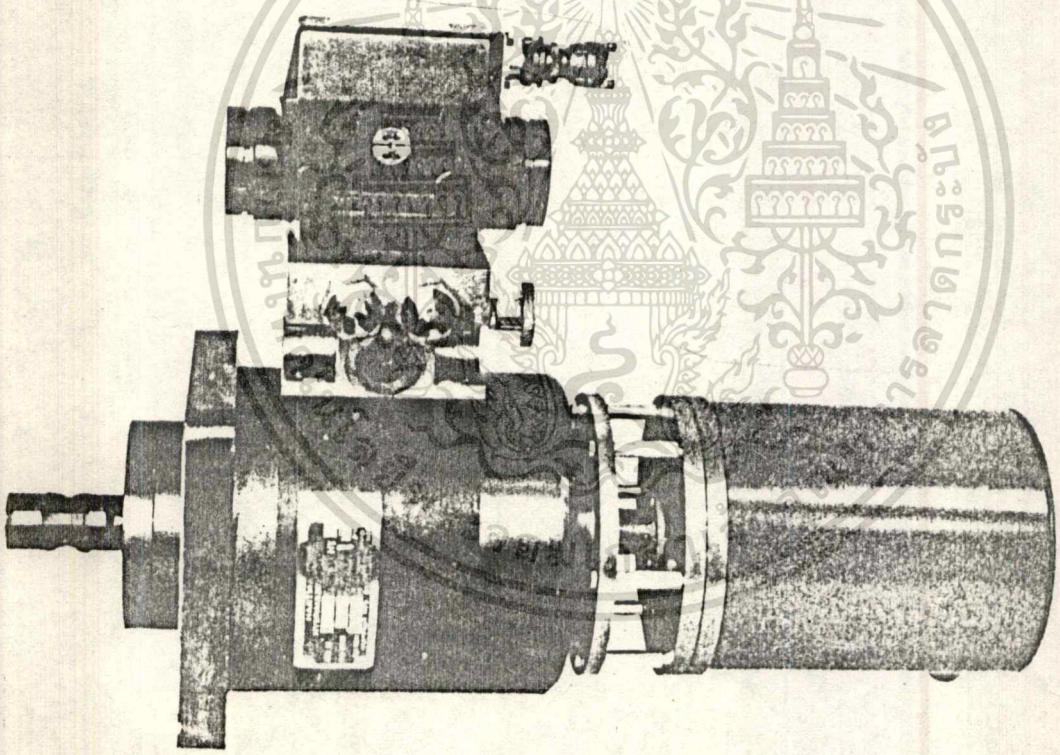


รูปที่ 3.35 ไฮดรอลิคมอเตอร์และเซอร์โววาล์วที่ติดตั้งทาโคมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 ผลตอบสนองของความเร็วต่อเวลาของวาล์ว



รูปที่ 3.37 ไฮดรอลิคมอเตอร์และเซอร์โววาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การโปรแกรมทางเดินของคมตัด (Cutter Caterline Programming)

4.1 การโปรแกรมแบบค่าสัมบูรณ์ (Absolute Programming)

ในการโปรแกรมแบบค่าสัมบูรณ์นี้ จุดกำเนิดจะถูกเลือกใช้ในแต่ละแกน เพื่อการอ้างอิงของโปรแกรม การเคลื่อนที่ทั้งหมดจะเริ่มจากจุดกำเนิด นั่นก็คือการเคลื่อนที่ทั้งหมดที่ถูกกำหนดในโปรแกรมจะเป็นตำแหน่งที่ค่าจริง ๆของจุดใดๆ ที่ต้องการเฉพาะ โดยทั่วไปการโปรแกรมแบบนี้จะต้องถูกกระทำโดยผู้โปรแกรมที่มีประสบการณ์ ซึ่งค่าต่าง ๆ ที่ถูกโปรแกรมจะสอดคล้องและใกล้เคียงกับค่าที่กำหนดทางวิศวกรรมเขียนแบบ จุดอ้างอิงในแบบพิมพ์ (Drawing) จะเป็นค่าที่เทียบกับจุดกำเนิด ในการโปรแกรมเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์นี้ โปรแกรมจะต้องเริ่มต้นด้วยรหัส G90 ก่อน และใช้รหัส X,Z แทนการเคลื่อนที่ตามแนวขวางและแนวตรงตามลำดับ

4.2 การโปรแกรมแบบค่าเพิ่ม (Incremental Programming)

เมื่อใช้โปรแกรมแบบค่าเพิ่ม จุดกำเนิดจะไม่มีความจำเป็นที่ต้องนำมาใช้ การเคลื่อนที่ทั้งหมดจะอ้างอิงจากตำแหน่งสุดท้ายของการเคลื่อนที่ของคมตัด ซึ่งหมายถึงว่าการเคลื่อนที่แบบค่าเพิ่มทั้งหมดคือ ระยะแทนที่ (Displacement) จากตำแหน่งที่ให้มา การเคลื่อนที่แบบพิกัดสั้น ๆ ในทันทีทันใดจะเป็นข้อได้เปรียบสำหรับการโปรแกรมแบบค่าเพิ่ม และการเคลื่อนที่จะสอดคล้องกันโดยตรงกับค่าที่เคลื่อนที่จริงของคมตัด ส่วนข้อได้เปรียบอื่น ๆ ก็คือ ทิศทางเครื่องหมายการเคลื่อนที่มีความสัมพันธ์กันโดยตรงกับการเคลื่อนที่ของคมตัด การโปรแกรมแบบค่าเพิ่มสำหรับพิกัดที่เป็นบวก จะให้คมตัดเคลื่อนที่ไปทิศทางหนึ่ง และพิกัดที่เป็นลบก็จะให้คมตัดเคลื่อนที่ไปในทิศทางตรงข้าม สำหรับการโปรแกรมแบบค่าสัมบูรณ์นั้นเครื่องหมายขึ้นอยู่กับ Quadrant ของการเคลื่อนที่คมตัด ทำนองเดียวกันในการโปรแกรมเครื่องกัด โปรแกรมแบบค่าเพิ่มจะใช้รหัสเริ่มต้นคือ G91 และการเคลื่อนที่จะใช้รหัส U,W แทน X,Z ตามลำดับ

4.3 การโปรแกรมแบบจุดกำเนิด

การเขียนโปรแกรมแบบนี้จะกำหนดจุดกำเนิดของคมตัดไว้ในส่วนของโปรแกรม โดยจะต้องเป็นการโปรแกรมแบบค่าสัมบูรณ์เท่านั้น และเรียกใช้พารามิเตอร์ต่าง ๆ เช่น รีจิสเตอร์ (Register) พรีเซ็ท (Preset) จุดเริ่มต้นของชิ้นงาน จุดอ้างอิงของคมตัดที่กำหนดขึ้น หรือ กำหนดโคออดิเนทแบบค่าสัมบูรณ์ต่าง ๆ เป็นต้น ขึ้นอยู่กับว่าเป็นเครื่องจักรชนิดใด ทั้งนี้เพราะแต่ละเครื่องมีการเรียกพารามิเตอร์เหล่านี้แตกต่างกันแล้วแต่ผู้ผลิต การโปรแกรมแบบจุดกำเนิด และระบบโคออดิเนทจะเริ่มต้นด้วยรหัส G92 สำหรับเครื่องกัด และ รหัส G50 สำหรับการเคลื่อนที่ใด ๆ รหัสจะบอกให้ระบบทราบว่าคมตัดอยู่ในตำแหน่งใดตลอดเวลา ซึ่งตำแหน่งนี้จะสัมพันธ์กับจุดกำเนิด ในความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นจริงเครื่องจักรจะรู้ว่าคมตัดอยู่ที่ใด แต่จะไม่รู้ว่าจุดกำเนิดว่าอยู่ที่ไหน เพราะฉะนั้นเมื่อบอกให้เครื่องจักรทราบว่าคมตัดอยู่ที่ใดเมื่อเทียบกับจุดกำเนิดแล้ว ก็จะสามารถหาค่าของจุดกำเนิดได้ ดังนั้นการโปรแกรมแบบค่าสัมบูรณ์จะให้ค่าของตำแหน่งที่เทียบกับจุดกำเนิดได้

4.4 ลักษณะการเคลื่อนที่แบบต่างๆ ของ CNC

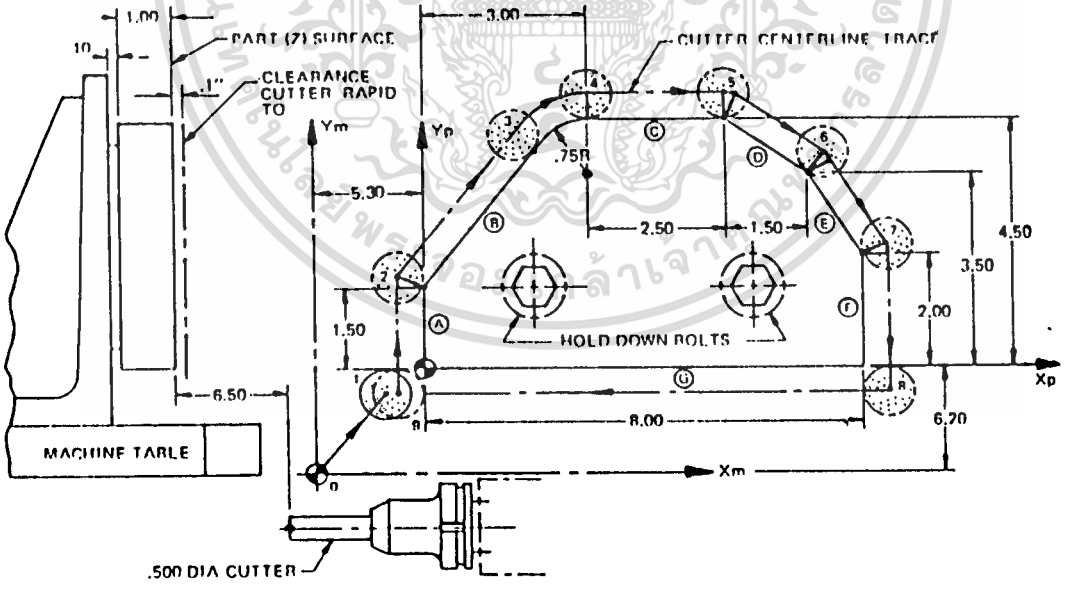
รหัส G00 หมายถึงการให้คมตัดเคลื่อนที่ไปด้วยความเร็วสูงสุดที่ถูกจำกัดไว้ เป็นผลให้เครื่องจักรเคลื่อนที่ไปอย่างรวดเร็วในตำแหน่งที่ต้องการ รหัสนี้จะถูกใช้ในบางกรณีที่ต้องการให้คมตัดเคลื่อนที่อย่างรวดเร็วโดยไม่มีการแมชชีน เพื่อที่จะไปปฏิบัติการเจาะหรือเคลื่อนที่เข้าหารู เป็นต้น คมตัดจะเคลื่อนที่ด้วยความเร็วที่เร็วที่สุด อย่างไรก็ตามในเครื่องจักรจริง ๆ จะต้องมีอุปกรณ์ป้องกันการเคลื่อนที่เกินพิกัด ในการควบคุมส่วนใหญ่การเคลื่อนที่แบบเร็วจะเป็นแบบจุดต่อจุดตัวอย่างเช่น ทุก ๆ แกนที่ถูกโปรแกรมจะเริ่มเคลื่อนที่แบบอิสระ และเมื่อแกนใดแกนหนึ่งถึงจุดที่ต้องการแล้ว มอเตอร์จะหยุด แต่ถ้าแกนใดยังไม่ถึงก็จะเดินต่อไปจนถึงจุดที่กำหนด นั่นคือมอเตอร์ขับเคลื่อนในแต่ละแกนจะมีความเร่งเดียวที่เร่งที่สุด และแน่นอนที่สุด ความปลอดภัยก็ยังคงเป็นสิ่ง จำเป็นที่จะให้ผู้ใช้ CNC ได้รู้สำหรับการเคลื่อนที่แบบรวดเร็วที่ประมาณ 254 มม.ต่อวินาทีการเคลื่อนที่แบบเชิงเส้นสัมพันธ์ (Linear Interpolation) การเคลื่อนที่แบบนี้จะใช้รหัสเริ่มต้นด้วย G01 ซึ่งจะเป็นผลให้เครื่องจักรปรับค่าของความเร็วในการควบคุมการขับเคลื่อนอย่างอิสระในแต่ละส่วน เพื่อที่จะรักษาการเคลื่อนที่ให้ได้เส้นตรง การเคลื่อนที่แบบวนรอบ (Circular Motion) จะใช้รหัส G02 ในการโปรแกรมส่วนโค้งของวงกลมแบบตามเข็มนาฬิกา และรหัส G03 แบบทวนเข็มนาฬิกา โดย CNC มีขีดความสามารถที่จะประมวลให้การทำงานระหว่างความสัมพันธ์ของตำแหน่ง และความเร็วของการเคลื่อนที่ประสานได้อย่างราบเรียบโดยเริ่มเคลื่อนที่และหยุดได้พร้อมกัน ที่จุดเริ่มต้นของการเคลื่อนที่แบบวนรอบการควบคุมจะรู้ตำแหน่งของคมตัดโดยจะต้องบอกตำแหน่งที่ต้องการที่จุดสุดท้ายของส่วนโค้ง ได้ทั้งในแบบค่าเพิ่มหรือค่าสัมบูรณ์ เพื่อที่จะรู้ถึงตำแหน่งจุดศูนย์กลางและรัศมีของส่วนโค้ง โดยจะใช้ I J และ K สำหรับเครื่อง CNC รุ่นเก่าๆ และใช้ r เป็นรัศมีของคมตัด R เป็นรัศมีของส่วนโค้งของชิ้นงาน การโปรแกรมเส้นกึ่งกลางของแนวคมตัด ซึ่งจะมีความสำคัญมากสำหรับปฏิบัติงานกับเครื่องจักรในระบบเก่า ผู้ปฏิบัติงานจะต้องมีความชำนาญเพียงพอและสามารถใส่ค่าชดเชยระยะคมตัด เพื่อให้ได้งานที่มีความถูกต้องแน่นอน สำหรับระบบใหม่ๆ จะมีการชดเชยระยะคมตัดไว้ให้แล้ว ทำให้ง่ายในการโปรแกรมถ้าเครื่องจักรสามารถทราบตำแหน่งของเส้นกึ่งกลางของแนวคมตัด

4.5 การคำนวณระยะของเส้นกึ่งกลางของแนวคมตัด

ส่วนของโปรแกรมการเคลื่อนที่รอบนอกงานของเครื่องกัดแสดงได้ในรูปที่ 4.1 โดยจะเริ่มจากการเลือกใช้เส้นผ่าศูนย์กลางของคมตัด และบอกถึงรูปร่างที่แท้จริงของชิ้นงานเข้าไปโดยบล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม CNC จะสามารถหาระยะชดเชยของคมตัดเทียบกับรูปร่างจริงของชิ้นงาน โดยคมตัดจะต้องเคลื่อนที่ไปจากจุดที่ 1,2,3,.... ถึง 8 และกลับมาที่ 9 ระยะแทนที่คือวงกลม (คมตัด) เส้นสัมผัสของวงกลมจะเคลื่อนที่ไปรอบ ๆ ชิ้นงานเส้นระหว่างจุดศูนย์กลางคมตัดกับขอบของคมตัดที่ชิ้นงาน (รัศมีของคมตัด) จะตั้งฉากกับส่วนของชิ้นงานที่กำลังตัด ทำให้สามารถที่จะคำนวณตำแหน่งจุดศูนย์กลางของคมตัดแต่ละจุดโดยการหาจากสามเหลี่ยมมุมฉาก หลังจากคำนวณขนาดเหล่านี้ได้แล้ว จะต้องใส่ระยะขนาดเหล่านี้ในโปรแกรม ค่าที่เป็นจริงของระยะจะแตกต่างกันขึ้นอยู่กับการเลือกใช้โปรแกรมว่าเป็นแบบค่าเพิ่มหรือค่าสัมบูรณ์ แต่เนื่องจากว่าการคำนวณจะใช้เวลามากกว่าการโปรแกรม ดังนั้นจึงควรจะใช้วิธีการแบบการชดเชยคมตัด หรือใช้คอมพิวเตอร์ช่วยคำนวณในการทำโปรแกรม อย่างไรก็ตามจะต้องเข้าใจวิธีการคำนวณเหล่านี้ให้ดีก่อน ต่อไปให้พิจารณาขั้นตอนต่าง ๆ ที่จำเป็น เพื่อที่จะคำนวณหาตำแหน่งจุดศูนย์กลางของคมตัดที่มีเส้นผ่าศูนย์กลางขนาด 12.7 มม. การโปรแกรมแบบค่าเพิ่ม จะกำหนดโดยเสมือนว่าชิ้นงานได้ถูกตัดออกอย่างหยาบ ๆ ก่อนให้เป็นรูปเป็นร่าง และให้มีส่วนเผื่อเหลืออีกประมาณ3.81 มม. สำหรับการตกแต่งสุดท้าย จะใช้วัสดุเป็นมาตรฐานทางวิศวกรรม หมายเลข 1030 (SAE 1030)



รูปที่ 4.1 โปรแกรมการเคลื่อนที่รอบนอกงานของเครื่องกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการโปรแกรมข้างล่างนี้ สามารถใช้ได้กับทุกๆ ชิ้นงานที่มีรูปร่างที่คล้ายกันนี้ รวมทั้งรหัสต่างๆ ด้วย แต่จะต้องตรวจสอบซ้ำสำหรับระบบที่ใช้จริงๆ ก่อน

START UP

N0010 G70 G40 G80 G91

G70 เป็นระบบมิลลิเมตร (มม.)

G40 "ยกเลิก" การชดเชยคมตัด

G80 "ยกเลิก" การทำซ้ำ

G91 ใช้แบบค่าเพิ่ม * "ยกเลิก" เพื่อความปลอดภัย

ผู้โปรแกรมที่มีประสบการณ์ส่วนใหญ่จะใช้การโปรแกรมแบบค่าเพิ่ม การเคลื่อนที่แบบค่าเพิ่มคือ ระยะการแทนที่ไปเรื่อย ๆ ตามแนวของคมตัด จากตำแหน่งปัจจุบัน ส่วนการโปรแกรมแบบค่าสัมบูรณ์ จะนำคมตัดไปที่ตำแหน่งใด ๆ ที่กำหนดไว้โดยสัมพันธ์กับจุดกำเนิด

โปรแกรมแบบค่าเพิ่มต่อไปนี้ เป็นตัวอย่างที่แสดงถึงความสัมพันธ์โดยตรง ระหว่างลำดับของโปรแกรมกับการเคลื่อนที่ของคมตัด

N0020 G92 X0 Y0 Z0 ปรับตัวแสดงผลเป็นศูนย์เพื่อให้รู้ว่าอยู่ที่จุดกำเนิด

N0030 S800 M03 ความเร็วตัด 800 แบบหมุนตามเข็มนาฬิกา

N0040 G00 X120.65 Y151.13 เคลื่อนที่แบบเร็วไปที่ x,y

N0050 Z-162.56 เคลื่อนที่เร็วไปที่ z มีระยะ 2.54 มม.

N0060 G01 Z-29.21 F76.2 ป้อนไป 1.27 มม.เมื่อถึงแล้วกัดชิ้นงานที่ 76.2 มม.ต่อนาที

N0070 X7.62 M08 ป้อนไปที่จุด 1 ให้ระบายความร้อน

4.5.1 การเคลื่อนที่ไปบนพื้นผิว "A" ระหว่างจุด 1 กับ 2

การเคลื่อนที่ในแนวแกน y ในทิศทางเดียว โดยจะคำนวณจากขนาดที่กำหนดให้ และไม่มีการเคลื่อนที่ในแนวแกน x จากจุด 1 ไป 2 อย่างไรก็ตาม การเคลื่อนที่ในแนวแกน y จะต้องยาวกว่า 38.1 มม. ซึ่งเป็นความยาวของระยะ "A" ด้วยค่าไม่ต่ำกว่าค่าของ y_2 (ดูรูปที่ 4.2) การคำนวณนี้มีความจำเป็นถึงแม้ว่าตำแหน่งรัศมีของคมตัด "r" ได้ตั้งฉากพื้นผิว "B" แล้วก็ตาม

$$y = r - \tan(45^{o} - \alpha_1/2)$$

$$= 2.41 \quad \text{มม.}$$

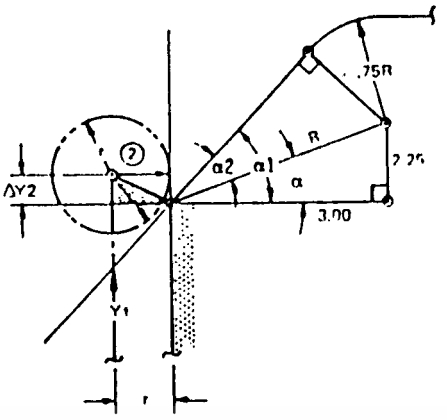
เมื่อคำนวณ y ได้แล้ว ก็สามารถที่จะหาค่า y_2 โดยการโปรแกรมค่าyจากจุด 1 ไป 2

$y_2 = 46.86$ มม.

และโปรแกรมในบรรทัดต่อไปคือ

N0080 Y 46.86 ย้ายคมตัดไปที่จุด 2

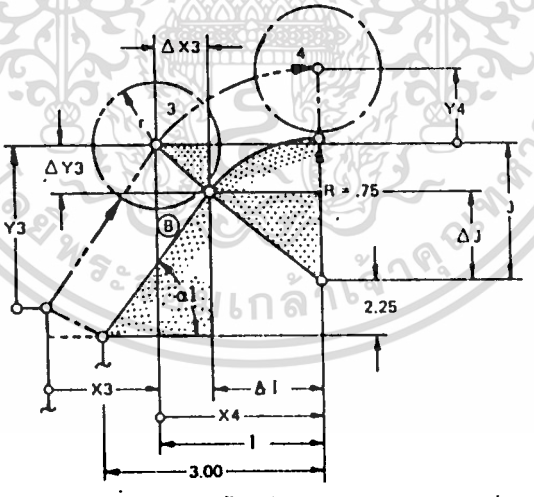
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 จุดกึ่งกลางคมตัดที่จุด 2

4.5.2 การเคลื่อนที่ไปบนพื้นผิว "B" ไปที่จุด 3 และโค้งไปหาจุด 4

ในที่นี้จะต้องคำนวณขนาดของ x_3 และ y_3 จากรูปที่4.3 ในช่วงของโปรแกรมการเคลื่อนที่ของคมตัดจากจุด 2 ไป 3 โดยขนาดเหล่านี้ไม่ได้หายไปจากรูปร่างจริงของชิ้นงาน แต่จะทำการคำนวณพร้อมกันไปกับการคำนวณจากจุด 3 ไปหาจุด 4 ที่เป็นส่วนโค้งโดยวิธีการเดียวกันกับที่ผ่านมา



รูปที่ 4.3 จุดเปลี่ยนแนวจากเส้นตรงเป็นเส้นโค้ง

เพื่อที่จะกัดชิ้นงานในแนวรัศมีโค้ง 19.05 มม. โปรแกรมการเคลื่อนที่ของ x และ y จากจุดเริ่มต้นของส่วนโค้งถึงจุดสุดท้ายของส่วนโค้งนั้น เป็นการแทนค่าของ I และ J ที่เป็นยูนิทเวคเตอร์ให้ถูกต้อง เพื่อที่จะเป็นจุดศูนย์กลางของส่วนโค้งที่สัมพันธ์กัน สำหรับจุด 3 ขณะนี้ไม่ได้เป็นส่วนหนึ่งส่วนใดของจุดโคออดิเนทแล้ว จากนั้นระบบจะต้องเปลี่ยนวิถีการเคลื่อนที่จากเส้นโค้งเป็นเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นตรงที่จุด 4 ซึ่งเครื่องจักรแบบเก่า ๆ ยังคงใช้วิธีนี้อยู่ สำหรับเครื่องจักรในรุ่นใหม่ๆ ส่วนใหญ่จะไม่ใช้วิธีนี้แล้ว แต่ในบางครั้งยังคงยอมรับในระบบเก่าอยู่ คือใช้ได้กับรูปแบบการคำนวณเดิมได้ การเคลื่อนที่แบบวนรอบ โดยปกติจะทำการโปรแกรมโดยใช้ค่ารัศมีซึ่งทำให้ง่าย อย่างไรก็ตาม จะทำแบบยากก่อน การคำนวณสำหรับการเคลื่อนที่ของคมตัดในรูปที่ 4.3 จะเป็นดังนี้

$$x_4 = \Delta I + \Delta x_3$$
$$= 19.0525 \quad \text{มม.}$$

ซึ่งจะเป็นระยะแทนที่ของ x จากจุด 3 ไป 4 และระยะแทนที่ของ y จากจุด 3 ไป 4 คือ

$$y_4 = R + r - (\Delta J + \Delta y_3)$$
$$= 8.542 \quad \text{มม.}$$

โดยที่ R = รัศมีของส่วนโค้งของชิ้นงานที่ต้องการตัด

r = รัศมีของคมตัด

ขนาดของ I จะถูกวัดจากจุดเริ่มต้น 3 ของส่วนโค้ง ขนานกับแกน x

$$I = \Delta I + \Delta x_3$$
$$= x_4$$
$$= 19.0525 \quad \text{มม.}$$

ทำนองเดียวกัน ขนาดของ J จะถูกวัดจากจุดเริ่มต้น 3 ของส่วนโค้ง ขนานกับแกน y

$$J = \Delta J + \Delta y_3$$
$$= 16.858 \quad \text{มม.}$$

คมตัดจะอยู่ในตำแหน่งจุดเริ่มต้นของส่วนโค้ง และส่วนควบคุมจะทราบถึงตำแหน่งนี้ตลอดเวลาจนกระทั่งถึงจุดสิ้นสุดของการเคลื่อนที่ที่ผ่านมา และจากจุดเริ่มต้นของส่วนโค้งในเส้นทางตามแนว J มาถึง I นี้ คอมพิวเตอร์จะคำนวณหาจุดศูนย์กลางของส่วนโค้ง โดยใช้ค่า I และ J เป็นด้านของสามเหลี่ยมมุมฉาก ส่วนควบคุมจะคำนวณหาด้านตรงข้ามมุมฉากของสามเหลี่ยมซึ่งก็คือรัศมีของส่วนโค้งทำให้รู้ถึงจุดศูนย์กลางและรัศมี จากนั้นส่วนควบคุมก็จะเดินไปกับคมตัดที่เคลื่อนที่ไปในระยะจริงที่ถูกกำหนดโดยโปรแกรมขนาดของ x และ y

ส่วนของเส้นวนรอบ สามารถเขียนได้ดังต่อไปนี้

N0100 G02 X19.0525 Y8.542 I19.0525 J-16.858

ยูนิทเวคเตอร์สามารถใช้ได้ทั้งอักษรตัวใหญ่และตัวเล็ก เช่น I หรือ I เป็นต้น

4.5.3 การเคลื่อนที่ไปบนพื้นผิว "C" จากจุด 4 ไปจุด 5

การคำนวณสำหรับส่วนพื้นผิว "C" จากจุด 4 ไปจุด 5 จะทำได้เช่นเดียวกับกรณีของ 4.5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

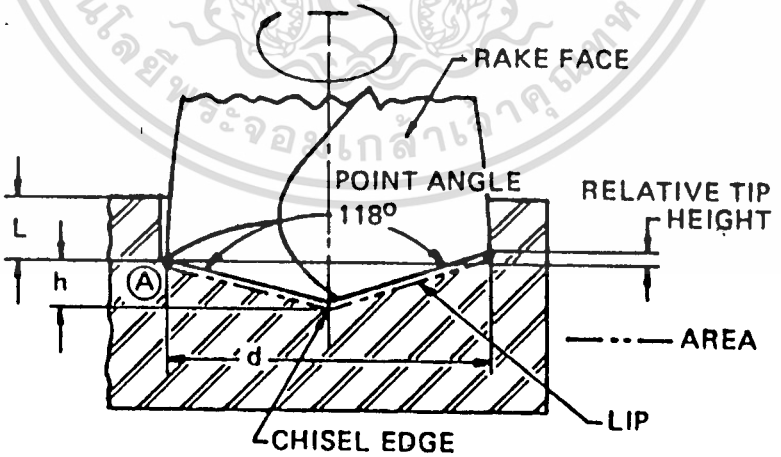
บทที่ 5
แรงกระทำที่เกิดขึ้นจากคมตัดขณะเครื่องจักรทำงาน (Machining Forces)

เครื่องจักรที่ใช้ CNC ในการควบคุมการทำงานของเครื่องจักร จำเป็นจะต้องทราบถึงโครงสร้างของเครื่องจักรกล เช่น น้ำหนักของส่วนเคลื่อนไหวทางกล ความสามารถของหน่วยขับกำลัง และระบบขับแกนของเครื่องจักร (ไฮดรอลิคมอเตอร์ หรือกระบอกสูบไฮดรอลิค) สิ่งเหล่านี้ถือว่าเป็นเรื่องที่สำคัญในการกำหนดการเคลื่อนที่ของแกนเครื่องจักร ในการตัดชิ้นงาน และผลตอบสนองความเร็วในการเคลื่อนที่ของแกนเครื่องจักรอย่างมีประสิทธิภาพ เพื่อให้ได้ผลลัพธ์ของชิ้นงานออกมามีความเที่ยงตรงสูงไม่ก่อให้เกิดการเคลื่อนที่แบบมีการ "Overshoot" แต่อย่างใด ในบทนี้จะกล่าวเฉพาะสูตรสำคัญ ๆ เพื่อจะเป็นประโยชน์ในการเลือกค่าความเร็วของคมตัด (Speed and Feed) กำลังงานที่ใช้ในการต่อสู้กับแรงด้านเนื่องจากคมตัดในขณะตัดชิ้นงาน ซึ่งพบเห็นบ่อย ๆ ในงานเจาะกัดและงานกลึงจะแยกเป็นหัวข้อย่อย ๆ ดังนี้

5.1 การเจาะ(Drilling)

5.1.1 ความเร็วในการตัด (Cutting Speed) V_c

พิจารณารูปที่ 5.1 จากรูปแสดงถึงพื้นที่ที่จะเจาะที่จุด A



รูปที่ 5.1 แสดงพื้นที่เจาะ(Drill Area)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_c = \frac{d \cdot \pi \cdot n}{1000} \quad \text{เมตรต่อนาที}$$

โดยที่ V_c = ความเร็วในการตัด

n = ความเร็วรอบของดอกสว่าน (RPM)

d = เส้นผ่าศูนย์กลางของดอกสว่าน มีหน่วยเป็นมิลลิเมตร

5.1.2 Rate Metal Removal (Qd)

หมายถึง ปริมาณของเศษโลหะที่ถูกตัดโดยคมตัดของดอกสว่าน คิดเป็นปริมาตรของเศษโลหะต่อหน่วยเวลา สามารถคำนวณได้ดังนี้

$$Q_d = \frac{A \cdot n \cdot F}{1000} \quad \text{ลูกบาศก์เซนติเมตร}$$

โดยที่ Q_d = ปริมาตรของโลหะที่ถูกขจัดโดยคมตัดของดอกสว่าน

n = ความเร็วรอบของดอกสว่านขณะทำการเจาะ

F = ความเร็วในการป้อนดอกสว่านเข้าสู่ชิ้นงาน

A = พื้นที่ระนาบของคมตัด มีหน่วยเป็นตารางมิลลิเมตร

จากที่กล่าวมาแล้วนี้ สามารถสรุปเป็นค่าคงที่สำหรับงานเจาะวัสดุประเภทโลหะชนิดต่าง ๆ ได้ดังตารางที่ 5.1

MATERIALS	CONSTANTS (k)		
	DRILLING	MILLING	TURNING
MILD STL. 25 RC	1.0	1.0	.9
MILD STL. 25 - 30 RC	1.6	1.8	1.3
HARD STL. 50 RC.	1.9	2.1	1.5
SOFT CAST IRON	.8	.7	.5
HARD CAST IRON	.9	1.1	1.0
ALUMINUM	.35	.4	.3
BRASS	.5	.6	.4
BRONZE	.6	.8	.7
STAINLESS 400	1.3	1.3	1.1
STAINLESS 300	1.6	1.9	1.7
TITANIUM	1.0	1.0	1.1
NICKEL ALLOYS	1.6	1.6	1.5

ตารางที่ 5.1 ตารางค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 กำลังงานคิดเป็นหน่วยแรงม้าที่ให้กับสปินเดิล (Spindle) (E)
สามารถคำนวณหาได้ดังสมการดังนี้

$$HP_s = K \cdot Q_d \cdot 100\% \quad \text{แรงม้า}$$

โดยที่ HP_s = กำลังงาน คิดเป็นหน่วยแรงม้าที่ให้กับเครื่องจักรขณะทำงานแมชชีน
Q_d = ปริมาตรของโลหะที่ถูกขจัดโดยคมตัด
K = ค่าคงที่ของโหละชนิดต่างๆ ที่ทำการแมชชีน แสดงดังตารางที่ 5.1

ในทางทฤษฎีถือว่าจะไม่เกิดการสูญเสียกำลังงานใด ๆ ในการแมชชีน นอกเหนือจากกำลังงานที่ให้กับคมตัด แต่ในความเป็นจริงแล้ว จะเกิดการสูญเสียเกิดขึ้นในทางปฏิบัติ ดังนั้นสมการข้างต้นจึงมองในรูปของประสิทธิภาพของการขับเคลื่อนเพลาของสปินเดิล ซึ่งจะมีค่าในการใช้งาน 0.7-0.85 ขึ้นอยู่กับเงื่อนไขของคมตัดที่ใช้ในการแมชชีน ค่ากำลังงานที่ให้กับเพลาของสปินเดิล คิดเป็นหน่วยแรงม้า จะคำนวณได้จากสมการ

$$HP_m = \frac{K \cdot Q}{E} \quad \text{แรงม้า}$$

โดยที่ HP_m = กำลังงานที่ใช้ในการขับเคลื่อนเพลาของสปินเดิลขณะทำการแมชชีน

5.1.4 แรงบิดของสปินเดิลที่เกิดขึ้นเนื่องจากการเจาะ (TS)
หมายถึงแรงต้านของคมตัดที่มีต่อชิ้นงานขณะที่ดอกสว่านกำลังตัดชิ้นงาน สามารถคำนวณหาแรงบิดได้จากสมการ

$$TS = \frac{663030 \cdot HP_s}{n} \quad \text{แรงม้า}$$

โดยที่ HP_s = หน่วยของกำลังงานคิดเป็นแรงม้า

ในทางวิศวกรรม หนึ่งแรงม้า(HP) มีค่าเท่ากับ 33.00 ฟุตปอนด์ต่อนาที หรือ 550 ฟุตปอนด์ต่อวินาทีในระบบอังกฤษ หนึ่งแรงม้ามีค่าเป็น 75 กิโลกรัมเมตรต่อวินาที หรือ 542.5 ฟุตปอนด์ต่อวินาทีในระบบเมตริก

5.1.5 เวลาที่ใช้ในการแมชชีน
หมายถึงเวลาที่ใช้ในการขจัดวัสดุ ยกตัวอย่างเช่น ระยะเวลาที่ใชในการเจาะชิ้นงาน คิดเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยเวลา (นาที) สามารถคำนวณได้ดังสูตรต่อไปนี้

$$T = \frac{L}{F} \quad \text{นาที}$$

โดยที่ L = ความลึกของขุองการเจาะ หน่วยเป็นมิลลิเมตร

F = อัตราการป้อนสว่านเข้าสู่วัสดุที่เป็นชิ้นงาน มีหน่วยเป็นมิลลิเมตรต่อนาที

5.2 งานที่เกี่ยวกับการกลึง (TURNING)

5.2.1 ความเร็วในการตัดชิ้นงาน V_t

รูปที่ 5.2แสดงถึงลักษณะของการตัดชิ้นงานของคมตัดโดยวิธีการกลึง

$$V_t = \frac{D \cdot \pi \cdot n}{1000} \quad \text{เมตรต่อนาที}$$

5.2.2 ปริมาณของวัสดุที่ถูกตัดออก (Q_t)

หมายถึง ปริมาตรของวัสดุที่ถูกขจัดโดยคมตัด คิดเป็นปริมาตรต่อหน่วยเวลา สามารถคำนวณหาได้ดังสูตรต่อไปนี้

$$Q_t = T \cdot F \cdot n \cdot c \quad \text{ลูกบาศก์เซนติเมตร}$$

โดยที่ Q_t = ปริมาตรของวัสดุที่ถูกตัดออกโดยวิธีการกลึง

F = อัตราความเร็วในการป้อนมีดกลึงผ่านชิ้นงาน

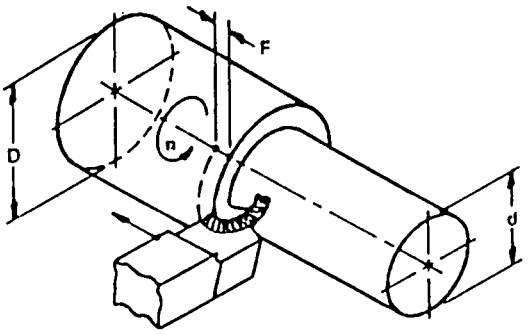
n = ความเร็วในการหมุนของชิ้นงาน คิดเป็นรอบต่อนาที

T = ความลึกของคมมีดที่ใช้ตัด คิดอ้างอิงจากผิวงานที่ยังไม่ได้ถูกตัด โดยวิธีการกลึง

โดยที่ $T = \frac{D-d}{2}$

c = เส้นรอบวงของชิ้นงานก่อนทำการตัดโดยวิธีการกลึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 งานกลึง

5.2.3 กำลังงานที่ใช้ขับเคลื่อสปินเดิล

หมายถึงกำลังงานที่ใช้ในการขับเคลื่อนสปินเดิล ขณะทำการตัดชิ้นงานด้วยมีดกลึงด้วยวิธีการกลึง

$$HP = \frac{K \cdot Q}{E} \qquad \text{แรงม้า}$$

โดยที่ HP_m = กำลังงานที่ใช้ในการขับเคลื่อนสปินเดิลด้วยมอเตอร์

5.2.4 แรงบิดที่หัวจับบนสปินเดิล

หมายถึง แรงต้านที่เกิดจากการตัดโลหะที่เป็นชิ้นงาน ถูกจับยึดอยู่บนหัวจับของสปินเดิลกับมีดกลึงสามารถคำนวณหาได้จากสูตรดังนี้

$$T_s = \frac{63030 \cdot HP_s}{n} \qquad \text{แรงม้า}$$

โดยที่ T_s = แรงบิด

5.2.5 ความราบเรียบของผิวงาน

หมายถึงความเรียบบนชิ้นงานหลังจากการตัดชิ้นงานด้วยวิธีการกลึง ในทางทฤษฎีสามารถคำนวณหาค่าความราบเรียบของผิวชิ้นงาน(Surface Roughness) S_r ซึ่งจะหาได้จากสูตรดังนี้

$$S_r = \frac{F}{8 \cdot r} \qquad \text{หนึ่งในล้านของเมตร}$$

โดยที่ F = ความเร็วในการป้อนมีดกลึงไปบนผิวงาน มีหน่วยเป็นมิลลิเมตรต่อรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

r = รัศมีความโค้งของคมตัดของมีดกลึง มีหน่วยเป็นมิลลิเมตร

5.2.6 ความเร็วและความหน่วงขณะเข้าสู่ตำแหน่งเริ่มต้นของการแมชชีนโดยวิธีการกลึง

ผู้ปฏิบัติงานจะต้องมีการเผื่อระยะก่อนที่คมตัดจะเข้ากระทำการตัดชิ้นงานโดยวิธีการกลึง เนื่องจากการเข้าสู่สภาวะเสถียรภาพของระบบควบคุมอาจมีผลกระทบต่อชิ้นงาน ตัวอย่างเช่น แกนของเครื่องจักรเคลื่อนที่พามีดกลึงจากจุดอ้างอิงของเครื่องจักรเข้ามาถึงจุดเริ่มต้นของการแมชชีน การเคลื่อนด้วยความเร็วสูงสุดของแกนเครื่องจักรดังกล่าวอาจเกิดการเคลื่อนตัวเกินตำแหน่งที่ต้องการควบคุม ซึ่งมาจากสาเหตุหลายประการ เช่น น้ำหนักของแกนเครื่องจักร กำลังของมอเตอร์ขับแกน ความฝืดของชิ้นส่วนทางกล เป็นต้น ผลจากสิ่งเหล่านี้อาจทำให้มีดกลึงเข้าตัดชิ้นงาน เกิดความเสียหายต่อชิ้นงานหรือคมตัดได้ สูตรต่อไปนี้อาจช่วยแก้ปัญหาต่างๆ ให้หมดไปหากได้มีการกำหนดขีดจำกัดของ CNC หรือข้อมูลทางเครื่องจักรจากโรงงานผู้ผลิต ตัวอย่างเช่น เครื่องควบคุม CNC กำหนดค่าค่าคงที่ของเวลา t_1 = 0.12 ดังนั้นสามารถจะหาระยะของ A_1 ในรูปที่ 5.3 ได้โดยการคูณค่าคงที่ของเวลาเข้ากับค่าความเร็ว สมมติกำหนดให้ความเร็วของสปินเดิลคือ 350 RPM. ความเร็วของการป้อนงานและความเร็วในการเข้าตัดชิ้นงาน 8 รอบของสปินเดิลต่อระยะการเคลื่อนที่ 25.4 มิลลิเมตร = 3.175

ถ้าความเร็วที่ปลายของคมตัดมีค่า 3.175 มิลลิเมตรในทุกๆ รอบของการหมุนของสปินเดิล

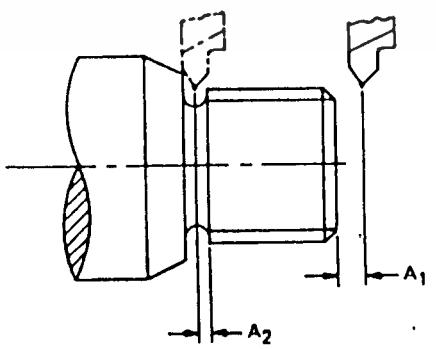
$$V_L = 3.175 \cdot 350$$

$$= 1,111.25 \quad \text{มิลลิเมตรต่อนาที}$$

$$A_2 = t_1 \cdot V_L$$

$$= \frac{(0.12 \cdot 1,111.25)}{60}$$

$$= 2.2225 \quad \text{มิลลิเมตร}$$



รูปที่ 5.3 แสดงค่าความเร็วและความหนาที่มีผลต่อการเคลื่อนที่ของแกนเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเวลาคงที่ของเวลาสามารถจะคำนวณได้ โดยกำหนดให้ค่าคงที่ของเวลาสำหรับความเร็วแสดงด้วย t_2

$$t_2 = t - t_2 - (t_1 \cdot EXP(\frac{-t}{t_1}))$$

ถ้ากำหนดให้

$$\frac{-t}{t_1} = \ln 0.05$$

$$= -2.99573$$

หรือ

$$t = 0.12 \cdot 2.99573$$

$$= 0.3595$$

แทนค่าสมการข้างต้นด้วย t เพื่อหา t_2 จะได้

$$t_2 = 0.359 - 0.12 + 0.35940.05$$

$$= 0.2455$$

ระยะทางที่เหมาะสมของ A_1 คือ

$$A_1 = t_2 \cdot V_L$$

จากข้อกำหนดในตัวอย่าง การหาค่าความหน่วงจะคำนวณได้คือ ถ้าความเร็วในการป้อนชิ้นงาน
มีค่า 8 รอบต่อระยะการเคลื่อนที่ 25.4 มิลลิเมตร ความเร็วของสปินเดิล 350 สามารถคำนวณได้ดังนี้

$$V_L = 3.175 \cdot 350$$

$$= 1,111.25 \quad \text{มิลลิเมตรต่อนาที}$$

$$A_2 = t_2 \cdot V_L$$

$$= (\frac{0.245 \cdot 1,111.25)}{60}$$

$$= 4.54 \quad \text{มิลลิเมตร}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ระยะทางสำหรับความเร็วในการเคลื่อนที่ของแกนเครื่องจักร สามารถพิจารณาเป็นสัดส่วนกับค่าของ A_2 ในทางปฏิบัติวิศวกรจะกำหนดค่าระยะ A_1 ให้เปลี่ยนแปลงได้เป็นอัตราส่วน 2 ถึง 4.4 เท่าของ A_2 ดังนั้น

$$A_1 = (2 \sim 4.4) \cdot A_2$$

5.3 การตัดชิ้นงานด้วยคมตัดโดยวิธีการกัด (Milling)

5.3.1 ความเร็วในการตัด Vm คือความเร็วของคมตัด โดยพฤติกรรมของคมตัดจะเคลื่อนที่แต่ชิ้นงานหยุดอยู่กับที่

$$V_m = \frac{D \cdot \pi \cdot n}{1000} \quad \text{เมตรนาที}$$

โดยที่ D = เส้นผ่าศูนย์กลางของคมตัด มีหน่วยเป็นมิลลิเมตร

5.3.2 อัตราการตัดชิ้นงานคิดเป็นหน่วยปริมาตร (Q_m)

$$Q_m = W \cdot T \cdot F \quad \text{ลูกบาศก์มิลลิเมตรต่อนาที}$$

โดยที่ Q_m = ปริมาตรของโลหะที่ถูกตัดโดยคมตัด มีหน่วยเป็นลูกบาศก์เซนติเมตรต่อนาที

W = หน้ากว้างของคมตัด มีหน่วยเป็นมิลลิเมตร

T = ความลึกของคมตัดคิดจากผิวโลหะก่อนตัด มีหน่วยเป็นมิลลิเมตร

F = ความเร็วในการป้อนคมตัดเข้าตัดชิ้นงานในแนวราบ มีหน่วยเป็นมิลลิเมตรต่อนาที

5.3.3 กำลังงานที่ใช้ในการตัดชิ้นงานสำหรับคมตัด

$$HP_m = \frac{K \cdot Q}{E} \quad \text{แรงม้า}$$

5.3.4 แรงบิดที่ใช้ในการหมุนคมตัดขณะตัดชิ้นงานโดยวิธีการกัด
หมายถึงแรงบิดที่เกิดขึ้นที่เพลาของสปินเดิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_s = \frac{63030 \cdot HP_m}{n} \qquad \text{แรงม้า}$$

สูตรการคำนวณที่เสนอนี้ เป็นเพียงการสรุปผลที่เกิดจากกรณีที่คมตัดกระทำต่อชิ้นงานในการเจาะการกลึง และการกัด ซึ่งผลดังกล่าวอาจเป็นผลต่อการทำงานของ CNC ที่จะเอาชนะแรงต้านที่เกิดขึ้นในภาวะต่างๆ ดังกล่าวมาแล้ว ดังนั้นการออกแบบอุปกรณ์ขับเคลื่อนสำหรับติดตั้งกับเครื่องจักรจึงควรอ้างอิงถึงเทอมต่างๆ ที่กล่าวมาแล้วนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เพื่อการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ลบรอยน้ำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การออกแบบส่วนประกอบของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์(CNC)

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ศึกษาวิจัยและออกแบบนี้ จะเป็นเครื่องควบคุมเชิงเลขขนาดสองแกน ซึ่งมีการควบคุมที่เป็นอิสระต่อกัน (Two Axis Simultaneous) และใช้เครื่องกลึงเป็นกรณีศึกษา โดยในบทนี้จะกล่าวถึงโครงสร้างของหน่วยคอมพิวเตอร์กลางและหน่วยควบคุมตำแหน่งที่ได้ออกแบบและจัดสร้างขึ้นมา สำหรับโปรแกรมต่างๆ ที่ใช้ควบคุมการทำงานของ CNC ที่ได้พัฒนาเขียนขึ้นมาจะบรรจุไว้ในภาคผนวก

6,1 หน่วยคอมพิวเตอร์กลาง

CNC ที่วิจัยออกแบบขึ้นมานี้ ประกอบด้วยส่วนต่างๆ คือ หน่วยคอมพิวเตอร์กลาง ทำหน้าที่ประมวลผลกลาง โดยใช้หน่วยประมวลผลกลาง 8088 ที่มีความเร็วของสัญญาณนาฬิกา 4 เม็กกะเฮิร์ท (MHz) มีหน่วยความจำแบบอ่านได้อย่างเดียว 128 กิโลไบท์ (Kilo-bytes) หน่วยความจำแบบอ่านและเขียนได้ขนาด 128 กิโลไบท์ ฮาร์ดแวร์ (Hardware) สามารถใช้อุปกรณ์เชื่อมต่อของคอมพิวเตอร์ส่วนบุคคลที่เทียบได้กับรุ่น XT ของ IBM การพัฒนาโปรแกรมจะใช้ภาษาแอสแซมบลี (Assembly Language) กระทำบนเครื่องคอมพิวเตอร์ส่วนบุคคล การติดต่อกับระบบภายนอกจะผ่านระบบเชื่อมต่ออินพุทเอาท์พุทดังนี้

1. ฮาร์ดแวร์หน่วยแสดงผล แสดงสภาวะการทำงานของระบบควบคุมขณะป้อนโปรแกรมเข้าเครื่องควบคุม หรือแสดงสภาวะขณะที่ระบบควบคุมกำลังทำงานเป็นรหัสตัวเลขฐานสิบ

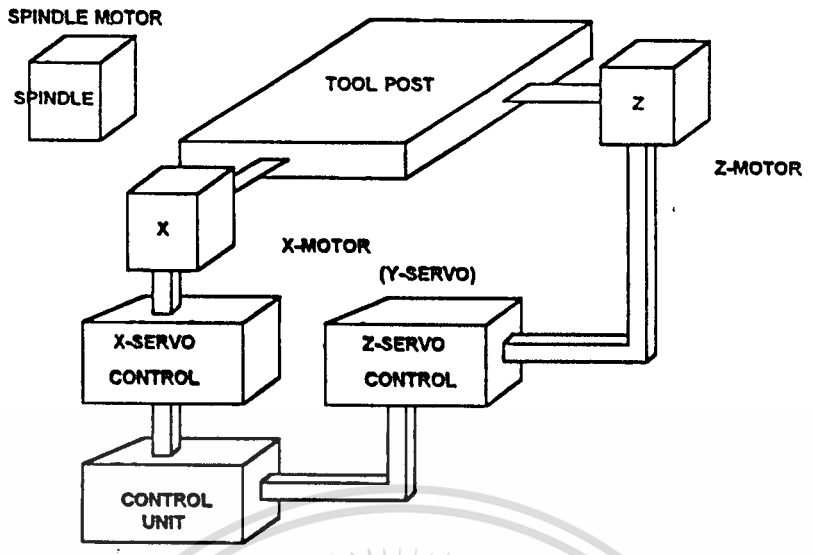
2. ฮาร์ดแวร์สำหรับป้อนข้อมูลเข้าเครื่องควบคุม ประกอบด้วยแป้นคีย์ที่เป็นรหัสตัวเลข และตัวอักษรเท่าที่จำเป็น

3. ฮาร์ดแวร์สำหรับเชื่อมต่อกับระบบภายนอกแบบดรรก ประกอบด้วยส่วนที่เป็นอินพุท และเอาท์พุท

4. ฮาร์ดแวร์สำหรับควบคุมตำแหน่งของแกนเครื่องจักร มีสัญญาณควบคุมแบบอนาล็อกขนาด $\pm$10 โวลท์ และมีสัญญาณป้อนกลับเป็นแบบดรรก

การพัฒนาทางซอฟท์แวร์ (Software) จะทำที่หน่วยคอมพิวเตอร์กลาง โดยแบ่งหมวดซอฟท์แวร์ที่ใช้ในการทำงานออกเป็นส่วนย่อย (Module) ดังแสดงในรูปที่ 6.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



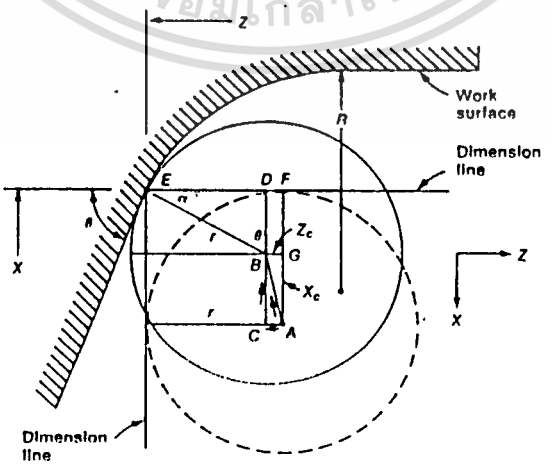
รูปที่ 6.1 บล็อกไดอะแกรมของ CNC ที่สร้างขึ้น

จากรูปที่ 6.1 สามารถจะอธิบายได้ดังนี้

6.1.1 ส่วนกำเนิดฟังก์ชันต่าง ๆ (Function Generator) ของแกนเครื่องจักร

เป็นส่วนกำเนิดจุดของการเคลื่อนที่ให้กับหน่วยควบคุมตำแหน่งเชิงตัวเลขแบบ 2 แกน แบ่งออกได้เป็นสองฟังก์ชัน คือ ฟังก์ชันที่เป็นเชิงเส้นตรง (Linear Interpolation) และฟังก์ชันที่เป็นเชิงเส้นโค้ง(Circular Interpolation)

เนื่องจากการควบคุมตำแหน่งเชิงเลขมีความต่อเนื่องและมีการซิงโครไนซระหว่างแกนทั้งสอง จึงเรียกพฤติกรรมการเคลื่อนที่แบบนี้ว่า วิถีการเคลื่อนที่(Trajectory) ดังแสดงในบล็อกไดอะแกรม รูปที่ 6.2

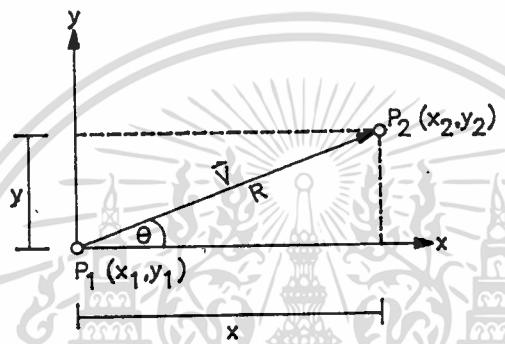


รูปที่ 6.2 แสดงวิถีการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปของชิ้นงานที่ต้องการจะขึ้นอยู่บนพื้นฐานความยืดหยุ่น ได้แก่ ความสามารถในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของวิถีการเคลื่อนที่ได้อย่างเที่ยงตรง ความสามารถในการกำหนดความเร็วของการเคลื่อนที่ของวิถีนั้นได้อย่างแม่นยำ และความสามารถที่จะกำหนดเงื่อนไขการทำงานขณะที่วิถีการเคลื่อนที่ยังดำเนินอยู่

6.1.1.1 การกำเนิดวิถีที่เป็นฟังก์ชันเส้นตรงเชิงความเร็ว เป็นวิธีการที่เลือกใช้ในการควบคุมการเคลื่อนที่ของคมตัดสำหรับ CNC ที่สร้างขึ้น แสดงได้ดังรูปที่ 6.3



รูปที่ 6.3 การกำเนิดวิถีที่เป็นฟังก์ชันเส้นตรงเชิงความเร็ว

การเคลื่อนที่เป็นฟังก์ชันเส้นตรง ถ้า P_1 เป็นจุดเริ่มต้นของการเคลื่อนที่ ที่เป็นฟังก์ชันเส้นตรงที่พิกัด x_1,y_1 และ P_2 คือจุดสิ้นสุดของการเคลื่อนที่ ที่พิกัด x_2,y_2 ความเร็วในการเคลื่อนที่จาก P_1 ไปยัง P_2 จะมีค่าเป็น

$$V = \frac{\sqrt{((x_2-x_1)^2+(y_2-y_1)^2)}}{t}$$

ความเร็วสัมพัทธ์ของ y และ x ที่ทำให้เกิด V จะมีค่าเป็น

$$V_x = V\cdot\cos\theta$$
$$V_y = V\cdot\sin\theta$$

ถ้า R คือความยาวของเส้นตรงที่ลากจากจุด P_1 มายัง P_2 หรือหมายถึงการเคลื่อนที่ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คมตัดในเชิงเส้นตรง จะได้ว่า

$$\sin\theta = \frac{(y_2 - y_1)}{\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}}$$

$$= \frac{(y_2 - y_1)}{R}$$

ทำนองเดียวกัน

$$\cos\theta = \frac{(x_2-x_1)}{\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}}$$

$$= \frac{(x_2-x_1)}{R}$$

โดยการแทนค่า จะได้

$$V_x = \frac{V\cdot(x_2 - x_1)}{R}$$

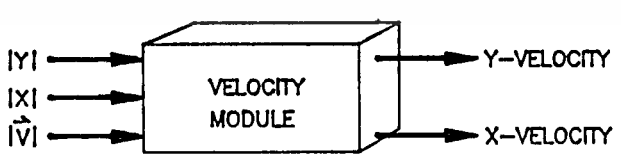
$$V_y = \frac{V\cdot(y_2 - y_1)}{R}$$

โดยที่ V = ความเร็วของการเคลื่อนที่ของคมตัด ที่ต้องการกำหนดในการใช้งาน

R = การเคลื่อนที่ของคมตัดในเชิงเส้นตรง

$$= \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$$

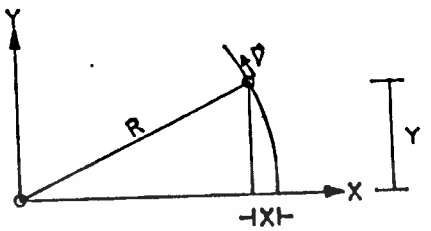
บล็อกไดอะแกรมของระบบซอฟท์แวร์แสดงได้ดังรูปที่ 6.4



รูปที่ 6.4 บล็อกไดอะแกรมของระบบซอฟท์แวร์กรณีเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.1.2 การกำเนิดวิถีที่เป็นฟังก์ชันเส้นโค้งเชิงความเร็ว แสดงได้ดังรูปที่ 6.5



รูปที่ 6.5 วิถีการเคลื่อนที่ของคมตัดที่เป็นฟังก์ชันเส้นโค้ง

การกำเนิดส่วนโค้งของวงกลมสามารถสังเคราะห์ขึ้นได้หลายวิถี เช่น วิธีการแทนค่าตัวแปรลงในสมการวงกลม

$$r = \sqrt{(x^2 + y^2)}$$

และ $$y = r \cdot \sin\theta$$

$$x = r \cdot \cos\theta$$

การคำนวณโดยวิธีการแทนค่าแบบตรงไปตรงมา จะทำให้ความเร็วในการทำงานของหน่วยประมวลผลกลางสูญเสียไปกับการคำนวณเป็นส่วนใหญ่ ดังนั้น เพื่อเป็นการแก้ปัญหานี้จึงได้ทำการสังเคราะห์ตารางค่าของซายน์ และโคซายน์ขึ้น เพื่อสร้างส่วนโค้งตามสมการข้างต้น ค่าของ x และ y ที่เกิดขึ้นจึงอยู่ในรูปของจุดที่อยู่บนส่วนโค้งที่มีรัศมีความโค้ง r และความเร็วของ x,y จะหาได้ดังนี้

$$V_x = \frac{V \cdot \Delta x}{r}$$

$$V_y = \frac{V \cdot \Delta y}{r}$$

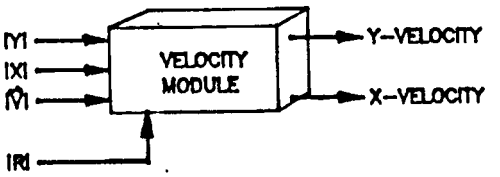
โดยที่ r = รัศมีความโค้งของวงกลม

V_y = ความเร็วของการเคลื่อนที่ในแนวแกน y

V_x = ความเร็วของการเคลื่อนที่ในแนวแกน x

$\Delta x, \Delta y$ = ระยะสัมพัทธ์ของการเคลื่อนที่ในแนวแกน x,y (หาค่าได้จากตาราง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 บล็อกไดอะแกรมของระบบซอฟท์แวร์กรณีเส้นโค้ง

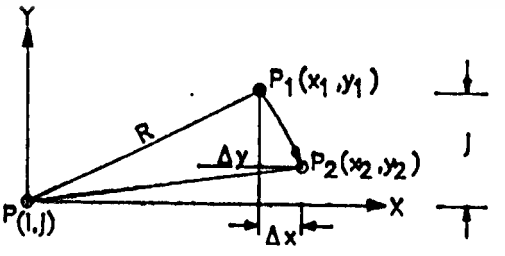
จะเห็นได้ว่า วิถีการเคลื่อนที่ของคมตัดที่เป็นฟังก์ชันเส้นตรงและเส้นโค้งนั้น จะแตกต่างกันที่ค่าพารามิเตอร์ r ซึ่งที่กล่าวมานี้จุดศูนย์กลางความโค้งของ r จะต้องอยู่ที่จุดกำเนิดเท่านั้น แต่ในกรณีที่จุดศูนย์กลางความโค้งอยู่ที่จุดใด ๆ ลักษณะการอ้างจุดเริ่มต้นของส่วนโค้งจะเป็นดังนี้ (ดูรูปที่ 6.7)

$$r^2 = I^2 + J^2$$

โดยที่ I = จุดเริ่มต้นของส่วนโค้งบนแกน x ที่อ้างอิงจากจุดศูนย์กลางของส่วนโค้ง

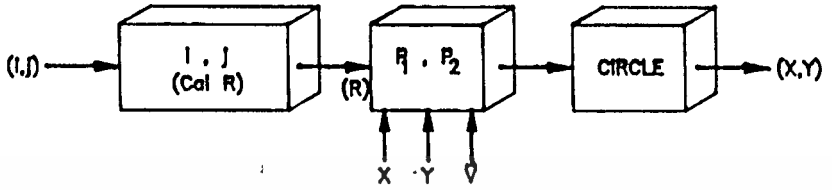
J = จุดเริ่มต้นของส่วนโค้งบนแกน y ที่อ้างอิงจากจุดศูนย์กลางของส่วนโค้ง

รูปที่ 6.8 แสดงบล็อกไดอะแกรมของระบบซอฟท์แวร์กรณีเส้นโค้ง ที่จุดศูนย์กลางความโค้งอยู่ที่จุดใด ๆ



รูปที่ 6.7 การอ้างจุดศูนย์กลางความโค้งของวงกลมสำหรับ CNC ที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.8 บล็อกไดอะแกรมของระบบซอฟท์แวร์กรณีเส้นโค้งในรูปที่ 6.7

6.1.1.3 การสร้างตารางฟังก์ชันซายน์และโคซายน์ การสร้างฟังก์ชันของส่วนโค้งโดยวิธีการสังเคราะห์ตาราง จะเป็นการลดเวลาที่ใช้ในการคำนวณลงได้มาก แต่การที่ได้มาซึ่งตารางที่จะครอบคลุมฟังก์ชันทั้งหมดของวงกลมนั้นจำเป็นจะต้องอาศัยการคำนวณค่าต่าง ๆ ไว้ล่วงหน้าเป็นจำนวนมากซึ่งในทางปฏิบัติแล้วตารางควรจะมีขนาดเล็กที่สุด เพื่อเป็นการประหยัดขนาดของหน่วยความจำ และหลีกเลี่ยงการคำนวณที่ยุ่งยากและต่อเนื่องให้มากที่สุด ซึ่งจะทำให้ความเร็วของการทำงานของระบบทั้งหมดเร็วขึ้น ในรูปที่ 6.9 ถ้า r คือรัศมี L คือเส้นที่เกิดจากการเคลื่อนที่ของแกนรัศมีเชิงมุมการเลือกแบ่งส่วนย่อยๆ (Strip) ของพื้นที่ที่เกิดจากเส้นโค้งนี้เป็น n และ m โดยที่ n เป็นความกว้างของเส้นแบ่งย่อยในแนวตั้ง (Vertical Strip) และ m เป็นความสูงบนแกน y ที่เกิดจากเส้นแบ่งย่อยในแนวตั้งแต่ละเส้น โดยวัดระยะจากแกน x ไปยังจุดที่เส้นแบ่งย่อยในแนวตั้งตัดกับส่วนโค้งของวงกลม การเก็บข้อมูลค่าพิกัดต่าง ๆ ของส่วนโค้งมีขีดจำกัดของความละเอียด ซึ่งพิกัดของส่วนโค้งสามารถที่จะบอกได้ 2 ลักษณะคือ การบอกพิกัดแบบอ้างระยะสัมพัทธ์ (Relative) และการบอกพิกัดแบบสัมบูรณ์ (Absolute) ซึ่งเป็นการบอกพิกัดที่มีการอ้างอิงจากจุดกึ่งกลางของส่วนโค้งนั้น และเนื่องจากการบอกพิกัดทั้ง 2 แบบนี้จะต้องใช้ปริมาณตัวเลขแทนขนาดพิกัด ดังนั้นค่าความละเอียด (Resolution) จะขึ้นอยู่กับจำนวนตัวเลขที่ใช้อ้างระยะหรือขนาดของพิกัด เช่น การอ้างระยะพิกัดสูงสุด 1000 หน่วย มีความละเอียดเป็น 1/1000 หน่วย

จากค่าของความละเอียดข้างต้น สามารถสรุปเป็นสมการได้ว่า

$$D = \frac{n}{B}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

n = ความกว้างของเส้นแบ่งย่อยในแนวตั้ง

B = จำนวนของข้อมูลย่อยสูงสุดที่สามารถจะเก็บไว้เพื่อการคำนวณ

เช่นถ้า B เป็นหน่วยความจำขนาด 8 บิท จำนวนข้อมูลย่อยสูงสุดที่ยอมรับได้จะมีค่าเท่ากับ 256 ข้อมูล และถ้า n = 1

$$D = \frac{1}{256}$$

$$= 0.0039$$

นั่นคือ ในรูปของสมการทางคณิตศาสตร์สำหรับเลขฐานสอง

$$D = \frac{1}{1+1\cdot 2^{1}+1\cdot 2^{2}+1\cdot 2^{3}+1\cdot 2^{4}+1\cdot 2^{5}+1\cdot 2^{6}+1\cdot 2^{7}}$$

เมื่อจัดให้อยู่ในรูปสมการทั่วไป จะได้

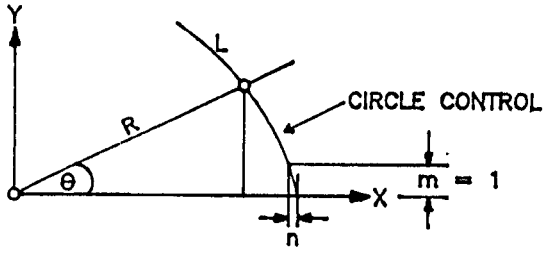
$$D = \frac{n}{a_0+a_1\cdot b^{1}+a_2\cdot b^{2}+a_3\cdot b^{3}+a_4\cdot b^{4}+a_5\cdot b^{5}+a_6\cdot b^{6}+a_7\cdot b^{7}}$$

$$= \frac{n}{1+\sum_{p=0}^{q}(a_p\cdot b^{p})}$$

โดยที่ a = สัมประสิทธิ์ของเลขฐานสอง มีค่า 0 หรือ 1

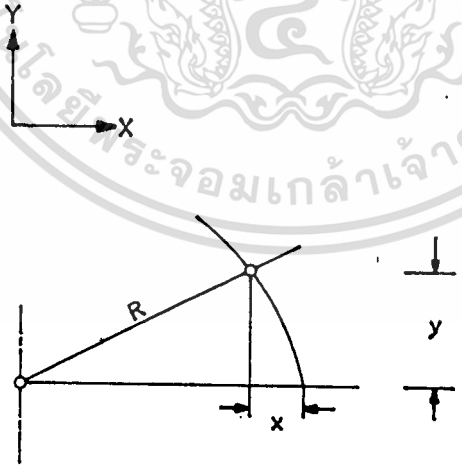
b = เลขฐานสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แสดงการสังเคราะห์ตารางไซน์และโคซายน์

จากสมการข้างต้น q จะบอกถึงขนาดความกว้างของฐาน เช่น ถ้า q = 2 ความกว้างของฐานจะมีค่าเท่ากับ q+1 หรือเท่ากับ 3 ซึ่งหมายถึงเลข 3 หลัก (3 บิท) นั่นก็คือ ถ้าความกว้างของฐานมีค่าน้อย ค่าความละเอียดจะลดลง ค่าของ q จึงควรมีค่าที่เหมาะสมกับการใช้งาน เช่น ถ้ายอมรับให้การกำหนดส่วนโค้งมีค่าความคลาดเคลื่อนไม่เกิน 1/1000 หน่วย q จึงต้องมีค่าเท่ากับ 1000 ซึ่งเมื่อเปลี่ยนเป็นเลขฐานสอง ขนาดของ q จะมีทั้งหมดสิบหลัก และเนื่องจาก q จะเป็นตัวกำหนดค่าขนาดของหน่วยความจำต่อหนึ่งชุดข้อมูลที่ใช้ในคอมพิวเตอร์ ดังนั้น q จึงเป็นตัวกำหนดค่าความเร็วในการโยกย้ายข้อมูลด้วย



รูปที่ 6.10 แสดงการหาค่าของ n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m = f(n)$$

เช่นถ้า n=1 และ m=10 ค่าของ q หาได้ดังนี้

$$b^q = m$$

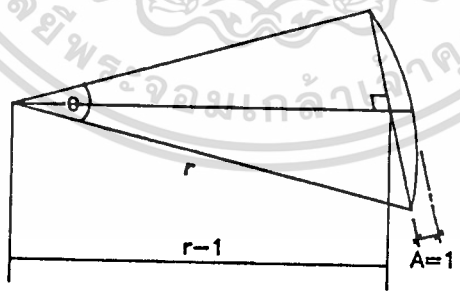
$$q = \frac{\ln m}{\ln b}$$

$$= \frac{\ln 10}{\ln 2}$$

$$= 3.3219$$

= เลขฐานสองสี่หลัก

วิธีการกำหนดค่าความละเอียดที่กล่าวมาข้างต้นนี้ เป็นวิธีการที่มีใช้กันอยู่ทั่วไป ทั้งในกรณีของเส้นตรงและเส้นโค้ง อย่างไรก็ตาม ในกรณีของเส้นโค้งนั้น ในวิทยานิพนธ์ฉบับนี้ ผู้วิจัยได้คิดค้นวิธีการออกแบบเพื่อกำหนดค่าความละเอียดขึ้นมาอีกวิธีหนึ่ง โดยการพิจารณาแบ่งมุมของส่วนส่วนโค้งออกเป็นส่วนย่อย ซึ่งสามารถให้ผลลัพธ์ที่ดีกว่าวิธีแรก เพราะในกรณีของเส้นโค้งนั้น เมื่อใช้วิธีใหม่นี้จะได้ความเป็นเชิงเส้น (Linearity) ที่ดี ซึ่งวิธีนี้สามารถจะอธิบายได้ดังต่อไปนี้



รูปที่ 6.11 แสดงการแบ่งส่วนโค้งเป็นส่วนย่อย ๆ

จากรูปที่ 6.11 ส่วนโค้งของวงกลมซึ่งมีรัศมีความโค้ง r จะถูกแบ่งออกเป็นส่วนย่อย โดยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 6.11 ส่วนโค้งของวงกลมซึ่งมีรัศมีความโค้ง r จะถูกแบ่งออกเป็นส่วนย่อย โดยที่ส่วนย่อย ๆ นี้จะถูกกำหนดจากค่าของมุม และทำการแบ่งมุมออกเป็น 2 ส่วนเท่า ๆ กัน หลังจากนั้นต้องลากเส้น CORD และกำหนดให้ระยะห่างระหว่างเส้น CORD กับส่วนโค้งมีค่าเป็นเลขจำนวนเต็มที่น้อยที่สุด คือเท่ากับ 1 (ดูรูปที่ 6.11 ประกอบ) นั่นคือ

$$\theta = 2\cdot\cos^{-1}\frac{(r-1)}{r}$$

จำนวนของข้อมูลย่อย B ในหนึ่ง Quadrant จะขึ้นอยู่กับค่าของมุม คือ

$$B = \frac{90^\circ}{\theta}$$

$$= \frac{\pi}{4\cdot\cos^{-1}\frac{(r-1)}{r}}$$

เช่นเดียวกันกับวิธีแรก m จะเป็นความสูงบนแกน y ที่เกิดจากการแบ่งมุม และลากเส้น CORD ดังในรูปที่ 6.11

ดังนั้น

$$m_1 = \Delta y_1$$

$$m_2 = \Delta y_2$$

$$= r\cdot\sin\theta$$

$$= r\cdot\sin 2\theta - r\cdot\sin\theta$$

$$= r\cdot[\sin 2\theta - \sin\theta]$$

ทำนองเดียวกัน

$$m_q = \Delta y_q$$

$$= r\cdot[\sin q\theta - \sin(q-1)\theta]$$

โดยที่ m_q = ความสูงบนแกน y ที่เกิดจากการแบ่งมุม

q = 1,2,3,.....,B

θ = ความละเอียดของสเกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ถ้าให้ r มีค่ามากที่สุดเท่ากับ 10 เมตร มีค่าความคลาดเคลื่อนจากเส้น CORD เท่ากับ 1 ไมครอน (Micron) จากรูปที่ 6.11

$$A = r\cdot(1-\cos\frac{\theta}{2})$$

$$1 = 10^{7}\cdot(1-\cos\frac{\theta}{2})$$

$$\theta = 2\cdot\cos^{-1}\frac{10^{7}-1}{10^{7}}$$

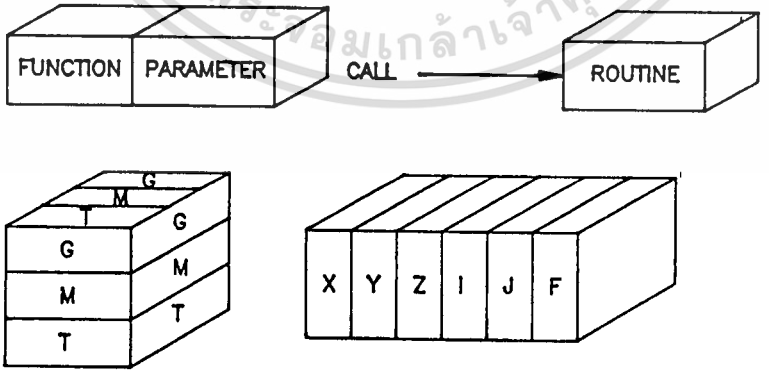
$$= 0.0512469036^{\circ}$$

แบ่งออกเป็นส่วนโค้งย่อยได้เท่ากับ

$$B = \frac{90^{\circ}}{0.0512469036}$$

$$= 1756.20366813 \quad \text{ส่วนย่อย}$$

ซึ่งในการใช้งานจริง จะใช้ค่า 2000 ส่วนย่อย รูปที่ 6.12 แสดงถึงบล็อกไดะแกรมการทำงานของโปรแกรม

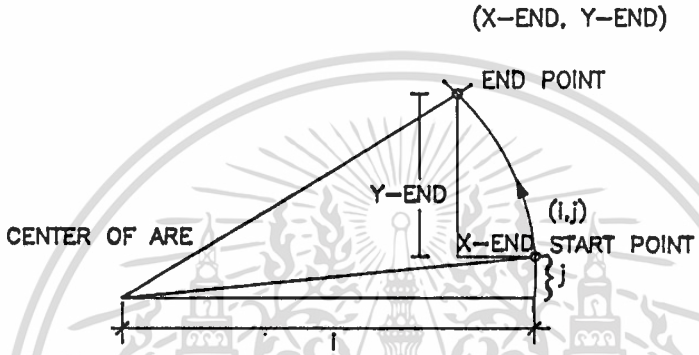


รูปที่ 6.12 บล็อกไดะแกรมการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระนาบ x-y สามารถจะทำได้ดังนี้

- กำหนดจุดเริ่มต้นของส่วนโค้งบนระนาบ x,y
- กำหนดจุดสิ้นสุดของส่วนโค้งบนระนาบ x,y
- เรียกใช้ฟังก์ชันส่วนโค้ง



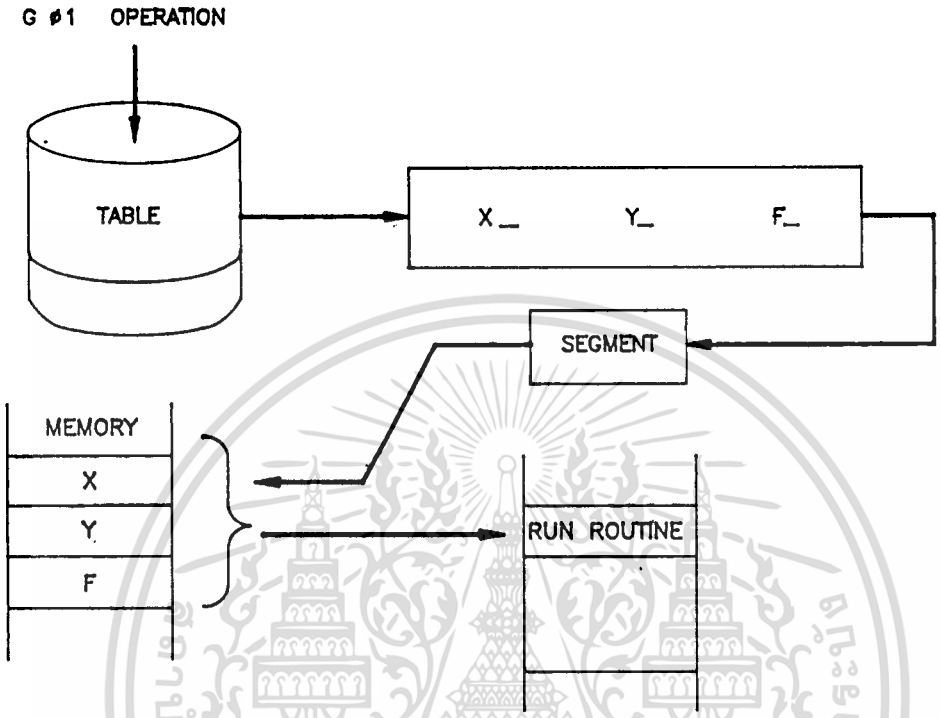
รูปที่ 6.13 รูปแสดงส่วนโค้งที่ต้องการสร้าง

6.1.1.4 ฟังก์ชันการทำงาน (Operating Function) ซึ่งประกอบด้วยฟังก์ชันอินพุท ได้แก่ รหัส G,M หรือ T ตารางรหัสพารามิเตอร์ต่างๆ ของอินพุท ซึ่งแสดงได้ดังรูปที่ 6.14

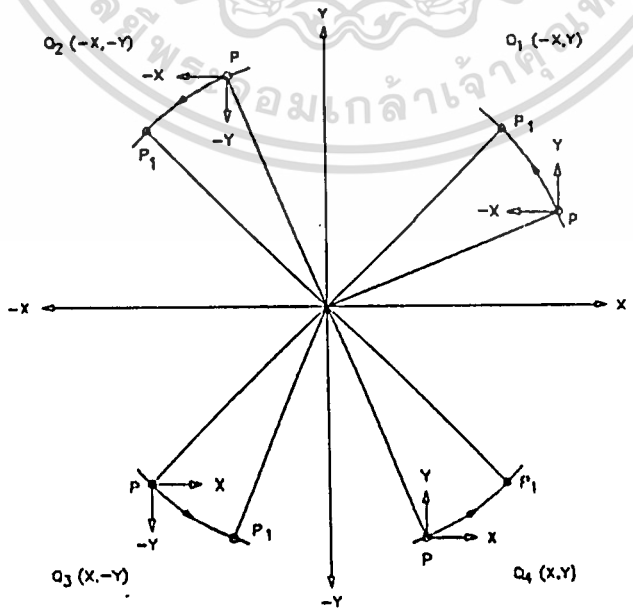
ก. การออกแบบรหัส GO2 และ GO3 แสดงได้ดังรูปที่ 6.15 ซึ่งแสดงส่วนของวงกลมที่ถูกแบ่งออกเป็นส่วนโค้งย่อย ๆ 4 ส่วนคือ Q_1, Q_2, Q_3 และ Q_4 ระบบทิศทางการเปลี่ยนแปลงจากจุดอ้างอิง P ไปยัง P_1 ถ้าพิจารณาทิศทางโดยแทนเครื่องหมาย จะระบุทิศทางได้ดังตารางที่ 6.1

ข. การพิจารณาหาตำแหน่งของส่วนโค้ง การกำเนิดส่วนโค้งของวงกลมจะต้องระบุค่าพารามิเตอร์ต่าง ๆ ได้แก่ จุดเริ่มต้นของส่วนโค้ง รัศมีของส่วนโค้ง และจุดสิ้นสุดของส่วนโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.14 รูปแสดงฟังก์ชันการทำงาน

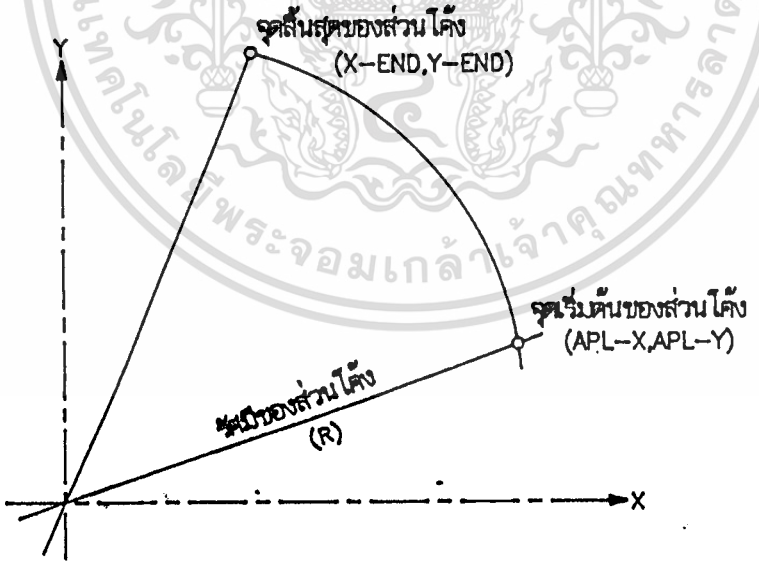


รูปที่ 6.15 การออกแบบรหัส GO2 และ GO3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sin	x	y	sin	x	y	Code
Q_1	-	+	Q_1	0	1	1
Q_2	-	-	Q_2	0	0	0
Q_3	+	-	Q_3	1	0	2
Q_4	+	+	Q_4	1	1	3

ตารางที่ 6.1 แสดงรหัสการเคลื่อนที่ในทิศทางต่าง ๆ



รูปที่ 6.16 รูปแสดงการกำหนดค่าพารามิเตอร์ต่าง ๆ เพื่อหาตำแหน่งของส่วนโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รัศมี r สามารถคำนวณได้จาก

$$r = \sqrt{(x^2 + y^2)}$$

ทิศทางของส่วนโค้งที่กำเนิดขึ้นสามารถระบุได้จากรหัสคือ G02 และ G03 G02 หมายถึงการกำเนิดส่วนโค้งในทิศทางตามเข็มนาฬิกา G03 หมายถึงการทำเนิดส่วนโค้งในทิศทางทวนเข็มนาฬิกา จุดเริ่มต้นของส่วนโค้งจะอ้างจากจุดสุดท้ายของการกำเนิดเส้นโค้งหรือเส้นตรงใด ๆ และจะแทนด้วยสัญญลักษณ์ ALP_X (Absolute Last Point) และ ALP_Y จุดสุดท้ายของการกำเนิดส่วนโค้งจะแทนด้วย X_END และ Y_END การระบุตำแหน่งของส่วนโค้งสามารถหาได้ โดยการตรวจสอบเครื่องหมายที่เกิดจากการหาผลต่างของ X_END Y_END และ ALP_X ALP_Y ดังนี้ ถ้า

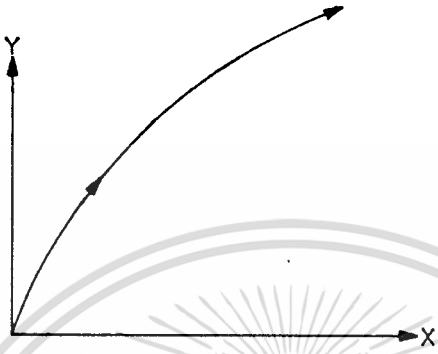
ALP_X - X_END ; x-axis
ALP_Y - Y_END ; y-axis

ซึ่งสรุปได้ดังตารางที่ 6.2

x	y	Quadrant
-	+	Q_1
-	-	Q_2
+	-	Q_3
+	+	Q_4

ตารางที่ 6.2 แสดงรหัสเครื่องหมายที่สัมพันธ์กับส่วนต่าง ๆ ของวงกลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



รูปที่ 6.17 แสดงการกำเนิดส่วนโค้งใช้วิธีตาราง

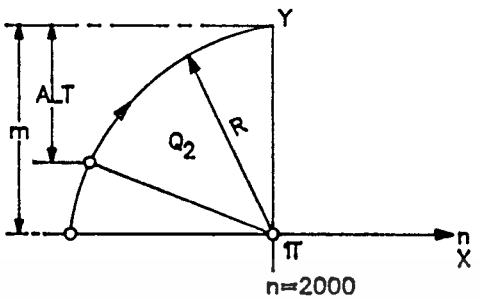
การกำเนิดส่วนโค้งต่าง ๆ สามารถสร้างอย่างง่ายๆ โดยการแบ่งส่วนโค้งออกเป็น 4 ส่วน คือ Q_1, Q_2, Q_3 และ Q_4 โดยจะถือส่วนโค้งที่สร้างง่ายที่สุดเป็นพื้นฐานในการสร้าง และส่วนโค้งต่าง ๆ ที่เหลือจะหาได้จากการแทนค่าเครื่องหมายแสดงการเคลื่อนที่ของแกนต่างๆ ที่กล่าวมาแล้วข้างต้น ในที่นี้ส่วนโค้งที่สร้างง่ายที่สุดคือ Q_2 ซึ่งสามารถสร้างได้โดยใช้ตาราง ค่าที่ได้จากตารางจะเป็นค่าสัมพัทธ์ ดังนั้น การกำเนิดส่วนโค้งจะเป็นการรวมค่าสัมพัทธ์เพื่อประกอบเป็นความสูงจากจุดเริ่มต้น ALP_X และ ALP_Y และเรียกค่าความสูงในแนวแกน x และ y ว่า ALT_X (Altitude) และ ALT_Y ตามลำดับ ค่าของ ALT_X และ ALT_Y จะหาได้จากสมการ

$$ALT_X = r \cdot [1 - \sum_{q=ALP_X}^{X_END} (\cos q\theta - \cos(q-1)\theta)]$$

$$ALT_Y = r \cdot [1 - \sum_{q=ALP_Y}^{Y_END} (\sin q\theta - \sin(q-1)\theta)]$$

และสามารถเขียนเป็นส่วนโค้งได้ดังรูปที่ 6.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.18 ส่วนโค้งที่สร้างขึ้น

ง. การหาจุดเริ่มต้นและจุดสิ้นสุดของส่วนโค้งของดาราง

การหาค่าของจุดเริ่มต้น ซึ่งอยู่ที่จุดใด.ๆ จะสามารถหาได้โดยการคำนวณความสูง และนำมาเปรียบเทียบกับค่าของ ALP_X และ/หรือ ALP_Y ซึ่งถือว่าเป็นตำแหน่งอ้างอิง จากสมการ

$$r = \sqrt{(I^2+J^2)}$$

โดยที่ r = รัศมีของส่วนโค้ง

I = จุดเริ่มต้นของส่วนโค้งบนแกน x ที่อ้างอิงจากจุดศูนย์กลางของส่วนโค้ง

J = จุดเริ่มต้นของส่วนโค้งบนแกน y ที่อ้างอิงจากจุดศูนย์กลางของส่วนโค้ง

ค่าของ I หรือ J จะเป็นศูนย์ได้เพียงค่าเดียว แต่อาจจะไม่เท่ากับศูนย์ทั้งสองค่าก็ได้ ถ้า I หรือ J ค่าใดค่าหนึ่งมีค่าเท่ากับศูนย์ แสดงว่าจุดเริ่มต้นของส่วนโค้งจะต้องอยู่ที่ค่าศูนย์ของแกนนั้น ๆ และมีรัศมีเท่ากับค่าของจุดเริ่มต้นที่ไม่เท่ากับศูนย์ในกรณีที่ค่าของ I และ/หรือ J ไม่เท่ากับศูนย์

อ้างถึง B = 2000 ณ จุดที่

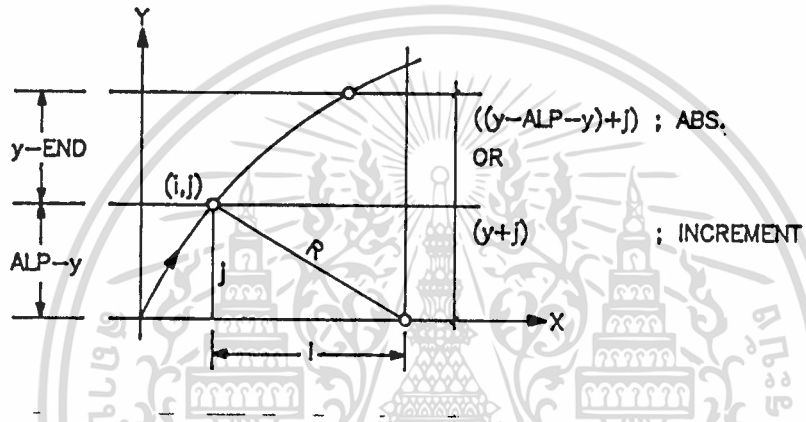
$$I = r\cdot[1-\sum_{q=1}^{2000}(\cos q\theta-\cos(q-1)\theta)]$$

และ/หรือ

$$J = r\cdot[1-\sum_{q=1}^{2000}(\sin q\theta-\sin(q-1)\theta)]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีค่าเป็นศูนย์ หรือใกล้เคียงศูนย์มากที่สุด (เริ่มมีค่าเป็นลบ) ค่าของ q ในขณะนั้นจะเป็นค่าที่ทำให้เกิดค่าของ ALP_X และ/หรือ ALP_Y ตามลำดับ หรือเป็นค่าของจุดเริ่มต้นของส่วนโค้งแรกในตาราง



รูปที่ 6.19 การหาจุดเริ่มต้นและจุดสิ้นสุดของส่วนโค้ง

ในกรณีที่ส่วนโค้งไม่ได้เริ่มต้นที่ศูนย์ นั่นคือ ALP_X และ ALP_Y จะอยู่ที่จุดใด ๆ กรณีนี้จะไม่สามารถใช้กฎเกณฑ์นี้ได้ ดังนั้นจึงต้องใช้เทคนิคประกอบ โดยการกำหนดจุดเริ่มต้นของส่วนโค้ง ให้มีจุดเริ่มต้นของส่วนโค้งเทียบกับ ALP_X และ ALP_Y ที่จุด I,J ซึ่งวิธีนี้สามารถใช้กับส่วนโค้งที่มีจุดศูนย์กลางความโค้งที่จุดใด ๆ ได้

จุดสิ้นสุดของส่วนโค้งหาได้จากการเปรียบเทียบค่าของ X_END และ Y_END ได้เช่นเดียวกันกับที่ได้กล่าวมา นั่นคือ

$$X_END = r\cdot[1-\sum_{q=ALP_X}^{2000}(\cos q\theta-\cos(q-1)\theta)]$$

และ/หรือ

$$Y_END = r\cdot[1-\sum_{q=ALP_Y}^{2000}(\sin q\theta-\sin(q-1)\theta)]$$

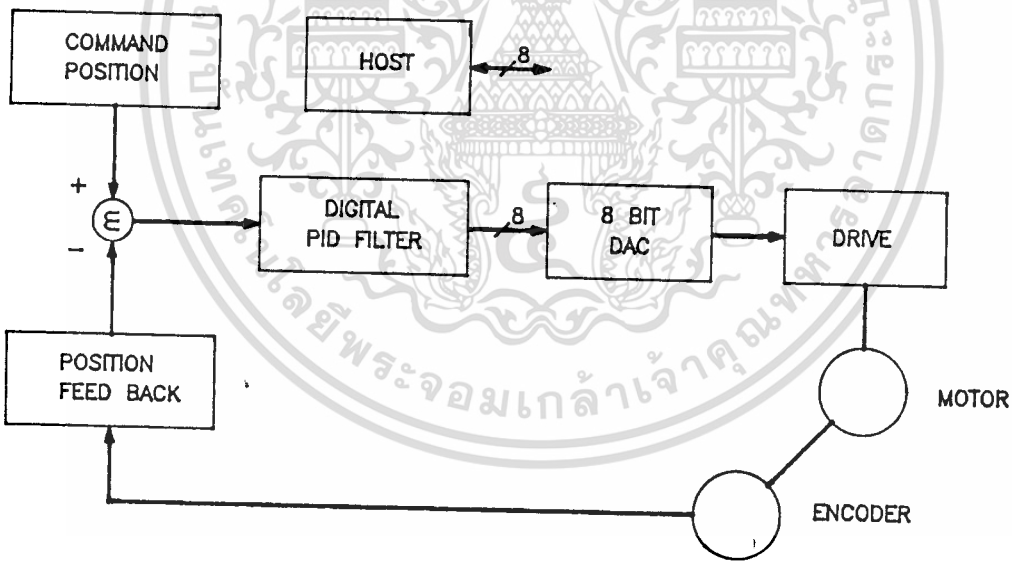
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

มีค่าเป็นศูนย์ หรือใกล้เคียงศูนย์มากที่สุด (เริ่มมีค่าเป็นลบ) ค่าของ q ในขณะนั้นจะเป็นค่าที่ทำให้เกิดค่าของ ALT_X และ/หรือ ALT_Y ตามลำดับ หรือเป็นค่าของจุดสิ้นสุดของส่วนโค้งแรกในตาราง อนึ่งในระหว่างที่สมการข้างต้นนี้ยังมีค่าเป็นบวก จะแสดงถึงว่าเครื่องจักรกำลังอยู่ในขั้นตอนการทำงานจากจุดเริ่มต้นไปยังจุดสุดท้ายของส่วนโค้งนั้น

6.2 หน่วยควบคุมตำแหน่ง (Position Control)

หน่วยควบคุมตำแหน่งจะมีการทำงานแยกเป็นอิสระจากหน่วยคอมพิวเตอร์กลาง กล่าวคือ หน่วยควบคุมตำแหน่งจะใช้วงจรรวมที่ออกแบบมาเฉพาะ เพื่อใช้ในการควบคุมความเร็ว และตำแหน่ง ซึ่งจะประกอบด้วยหน่วยคำนวณและหน่วยที่เป็นศูนย์กลางในการทำงานได้แก่ การแปลคำสั่งที่ได้รับจากหน่วยคอมพิวเตอร์กลาง คำสั่งต่าง ๆ จะเป็นคำสั่งที่เกี่ยวกับการควบคุม และการปรับพารามิเตอร์ของระบบ และแสดงได้ในตารางที่ 6.3

ในทางปฏิบัติ วงจรรวมที่ใช้เป็นหน่วยควบคุมตำแหน่ง สามารถแสดงได้ด้วยบล็อกไดอะแกรม ดังรูปที่ 6.20



รูปที่ 6.20 บล็อกไดอะแกรมของหน่วยควบคุมตำแหน่งที่เป็นวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command	Type	Description	Hex	Data Bytes	Note
RESET	init	reset LM628	00h	none	1
PORT8	init	select 8-bit output	05h	none	2
PORT12	init	select 12-bit output	06h	none	2
DFH	init	define home	02h	none	1
SIP	init	set index position	03h	none	1
LPEI	intr	interrupt on error	1Bh	two	1
LPES	intr	stop on error	1Ah	two	1
SBPA	intr	set breakpoint, abs	20h	four	1
SBPR	intr	set breakpoint, rel	21h	four	1
MSKI	intr	mask interrupts	1Ch	two	1
RSTI	intr	reset interrupts	1Dh	two	1
LFIL	filt	load filter params	1Eh	2 to 10	1
UDF	filt	update filter	04h	none	1
LTRJ	traj	load trajectory	1Fh	2 to 14	3
STT	traj	start motion	01h	none	1
RDSTAT	report	read status byte	none	one	1,4
RDSIGS	report	read signals reg	0Ch	two	1
RDIP	report	read index position	09h	four	1
RDDP	report	read desired pos	08h	four	1
RDRP	report	read real position	0Ah	four	1
RDDV	report	read desired vel	07h	four	1
RDRV	report	read rel velocity	0Bh	two	1
RDSUM	report	read integration sum	0Dh	two	1

Note 1: Commands may be executed "on the fly" during motion.
Note 2: Commands not applicable to execution during motion.
Note 3: Command may be executed during motion if acceleration parameter has not been changed.
Note 4: Command needs no code because the command port status-byte read is totally supported by hardware.

ตารางที่ 6.3 คำสั่งของวงจรรวม LM628

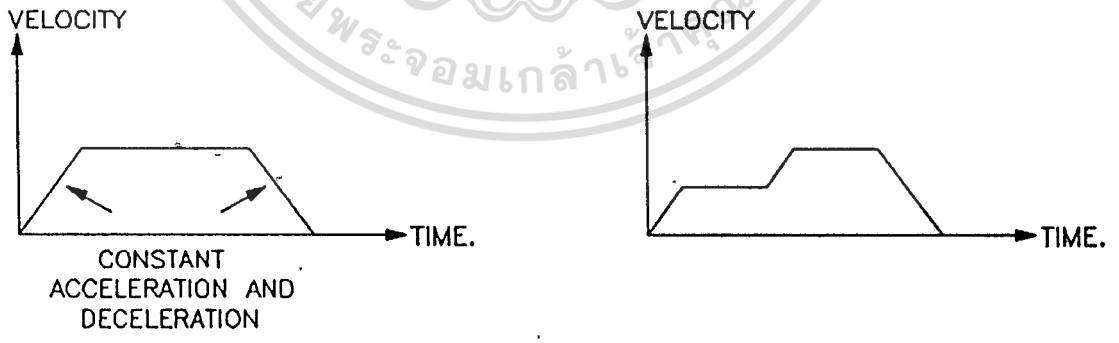
จากรูป แสดงถึงระบบเซอร์โวที่ถูกบรรจุอยู่ในวงจรรวมหมายเลข 628 การติดต่อกับวงจรรวมสำหรับควบคุมตำแหน่ง สามารถติดต่อกับหน่วยคอมพิวเตอร์กลางที่ใช้เป็นศูนย์กลางการควบคุมโดยผ่านทางหน่วยอินพุทเอาท์พุท โดยหน่วยคอมพิวเตอร์กลางจะทำการโปรแกรมค่าของความเร็ว ค่าชดเชยเกี่ยวกับวงจรกรองแบบดิจิตอล (Digital Filter) เอาท์พุทของวงจรรวมที่ใช้ในการควบคุมตำแหน่งนี้จะถูกต่อกับวงจรรวมสำหรับเปลี่ยนสัญญาณรหัสตัวเลขฐานสองเป็นสัญญาณไฟฟ้าแบบอนาล็อก เพื่อป้อนให้กับระบบขับเคลื่อน เรียกว่าสัญญาณควบคุมความเร็ว (Speed Command) ระบบขับเคลื่อนจะขับกำลังงานไฟฟ้าเข้าสู่มอเตอร์ เป็นผลให้เพลาของมอเตอร์ที่มีอุปกรณ์เข้ารหัส (Encoder) เป็นหน่วยป้อนกลับ นำสัญญาณป้อนกลับเข้าสู่วงจรรวมเพื่อควบคุมตำแหน่ง หรือความเร็วของการหมุนของเพลามอเตอร์ตามค่าที่ถูกโปรแกรมจากหน่วยคอมพิวเตอร์กลางที่เป็นศูนย์การควบคุม การควบคุมตำแหน่งหรือความเร็วที่กล่าวมาแล้วนี้เป็นการควบคุมแบบลูปปิด ขีดจำกัดในการทำงานของวงจรรวมหมายเลข 628 แสดงดังตารางที่ 6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Position Range	-1,073,741,824 to 1073,741,824
Velocity Range :	(0 to 1,073,741,824)/2^{16} Count/Sample
Acceleration Range	(0 to 1,073,741,824)/2^{16} Count/Sample
Moter Drive Output	8, 12 bit Parllel output to DAC
Feedback Device	Incremental Optical Encoder
Control Algorithm	PID

ตารางที่ 6.4 แสดงขีดจำกัดการทำงานของวงจรรวมหมายเลข 628

การควบคุมตำแหน่งของวงจรรวมหมายเลข 628 สามารถจะจำกัดความเร็วในการเข้าสู่ตำแหน่งที่กำหนดได้ โดยจะต้องกำหนดความเร็วและจุดหมายของตำแหน่ง รวมทั้งความเร็วของการเคลื่อนที่ ให้กับวงจรรวม วงจรรวมจะทำการคำนวณ และควบคุมการเคลื่อนที่ให้เป็นไปตามค่ากำหนดรูปที่ 6.21 แสดงถึงความสัมพันธ์ของความเร็วในการเคลื่อนที่ต่อระยะเวลา



รูปที่ 6.21 ความสัมพันธ์ระหว่างความเร็วต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ตัวอย่างเช่น ถ้าต้องการเคลื่อนตำแหน่งเชิงการหมุนของเพลามอเตอร์ไป 100 รอบ ด้วยความเร็ว 600 รอบต่อนาที ความละเอียดของอุปกรณ์เข้ารหัส 500 พัลซ์ (Pulse) ต่อการหมุน 1 รอบ ตัวแปรในการเคลื่อนที่สำหรับวงจรรวมหมายเลข 628 สามารถจะคำนวณได้ดังนี้

$$P = D \cdot N \quad \text{Count}$$

โดยที่ P = ค่าของตำแหน่งควบคุมที่จะป้อนให้กับวงจรรวมหมายเลข 628 มีค่าเป็นการนับของอุปกรณ์เข้ารหัส

D = ค่าความละเอียด มีค่าเป็นสี่เท่าของจำนวนพัลซ์ต่อรอบ

N = จำนวนรอบของการหมุนของเพลา

นั่นคือ

$$P = 500 \cdot 4 \cdot 100 \quad \text{Count}$$
$$= 200{,}000 \quad \text{Count}$$

และ

$$V = D \cdot T \cdot C \cdot Rev \quad \text{Count/Sample time}$$

โดยที่ V = ค่าของความเร็วรอบที่จะป้อนให้กับวงจรรวมหมายเลข 628 มีหน่วยเป็น Count/Sample time

T = Sample Time มีหน่วย 256 s ที่สัญญาณนาฬิกา 8 MHz

C = ค่าสำหรับแปลงหน่วย มีค่า 1 นาที/60 วินาที

Rev = ความเร็วรอบต่อนาทีของเพลามอเตอร์

นั่นคือ

$$V = \frac{2000 \cdot 256 \cdot 10^{-6} \cdot 600}{60} \quad \text{Count/Sample}$$
$$= 5.12 \quad \text{Count/Sample}$$

และค่าที่จะอ่านได้จากวงจรรวม คือ

$$V = 5.12 \cdot 65536$$
$$= 335544.32$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$A = D \cdot T \cdot Acc$ Count/Sample^2

โดยที่ A = ค่าความเร่งของการเคลื่อนที่ ที่ต้องการจะป้อนให้กับวงจรรวมหมายเลข 628 มีหน่วยเป็น Count/Sample^2

Acc = ค่าความเร่งของเพลามอเตอร์ที่ต้องการ มีหน่วยเป็นรอบ/วินาที^{2}

= 1 รอบ/วินาที^{2}

นั่นคือ

$A = 2000 \cdot 256 \cdot 10^{-6} \cdot 1$

$= 0.512$ Count/Sample^2

และค่าที่จะอ่านได้จากวงจรรวม คือ

$A = 0.512 \cdot 65536$

$= 33554.432$

ค่าที่ใส่ให้กับวงจรรวมในส่วนของความเร่ง สำหรับกำหนดความเร่ง คือ

$A = 33554$

จากตัวอย่างที่กล่าวมานี้ ค่าจากการคำนวณจะต้องถูกเปลี่ยนเป็นเลขฐานสองก่อนที่จะป้อนให้กับวงจรรวมหมายเลข 628 สำหรับเลขหลังจุดทศนิยมจะไม่สามารถป้อนให้กับ วงจรหมายเลข 628 ได้ ค่าพารามิเตอร์สำหรับวงจรกรองแบบดิจิตอลรวมหมายเลข 628 จะใช้วิธีการของ พีไอดี (PID : Proportional+Integral+Derivative) ในการชดเชยลูปการควบคุม กำลังงานจะถูกขับเข้ามอเตอร์อย่างเป็นสัดส่วนกับค่าความคลาดเคลื่อน รวมกับค่าของการ Integral และ Derivative ของค่าความคลาดเคลื่อนจากตำแหน่ง ซึ่งแสดงดังสมการต่อไปนี้

$$U(n) = K_P \cdot e(n) + K_i \cdot T \cdot \sum_{x=0}^{n} e(x) + \frac{K_d}{T_d} \cdot (e(n') - e(n'-1))$$

โดยที่

U(n) = สัญญาณควบคุมมอเตอร์

e(n) = ค่าความคลาดเคลื่อนจากตำแหน่งอ้างอิงที่กำหนดขึ้น

T = คาบเวลาของการ Sampling ของส่วน Proportional และ Integral มีค่าเป็น 250 μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

$U(n)$ = สัญญาณควบคุมมอเตอร์

$e(n)$ = ค่าความคลาดเคลื่อนจากตำแหน่งอ้างอิงที่กำหนดขึ้น

T = คาบเวลาของการ Sampling ของส่วน Proportional และ Integral มีค่าเป็น 250 μs ที่สัญญาณนาฬิกา 8 MHz

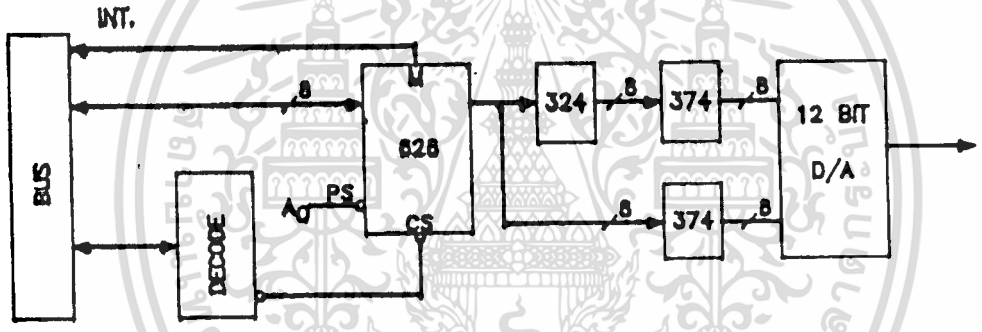
T_d = คาบเวลาของการ Sampling สำหรับ Derivative ที่ถูกกำหนดโดยผู้ควบคุม

n' = คือการ Sampling ขึ้นมาในคาบเวลา ของ Derivative Rate

K_p = Discrete-time filter parameter ของส่วน Proportional ที่ถูกกำหนดโดยผู้ควบคุม

$K_i \cdot T$ = Discrete-time filter parameter ของส่วน Integral ที่ถูกกำหนดโดยผู้ควบคุม

$\frac{K_d}{T_d}$ = Discrete-time filter parameter ของส่วน Derivative ที่ถูกกำหนดโดยผู้ควบคุม



รูป 6.22 แสดงการเชื่อมต่อหน่วยคอมพิวเตอร์กลางเข้ากับวงจรรวมหมายเลข 628

หน่วยคอมพิวเตอร์กลางจะเปรียบเสมือนหน่วยสั่งการให้กับวงจรรวม 628 เพื่อควบคุมตำแหน่งหรือความเร็ว โดยที่หน่วยคอมพิวเตอร์กลางจะติดต่อกับวงจรรวมหมายเลข 628 ผ่านทางพอร์ท (Port) ซึ่งประกอบด้วยพอร์ทที่สำคัญสองพอร์ทคือ 1. พอร์ทที่ใช้เป็นทางเข้าของคำสั่งสำหรับวงจรรวมหมายเลข 628 และยังเป็นพอร์ทที่ใช้อ่านสัญญาณบอกสภาวะได้ด้วย โดยที่สัญญาณบอกสภาวะจะบอกเป็นสภาวะแบบตรรก มีขนาดของข้อมูล 8 บิท แต่ละบิทจะบอกความหมายของสภาวะต่างๆ ดังตารางที่ 6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. พอร์ทที่ใช้เป็นทางเข้าออกของข้อมูล ได้แก่ข้อมูลเกี่ยวกับตำแหน่งของการควบคุมความเร็วตลอดจนค่าของตัวแปรสำหรับวงจรกรองพีไอดี พอร์ททั้งสองนี้จะถูกแยกออกจากกันด้วยสัญญาณ PS เมื่อ PS มีค่าเป็น 0 แสดงว่าเป็นพอร์ทคำสั่ง ถ้า PS มีค่าเป็น 1 แสดงว่าพอร์ทจะเป็นทางเข้าออกข้อมูล รูปที่ 6.23 แสดงถึงการเชื่อมต่อวงจรรวมหมายเลข 628 เข้ากับหน่วยคอมพิวเตอร์กลางที่ได้ออกแบบขึ้น

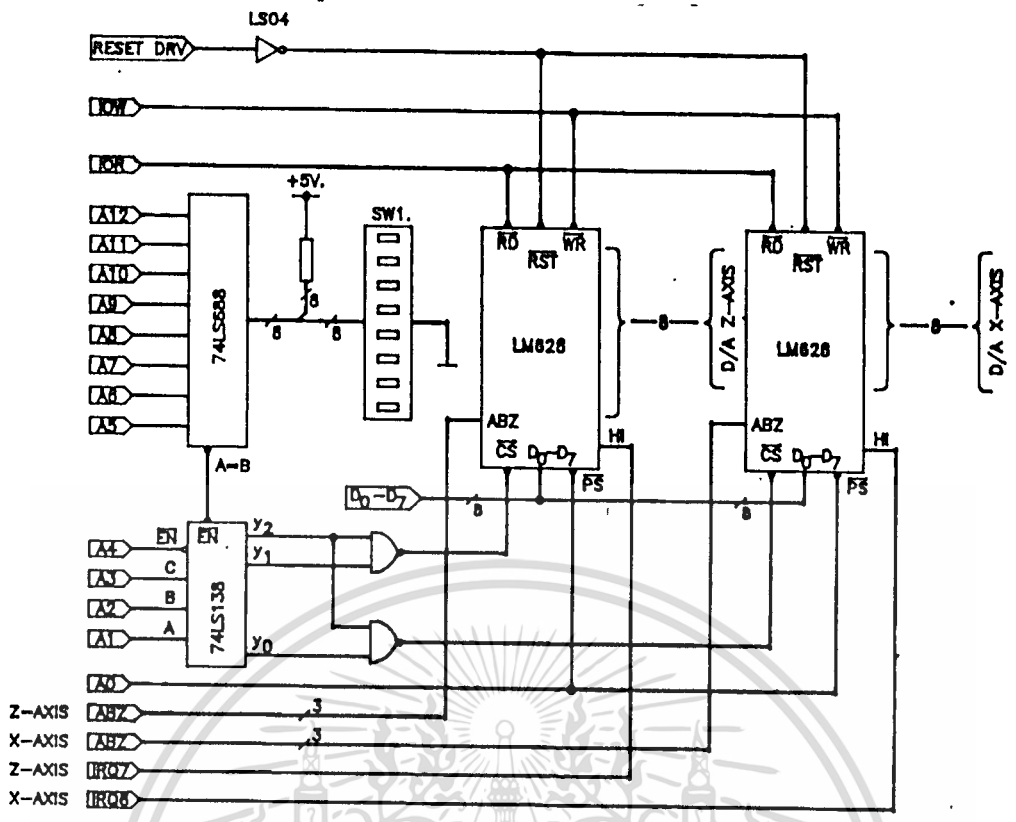
bit position	ฟังก์ชัน
bit 7	Motor Off
bit 6	Breakpoint Reached
bit 5	Excessive Position Error
bit 4	Wrap Round Occurend
bit 3	Index Pulse Observed
bit 2	Trajectory Complete
bit 1	Command Error
bit 0	Bussy bit

ตารางที่ 6.5 แสดงสภาวะของบิทต่างๆ

การกำหนดค่าต่างๆ สำหรับวงจรรวมหมายเลข 628 ที่ได้จัดทำขึ้น สามารถแสดงได้ดังโฟลชาร์ท (Flow Chart) ในรูปที่ 6.24 และรูปที่ 6.25 ค่าของความเร็วจะถูกโปรแกรมลงในวงจรรวมหมายเลข 628 ผ่านทางหน่วยคอมพิวเตอร์กลาง เมื่อวิถีการเคลื่อนที่ผิดพลาดไป ค่าความคลาดเคลื่อนที่เกิดขึ้นจะทำให้วงจรรวมหมายเลข 628 ทำการอินเทอร์รัพท์ (Interrupt) เพื่อให้หน่วยคอมพิวเตอร์กลางทำการแก้ไข หรือสั่งให้มอเตอร์ขับแกนหยุดการเคลื่อนที่ทันที

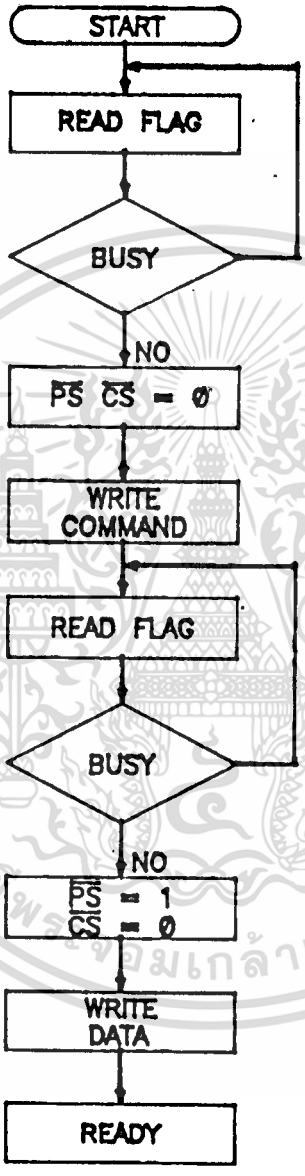
การทดสอบความถูกต้องในการทำงาน สามารถตรวจสอบได้โดยการนำส่วนที่ได้ออกแบบ และจัดสร้างขึ้นมาทำการทดสอบบนเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่นเทียบเท่า IBM AT โดยการควบคุมตำแหน่งแบบ Manual โดยมีลักษณะการทำงานแสดงได้ดังรูปที่ 6.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



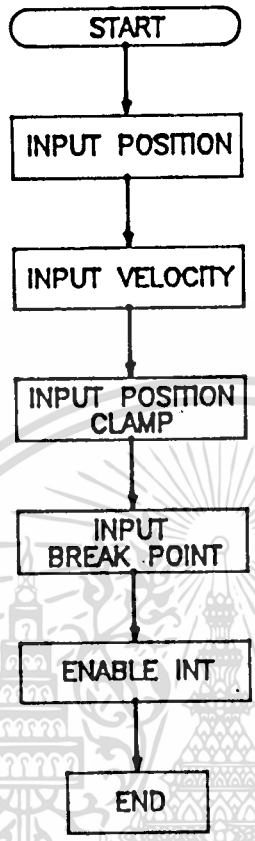
รูปที่ 6.23 แสดงถึงการเชื่อมต่อวงจรรวมหมายเลข 628
เข้ากับหน่วยคอมพิวเตอร์กลางที่ได้ออกแบบขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.24 โฟลชาร์ทการโปรแกรมวงจรรวมหมายเลข 628

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



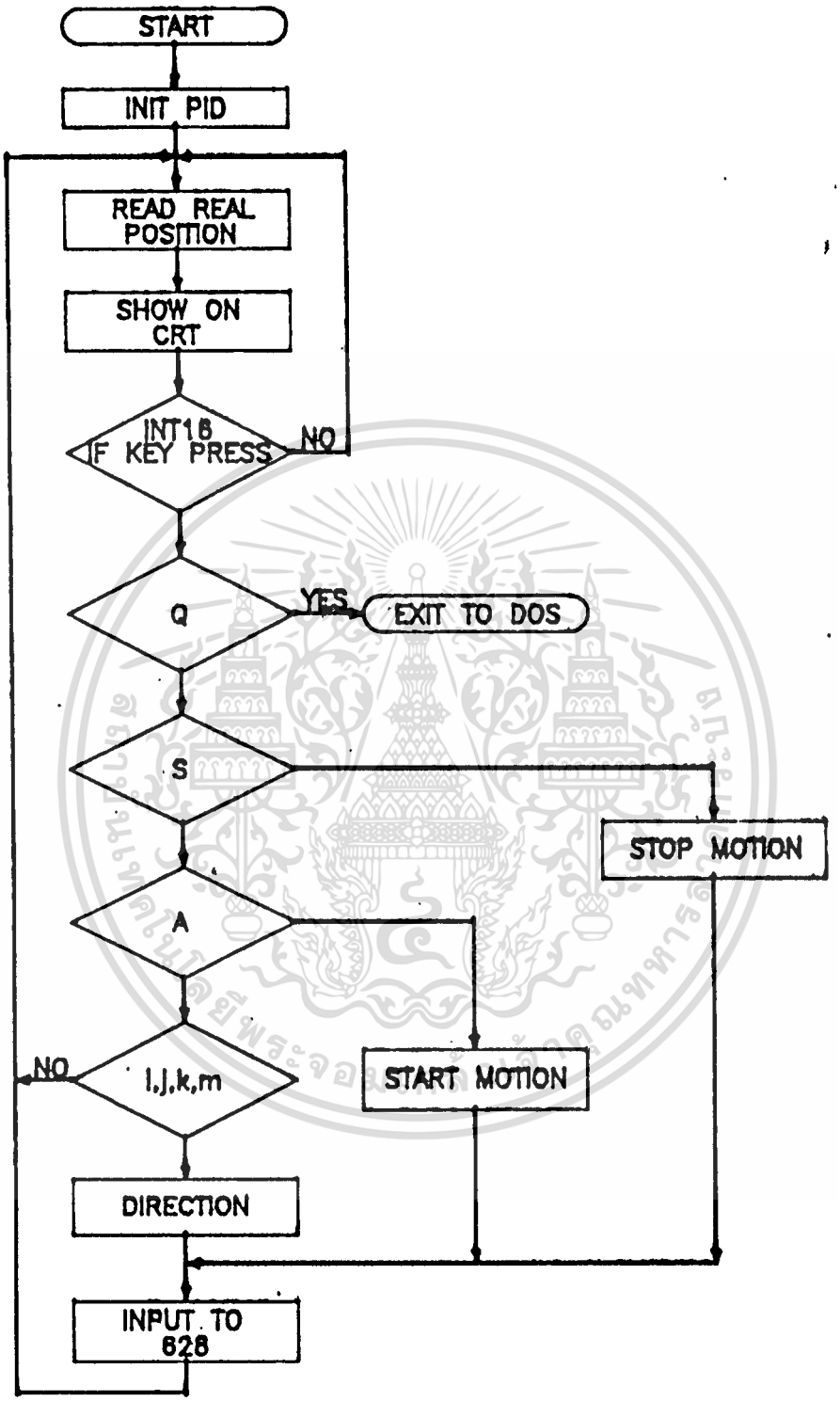
รูปที่ 6.25 โฟลชาร์ทแสดงการทดสอบการทำงานของวงจรรวมหมายเลข 628

6.3 หน่วยแสดงผล

หน่วยแสดงผลจะใช้แสดงค่าสภาวะการทำงานของ CNC เช่น ตำแหน่งของแกนเครื่องจักรสภาวะทางตรรกของระบบ ตลอดจนข้อความต่างๆ เพื่อสื่อความหมายสำหรับผู้ปฏิบัติงาน โดยการนำเสนอทางจอภาพ ในที่นี้จะใช้ฮาร์ดแวร์ของระบบเชื่อมต่อจอภาพของคอมพิวเตอร์ส่วนบุคคลที่เทียบเท่า PC XT ของบริษัท IBM ส่วนประกอบของหน่วยแสดงผลทางจอภาพแสดงได้ดังบล็อกไดอะแกรมในรูปที่ 6.27

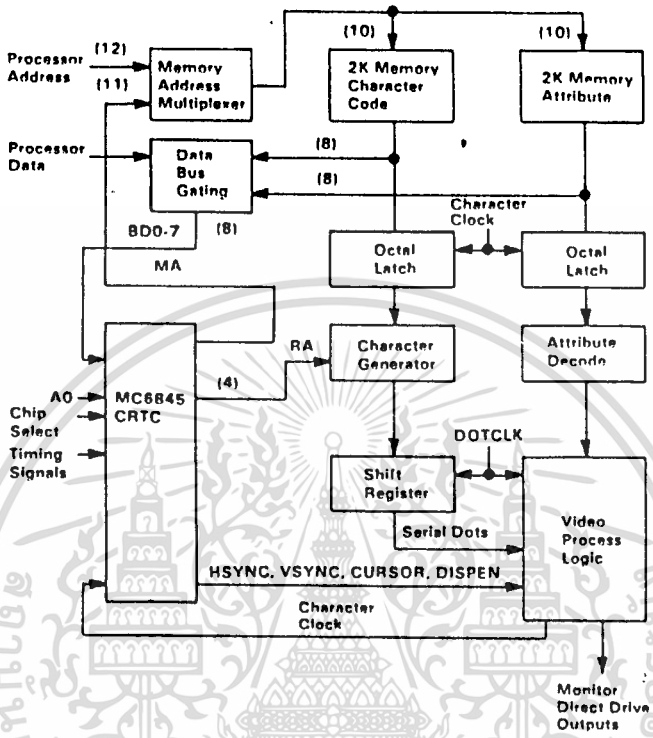
หน่วยแสดงผลทางจอภาพสามารถทำงานได้โดยอิสระ โดยการป้อนข้อมูลที่เป็นรหัส ASCII ตั้งแต่ตำแหน่งที่ B060H เข้าไป หลักนั้นฮาร์ดแวร์ของหน่วยแสดงผลทางจอภาพจะทำหน้าที่แสดงผลออกมาทางจอภาพตามข้อมูลคำสั่งที่ถูกป้อนเข้าไป โดยมีกฎเกณฑ์ต่างๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.26 โฟลชาร์ทแสดงการทดสอบการควบคุมตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.27 บล็อกไดอะแกรมของหน่วยแสดงผล

1. โปรแกรมพารามิเตอร์สำหรับวงจรรวมหมายเลข 6845 ที่อยู่บนฮาร์ดแวร์ของหน่วยเชื่อมต่อจอภาพ ตั้งแต่ตำแหน่ง 3B4H และ 3B5H โดยที่ตำแหน่ง 3B4H จะกำหนดหมายเลขรีจิสเตอร์สำหรับตำแหน่ง 3B5H จะใช้เป็นพอร์ทสำหรับผ่านข้อมูล จากหน่วยคอมพิวเตอร์กลางเข้าสู่รีจิสเตอร์ที่ถูกเลือก ณ ตำแหน่ง 3B4H ข้อมูลต่างๆ ที่ป้อนลงในรีจิสเตอร์จะแสดงในตารางที่ 6.6

2. ป้อนข้อมูลที่ต้องการแสดงออกทางจอภาพด้วยรหัส ASCII เริ่มตั้งแต่ตำแหน่ง 0B000H เป็นต้นไป และมีลักษณะการจัดเรียงข้อมูลต่างๆ ดังแสดงในรูปที่ 6.28 การป้อนข้อมูลที่เป็นรหัส ASCII ลงไปในหน่วยความจำของหน่วยแสดงผลทางจอภาพ จะประกอบด้วยข้อมูล 2 ประเภทคือ

ก. ข้อมูลที่เป็นรหัสของตัวอักษร A-Z และสัญญลักษณ์ต่างๆ รวม 256 รหัส เป็น ASCII อยู่ในตำแหน่งเลขคี่

ข. ข้อมูลบอกลักษณะการแสดงผลบนจอภาพ ซึ่งประกอบด้วยรหัสต่างๆ แสดงดังตารางที่ 6.7 โดยจะอยู่ในตำแหน่งเลขคู่

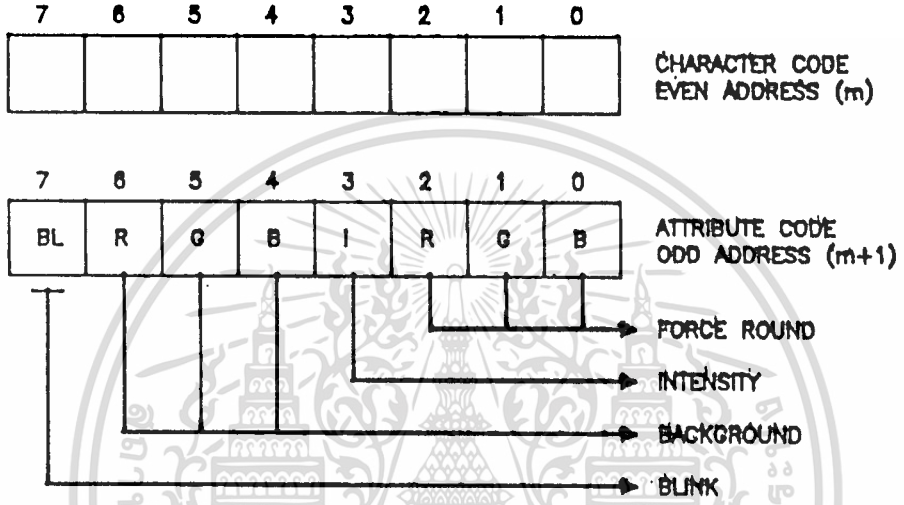
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Number	Register File	Program Unit	IBM Monochrome Display (Address in hex)
R0	Horizontal Total	Chracters	61
R1	Horizontal Displayed	Chracters	50
R2	Horizontal Sync Position	Chracters	52
R3	Horizontal Sync Width	Characters	0F
R4	Vertical Total	Character Rows	19
R5	Vertical Total Adjust	Scan Line	06
R6	Vertical Displayed	Character Row	19
R7	Vertical Sync Position	Character Row	19
R8	Interlace Mode	----------	02
R9	Maximum Scan Line Address	scan Line	0D
R10	Cursor Start	scan Line	0B
R11	Cursor End	scan Line	0C
R12	Start Address (H)	----------	00
R13	Start Address (L)	----------	00
R14	Cursor (H)	----------	00
R15	Cursor (L)	----------	00
R16	Reserved	----------	---
R17	Reserved	----------	---

ตารางที่ 6.6 ข้อมูลต่าง ๆ ที่ป้อนให้กับรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้หน่วยแสดงผลทางจอภาพ จะต้องมีการใส่ค่าเริ่มต้น ให้กับหน่วยความจำของหน่วยแสดงผลทางจอภาพ เนื่องจากหน่วยความจำบนหน่วยแสดงผลทางจอภาพเป็นหน่วยความจำชนิดอ่านและเขียนได้ซึ่งเมื่อไม่มีแรงดันไฟฟ้าจ่ายให้ ข้อมูลทั้งหมดที่มีอยู่จะสูญหายไป ดังนั้นจึงจำเป็นต้องตั้งค่าเริ่มต้นทุกครั้งเมื่อเริ่มใช้หน่วยแสดงผลทางจอภาพ ขั้นตอนการตั้งค่าเริ่มต้นจะแสดงได้ด้วยโฟลชาร์ทในรูปที่ 6.29

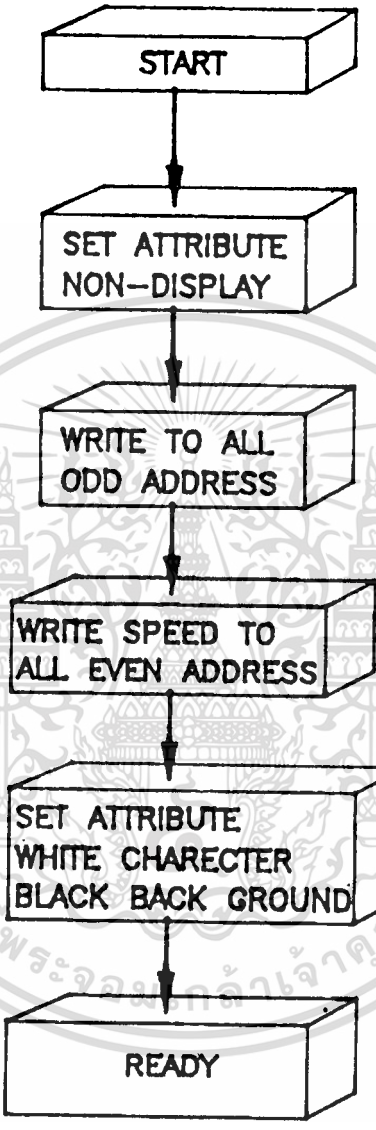


รูปที่ 6.28 ลักษณะการจัดเรียงข้อมูลต่างๆ ในหน่วยแสดงผลทางจอภาพ

Background R G B	Foreground R G B	ฟังก์ชันในแต่ละแอทริบิว (Attribute)
0 0 0	0 0 0	Non-Display
0 0 0	0 0 1	Underline
0 0 0	1 1 1	White Character/Background
1 1 1	0 0 0	Reverse Video

ตารางที่ 6.7 ลักษณะของการแสดงผลทางจอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.29 แสดงลำดับขั้นตอนการตั้งค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างตำแหน่งของตัวอักษรหรือสัญญลักษณ์บนจอภาพ สำหรับหน่วยแสดงผลทางจอภาพสามารถคำนวณหาได้จากลักษณะการวางข้อมูลของหน่วยแสดงผลทางจอภาพ โดยใช้รหัส ASCII 80 ไบท์ต่อหนึ่งบรรทัด รวม 24 บรรทัด ตำแหน่งแรกของข้อมูล (0B001H) จะอยู่ทางมุมซ้ายบนของจอภาพ และตำแหน่งของตัวอักษรจะนับเรียงจากซ้ายไปขวา

ตำแหน่งของตัวอักษร = $2 \cdot R \cdot C + 2K$

และ

ตำแหน่งของแอทริบิว = $R \cdot (2C-1) + (2K-1)$

โดยที่ R = จำนวนบรรทัดก่อนถึงบรรทัดที่ต้องการแสดงผลทางจอภาพ

C = จำนวนตัวอักษรต่อบรรทัด

K = ตำแหน่งตัวอักษรที่ต้องการแสดงที่นับจากซ้ายไปขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

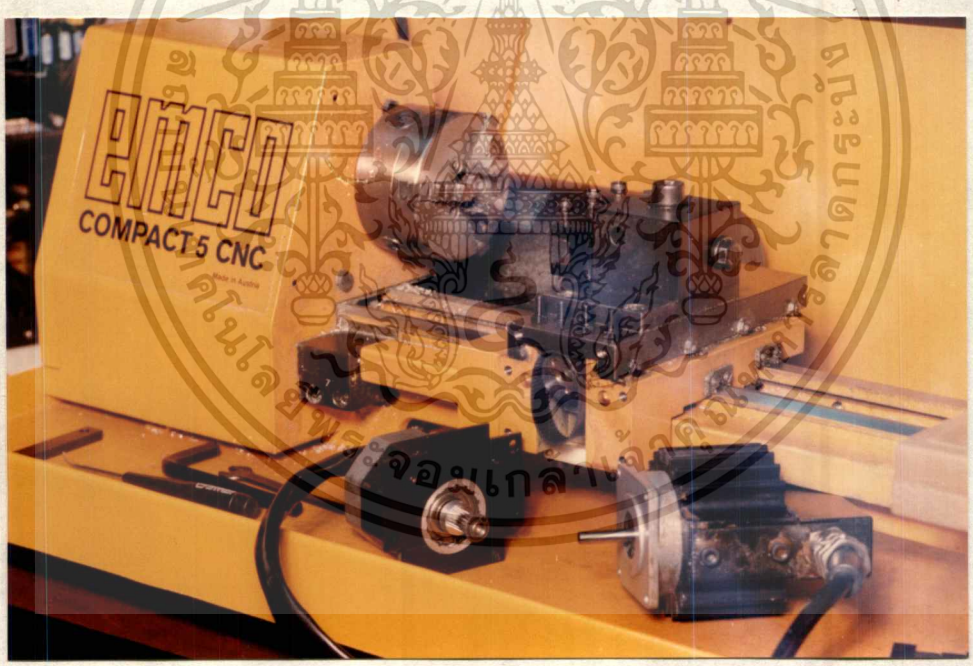
บทที่ 7
การทดสอบการทำงานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์

7.1 ขั้นตอนเบื้องต้นก่อนการทดสอบการกลึง

โดยการนำเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ได้ออกแบบและพัฒนาขึ้น มาประยุกต์ใช้กับเครื่องกลึงที่มีอยู่ภายในภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม และจะใช้เป็นกรณีศึกษาเพราะไม่สามารถจัดหาเครื่องจักรที่ใช้ในงานอุตสาหกรรมตลอดจนอุปกรณ์อื่นมาใช้ในการทดลอง ดังนั้น ในขั้นแรกจึงต้องทำการดัดแปลงอุปกรณ์ขับเคลื่อนแกน เพื่อให้สามารถต่อเข้ากับแกนกลึงขนาดเล็ก และอุปกรณ์อื่นๆ คือ

1. การดัดแปลงอุปกรณ์ขับเคลื่อนแกนเครื่องจักรโดยการเปลี่ยนมอเตอร์

เนื่องจากมอเตอร์ขับเคลื่อนแกนที่ใช้อยู่เป็นมอเตอร์แบบสเต็ปปิงมอเตอร์ธรรมดา จึงได้ทำการเปลี่ยนเป็นสเต็ปปิงมอเตอร์ชนิดที่มีความละเอียดสูง และปรับความละเอียดไว้ที่ 1000 สเต็ปต่อการหมุนของมอเตอร์ 1 รอบ (1000 Step/Revolution) และเพราะว่ามอเตอร์ทั้งสองมีเพลาขนาดเดียวกัน ดังนั้น อัตราทดของร่องสายพานแบบฟันยังคงเป็นค่าเดิมทั้งแกน X และแกน Z

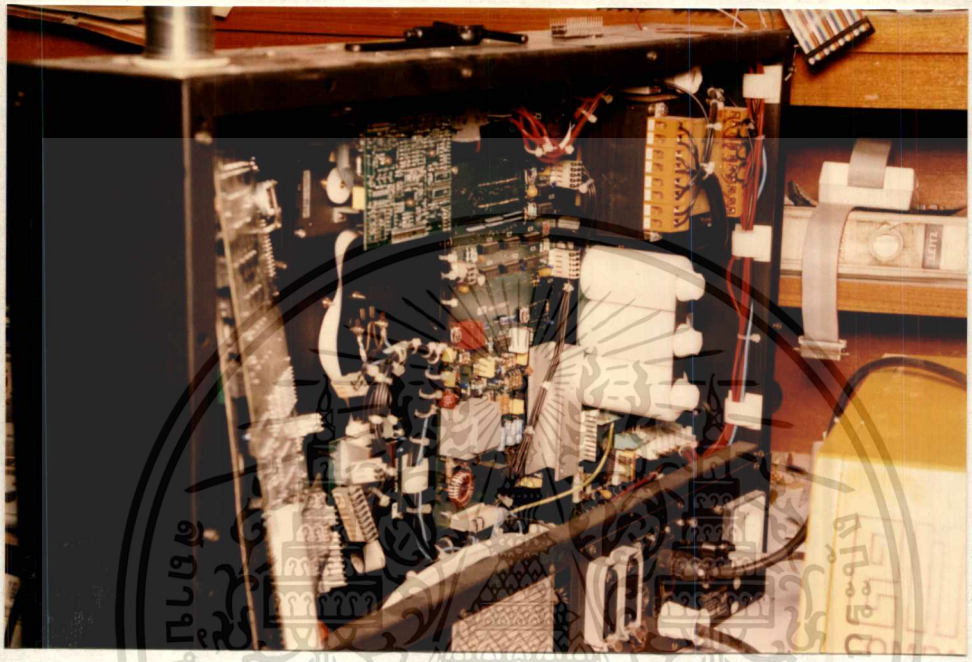


รูปที่ 7.1 การดัดแปลงอุปกรณ์ขับเคลื่อนแกนเครื่องจักรโดยการเปลี่ยนมอเตอร์ให้เป็นแบบที่มีความละเอียดสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ต่อระบบขับเคลื่อนหน้าจานของเครื่องกลึงเข้ากับระบบควบคุมเดิม

ในส่วนของระบบขับเคลื่อนหน้าจานของเครื่องกลึงนั้น จะใช้ระบบขับเคลื่อนเดิมที่มีอยู่ ทั้งนี้เพื่อหลีกเลี่ยงการดัดแปลงระบบไฟฟ้าในส่วนนี้ของระบบควบคุมเดิม ซึ่งอาจจะเกิดความเสียหายจากการดัดแปลงอุปกรณ์ป้อนกลับความเร็วและขั้วต่อต่าง ๆ ได้



รูปที่ 7.2 แสดงการต่อระบบขับเคลื่อนหน้าจานของเครื่องกลึงเข้ากับระบบควบคุมเดิม

3. ปลดเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ของระบบควบคุมเดิม

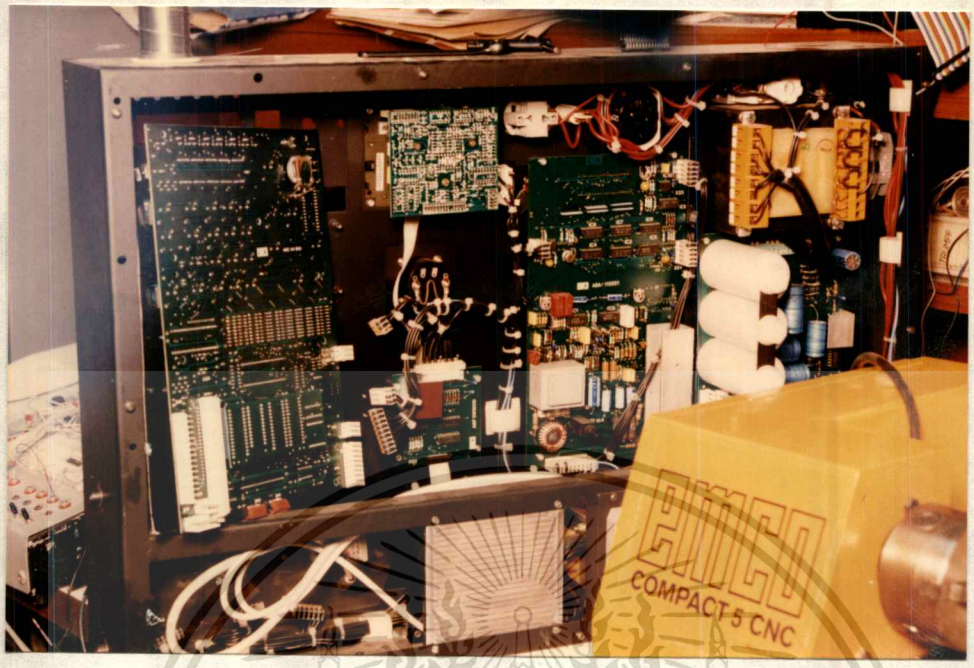
เมื่อนำเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ได้ออกแบบและพัฒนาขึ้นมาต่อกับเครื่องกลึงเป็นที่เรียบร้อย จึงได้ทำการปลดเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ของระบบควบคุมเดิมออก เพื่อป้องกันการชำรุดเสียหายอันอาจจะเกิดจากความผิดพลาดในขณะทำการทดสอบการทำงาน

4. ต่อสัญญาณอินพุทของระบบขับเคลื่อนแกน

ขั้นตอนต่อไปได้ทำการต่อสัญญาณอินพุทของระบบขับเคลื่อนแกน เข้ากับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่สร้างขึ้น สัญญาณอินพุทนี้ประกอบด้วยสัญญาณต่าง ๆ คือ

- สัญญาณอินาเบิล (Enable) ของระบบขับเคลื่อนแกน
- สัญญาณบอกทิศทางการหมุนของมอเตอร์
- สัญญาณบอกการเคลื่อนที่ โดยสัญญาณนี้จะเป็นสัญญาณนาฬิกาซึ่งมอเตอร์จะเคลื่อนที่ 1 สเต็ปต่อสัญญาณนาฬิกา 1 ลูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.3 การเคลื่อนย้ายเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ของระบบเดิมออก
เพื่อป้องกันความเสียหายขณะทำการทดสอบ

7.2 การทดสอบการกลึง

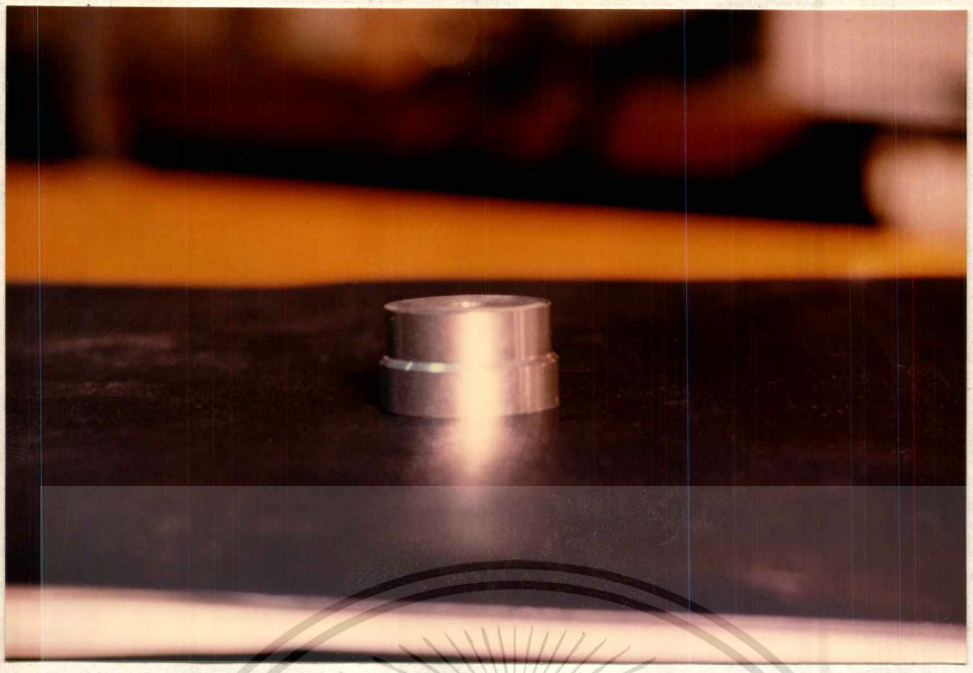
เมื่อได้เตรียมการเป็นที่เรียบร้อยแล้ว จึงได้เริ่มทำการทดสอบการกลึง โดยการใช้ฮาร์ดแวร์ประกอบของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ได้สร้างขึ้น ต่อเข้ากับเครื่องคอมพิวเตอร์ส่วนบุคคลรุ่น IBM AT มีความเร็วสัญญาณนาฬิกา 8 MHz เพื่อความสะดวกในการขนย้ายอุปกรณ์ประกอบอื่นๆ ตลอดจนการแก้ไขโปรแกรมเมื่อเกิดการผิดพลาด

ลักษณะการกลึงที่ทำการทดสอบแบ่งออกเป็น 3 แบบคือ

7.2.1 การกลึงในแนวตรงแบบซิงโครไนซ

เป็นการกลึงที่ยกระดับขึ้นเป็นแนวลาดเอียง โดยใช้คมตัดแบบตัดงานตรง เพื่อทดสอบความราบเรียบของการเคลื่อนที่ของความเร็วเวคเตอร์ ชิ้นงานที่ได้จากการกลึงนี้แสดงได้ดังรูปที่ 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 ชิ้นงานที่ได้จากการทดสอบการกลึงในแนวตรง

7.2.2 การกลึงในแนวโค้งแบบซิงโครไนซ

เป็นการกลึงในลักษณะเดียวกับการกลึงที่ได้กล่าวมาข้างต้น แต่เปลี่ยนจากแนวตรงเป็นแนวโค้งเพื่อทดสอบความราบเรียบของความเร็วของคมตัด ซึ่งเป็นความเร็วเวคเตอร์ ชิ้นงานที่ได้จากการทดสอบการกลึงในแนวโค้งนี้สามารถแสดงได้ดังรูปที่ 7.5

รูปที่ 7.5 ชิ้นงานที่ได้จากการทดสอบการกลึงในแนวโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2.3 การกลึงในแนวโค้งสัมผัสกับแนวโค้งที่ไม่มีการชดเชยค่ารัศมีของคมตัด

การกลึงในลักษณะนี้ เป็นการกลึงเพื่อให้ทราบถึงผลของรัศมีของคมตัดที่มีต่อผิวงาน ในกรณีที่มีการเปลี่ยนแปลงของแนวการกลึงเป็นเส้นโค้ง ชิ้นงานที่ได้จากการทดสอบการกลึงในลักษณะนี้แสดงได้ดังรูปที่ 7.6

รูปที่ 7.6 ชิ้นงานที่ได้จากการทดสอบการกลึงในแนวโค้งสัมผัสกับแนวโค้ง

7.2.4 การกลึงในแนวโค้งสัมผัสกับแนวโค้งที่มีการชดเชยค่ารัศมีของคมตัด

เป็นการกลึงเพื่อให้ทราบถึงผลของรัศมีของคมตัดที่มีต่อผิวงาน ในกรณีที่มีการเปลี่ยนแปลงของแนวการกลึงเป็นเส้นโค้ง แต่มีการชดเชยค่ารัศมีของคมตัด ซึ่งจะทำให้ชิ้นงานมีขนาดที่ถูกต้องตามขนาดที่ต้องการ ชิ้นงานที่ได้จากการทดสอบการกลึงในลักษณะนี้แสดงได้ดังรูปที่ 7.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.7 ชิ้นงานที่ได้จากการทดสอบการกลึงในแนวโค้งสัมผัสกับแนวโค้งที่มีการชดเชยค่ารัศมีของคมตัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลและวิจารณ์

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ใช้อยู่ตามโรงงานอุตสาหกรรมต่าง ๆ นั้นมีความหลากหลายเชิงการใช้งาน อีกทั้งมีระบบการทำงานที่ซับซ้อน และมักจะถูกติดตั้งกับเครื่องจักรที่มีราคาสูง จึงทำให้มีขอบเขตการใช้งานในกลุ่มของผู้ปฏิบัติงานกลุ่มเล็ก ๆ เท่านั้น และเนื่องจากระบบอุตสาหกรรมมีการปกปิดขั้นตอน และวิธีการผลิต ตลอดจนวิธีการใช้งาน ทำให้การหาข้อมูลเกี่ยวกับเครื่องควบคุมเชิงเลขแบบใหม่เป็นไปได้ยาก วิทยานิพนธ์ฉบับนี้จึงเน้นหนักเกี่ยวกับพื้น ฐานการทำงานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่ได้ออกแบบ และพัฒนาขึ้นมา โดยมีการทำงานเป็นสองลักษณะคือ การควบคุมวิถีการเคลื่อนที่ของแกนเครื่องจักรในแนวตรง และในแนวโค้งแบบซิงโครไนซ และสามารถกำหนดความเร็วในการเคลื่อนที่ที่เป็นเชิงเวคเตอร์ได้เป็นผลสำเร็จเป็นผลให้สามารถ ประยุกต์ไปใช้กับเครื่องจักรอุตสาหกรรมชนิด 2 แกนได้อย่างกว้างขวาง เช่น เครื่องกลึง เครื่องเจาะ จากการทดสอบเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่จัดสร้างขึ้นกับเครื่องกลึง ที่มีอยู่ภายในภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่มีความละเอียดประมาณ +100 ไมครอนโดยดัดแปลงระบบขับเคลื่อนของเครื่องกลึง ให้สามารถต่อเข้ากับเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่สร้างขึ้นผลการทดสอบการกลึงชิ้นงาน สามารถให้ความละเอียดเกินความสามารถของเครื่องจักรทั่วไป คือ ดีกว่า 100 ไมครอน ซึ่งถ้านำไปใช้กับเครื่องจักรที่มีความละเอียดดีกว่านี้ คาดว่าสามารถให้ความถูกต้องประมาณ +5 ไมครอน สำหรับความราบเรียบของผิวงานไม่สามารถตรวจสอบได้ เนื่องจากขีดจำกัดด้านเครื่องมือวัด แต่จากการตรวจสอบด้วยสายตาพบว่า ชิ้นงานมีระยะของวงที่เกิดจากคมตัดของมีดกลึงที่มีความสม่ำเสมอดี ไม่แตกต่างจากชิ้นงานที่กลึงจากเครื่องกลึงที่ใช้เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่มีอยู่ภายในภาควิชาฯ อย่างไรก็ตาม เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่สร้างขึ้นยังมีลำดับขั้นตอนในการใช้งานค่อนข้างยุ่งยาก เนื่องจากการออกแบบเน้นหนักความถูกต้องของการทำงานของเครื่องจักร โดยไม่คำนึงถึงการใช้งานระดับผู้ปฏิบัติงาน และอาจจะขึ้นอยู่กับชนิดของเครื่องจักรด้วย อีกทั้งวิธีการใช้งานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่มีใช้อยู่มีความแตกต่างกันไปขึ้นอยู่กับบริษัทผู้ผลิต ซึ่งจะต้องทำการปรับปรุงต่อไป

ข้อเสนอแนะ

เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่มีใช้อยู่ทั่วไปนั้น ส่วนควบคุมจะถูกออกแบบให้มีการทำงานโดยอาศัยประสบการณ์ของผู้ปฏิบัติงานในสาขาต่าง ๆ บางสาขาเท่านั้น ดังนั้นการออกแบบเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ จึงมีฟังก์ชันการทำงานที่อาจไม่เหมาะสมสำหรับงานบางงานหากได้มีการพัฒนาเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ ให้สามารถสร้างรูปแบบของฟังก์ชันการทำงานได้โดยผู้ใช้งานในสาขาต่าง ๆ ตามความชำนาญกับงานที่ตนเองสัมผัสอยู่จะทำให้ประสิทธิภาพการทำงานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์มีความสามารถในการทำงานสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำวิจัยในวิทยานิพนธ์ฉบับนี้เป็นไปด้วยความลำบาก เนื่องจากข้อมูลต่างๆ เกี่ยวกับการทำงานของระบบฮาร์ดแวร์ ตลอดจนเทคนิคการออกแบบอุปกรณ์ฮาร์ดแวร์ไม่เป็นที่เปิดเผย จึงต้องค้นคว้าจากหนังสือหรือเอกสารอ้างอิงต่างๆ เท่าที่จะหาได้ ซึ่งส่วนใหญ่จะเป็นแต่เพียงหลักการโดยทั่วไปเท่านั้นอย่างไรก็ตาม ผู้ทำวิจัยได้รับความกรุณาเอื้อเฟื้อจากผู้ประกอบการเกี่ยวกับการขายและการใช้งานของเครื่องจักรที่ใช้เครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ในอุตสาหกรรมบางหน่วยงาน ให้โอกาสผู้ทำวิจัยได้ทำการศึกษาวิธีการทำงานของฮาร์ดแวร์ และวิธีการใช้งานของเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ที่มีใช้อยู่จริงเป็นอย่างดี ทำให้การศึกษาวิธีการออกแบบเครื่องควบคุมเชิงเลขแบบใช้คอมพิวเตอร์ซึ่งได้ดำเนินการมาเป็นระยะเวลานานหลายปี ได้ประสบผลสำเร็จ และสามารถเขียนเป็นวิทยานิพนธ์ฉบับนี้ขึ้นมาได้

ผู้ทำวิจัยขอขอบพระคุณผู้มีอุปการะคุณทุกท่านที่กรุณาให้คำปรึกษา ร่วมมือ และคอยให้กำลังใจในการทำวิจัยนี้ ได้แก่

รองศาสตราจารย์ กิตติ ตีรเศรษฐ หัวหน้าภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

รองศาสตราจารย์ ดร. พีรศักดิ์ วรสุนทโรสถ

กรรมการผู้จัดการ บริษัท GKS Mechanic

กรรมการผู้จัดการ บริษัท วิศววุธ จำกัด

คุณพ่อนิยม คุณแม่นงเยาว์ อุดคกิมาพันธุ์

อาจารย์และเพื่อนร่วมงานทุกท่าน ในภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม และผู้ทำวิจัยขอขอบพระคุณกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่ได้ให้ความกรุณาเป็นกรรมการสอบและชี้แนะข้อบกพร่องต่างๆ จนวิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

รองศาสตราจารย์ กิตติ ตีรเศรษฐ

รองศาสตราจารย์ ดร.โยธิน เปรมปราณีรัชต์

ผู้ช่วยศาสตราจารย์ ดร. จงกล งามวิวิทย์

ผู้ช่วยศาสตราจารย์ พิพัฒน์ เลาหสงคราม

ผู้ช่วยศาสตราจารย์ ดร. รัตติกร วรากูลศิริพันธุ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Assembly Language Programming for the IBM PC AT
Leo J. Scanlon
Brandy Communications Company, Inc.
A Simon & Schuster Publishing Company

[2] The 8086 Book
Rusell Rector-Grorge Alexy
OSBORNE/McGraw-Hill. Berkeley, California

[3] Interfacing to the IBM Personal Computer
Lewis C. Eggebrecht
Howard W.Sams & Co., Inc.

[4] IBM Technical Reference Personal Computer XT, 6936808

[5] IBM Personal Computer; Computer Language Series
Macro Assembly by Microsoft

[6] Computer Numerical Control Programming
Michael Sava, Joseph Pusztai
Humber College, Toronto, Cannada
Prentice-Hall International Editions

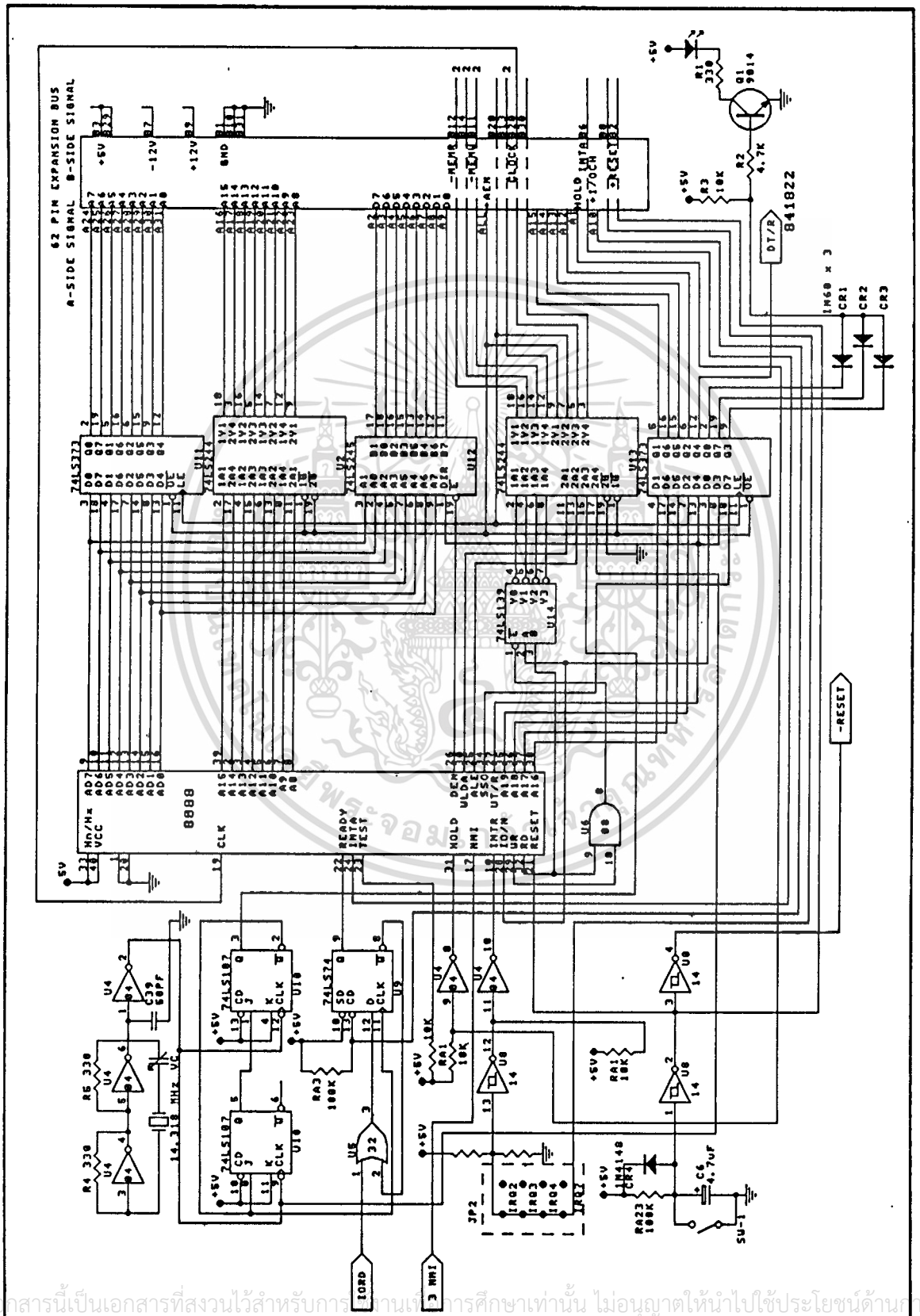
[7] GEK-25290 1050 Microprocessor CNC
Industrial Control Department
General Electric Company

[8] FANUC System 12M-Model A
Operator's Manual
B54824E/02 FANUC LTD,1984

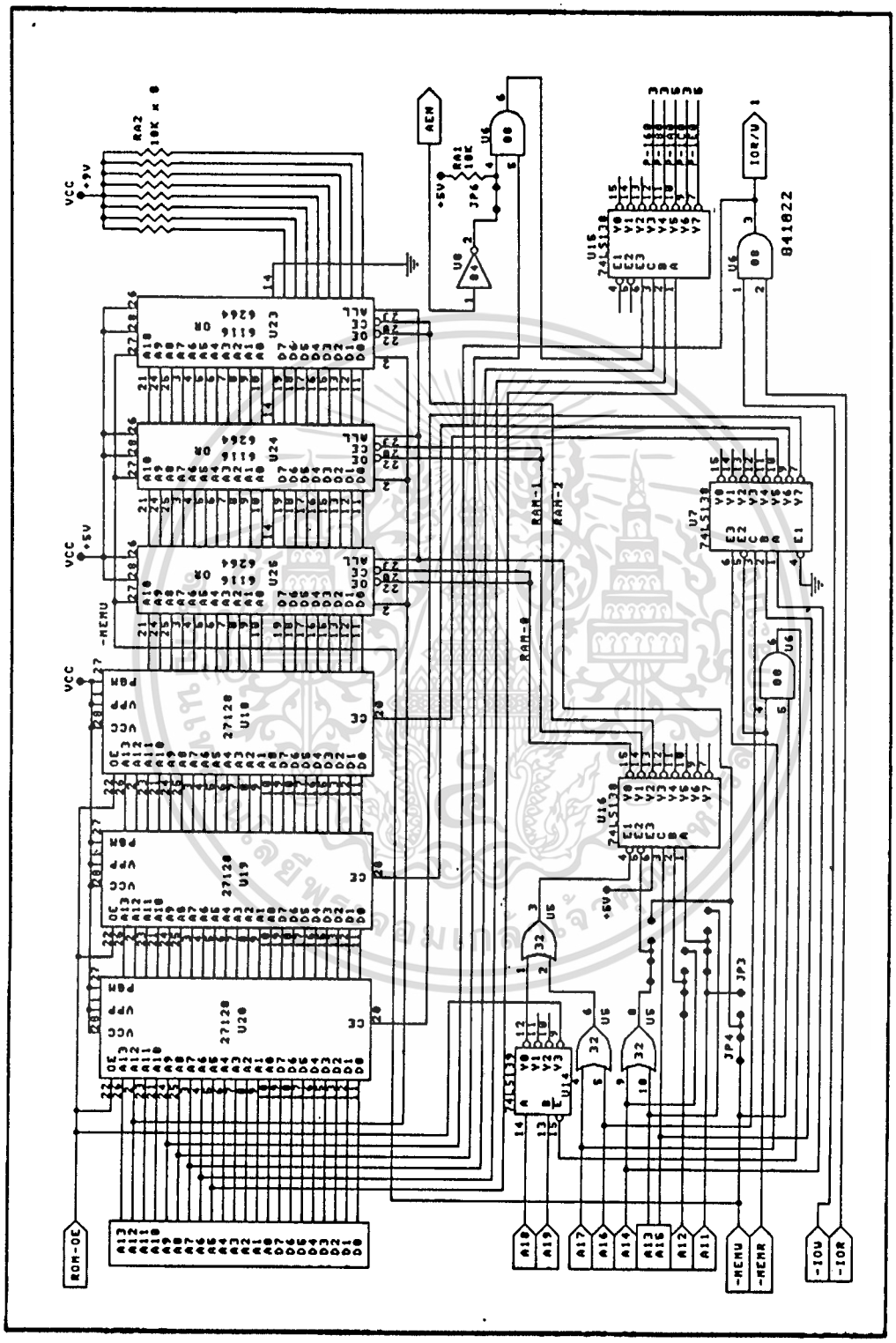
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ภาคผนวก ก.

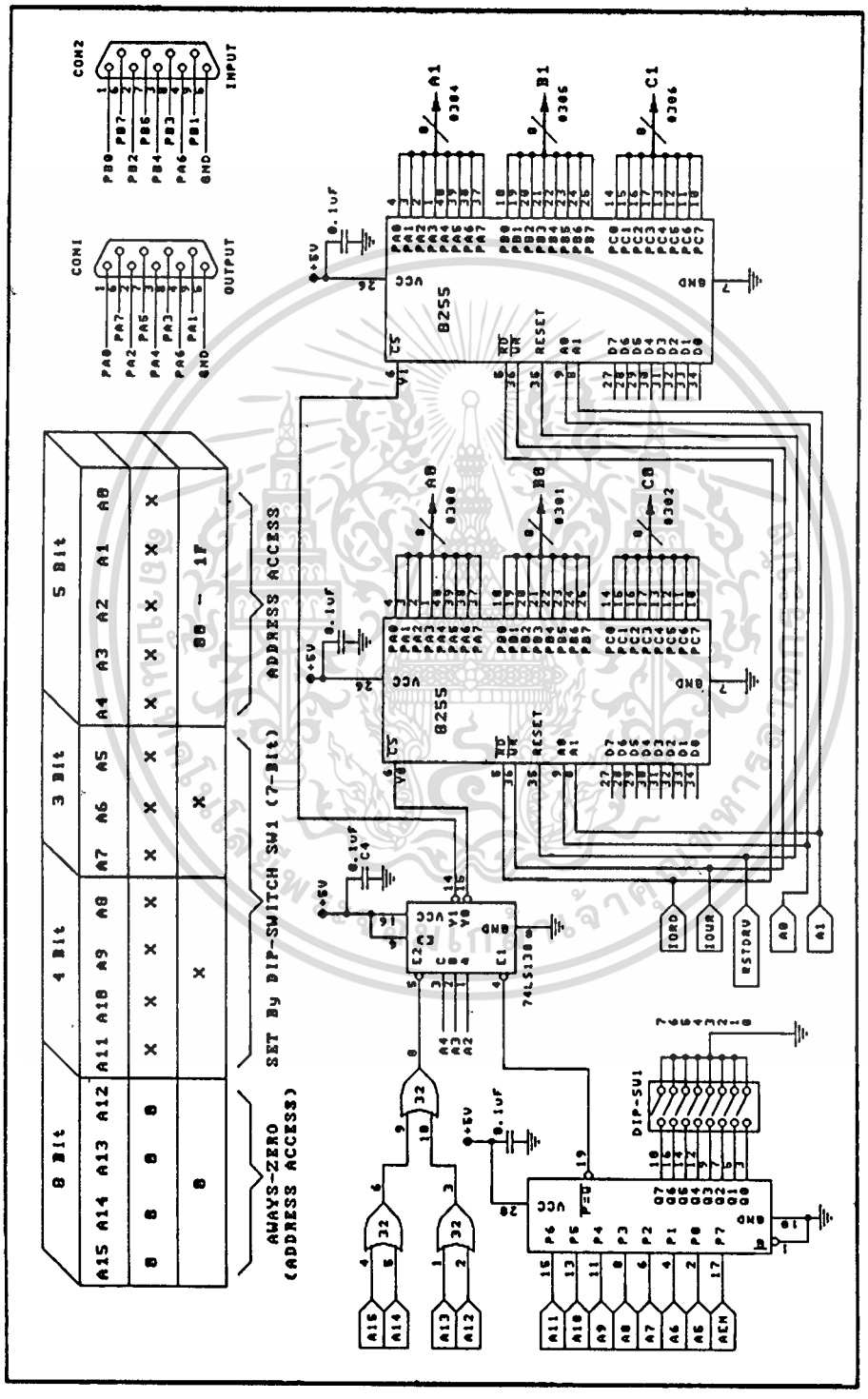
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



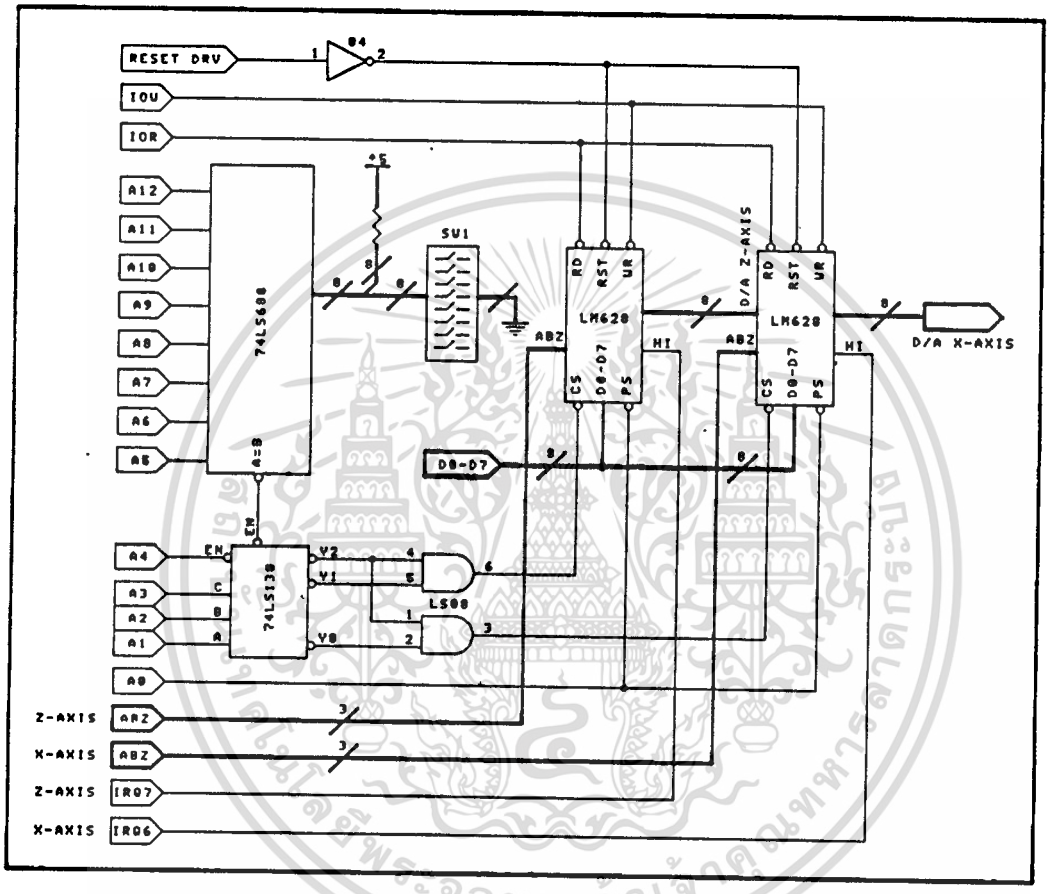
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



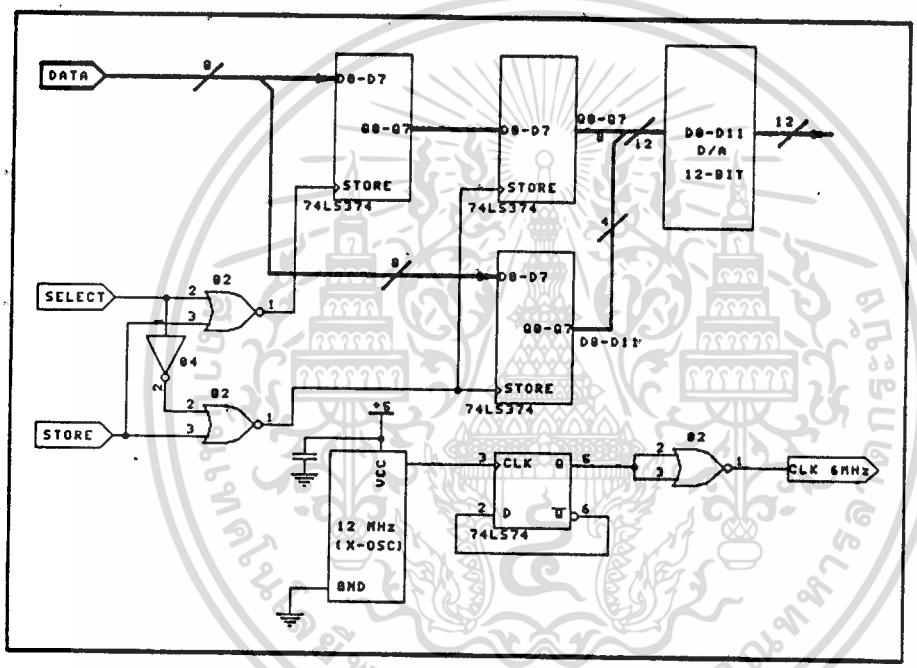
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;NAME: MOTION2.ASM
        .8086
ATTR    EQU   07          ;ATTRIBUTE OF DISPLAY
BLANK   EQU   20H         ;ASCII BLANK
MONO_SCN EQU  0B800H      ;BASE ADDRESS DISPLAY
ROW     EQU   25          ;NUMBER OF ROW OF CHAR
COL     EQU   80          ;NUMBER OF COL OF CHAR
NC_DSP  EQU   ROW*COL     ;ALL THE CHAR DISPLAY ON SCREEN
RANG    EQU   15          ;AMOUNT OF BYTES OF RESULT

RESET   EQU   00H         ;RESET COMMAND
PORT8   EQU   05H         ;8-BIT D/A
PORT12  EQU   06H         ;12-BIT D/A
DFH     EQU   02H.        ;DEFIND HOME
SIP     EQU   03H         ;SET INDEX POSITION
LPEI    EQU   1BH         ;INTERUPT ON ERROR
LPES    EQU   1AH         ;STOP ON ERROR
SBPA    EQU   20H         ;SET BRAKE POINT, ABS
SBPR    EQU   21H         ;SET BRAKE POINT, REAL
MSKI    EQU   1CH         ;MAKE INTERUPT
RSTI    EQU   1DH         ;RESET INTERUPT
LFIL    EQU   1EH         ;LOAD PARAMS
UDF     EQU   04H         ;UPDATE FILTER
LTRJ    EQU   1FH         ;LOAD TRAJECTORY
STI     EQU   01H         ;START MOTION
RDSIGS  EQU   0CH         ;READ SIGNAL REG
RDIP    EQU   09H         ;READ INDEX POSITION
RDDP    EQU   08H         ;READ DESIER POS
RDRP    EQU   0AH         ;READ REAL POSITION
RDDV    EQU   07H         ;READ DESIRED VEL
RDRV    EQU   0BH         ;READ REAL VELOCITY
RDSUM   EQU   0DH         ;READ INTEGRATION SUM

CTRLW1  EQU   00H         ;TRAJECTORY CONTROL WORD MSB
CTRLW2  EQU   02H         ;TRAJECTORY CONTROL WORD LSB
CTRFL1  EQU   00H         ;CONTROL WORD FILTER MSB
CTRFL2  EQU   08H         ;CONTROL WORD FILTER LSB

STACK   SEGMENT     PARA  STACK 'STACK' ;1K STACK SPACE
        DB          256 DUP (0)    ;256 BYTE OF STACK SPACE
STACK   ENDS

DATA    SEGMENT     PARA  PUBLIC  'DATA'
THIRTY  DB          30            ;VALUE FOR MUL INSTRUC
SIGTEEN DB          10H
PARM    DB          128 DUP (0)   ;MOVE PARAMETER TO HERE

; MASSAGE  TABLE:

MSG0    DB          '                                '
MSG1    DB          '                                '
MSG2    DB          '                                '
MSG3    DB          '                                '
MSG4    DB          '                                '
MSG5    DB          '                                '
MSG6    DB          '                                '
MSG7    DB          '                                '
MSG8    DB          '                                '

CHAR_BUFF DB        'POSITION                        ';DEFIND BUFFER FOR CHAR
VELOCITY  DB        'VELOCITY                        '
HEADING   DB        '          B7 B6 B5 B4 B3 B2 B1 B0 '
STATUS1   DB        'RDSTAT :                        '
STATUS2   DB        'RDSIGS :                        '
STATUS3   DB        '       :                        '
STATUS4   DB        'RDIP   :                        '
STATUS5   DB        'RDDP   :                        '
STATUS6   DB        'RDRP   :                        '
STATUS7   DB        'RDDV   :                        '
STATUS8   DB        'RDRV   :                        '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STATUS9   DB          'RDSUM  :                        '

MSG9      DB     'LAST MESSAGE OF THE DAY-------'

OUT_WORD  DW      0FFFFH
OUT_WORD1 DW      00000H
SCALE     DD      10000H

BUFF      DB   30 DUP (0)          ;DEFIND BUFFER AND CLEAR.
VBUFF     DB   30 DUP (0)          ;DEFIND BUFFER FOR VELOCITY.
DBUFF     DB   30 DUP (0)          ;DEFIND DATA BUFFER.
INPUT     DW    1 DUP (0)
          DW    1 DUP (0)

STATUS    DB    1 DUP (0)          ;STATUS REG OF LM 628.
RDSIGNAL  DB    1 DUP (0)          ;STSIG HIGH BYTE.
RDSIGNAL1 DB    1 DUP (0)          ;STSIG LOW BYTE.
FORWD     DB      00H,10H,40H,00H  ;FORWARD DIRECTION.
REWRD     DB      00H,00H,01H,00H  ;REWORD DIRECTION.
SPEEDS    DB      00H,00H,00H,00H  ;SPEED COMMAND.
JOG       DB      00H,00H,00H,00H  ;JOGING SPEED
ASCII_IBL DB     '0123456789ABCDEF'
PEN_UP    DB     'PD'
DATA_X    DB     '000000,'
DATA_Y    DB     '000000;'
          DB     'QQ'
P_SPEED   DB     ';IN;SP1;VS80;'
          DB     'QQ'
          DB     'PD1000,1000;'
          DB     'QQ'
HI_MCND   DW     ?
LO_MCND   DW     ?
HI_PP1    DW     ?
LO_PP1    DW     ?
HI_PP2    DW     ?
LO_PP2    DW     ?
HI_PP3    DW     ?
LO_PP3    DW     ?
HI_PP4    DW     ?
LO_PP4    DW     ?
NEG_IND   DB     ?
VAR_Z     DW     2 DUP (0)
LAST      DW     1 DUP (0)
LAST_X    DW     1 DUP (0)
CNT_I     DW     1 DUP (0)
X_CNT     DW     1 DUP (0)
RESERV    DW     ?
;---------------------------------
;   ARC GENERATED.
;---------------------------------
ARC_1 DW     1 DUP (0)
ARC_2 DW     1 DUP (0)
ARC_3 DW     1 DUP (0)
ARC_4 DW     1 DUP (0)
RES_2 DW     1 DUP (0)
RES_1 DW     1 DUP (1)

SQR_1 DW     1 DUP (1)     ;INITIAL BY ONE.
SQR_2 DW     1 DUP (0)
SQR_3 DW     1 DUP (0)
SQR_4 DW     1 DUP (0)

SQ_R1 DW     1 DUP (0)     ;SUCESSIVE APPROXIMATION
SQ_R2 DW     1 DUP (0)     ;REMAIN.
SQ_R3 DW     1 DUP (0)     ;(LAST SIATE OF ARC_1 - ARC_4.)
SQ_R4 DW     1 DUP (0)
XN_   DW     1 DUP  (0)
TWO   DW     1 DUP  (0)
;-----------------------
;   DISK I/O DEVICE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------
FCB       DB    36 DUP (0)    ;FILE CONTROL BOX.
DTA       DB    0             ;DISK TRANSFER AREA.
CNT_DATA  DW    DATA_BUF
TBL_FILE  DB    'SINE    DAT' ;DATA FILE NAME 'SINE.DAT'.
DISK_NO   DB    03            ;DEFIND DRIVE C.

DATA_BUF DB 4000H DUP (0) ;DATA BUFFER FOR DTA.
;-------------------
;   TEST PLOT
;-------------------

Y_AXITL   DW    1  DUP (0)    ; Y AXIT ON PLOTTER.
Y_AXITH   DW    1  DUP (0)
Y_STARTL  DW    1  DUP (0)    ;START OF ARC CONTOUR Y
Y_STARTH  DW    1  DUP (0)
Y_ENDL    DW    1  DUP (0)    ;END OF ARC CONTOUR Y.
Y_ENDH    DW    1  DUP (0)
Y_ALPH    DW    1  DUP (0)
Y_ALPL    DW    1  DUP (0)
Y_ABSH    DW    1  DUP (0)    ;LAST POSITION OF CONTOUR Y.
Y_ABSL    DW    1  DUP (0)
Y_AXIT1H      DW   1  DUP (0)     ;ALTITUDE OF Y FIRST POINT (MSW).
Y_AXIT1L      DW   1  DUP (0)     ;ALTITUDE OF Y FIRST POINT (LSW).
Y_ANGLE       DW   1  DUP (0)     ;FIRST ANGLE POINTER Y.

X_AXITL   DW    1  DUP (0)    ; X AXIT ON PLOTTER.
X_AXITH   DW    1  DUP (0)
X_STARTL  DW    1  DUP (0)    ;START OF ARC CONTOUR X
X_STARTH  DW    1  DUP (0)
X_ENDL    DW    1  DUP (0)    ;END OF ARC CONTOUR X.
X_ENDH    DW    1  DUP (0)
X_ALPH    DW    1  DUP (0)
X_ALPL    DW    1  DUP (0)
X_ABSH    DW    1  DUP (0)    ;LAST POSITION OF CONTOUR X.
X_ABSL    DW    1  DUP (0)
X_AXIT1H      DW   1  DUP (0)     ;ALTITUDE OF X FIRST POINT (MSW).
X_AXIT1L      DW   1  DUP (0)     ;ALTITUDE OF X FIRST POINT (LSW).
X_ANGLE       DW   1  DUP (0)     ;FIRST ANGLE POINTER X.
;TEST ONLY
OUTPUT_X      DW   ?              ;OUTPUT TO DSP.
OUTPUT_Y      DW   ?
INTVAL_X      DW   ?              ;ABS INTERVAL DISTANCE.
INTVAL_Y      DW   ?
V_SPEED       DW   ?
QUARDANT      DB   ?              ;DIRECTION OF ARC.

T_SCALE       DW   2000H          ;SCALE PLOT.
CNT_DATAY     DW   1  DUP (0)     ;DATA POINTER Y.
CNT_DATAX     DW   1  DUP (0)     ;DATA POINTER X.
REMAINW2      DW   1  DUP (0)     ;MSW.
REMAINW1      DW   1  DUP (0)     ;
REMAINW0      DW   1  DUP (0)     ;LSW.
FIRST_DIVISOR DW   51329
SECOND_DIVISOR DW  1617           ;DEFAULT VALUE 1617
ANGLE_0       DW   1              ;ANGLE COUNTER.
ANGLE_1       DW   1
;---------------------
;   ARC PARAMETER.
;---------------------

X_RECORD      DW   2  DUP (1000)
Y_RECORD      DW   2  DUP (1000)
I_RECORD      DW   2  DUP (1000)
J_RECORD      DW   2  DUP (1000)
STORE         DW   ?
SHFT          DW   1  DUP (0)

 ;ERROR MESSAGE:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ERRMSG    DB          'ERROR !!! INVALID PARAMETER!!'

DATA      ENDS

CODE      SEGMENT     PARA PUBLIC 'CODE'

START     PROC    FAR

;STANDARD PROGRAM PROLOGUE EXCEPT RETRAIN DS AS PTR

          ASSUME  CS:CODE
          PUSH    DS        ;SAVE PSP SEG ADDR
          MOV     AX,0
          PUSH    AX        ;SAVE RET ADDR OFFET (PSP+0)
          MOV     AX,DATA
          MOV     ES,AX     ;ESTABLISH EXTRA SEG ADDRESSABILITY
          ASSUME  ES:DATA
;
; MOVE FCB PARAMETER FROM PSP TO OUR DATA SEGMENT:
;
          MOV     SI,5CH              ;SOURCE STRING OFFSET (WITH PSP).
          MOV     DI,OFFSET FCB       ;DEST STRING OFFSET.
          MOV     CX,12               ;STRING RANG.
          CLD                         ;SET 'FORWARD' STRING OPERATION.
          REP     MOVSB               ;MOVE PARM IN TO DATA AREA
;
; ESTABLISH NORMAL DATA SEGMENT ADDRESSABLITY
;
          MOV     DS,AX
          ASSUME  DS:DATA
;
;   SET DATA AND OPEN FILE:
;
NAME:
          MOV     AL,DISK_NO          ;DEFIND DISK NO.
          MOV     FCB,AL
          MOV     SI,OFFSET TBL_FILE  ;SINE TABLE FILE.
          MOV     DI,OFFSET FCB+1
          MOV     CX,11
          CLD
          REP     MOVSB               ;INITIAL PRIMARY FILE NAME
                                      ;AND FILE NAME EXTENSION.
          MOV     DX,OFFSET DTA       ;ADDRESS OF DTA
          MOV     AH,1AH              ;DOS FUNCTION = 'SET DTA'
          INT     21H                 ;INVOKE DOS FUNCTION
          MOV     DX,OFFSET FCB       ;ADDRESS OF FCB
          MOV     AH,0FH              ;DOS FUNCTION = 'OPEN FILE'
          INT     21H                 ;INVOKE DOSFUNCTION
          CMP     AL,0                ;DID THE OPEN WORK?
          JNZ     ERROR2              ;BRANCH IF NOT
;
;   INITIALIZE THE RECORD SIZE','CURRENT BLOCK',AND
;   'CURRENT RECORD' FILE WITHIN THE FCB:
;
          MOV     WORD PTR FCB+0CH,0  ;'CURRENT BLOCK' = 0
          MOV     WORD PTR FCB+0EH,1  ;'RECORD SIZE'   = 1
          MOV     FCB+20H,0           ;'CURRENT RECORD'= 0

; 'TAB' (09H),END OF FILE' (1A),AND 'LINE FEED' (0AH)

          MOV     CX,4000             ;READ DATA 4000 BYTE
RD_AGAIN: MOV     DX,OFFSET FCB       ;ADDRESS OF FCB
          MOV     AH,14H              ;DOS FUNCTION = 'SEQUENTIAL RED'
          INT     21H                 ;INVOKE DOS FUNCTION
          CMP     AL,0                ;DID THE 'READ' WORK?
          JNZ     EOF                 ;BRANCH IF NOT
          MOV     AL,DTA              ;GET THE CHARACTER JUST READ
          CALL    SAVE_TBL               ;SAVE TO TEXT BUFFER.
          LOOP    RD_AGAIN
          JMP     EOF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;
; AFILE ACCESS ERROR HAS OCCUREED, IELL USER AND QUIT:
;

ERROR2:   RET                              ;RETURN TO DOS
;
; END OF FILE HAS BEEN ENCOUNTED, CLOSE THE FILE AND
;   RETURN TO DOS:
;
EOF:      MOV      DX,OFFSET FCB           ;ADDRESS OF FCB
          MOV      AH,10H                  ;DOS FUNCTION = 'CLOSE FILE'
          INT      21H                     ;INVOKE DOS FUNCTION


               CALL    CLRSCN  ;WRITE CHAR BLANK ON SCREEN

               JMP     INT8255

               MOV     DX,00
               MOV     AH,1
               INT     17H

               MOV     BX,OFFSET P_SPEED   ;INITIAL PLOTER.
     PLT_INIT:
               MOV     AL,[BX]
               CMP     AL,'Q'
               JZ      PLT_EXIT
               CALL    PLOT
               INC     BX
               JMP     PLT_INIT
     PLT_EXIT:

     INT8255:
               MOV     DX,304H
               IN      AL,DX
               CMP     AL,00       ;CHECK DSP POWER UP
               JZ      COLD
               MOV     DX,303H
               MOV     AL,80H
               OUT     DX,AL
               MOV     AL,0FFH
               MOV     DX,302H
               OUT     DX,AL

               MOV     DX,307H
               MOV     AL,93H
               OUT     DX,AL
               ;CALL    INIT       ;INITIAL LM628.
     COLD:     ;CALL     IO_TEST   ;TEST SERVO CARD.

               CALL     CUR_OFF   ;OFF CURSOR ON CRT.

     LOOP:     MOV      AH,01     ;FUNCTION = READ KEYBOARD.
               INT      16H       ;CALL BIOS KEYBOARD ROUTINE.

               ;SUB     AL,'0'    ;CONVERT FROM ASCII TO BINARY
               ;JC      ERROR     ;BRANCH IF NOT NUMERIC (<'0')
               ;CMP     AL,9      ;CHECK FOR VALID NUMERIC

               JNZ      ERROR     ;BRUNCH IF NOT VALID(>'9')

               JMP      MSG



     ;SELECT APPROPRIATE MASSAGE FROM MASSAGE TABLE

     MSG:      MOV      BX,OFFSET MSG0  ;POINT TO FIRST MESSAGE
               MOV      AX,09H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MUL     THIRTY      ;AX=AX*30
            ADD     BX,AX       ;POINT TO DESIRED MESSAGE
            CALL    DISPLAY     ;DISPLAY MASAGE AT [BX]
            CALL    ASCII

            ;CALL    READ       ;READ STATUS.
            ;CALL    REPORT1    ;DISPLAY IT.

        ;   CALL    COSINT
            CALL    TEST

            JMP     LOOP        ;LOOP

    ;DISPLAY ERROR MESSAGE FOR VALID PARAMETER:

    ERROR:  MOV     AH,0        ;DEFIND AH READ CHARACTER.
            INT     16H
            SUB     AL,'0'
            JC      ERROR1
;            CMP     AL,1DH     ;CHARACTER M.
;            JZ      FORWARD
            CMP     AL,19H      ;CHARACTER I.
            JZ      VAR_ADJ
;            CMP     AL,23H     ;CHARACTER S.
;            JZ      STOP

    ERROR1: MOV     BX,OFFSET ERRMSG
            CALL    DISPLAY     ;DISPLAY THE ERROR MESSAGE AT [BX]
            RET                 ;RETURN TO DOS


;-----------------------------------------

    ;----------------------
    ; X-VARIABLE ADJUSTABLE.
    ;----------------------
    ;ADJUST VARIABLE ON X-VALUE.
    ; Z = SQRT(R_SQR - X_SQR)

    VAR_ADJ PROC    NEAR
            PUSH    AX
            PUSH    BX

            MOV     BX,OFFSET INPUT
            MOV     AX,[BX]
            INC     AX
            MOV     [BX],AX
            MOV     BX,OFFSET VBUFF
            MOV     [BX],AX

            POP     BX
            POP     AX

            JMP     LOOP

    VAR_ADJ ENDP

;-----------------------------------------

    FCODE     DB   2048  DUP (0) ;DEFIND MEMORY FOR FIRST CODE.
    SCODE     DB   1024  DUP (0) ;DEFIND MEMORY FOR SECOND CODE.
    CMD_INDEX DW   1     DUP (0) ;COMMAND POINTER.
    F_INDEX   DW   1     DUP (0) ;FIRST CODE POINTER.
    TEMP      DB   1     DUP (0) ;TEMPOLARY USE FOR CONVERT DATA.
    TEMPH     DW   1     DUP (0) ;TEMPOLARY USE FOR CONVERT DATA.
    TEMPL     DW   1     DUP (0)
    DIST_X    DW   2     DUP (0) ;DISTANCE OF X-AXIS.
    DIST_Z    DW   2     DUP (0) ;DISTANCE OF Y-AXIS.
    F_PARA    DW   2     DUP (0) ;FEED VECTOR.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
S_PARA    DW   2   DUP (0)   ;SPEED OF SPINDLE.
DIR_MOVE  DB   1   DUP (0)   ;DIRECTION MOVEMENT REG.

INIT_FCODE: MOV     BX,OFFSET FCODE

SICFCNC   PROC     NEAR
          MOV      BX,OFFSET F_INDEX
          MOV      AL,[BX]
          CMP      AL,'G'
          JZ       G_CMD      ;IF DATA IS CHARECTER G THEN DO G_CMD.
          CMP      AL,'M'
;          JZ       M_CMD     ;IF DATA IS CHARECTER M THEN DO M_CMD.

G_CMD:
          INC      BX
          MOV      AL,[BX]
          CMP      AL,' '     ;CHECK SPACE.
          JZ       G_CMD      ;IF DATA = SPACE THEN SKIP TO NEXT DATA.
          SUB      AL,'0'     ;CHANGE ASCII TO BINARY.
          JC       CMD_ERR    ;CMD ERROR.
          CMP      AL,9
          JNZ      CMD_ERR    ;IF AL >9 INVALID.
          PUSH     BX
          MOV      BL,10
          MUL      BL
          POP      BX         ;RESTORE BX.
          MOV      TEMP,AL
          INC      BX
          MOV      AL,[BX]
          SUB      AL,'0'
          JC       CMD_ERR
          CMP      AL,9
          JNZ      CMD_ERR
          MOV      F_INDEX,BX ;STORE LAST POINTER.
          ADD      AL,TEMP    ;ADD WITH LSB.
          MOV      TEMP,AL
          CMP      AL,00
          JZ       G_00       ;G00
          CMP      AL,01
;          JZ       G_01       ;G01
          CMP      AL,02
;          JZ       G_02       ;G02
          JMP      G_END

CMD_ERR:  NOP                 ;CODE ERROR.
;NOTE.
;MUI BE HAVE ERROR MASAGE OR ALARM CODE SHOW.
          RET                 ;RETURN TO MAIN.

G_00:     INC      BX         ;NEXT DATA.
          MOV      AL,[BX]
          CMP      AL,' '     ;CHECK SPACE.
          JZ       G_00       ;IF SPACE APPEAR THEN SKIP TO NEXT DATA.
          MOV      CX,2       ;SET COUNTER = 2 FOR DATA SEQUENT X,Z.
G00_PARA:                     ;G00 PARAMETER INPUT.
          CMP      AL,'X'     ;CHECK X PARAMETER.
          JZ       G00_X      ;INPUT VALUE OF X.
          CMP      AL,'Z'     ;CHECH Z PARAMETER.
          JZ       GOTO_Z     ;INPUT VALUE OF Z.
          JMP      CMD_ERR    ;IF DATA IS NOT X OR Y THEN CMD ERROR.
GOTO_Z:   JMP      G00_Z


G00_X:
          INC      BX
          MOV      AL,[BX]
          CMP      AL,' '
          JZ       G00_X
          CMP      AL,'-'     ;CHECK SIGN INTEGER VALUE.
          JZ       GOTOXDIR   ;DIRECTION OF X MOVEMENT.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     AL,0FEH   ;CLEAR BIT 0.
                AND     DIR_MOVE,AL ;CLEAR DIRECTION MOVEMENT REG BIT 0 WITH 0.
                JMP     GOOX_SKIP1 ;SPACE SKIP.
    GOTOXDIR:   JMP     GOOX_DIR
    GOOX_NUM:                   ;X NUMERIC FIELD.
                MOV     F_INDEX,BX ;SAVE LAST POINTER AT LSB.
                CALL    DAC2BIN    ;DACIMAL TO BINARY CONVERT.

                MOV     BX,OFFSET DIST_X  ;DIISTANCE OF X-AXIS.
                MOV     AX,TEMPH
                MOV     [BX],AX
                ADD     BX,2       ;INCREMENT 2 BYTE.
                MOV     AX,TEMPL
                MOV     [BX],AX

                LOOP    GOO_PARA   ;COUNT INPUT SQUENT.
                CALL    RUN_GOO    ;EXECUTE GOO.

                JMP     STCFCNC    ;JMP TO STANDARD CODE FOR CNC
                                   ;PART PROGRAM.

    GOOX_DIR:                      ;DIRECTION FLAG X (BIT-0).
                MOV     AL,01      ;SET BIT 0.
                OR      DIR_MOVE,AL ;SET DIRECTION MOVEMENT REG BIT 0.
    GOOX_SKIP:                     ;CHECK SPACE.
                INC     BX         ;POINT NEXT DATA.
                MOV     AL,[BX]    ;READ NEXT DATA.
    GOOX_SKIP1:
                CMP     AL,' '
                JZ      GOOX_SKIP  ;SPACE SKIP.
                MOV     DX,00      ;SET COUNTER = 0.
                SUB     AL,'0'
                JC      INPUT_ERR  ;IF AL < 0 NUMBER FIELD ERR.
                CMP     AL,9
                JNZ     INPUT_ERR  ;IF AL > 9 NUMBER FIELD ERR.
                JMP     GOOX_LSB1
    INPUT_ERR:




    GOOX_LSB:                      ;CHECK DATA FIELD.
                SUB     AL,'0'
                JC      X_LSB      ;IF CARRY SET POINTER IS LSB+1.
                CMP     AL,9
                JNZ     X_LSB      ;IF AL > 9 DO X_LSB.
    GOOX_LSB1:
                INC     DX         ;COUNT DIGIT.
                INC     BX         ;POINTER IS NEXT DATA.
                MOV     AL,[BX]
                JMP     GOOX_LSB   ;READ AGAIN.
    X_LSB:
                DEC     BX         ;POINTER IS LSB.
                JMP     GOOX_NUM
    GOO_Z:
                LOOP    GOO_PARA

    G_END:                         ;END OF G-CODE.
                RET
    STCFCNC     ENDP

RUN_GOO         PROC    NEAR       ;EXECUTE GOO.


RUN_GOO         ENDP

DAC2BIN         PROC    NEAR       ;CONVERT ASCII NUMBER TO BINARY.

                MOV     BX,F_INDEX  ;LAST DATA POINT.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SUB     AX,AX       ;ZERO ACCUMULATOR.
                MOV     TEMPH,AX    ;ZERO TEMPH.
                MOV     AL,[BX]     ;READ NUMERIC.
                SUB     AL,'0'      ;CHANGE TO BINARY.
                MOV     TEMPL,AX    ;FIRST DIGIT.
                DEC     DX
                JNZ     MPY_TEN
                JMP     BIN_RDY
MPY_TEN:                            ;MULTIPLY BY 10.
                DEC     BX          ;POINT TO NEXT DIGIT.
                SUB     AX,AX
                MOV     AL,[BX]     ;READ IT.
                SUB     AL,'0'
                PUSH    DX
                PUSH    BX
                MOV     BX,10
                MUL     BX
                ADD     TEMPL,AX
                MOV     AX,0
                ADC     TEMPH,AX
                POP     BX
                POP     DX
                DEC     DX
                JNZ     MPY_HUN
                JMP     BIN_RDY
MPY_HUN:                            ;MULTIPLY BY 100.
                DEC     BX          ;POINT TO NEXT DIGIT.
                SUB     AX,AX
                MOV     AL,[BX]     ;READ IT.
                SUB     AL,'0'
                PUSH    DX
                PUSH    BX
                MOV     BX,100
                MUL     BX
                ADD     TEMPL,AX
                MOV     AX,0
                ADC     TEMPH,AX
                POP     BX
                POP     DX
                DEC     DX
                JNZ     MPY_THOU
                JMP     BIN_RDY
MPY_THOU:                           ;MULTIPLY BY 1000.
                DEC     BX          ;POINT TO NEXT DIGIT.
                SUB     AX,AX
                MOV     AL,[BX]     ;READ IT.
                SUB     AL,'0'
                PUSH    DX
                PUSH    BX
                MOV     BX,1000
&d@               MUL     BX
                ADD     TEMPL,AX
                MOV     AX,0
                ADC     TEMPH,AX
                POP     BX
                POP     DX
                DEC     DX
                JNZ     MPY_TENTHOU
                JMP     BIN_RDY
MPY_TENTHOU:                        ;MULTIPLY BY 10000
                DEC     BX          ;POINT TO NEXT DIGIT.
                SUB     AX,AX
                MOV     AL,[BX]     ;READ IT.
                SUB     AL,'0'
                PUSH    DX
                PUSH    BX
                MOV     BX,10000
                MUL     BX
                ADD     TEMPL,AX
                MOV     AX,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ADC     TEMPH,AX
                POP     BX
                POP     DX
            ;   DEC     DX
            ;   JNZ     BIN_OVFO    ;DACIMAL CONVERT TO BIN OVERFOLW.
BIN_RDY:                            ;DACIMAL TO BINARY READY.
                MOV     F_INDEX,BX  ;SAVE LAST POINTER.
                RET                 ;RETURN TO MAIN.
DAC2BIN         ENDP



  PLT_S     PROC    NEAR
            PUSH    AX
            PUSH    DX
            MOV     BX,OFFSET PEN_UP
  PLT_T:
            MOV     AL,[BX]
            CMP     AL,'Q'
            JZ      PL_END

            MOV     DX,378H
            OUT     DX,AL
            MOV     DX,379H
  S_WAIT:
            IN      AL,DX
            TEST    AL,80H
            JZ      S_WAIT
            MOV     DX,37AH
            MOV     AL,0DH
            OUT     DX,AL
            MOV     AL,0CH
            OUT     DX,AL
            INC     BX
            JMP     PLT_T
  PL_END:
            POP     DX
            POP     AX
            RET
  PLT_S     ENDP

PLT_S1      PROC    NEAR
            PUSH    BX
            PUSH    DX

            MOV     AH,1        ;PARAMETER TO INITIAL PRINTER.
            MOV     DX,0        ;SET FOR ONE PRINTER.
            INT     17H
            MOV     BX,OFFSET PEN_UP
PLTT1:
            MOV     AL,[BX]
            CMP     AL,'Q'
            JZ      PLTT2

            MOV     AH,0
            INT     17H
            INC     BX
            JMP     PLTT1
PLTT2:
            POP     DX
            POP     BX
            RET
PLT_S1      ENDP

;-------------------------------
;   WRITE DATA TO PRINTER PORT
;-------------------------------
; AL: INPUT
; DX: I/O POINTER.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ;
    PLOT      PROC    NEAR
              PUSH    AX
              PUSH    DX
              MOV     AH,0
              INT     17H
              POP     DX
              POP     AX
              RET
    PLOT      ENDP

;---------------------------------
; CONVERT BINARY NUMBER TO A STRING.
;---------------------------------
; CONVERT A SIGNED BINARY NUMBER TO SIX-BYTE
;ASCII STRING (SIGN PLUS FIVE DIGITS)
; INPUT:  AX = NUMBER TO CONVERTED.
; RESULT: DX = STARTING OF ADDRESS OF STRING BUFFER.
;         CX = CHARACTER COUNT.
; OTHER REGISTER ARE PRESERVED.

   BIN_ASCII  PROC NEAR
              PUSH DX
              PUSH BX
              PUSH CX
              PUSH SI
              PUSH AX
              MOV  BX,DX
              MOV  CX,6
   FILL_BUFF:
              MOV  BYTE PTR [BX],' '
              INC  BX
              LOOP FILL_BUFF
              MOV  SI,10              ;GET READY TO DIVIDED BY 10.
              OR   AX,AX              ;IF VALUE IS NEGATIVE.
              JNS  CLR_DVD
              NEG  AX                 ;MAKE IT POSITIVE.
   CLR_DVD:
              SUB  DX,DX              ;CLEAR UPPER HALF OF DIVIDEND.
              DIV  SI                 ;DIVIDE AX BY 10.
              ADD  DX,'0'             ;CONVERT REMAINDER TO ASCII DIGIT.
              DEC  BX                 ;BACK UP THROUGH BUFFER.
              MOV  [BX],DL            ;STORE CHAR. IN THE STRING
              INC  CX                 ;COUNT CONVERTED CHARACTER.
              OR   AX,AX              ;ALL DONE ?
              JNZ  CLR_DVD            ;IF NOT GET NEXT DIGIT.
              POP  AX                 ;YES. GET ORIGINAL VALUE.
              OR   AX,AX              ;WAS IT NEGATIVE.
              JNS  NO_MORE
              DEC  BX                 ;YES. STORE SIGN
              MOV  BYTE PTR [BX],'-'
              INC  CX                 ;AND INCREASE CHARACTER COUNT.
   NO_MORE:
              POP  SI                 ;RESTORE REGISTER
              POP  CX
              POP  BX
              POP  DX
              RET                     ;AND EXIT.

   BIN_ASCII  ENDP


;---------------------------------
;    INTEGER UNSIGNED MULTIPLY.
;---------------------------------
;32-BIT BY 32-BIT UNSIGNED MULTIPLY PROCEDURE.

;INPUT  CX:BX = MULTIPLIER
;       DX:AX = MULTIPLICAND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;OUTPUT DX,CX,BX AND AX (HIGH TO LOW ORDER.)

MULU32  PROC    NEAR
        PUSH    DS
        PUSH    DI
        MOV     HI_MCND,DX
        MOV     LO_MCND,AX
        MUL     BX
        MOV     HI_PP1,DX
        MOV     LO_PP1,AX
        MOV     AX,HI_MCND
        MUL     BX
        MOV     HI_PP2,DX
        MOV     LO_PP2,AX
        MOV     AX,LO_MCND
        MUL     CX
        MOV     HI_PP3,DX
        MOV     LO_PP3,AX
        MOV     AX,HI_MCND
        MUL     CX
        MOV     HI_PP4,DX
        MOV     LO_PP4,AX
        MOV     AX,LO_PP1
        MOV     BX,HI_PP1
        ADD     BX,LO_PP2
        ADC     HI_PP2,0
        ADD     BX,LO_PP3
        MOV     CX,HI_PP2
        ADC     CX,HI_PP3
        ADC     HI_PP4,0
        ADD     CX,LO_PP4
        MOV     DX,HI_PP4
        ADC     DX,0

        POP     DI
        POP     DS
        RET

MULU32 ENDP

;----------------------
;   ARC GENERAIED.
;----------------------
;INPUT 64-BIT:
;OUTPUT 32-BIT.
;VAR INPUT:   I_RECORD+2 - I_RECORD HIGH ORDER TO LOW (32-BIT).
;             J_RECORD+2 - J_RECORD
;    OUTPUT:  RES_2 - RES_1   HIGH ORDER TO LOW (32-BIT).

ARC   PROC  NEAR
      PUSH  AX
      PUSH  BX
      PUSH  CX
      PUSH  DX

      SUB  AX,AX
      MOV  CX,I_RECORD+2   ;READ MSW.
      MOV  DX,CX
      MOV  AX,I_RECORD     ;READ LSW.
      MOV  BX,AX

      CALL  MULU32       ;CALCULAIED I-SQUARE.

      PUSH  DX           ;STORE VALUE OF I_SQR
      PUSH  CX           ;64-BIT.
      PUSH  BX
      PUSH  AX
;---------------------------------------
; FIND
;     RADIUS=SQRT(I_SQUARE + J_SQUARE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;----------------------------------------
      SUB  AX,AX
      MOV  CX,J_RECORD+2   ;READ MSW.
      MOV  DX,CX
      MOV  AX,J_RECORD     ;READ LSW.
      MOV  BX,AX
      CALL MULU32          ;CALCULATED J-SQUARE.
      MOV   STORE,DX       ;TANSIENT DATA STORAGE.
      POP   DX             ;READ I-SQUARE FIRST WORD (LSW).
      ADD   AX,DX
      POP   DX             ;READ I-SQOARE SECOND WORD.
      ADC   BX,DX
      POP   DX             ;READ I-SQOARE THERD WORD.
      ADC   CX,DX
      POP   DX             ;READ I-SQOARE FOURTH WORD (MSW).
      PUSH  AX
      MOV   AX,STORE       ;DEFIND MSB=00.
      ADC   DX,AX
      POP   AX             ;RESTORE DATA ON AX.

SAVE_DATA:
      MOV   ARC_4,DX       ;SAVE HIGH ORDER TO LOW ON
      MOV   ARC_3,CX       ;ARC_4 TO ARC_1.
      MOV   ARC_2,BX
      MOV   ARC_1,AX

      ;JMP   SQRT
    ;--------------------------
    ; DIVISON 64-BIT OPERATION.
    ;--------------------------
NEWTON:
        MOV     TWO,2
        MOV     XN,16            ;INITIAL FIRST VALUE.
        MOV     DX,0
LOAD_XN:
        MOV     AX,ARC_2
        DIV     XN
        MOV     AX,ARC_1
        DIV     XN               ;DIVISION SECOND WORD.
        ADD     AX,XN            ; AX = Xn + X/Xn.
        MOV     DX,0
        DIV     TWO
        MOV     BX,XN            ; BX = 0.5(Xn + X/Xn).
        MOV     XN,AX
        CMP     AX,BX
        JNZ     LOAD_XN
        MOV     RES_1,AX         ;STORE DATA MSW.
        MOV     RES_2,0
CAL_ERR:
        POP    DX
        POP    CX
        POP    BX
        POP    AX

        RET


SQRT:
              MOV    RES_1,0          ;INITIAL ZERO ON RES_1(LSW)
              MOV    RES_2,0          ;AND RES_2 (MSW).
              MOV    SHFT,8000H
              MOV    CX,16            ;16-BIT MSW.
    SQRT_LOOP:
              PUSH   CX
              MOV    DX,RES_2
              ADD    DX,SHFT
              MOV    CX,DX
              SUB    AX,AX
              MOV    BX,AX
              CALL   MULU32           ;SQUARE (AX:DX).
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SUB     AX,ARC_1
                SBB     BX,ARC_2
                SBB     CX,ARC_3
                SBB     DX,ARC_4
                JNC      SQRT_NEXT
                MOV     AX,SHFT
                ADD     RES_2,AX
        SQRT_NEXT:
                ROR     SHFT,1          ;SHIFT RIGHT ONE BIT.
                POP     CX
                LOOP    SQRT_LOOP
                MOV     SHFT,8000H       ;INITIAL SHFT.
                MOV     CX,16            ;16-BIT LSW.
        SQRT_LOOP1:
                PUSH    CX
                MOV     DX,RES_2
                MOV     CX,DX
                MOV     AX,SHFT
                ADD     AX,RES_1
                MOV     BX,AX
                CALL    MULU32           ;SQUARE (AX:DX).
                SUB     AX,ARC_1
                SBB     BX,ARC_2
                SBB     CX,ARC_3
                SBB     DX,ARC_4
                JNC      SQRT_NEXT1
                MOV     AX,SHFT
                ADD     RES_1,AX
        SQRT_NEXT1:
                ROR     SHFT,1          ;SHIFT RIGHT ONE BIT.
                POP     CX
                LOOP    SQRT_LOOP1

                POP     DX
                POP     CX
                POP     BX
                POP     AX

                RET

 ARC   ENDP
 ;---------------------------------------------------------

I_FITING PROC NEAR

        MOV     HI_MCND,DX
        MOV     LO_MCND,AX
        MUL     BX
        MOV     HI_PP1,DX
        MOV     LO_PP1,AX
        MOV     AX,HI_MCND
        MUL     BX
        MOV     HI_PP2,DX
        MOV     LO_PP2,AX
        MOV     AX,LO_MCND
        MUL     CX
        MOV     HI_PP3,DX
        MOV     LO_PP3,AX
        MOV     AX,HI_MCND
        MUL     CX
        MOV     HI_PP4,DX
        MOV     LO_PP4,AX
        MOV     AX,LO_PP1
        MOV     BX,HI_PP1
        ADD     BX,LO_PP2
        ADC     HI_PP2,0
        ADD     BX,LO_PP3
        MOV     CX,HI_PP2
        ADC     CX,HI_PP3
        ADC     HI_PP4,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD     CX,LO PP4
        MOV     DX,HI PP4
        ADC     DX,0
    ;--------------------------
    ; DIVISON 64-BIT OPERATION.
    ;--------------------------
        PUSH    AX              ;STORE DATA AX.
        MOV     AX,CX
        DIV     FIRST DIVISOR   ;DIVISION FIRST WORD.
        MOV     REMAINW2,AX     ;STORE DATA MSW.
        MOV     AX,BX           ;READ DATA BX.
        DIV     FIRST DIVISOR   ;DIVISION SECOND WORDS.
        MOV     REMAINW1,AX     ;STORE DATA MSW.
        POP     AX              ;READ DATA AX.
        DIV     FIRST DIVISOR   ;DIVISION THERD WORDS.
        MOV     [REMAINW0],AX   ;STORE DATA LSW.

        MOV     DX,[REMAINW2]
        MOV     AX,[REMAINW1]
        DIV     SECOND DIVISOR
        MOV     REMAINW1,AX
        MOV     AX,REMAINW0
        DIV     SECOND DIVISOR
        MOV     [REMAINW0],AX
        RET

I_FITING  ENDP


SEARCH_J  PROC    NEAR

;----------------------
;  CHECK FORWARD PART
;----------------------
            MOV     Y ENDH,0
            SUB     AX,AX
            MOV     AX,Y ENDL
            SUB     AX,J RECORD
            MOV     AX,Y ENDH
            SBB     AX,J RECORD+2
            JC      EXCHANGE
            MOV     AX,I RECORD
            MOV     X STARTL,AX
            MOV     AX,I RECORD+2
            MOV     X STARTH,AX
            MOV     AX,J RECORD     ;IF Y END > J RECORD THEN PUSH
            MOV     Y STARTL,AX     ;VALUE OF J_RECORD TO Y_START.
            MOV     AX,J RECORD+2
            MOV     Y STARTH,AX
            JMP     PLT_REVERSE
EXCHANGE:
            MOV     AX,X ENDL
            MOV     X STARTL,AX
            MOV     AX,X ENDH
            MOV     X STARTH,AX
            MOV     AX,Y ENDL       ;IF Y END < J RECORD THEN PUSH
            MOV     Y STARTL,AX     ;VALUE BETWEEN Y_END AND Y_START.
            MOV     AX,Y ENDH
            MOV     Y_STARTH,AX

            MOV     AX,I RECORD
            MOV     X ENDL,AX
            MOV     AX,I RECORD+2
            MOV     X ENDH,AX
            MOV     AX,J RECORD
            MOV     Y ENDL,AX
            MOV     AX,J RECORD+2
            MOV     Y_ENDH,AX
PLT_REVERSE:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SUB     BX,BX           ;ZERO BX.
        MOV     Y_AXITL,BX
        MOV     Y_AXITH,BX
        MOV     X_AXITL,BX
        MOV     X_AXITH,BX
        CALL ARC                ;FIND RADIUS

        MOV     BX,OFFSET DATA_BUF
        MOV     CNT_DATAY,BX
        ADD     BX,3998         ;OFFSET TO END OF SINE
        MOV     CNT_DATAX,BX    ;TABLE 1999 WORD.
        MOV     CX,2000         ;SET COUNTER LIMIT.
FIND_LOOP:
        MOV     BX,CNT_DATAY    ;READ POINTER.
        MOV     AX,[BX]         ;READ SINE TABLE.
        ADD     CNT_DATAY,2     ;UPDATE COUNTER.
        ADD     Y_AXITL,AX      ;ADD LOW WORD Y-AXIT.
        ADC     Y_AXITH,0       ;ADD CARRY FROM LOW WORD Y-AXIT.

        MOV     BX,CNT_DATAX
        MOV     AX,[BX]
        SUB     CNT_DATAX,2
        ADD     X_AXITL,AX
        ADC     X_AXITH,0
      PUSH  CX
        MOV     CX,Y_AXITH      ;READ DATA ON Y_AXITH.
        MOV     BX,Y_AXITL      ;READ LOW WORD.
        MOV     DX,RES_2
        MOV     AX,RES_1
        CALL    T_FITING        ;TABLE FITTING SCALE.

        SUB     AX,AX
        MOV     AX,Y_STARTL     ;READ RADIUS MSW.
        SUB     AX,REMAINW0
        MOV     AX,Y_STARTH
        SBB     AX,REMAINW1
        JNC     FIND_NEXT       ;IF I RECORD > REMAINW
                                ;THEN READY.
        POP     CX
        MOV     ANGLE_0,CX      ;SAVE ANGLE COUNTER FIRST POINT.
        MOV     AX,CNT_DATAX    ;READ TABLE POINTER X.
        MOV     X_ANGLE,AX      ;SAVE TABLE POINTER X FIRST POINT.
        MOV     AX,CNT_DATAY    ;READ TABLE POINTER Y.
        MOV     Y_ANGLE,AX      ;SAVE TABLE POINTER Y FIRST POINT.
        MOV     AX,X_AXITH
        MOV     X_AXIT1H,AX     ;SAVE DATA MSW FIRST POINT.
        MOV     AX,X_AXITL
        MOV     X_AXIT1L,AX     ;SAVE DATA LSW FIRST POINT.
        MOV     AX,Y_AXITH
        MOV     Y_AXIT1H,AX
        MOV     AX,Y_AXITL
        MOV     Y_AXIT1L,AX
        DEC     CX              ;COUNT FOR NEXT COUNT.
        JMP     FINAL_POINT

FIND_NEXT:

        POP     CX
        LOOP    FIND_LOOP
        MOV     ANGLE_0,0FFFFH  ;SET OVER FLOW ANGLE.
        RET                     ;END OF PROCESS.

FINAL_POINT:

        MOV     BX,CNT_DATAY    ;READ POINTER.
        MOV     AX,[BX]         ;READ SINE TABLE.
        ADD     CNT_DATAY,2     ;UPDATE COUNTER.
        ADD     Y_AXITL,AX      ;ADD LOW WORD Y-AXIT.
        ADC     Y_AXITH,0       ;ADD CARRY FROM LOW WORD Y-AXIT.
        MOV     BX,CNT_DATAX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     AX,[BX]
          SUB     CNT_DATAX,2
          ADD     X_AXITL,AX
          ADC     X_AXITH,0
       PUSH    CX
          MOV     CX,Y_AXITH        ;READ DATA ON Y_AXITH.
          MOV     BX,Y_AXITL        ;READ LOW WORD.
          MOV     DX,RES_2
          MOV     AX,RES_1
          CALL    T_FITING          ;TABLE FITTING SCALE.
          SUB     BX,BX
          MOV     BX,Y_ENDH
          MOV     AX,Y_ENDL
          SUB     AX,REMAINW0
          SBB     BX,REMAINW1
          JNC     FIND_NEXT1        ;IF Y_END > REMAIN THEN
                                    ;FIND_NEXT1.
       POP     CX
          MOV     ANGLE_1,CX        ;SAVE COUNTER END POINT.

FIND_READY:
          RET

FIND_NEXT1:

       POP     CX
          LOOP    FINAL_POINT
          MOV     ANGLE_1,0FFFFH    ;SET OVER FLOW.
          RET

SEARCH_J  ENDP


TEST      PROC    NEAR

;-----------------------
;  INITIAL VALUE OF ARC
;-----------------------

          MOV     AX,1200H          ;DEFIND VECTOR SPEED.
          MOV     V_SPEED,AX

          MOV     X_ENDL,00         ;DEFAULT 200.
          MOV     X_ENDH,00
          MOV     Y_ENDL,00         ;2000
          MOV     Y_ENDH,00

          CALL    L_Q1

          MOV     X_ENDL,00         ;DEFIND END POINT OF ARC.
          MOV     X_ENDH,00
          MOV     Y_ENDL,10000      ;2000
          MOV     Y_ENDH,00

          CALL    L_Q4


          MOV     I_RECORD,1000     ;3000    ;DEFIND CENTER OF ARC.
          MOV     J_RECORD,0        ;00
          MOV     I_RECORD+2,0
          MOV     J_RECORD+2,0


          MOV     X_ENDL,1000       ;DEFIND END POINT OF ARC.
          MOV     X_ENDH,00
          MOV     Y_ENDL,1000       ;2000
          MOV     Y_ENDH,00

          CALL    FFW_Q3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     I_RECORD,1500    ;3000    ;DEFIND CENTER OF ARC.
MOV     J_RECORD,0       ;00
MOV     I_RECORD+2,0
MOV     J_RECORD+2,0


MOV     X_ENDL,1500      ;DEFIND END POINT OF ARC.
MOV     X_ENDH,00
MOV     Y_ENDL,1500      ;2000
MOV     Y_ENDH,00

CALL    RFW_Q3

MOV     X_ENDL,00
MOV     X_ENDH,00
MOV     Y_ENDL,10000
MOV     Y_ENDH,00

CALL    L_Q4

MOV     AX,0A000H        ;DEFIND VECTOR SPEED.
MOV     V_SPEED,AX

MOV     X_ENDL,100       ;DEFIND END POINT OF ARC.
MOV     X_ENDH,00
MOV     Y_ENDL,00        ;2000
MOV     Y_ENDH,00

CALL    L_Q2

MOV     X_ENDL,00        ;DEFIND END POINT OF ARC.
MOV     X_ENDH,00
MOV     Y_ENDL,10000     ;2000
MOV     Y_ENDH,00

CALL    L_Q1

MOV     X_ENDL,00        ;DEFIND END POINT OF ARC.
MOV     X_ENDH,00
MOV     Y_ENDL,12500     ;2000
MOV     Y_ENDH,00

CALL    L_Q1

MOV     X_ENDL,2600      ;DEFIND END POINT OF ARC.
MOV     X_ENDH,00
MOV     Y_ENDL,00        ;2000
MOV     Y_ENDH,00

CALL    L_Q4

RET


CALL    FFW_Q2
CALL    RFW_Q3
CALL    FFW_Q4
CALL    RFW_Q1
RET

CALL    FFW_Q1
CALL    RFW_Q4
CALL    FFW_Q3
CALL    RFW_Q2

RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     I_RECORD,2500    ;3000    ;DEFIND CENTER OF ARC.
                MOV     J_RECORD,00      ;00
                MOV     I_RECORD+2,0
                MOV     J_RECORD+2,0

                MOV     X_ENDL,00        ;DEFIND END POINT OF ARC.
                MOV     X_ENDH,00
                MOV     Y_ENDL,2500      ;2000
                MOV     Y_ENDH,00

                CALL    PLT_FORWARDQ4

                MOV     I_RECORD,1250    ;3000    ;DEFIND CENTER OF ARC.
                MOV     J_RECORD,00    ;00
                MOV     I_RECORD+2,0
                MOV     J_RECORD+2,0

                MOV     X_ENDL,00        ;DEFIND END POINT OF ARC.
                MOV     X_ENDH,00
                MOV     Y_ENDL,1250      ;2000
                MOV     Y_ENDH,00

                CALL    PLT_FORWARDQ4

                RET

;--------------------------------
;  PLT_S  CW  Q_
;--------------------------------

L_Q1:

                MOV     QUARDANT,80H     ;DEFIND QUARDANT.
                CALL    LINEAR
                CALL    LOAD_DSP
                RET

L_Q2:
                MOV     QUARDANT,40H     ;DEFIND QUARDANT.
                CALL    LINEAR
                CALL    LOAD_DSP
                RET
L_Q3:
                MOV     QUARDANT,0C0H    ;DEFIND QUARDANT.
                CALL    LINEAR
                CALL    LOAD_DSP
                RET
L_Q4:
                MOV     QUARDANT,00H     ;DEFIND QUARDANT.
                CALL    LINEAR
                CALL    LOAD_DSP
                RET


LINEAR          PROC    NEAR

                MOV     AX,X_ENDL
                MOV     BX,02
                MUL     BX
                MOV     OUTPUT_X,AX
                MOV     AX,Y_ENDL
                MUL     BX
                MOV     OUTPUT_Y,AX

                RET
LINEAR          ENDP

FFW_Q1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     QUARDANT,80H        ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ4
                RET

FFW_Q2:

                MOV     QUARDANT,0C0H       ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ4
                RET

FFW_Q3:

                MOV     QUARDANT,04FH       ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ4
                RET

FFW_Q4:

                MOV     QUARDANT,0FH      ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ4
                RET


RFW_Q1:

                MOV     QUARDANT,80H        ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ1
                RET

RFW_Q2:

                MOV     QUARDANT,0C0H       ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ1
                RET

RFW_Q3:

                MOV     QUARDANT,04FH       ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ1
                RET

RFW_Q4:

                MOV     QUARDANT,0FH      ;DEFIND QUARDANT.
                CALL    PLT_FORWARDQ1
                RET

PLT_FORWARDQ1:
                CALL    SEARCH_J            ;SEARCH START POINT AND END POINT.

                MOV     CX,X_AXIT1H          ;READ DATA ON Y_AXITH.
                MOV     BX,X_AXIT1L          ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                MOV     AX,REMAINWO
                MOV     BX,02
                MUL     BX
                MOV     INIVAL_Y,AX

                MOV     CX,Y_AXIT1H         ;READ DATA ON Y_AXITH.
                MOV     BX,Y_AXIT1L         ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                MOV     AX,REMAINWO
                MOV     BX,02
                MUL     BX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     INTVAL_X,AX

                MOV     CX,ANGLE_0      ;SET TBL COUNTER.
                SUB     CX,ANGLE_1
                DEC     CX
LOOP_PLTS3:
                PUSH    CX
                SUB     X_ANGLE,2       ;UPDATE COUNTER.
                MOV     BX,X_ANGLE      ;READ POINTER.
                SUB     AX,AX
                MOV     AX,[BX]         ;READ SINE TABLE.
                ADD     X_AXIT1L,AX
                ADC     X_AXIT1H,0
                MOV     CX,X_AXIT1H
                MOV     BX,X_AXIT1L
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING        ;TABLE FITTING SCALE.

                CALL    DATAFORY

                SUB     AX,AX
                ADD     Y_ANGLE,2       ;UPDATE COUNTER.
                MOV     BX,Y_ANGLE      ;READ POINTER.
                MOV     AX,[BX]         ;READ SINE TABLE.
                ADD     Y_AXIT1L,AX
                ADC     Y_AXIT1H,0

                MOV     CX,Y_AXIT1H
                MOV     BX,Y_AXIT1L
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING        ;TABLE FITTING SCALE.

                CALL    DATAFORX
                CALL    LOAD_DSP

                POP     CX
                LOOP    LOOP_PLTS3

                MOV     CX,X_AXIT1H       ;READ DATA ON Y_AXITH.
                MOV     BX,X_AXIT1L       ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                CALL    DATAFORY

                SUB     AX,AX
                MOV     CX,Y_AXIT1H       ;READ DATA ON Y_AXITH.
                MOV     BX,Y_AXIT1L       ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                CALL    DATAFORX
                CALL    LOAD_DSP

                RET


PLT_FORWARDQ4:
                CALL    SEARCH_J        ;SEARCH START POINT AND END POINT.

                MOV     CX,X_AXIT1H       ;READ DATA ON Y_AXITH.
                MOV     BX,X_AXIT1L       ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     AX,REMAINW0
                MOV     BX,02
                MUL     BX
                MOV     INTVAL_X,AX

                MOV     CX,Y_AXIT1H       ;READ DATA ON Y_AXITH.
                MOV     BX,Y_AXIT1L       ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                MOV     AX,REMAINW0
                MOV     BX,02
                MUL     BX
                MOV     INTVAL_Y,AX


                MOV     CX,ANGLE_0        ;SET TBL COUNTER.
                SUB     CX,ANGLE_1
                DEC     CX
LOOP_PLIS2:
                PUSH    CX
                SUB     X_ANGLE,2         ;UPDATE COUNTER.
                MOV     BX,X_ANGLE        ;READ POINTER.
                SUB     AX,AX
                MOV     AX,[BX]           ;READ SINE TABLE.
                ADD     X_AXIT1L,AX
                ADC     X_AXIT1H,0
                MOV     CX,X_AXIT1H
                MOV     BX,X_AXIT1L
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING          ;TABLE FITTING SCALE.

                CALL    DATAFORX

                SUB     AX,AX
                ADD     Y_ANGLE,2         ;UPDATE COUNTER.
                MOV     BX,Y_ANGLE        ;READ POINTER.
                MOV     AX,[BX]           ;READ SINE TABLE.
                ADD     Y_AXIT1L,AX
                ADC     Y_AXIT1H,0

                MOV     CX,Y_AXIT1H
                MOV     BX,Y_AXIT1L
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING          ;TABLE FITTING SCALE.

                CALL    DATAFORY
                CALL    LOAD_DSP

                POP     CX
                LOOP    LOOP_PLIS2

                MOV     CX,X_AXIT1H        ;READ DATA ON Y_AXITH.
                MOV     BX,X_AXIT1L        ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING

                CALL    DATAFORX

                SUB     AX,AX
                MOV     CX,Y_AXIT1H        ;READ DATA ON Y_AXITH.
                MOV     BX,Y_AXIT1L        ;READ LOW WORD.
                MOV     DX,RES_2
                MOV     AX,RES_1
                CALL    T_FITING
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    DATAFORY
                CALL    LOAD_DSP

                RET

TEST            ENDP

DATAFORY        PROC    NEAR

                MOV     AX,INTVAL_Y
                MOV     OUTPUT_Y,AX
                MOV     AX,REMAINWO
                MOV     INTVAL_Y,AX
                SUB     AX,OUTPUT_Y
                MOV     BX,02
                MUL     BX

                MOV     OUTPUT_Y,AX

                JNC     GOOD_Y
                MOV     AX,00
                MOV     OUTPUT_Y,AX

GOOD_Y:         RET
DATAFORY        ENDP

DATAFORX        PROC    NEAR
                MOV     AX,INTVAL_X
                MOV     OUTPUT_X,AX
                MOV     AX,REMAINWO
                MOV     INTVAL_X,AX
                SUB     AX,OUTPUT_X
                MOV     BX,02
                MUL     BX
                MOV     OUTPUT_X,AX
                JNC     GOOD_X
                MOV     AX,00
                MOV     OUTPUT_X,AX

GOOD_X:         RET

DATAFORX        ENDP

 LOAD_DSP       PROC    NEAR
                MOV     AX,OUTPUT_X
                ADD     AX,OUTPUT_Y
                JNZ     START_LOAD
                RET
  START_LOAD:
                MOV     DX,306H
  IW1:          IN      AL,DX
                AND     AL,01       ;CHECK BIT 0 PORT C.
                JZ      IW1
                MOV     AL,3FH
                OR      AL,QUARDANT
                OUT     DX,AL
  I32010:
                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
                OUT     DX,AL

                MOV     DX,304H     ;PORT A.
                IN      AL,DX
                AND     AL,03
                CMP     AL,00       ;X-PARAMETER.
                JZ      NEXT_POINT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JMP     I32010

NEXT_POINT:

                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
                OUT     DX,AL

WAIT_DSP4:
                MOV     DX,304H
                IN      AL,DX
                AND     AL,03
                CMP     AL,03       ;DSP CODE 03 = X_PARA.
                JNZ     WAIT_DSP4
NEX_DATA:

                MOV     AX,OUTPUT_X
                MOV     DX,300H     ;PORT A.
                OUT     DX,AL
                INC     DX          ;PORT B.
                MOV     AL,AH
                OUT     DX,AL

DATA_3:
                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
                OUT     DX,AL
WAIT_DSP1:
                MOV     DX,304H
                IN      AL,DX
                AND     AL,03
                CMP     AL,02       ;DSP CODE 02 = Y_PARA.
                JNZ     WAIT_DSP1
DATA_2:
                MOV     AX,OUTPUT_Y
                MOV     DX,300H     ;PORT A.
                OUT     DX,AL
                INC     DX          ;PORT B.
                MOV     AL,AH
                OUT     DX,AL

                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
                OUT     DX,AL

WAIT_DSP2:
                MOV     DX,304H
                IN      AL,DX
                AND     AL,03
                CMP     AL,01       ;DSP CODE 01 = V.
                JNZ     WAIT_DSP2
DATA_1:

                MOV     AX,V_SPEED
                MOV     DX,300H     ;PORT A.
                OUT     DX,AL
                INC     DX          ;PORT B.
                MOV     AL,AH
                OUT     DX,AL

                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                OUT     DX,AL

WAIT_DSP3:
                MOV     DX,304H
                IN      AL,DX
                AND     AL,03
                CMP     AL,00       ;DSP CODE 00 = DATA READY.
                JNZ     WAIT_DSP3

                MOV     AL,0FH
                OR      AL,QUARDANT
                MOV     DX,306H
                OUT     DX,AL

                MOV     AL,00
                MOV     DX,302H     ;PORT C
                OUT     DX,AL
                MOV     AL,03
                OUT     DX,AL
WAIT_DSP:
                NOP
                MOV     DX,306H     ;READY BIT ON DSP START RUN.
                IN      AL,DX
                AND     AL,01
                JNZ     WAIT_DSP

                MOV     AL,3FH
                OR      AL,QUARDANT
                OUT     DX,AL

FAIL_LOAD:      RET

LOAD_DSP        ENDP

   COSINI       PROC    NEAR

  START_TEST:

                MOV     I_RECORD,00     ;DEFIND CENTER OF ARC.
                MOV     J_RECORD,7000
                MOV     I_RECORD+2,0
                MOV     J_RECORD+2,0
                MOV     X_ENDL,7000     ;DEFIND END POINT OF ARC.
                MOV     X_ENDH,0

                MOV     Y_ABSH,0        ;DEFIND REFERENCE POSITION
                MOV     Y_ABSL,0        ;OF ARC.
                MOV     X_ABSH,0
                MOV     X_ABSL,0
                CALL    ARC
  END_TEST:
                SUB     BX,BX           ;ZERO BX.
                MOV     Y_AXITL,BX
                MOV     Y_AXITH,BX
                MOV     X_AXITL,BX
                MOV     X_AXITH,BX

                MOV     BX,OFFSET DATA_BUF
                MOV     CNT_DATAY,BX
                ADD     BX,3998         ;OFFSET TO END OF SINE TABLE
                MOV     CNT_DATAX,BX    ;(N+1 = END_TBL - START_TBL + 1).

                MOV     CX,2000
  SIN_LOOP:     MOV     BX,CNT_DATAY    ;READ POINTER.
                MOV     AX,[BX]         ;READ SINE TABLE.
                ADD     CNT_DATAY,2     ;UPDATE COUNTER.
                SUB     BX,BX           ;CLEAR CARRY FLAG.
                ADD     Y_AXITL,AX      ;ADD LOW WORD Y-AXIT AND
                ADC     Y_AXITH,0       ;ADD CARRY FROM LOW WORD Y-AXIT.
              PUSH    CX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     BX,CNT_DATAX        ;READ POINTER.
            MOV     AX,[BX]             ;READ SINE TABLE.
            SUB     CNT_DATAX,2         ;UPDATE COUNTER.
            SUB     BX,BX
            ADD     X_AXITL,AX          ;ADD LOW WORD Y-AXIT AND
                                        ;RESTORE DATA Y-AXIT.
            ADC     X_AXITH,0           ;ADD CARRY FROM LOW WORD Y-AXIT.
            MOV     CX,X_AXITH          ;READ DATA ON Y_AXITH.
            MOV     BX,X_AXITL          ;READ LOW WORD.
            MOV     DX,RES_2
            MOV     AX,RES_1
            CALL    T_FITING            ;TABLE FITTING SCALE.
            CALL    CLIP
          POP     CX
            LOOP    SIN_LOOP

            RET

 COSINT     ENDP

CLIP        PROC    NEAR
            SUB     BX,BX
            MOV     BX,I_RECORD+2
            MOV     AX,I_RECORD
            SUB     AX,REMAINW0
            SBB     BX,REMAINW1
            JNC     NO_PLOT             ;IF I_RECORD < REMAIN
                                        ;THEN NO PLOT.
            SUB     BX,BX
            MOV     BX,X_ENDH
            MOV     AX,X_ENDL
            SUB     AX,REMAINW0
            SBB     BX,REMAINW1
            JC      NO_PLOT

            MOV     DX,OFFSET DATA_X
            MOV     AX,X_ABSL
            ADD     AX,REMAINW0         ;ABSOLUTE OF POSITION X
            CALL    BIN_ASCII           ;AT REFERENCE POINT.
            MOV     CX,Y_AXITH
            MOV     BX,Y_AXITL
            MOV     DX,RES_2
            MOV     AX,RES_1
            CALL    T_FITING
            MOV     DX,OFFSET DATA_Y
            MOV     AX,Y_ABSL
            ADD     AX,REMAINW0         ;ABSOLUTE OF POSITION Y
                                        ;AT REFERENCE POINT.
            CALL    BIN_ASCII
            CALL    PLT_S
NO_PLOT:

CLIP        ENDP
;-----------------------
; INTEGER SQUARE ROOT.
;-----------------------
;INPUT   ARC_1,ARC_2,ARC_3,ARC_4
;OUTPUT  RES_1,RES_2



;-----------------------
; INTEGER SUBSTRACTION.
;-----------------------
;32-BIT BY 32-BIT SUBTRACTION.
;INPUT  DX:BX
;       CX:AX
;RESULT DX:AX

ISUB        PROC    NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PUSH    BX
            PUSH    CX
            SUB     BX,AX       ;SUB BX BY AX
            SBB     DX,CX       ;AND DX BY CX.
            MOV     DX,CX       ;STORE RESULT ON
            MOV     AX,BX       ;DX,AX HIGH ORDER TO LOW.
            POP     CX
            POP     BX
            RET
ISUB        ENDP

;-----------------------
; INTEGER ADDITION.
;-----------------------
;32-BIT BY 32-BIT ADDITION.
;INPUT  DX:BX
;       CX:AX
;RESULT DX:AX

IADD        PROC    NEAR
            PUSH    BX
            PUSH    CX
            ADD     AX,BX       ;SUB BX BY AX
            ADC     DX,CX       ;AND DX BY CX.
            POP     CX
            POP     BX
            RET
IADD        ENDP


;SUBROUTINE TO DISPLAY A MESSAGE ON THE SCREEN.

;ENTER TO BX -> MESSAGE TO BE DISPLAYED.
;MESSAGE IS ASSUME TO BE 30 CHARACTERS LONG.

DISPLAY     PROC    NEAR

            PUSH    AX
            PUSH    BX
            PUSH    CX
            PUSH    DX

            MOV     AH,02       ;ATTR OF SCREEN
            MOV     BH,00
            MOV     DX,0A14H    ;CURSOR POSITION ROW,COL
            INT     10H         ;CALL DOS ROUTINE DISPLAY

            POP     DX
            POP     CX
            POP     BX
            POP     AX

            MOV     CX,30       ;NUMBER OF CHARACTER TO DISPLAY
DISP1:      MOV     AL,[BX]     ;GET NEXT CHARACTOR TO DISPLAY
            CALL    DISPCHAR    ;DISPLAY IT
            INC     BX          ;POINT TO NEXT CHARACTER
            LOOP    DISP1       ;DO IT 30 TIME
            MOV     AL,0DH      ;CARRIAGE RETURN
            CALL    DISPCHAR
;            MOV     AL,0AH     ;LINE FEED
;            CALL    DISPCHAR
            RET                 ;RETURN TO CALLER OF 'DISPLAY'
DISPLAY     ENDP

;SUBROUTINE TO DISPLAY A CHARACTER ON THE SCREEN.
;ENTER WITH AL=CHARACTER TO BE DISPLAYED.
;USE VIDEO INTERFACE IN BIOS.

DISPCHAR PROC   NEAR
         PUSH   BX          ;SAVE BX REGISTER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     BX,0        ;SELECT DISPLAY PAGE0
        MOV     AH,14       ;FUNCTION CODE FOR 'WHITE'
        INT     10H         ;CALL VIDEO DRIVE TO BIOS
        POP     BX          ;RESTORE BX REGISTER
        RET                 ;RETURN TO CALLER OF 'DISPCHAR'

DISPCHAR ENDP

;-----------------------------------
; CALL BIOS CURSOR OFF  USED INT10.
;-----------------------------------
CUR_OFF  PROC
        PUSH    AX
        PUSH    CX

        MOV     AH,01
        MOV     CH,20H
        INT     10H

        POP     CX
        POP     AX

        RET

CUR_OFF  ENDP

;-----------------------------------
;       CLEAR SCREEN MONITOR
; BY WRITE CHARACTER SPACE ON SCREEN.
;-----------------------------------

CLRSCN   PROC                ;CLEAR SEREEN MONITOR.
        PUSH    AX
        PUSH    BX
        PUSH    CX
        MOV     AX,ES
        PUSH    AX

        MOV     AX,0B800H    ;DEFIND TAGET ADDRESS
        MOV     ES,AX        ;DEFIND EXTRA SEGMENT
        MOV     DI,0         ;INITIAL OFFSET ADDR INTO SEGMENT
        MOV     AL,' '       ;BLANK CHARACTER TO FILL ADAPTER MEMORY
        MOV     AH,02        ;THIS WILL BE THE ATTRIBUTE BYTE
                             ;ALL CHARACTERS DISPLAYED WILL BE
                             ;GREEN BACKGROUND
        MOV     CX,8000      ;INITIAL ALL OF MEMORY
        CLD                  ;SO STOSW GOES FORWARD
        REP     STOSW        ;BLANK OUT THE ADAPTER MEMORY

        POP     AX
        MOV     AX,ES
        POP     CX
        POP     BX
        POP     AX

        RET

CLRSCN   ENDP

;-----------------------------------
;     BINARY FORMAT TO ASCII
;   USING ASCII TABLE PROCDURE.
;-----------------------------------
ASCII    PROC
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX

        MOV     SI,OFFSET BUFF       ;DEFIND SOURCE POINTER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     DI,OFFSET CHAR_BUFF  ;DEFIND DI TO CHAR_BUFF
        CALL    ASCII1
        MOV     SI,OFFSET VBUFF      ;DEFIND SOURCE POINTER.
        MOV     DI,OFFSET VELOCITY   ;DEFIND DI TO VELOCITY ADDRESS.
        CALL    ASCII1
        MOV     BX,OFFSET VELOCITY
        MOV     DX,0C14H             ;CURSOR POSITION ROW,COL
        CALL    DISP_IT

        POP     DX
        POP     CX
        POP     BX
        POP     AX
        RET
ASCII   ENDP

;----------------------------
;  HEX  DECIMAL TO ASCII
;----------------------------

ASCII1  PROC                         ;CONVERSE BINARY TO ASCII
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DI
        PUSH    SI

        MOV     CX,08H               ;COUNT EIGHT BYTES
        MOV     AX,DI
        ADD     AX,28                ;ADJUST POINTER
        MOV     DI,AX                ;MODIFY DI TO DI + 28
        SUB     AX,AX                ;ZERO AX.
CONVERT: MOV    AL,[SI]              ;READ BUFF
        PUSH    AX                   ;SAVE AL TO BL
        AND     AL,0FH              ;MASK HIGH BYTE
        MOV     BX,OFFSET ASCII_TBL
        PUSH    BX
        ADD     BX,AX                ;ASCII TABLE LOOK UP.
        MOV     AL,[BX]              ;READ ASCII TABLE.
        MOV     [DI],AL              ;PUSH IT INTO CHAR BUFF
        DEC     DI                   ;DEC CHAR POINTER TO NEXT BYTE.
        POP     BX                   ;RESTORE TABLE OFFSET ON BX.
        POP     AX                   ;RESTORE AL
        AND     AX,0F0H              ;MASK LOW BYTE
        DIV     SIGTEEN              ;SHIFT DATA TO RIGHT 4 BIT.
        ADD     BX,AX                ;ASCII TABLE LOOK UP.
        MOV     AL,[BX]              ;READ TABLE.
        MOV     [DI],AL              ;PUSH IT INTO CHAR BUFF
        DEC     DI                   ;DEC CHAR BUFF POINTER
        INC     SI                   ;INC SOURCE POINTER
        LOOP    CONVERT              ;DO IT EIGHT TIME

        POP     SI
        POP     DI
        POP     CX
        POP     BX
        POP     AX
        RET

ASCII1  ENDP


REPORT1 PROC                         ;REPORT STATUS BYTE.

        MOV     SI,OFFSET STATUS     ;DEFIND SOURCE POINTER
        MOV     DI,OFFSET STATUS1    ;DEFIND DI TO STATUS1.
        MOV     DX,0E14H             ;DEFIND CURSOR POSITION ROW,COL.
        MOV     BX,OFFSET STATUS1
        CALL    REPORT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV     SI,OFFSET RDSIGNAL1
          MOV     DI,OFFSET STATUS2
          MOV     DX,1014H              ;DEFIND CURSOR POSITION ROW,COL.
          MOV     BX,OFFSET STATUS2
          CALL    REPORT
          MOV     SI,OFFSET RDSIGNAL
          MOV     DI,OFFSET STATUS2-9
          MOV     DX,1014H
          MOV     BX,OFFSET STATUS2           ;DEFIND DISPLAY LOCATION
          CALL    REPORT
          RET

REPORT1   ENDP

REPORT    PROC
          PUSH    AX
          PUSH    BX
          PUSH    CX
          PUSH    DX                    ;CURSOR POSITION ROW,COL

          MOV     CX,08                 ;COUNT 8

          MOV     AX,DI                 ;READ STATUS.
          ADD     AX,28                 ;ADJUST POINTER
          MOV     DI,AX                 ;MODIFY DI TO DI + 28
          MOV     AL,[SI]               ;READ STATUS.
REP_BIT:  MOV     BL,AL                 ;SAVE AL TO BL
          AND     AL,01                 ;MASK HIGH BYTE
          ADD     AL,30H                ;CHANGE TO ASCII
          MOV     [DI],AL               ;PUSH IT INTO CHAR BUFF
          DEC     DI                    ;DEC CHAR POINTER
          MOV     AL,BL                 ;RESTORE AL
          ROR     AL,1

          LOOP    REP_BIT               ;DO IT EIGHT TIME

          POP     DX
          POP     CX
          POP     BX
          POP     AX

DISP_IT:  PUSH    BX
          MOV     CX,60H
          MOV     AH,02       ;ATTR OF SCREEN
          MOV     BH,00
         ;MOV     DX,0C14H    ;CURSOR POSITION ROW,COL
          INT     10H         ;CALL DOS ROUTINE DISPLAY
          POP     BX

          MOV     CX,30       ;NUMBER OF CHARACTER TO DISPLAY
DISPST:   MOV     AL,[BX]     ;GET NEXT CHARACTOR TO DISPLAY
          CALL    DISPCHAR    ;DISPLAY IT
          INC     BX          ;POINT TO NEXT CHARACTER
          LOOP    DISPST      ;DO IT 30 TIME
          MOV     AL,0DH      ;CARRIAGE RETURN
          CALL    DISPCHAR

          RET

REPORT    ENDP


READ      PROC

          PUSH  DX
          PUSH  BX
          PUSH  AX

          MOV   BX,OFFSET BUFF+3 ;FOUR BYTE ( 2 WORDS )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   DX,200H          ;SET POINTER TO PORT COMMAND.
        MOV   AL,RDRP
        CALL  WAIT
        OUT   DX,AL            ;WRITE COMMAND READ REAL POSITION.
        INC   DX
        CALL  WAIT
        IN    AL,DX            ;READ DATA FIRST BYTE MSB.
        MOV   [BX],AL
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ DATA FIRST BYTE LSB.
        MOV   [BX],AL
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ DATA SECOND BYTE MSB.
        MOV   [BX],AL
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ DATA SECOND BYTE LSB.
        MOV   [BX],AL

BIN_BCD: MOV  BX,OFFSET BUFF
        FILD  QWORD PTR [BX]   ;PUSH HEX TO REG OF MATH CO.
        FBSTP TBYTE PTR BUFF   ;READ SHORT BCD FROM MATH CO.

RD_VEL:  MOV  BX,OFFSET VBUFF+3
        MOV   DX,200H
        MOV   AL,RDRV          ;READ REAL VELOCITY COMMAND.
        CALL  WAIT
        OUT   DX,AL
        INC   DX
        CALL  WAIT
        IN    AL,DX
        MOV   [BX],AL          ;SAVE MSB
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ SECOND BYTE.
        MOV   [BX],AL          ;SAVE SECOND BYTE.
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ THERD BYTE.
        MOV   [BX],AL          ;SAVE THERD BYTE.
        DEC   BX
        CALL  WAIT
        IN    AL,DX            ;READ LSB
        MOV   [BX],AL          ;SAVE LSB

        MOV   BX,OFFSET VBUFF
        FILD  QWORD PTR [BX]   ;PUSH HEX TO REG OF MATH CO.
        FIMUL DWORD PTR [SCALE]
        FBSTP TBYTE PTR VBUFF  ;READ SHORT BCD FROM MATH CO.

RD_STAT: MOV  BX,OFFSET STATUS
        MOV   DX,200H
        CALL  WAIT
        IN    AL,DX            ;READ STATUS FROM LM 628.
        MOV   [BX],AL          ;PUSH IT INTO MEMORY.
        MOV   AL,RDSIGS
        CALL  WAIT
        OUT   DX,AL            ;WRITE COMMAND RDSIG TO LM 628.
        INC   DX
        CALL  WAIT
        IN    AL,DX            ;READ DATA FROM LM 628
        INC   BX
        MOV   [BX],AL
        CALL  WAIT
        IN    AL,DX
        INC   BX
        MOV   [BX],AL
        DEC   DX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
        POP   AX
        POP   BX
        POP   DX

        RET

READ    ENDP

WAIT    PROC

        PUSH  AX
        PUSH  DX

        MOV   DX,200H
WAIT1:  IN    AL,DX
        AND   AX,01
        JNZ   WAIT1

        POP   DX
        POP   AX

        RET

WAIT    ENDP

;INITIAL PERIPHERAL INTERFACE ADAPTER

INIT    PROC
        PUSH DX
        PUSH CX
        PUSH BX
        PUSH AX

        MOV   DX,20DH          ;DEFIND PORT ADDRESS OF 6821
        MOV   AX,00            ;DEFIND MODE OF CONTROL REG
        OUT   DX,AL            ;WRITE ZERO INTO CONTROL REG OF 6821
        MOV   AX,0FFFFH          ;DEFIND MODE OF DIRECTION REG
        DEC   DX               ;RS1 RS2 = 00
        OUT   DX,AL            ;WRITE OFFH IN TO DIRECTION REG
        INC   DX               ;CHANGE RS1 RS2 TO ADDRESS 01
        OUT   DX,AL          ;WIRTE OFFH IN TO CONTROL REG

        POP   AX
        POP   BX
        POP   CX
        POP   DX

        RET
INIT    ENDP

SAVE_TBL PROC    NEAR

        PUSH    BX
        MOV     BX,[CNT_DATA]
        MOV     [BX],AL
        INC     CNT_DATA
        POP     BX
        RET

SAVE_TBL ENDP

START   ENDP
CODE    ENDS
        END     START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้