การสร้างต้นแบบคอมพิวเตอร์ควบคุมเครื่องกลึง

INDUSTRIAL LATHE COMPUTER CONTROLLER

โดย

นายพิสุทธิ์ อภิชยกุล 36014297

นายอนุรัตน์ ธนะโสธร 36014541

อาจารย์ที่ปรึกษา

รศ.ดร.โยธิน เปรมปราณีรัชต์

อาจารย์กวิน สนธิเพิ่มพูน

วัน เดือน ปี ๑๑ ต.ค. 2541

เลขทะเบียน 038044

เลขเรียกหนังสือ T39064 พ 486 ก.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

038044

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2539

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ต้นแบบคอมพิวเตอร์ควบคุมเครื่องกลึง

INDUSTRIAL LATHE COMPUTER CONTROLLER

ผู้จัดทำ

1. นายพิสุทธิ์ อภิชยกุล
2. นายอนุรัตน์ ธนะโสธร

อาจารย์ที่ปรึกษา
(รศ.ดร.โยธิน เปรมปราณีรัชต์)

อาจารย์ที่ปรึกษา
(อาจารย์กวิน สนธิเพิ่มพูน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างต้นแบบคอมพิวเตอร์ควบคุมเครื่องกลึง

นายพิสุทธิ์ อภิชยกุล

นายอนุรัตน์ ธนะไสธร

รศ.ดร.โยธิน เปรมปราณีรัชต์ อาจารย์ที่ปรึกษา

อาจารย์กวิน สนธิเพิ่มพูน อาจารย์ที่ปรึกษา

ปีการศึกษา 2539

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาถึงการควบคุมตำแหน่งและความเร็วของดีซีเซอร์โวมอเตอร์ และอินดัคชั่นมอเตอร์ โดยอาศัยการควบคุมผ่านทางการ์ดควบคุมสำเร็จรูป PMD 5650 ซึ่งสามารถทำงานและสั่งงานบนเครื่องไมโครคอมพิวเตอร์โดยตรงได้ เพื่อนำไปประยุกต์ใช้ในการควบคุมเครื่องกลึง บนพื้นฐานไมโครคอมพิวเตอร์ ซึ่งต้องเขียนซอฟท์แวร์ขึ้นมาเพื่อควบคุมการ์ดให้สามารถรับและทำตามคำสั่งที่เข้ามาในรูปคำสั่งมาตราฐาน G Code นอกจากนั้นในส่วนของเครื่องกลึงที่จะนำมาใช้ก็ต้องอยู่ในสภาพพร้อมที่จะติดตั้งระบบควบคุมแบบคอมพิวเตอร์ได้

จากผลงานที่ได้ทำขึ้นจะได้ต้นแบบคอมพิวเตอร์ควมคุมเครื่องกลึงซีเอ็นซี ที่มีราคาไม่แพง ใช้ต้นทุนในการผลิตต่ำ แต่สามารถใช้งานได้อย่างมีประสิทธิภาพทัดเทียมกับเครื่องกลึงที่สั่งเข้ามาจากต่างประเทศ อีกทั้งยังนำไปประยุกต์ใช้งานในเชิงอุตสาหกรรมได้อย่างกว้างขวาง

ABSTRACT

This thesis is studying the DC servo motor and induction motor velocity and position control, by using the Servo-motor controller card model PMD 5650 which can function and be operated directly via microcomputer. It is applied for the designing of CNC lathe machine based on microcomputer. Since the controlling card cannot be input by the standard G code, it is essential to program the specific software for making the card adaptable and operatable by those standard instructions. Moreover, the CNC lathe machine should also be ready for any computer control system installation, indeed.

From the result of the tedious work, we acquired the prototype of the CNC lathe machine, which is inexpensive, low operating cost, and perform machining job efficiently as well as other import CNC lathe machines from abroad. It can be used in wide range industry fields.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญรูป	II
สารบัญตาราง	IV
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	2
บทที่ 2 ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี	3
2.1 พัฒนาการของเครื่องจักรกลเอ็นซีและซีเอ็นซี	3
2.1.1 ความหมายของเอ็นซีและซีเอ็นซี	3
2.1.2 ความแตกต่างระหว่างเครื่องจักรกลเอ็นซีกับเครื่องจักรกลทั่วไป	3
2.1.3 ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็น	4
2.1.4 ข้อดีและข้อเสียของเครื่องจักรกลซีเอ็นซี	4
2.2 เครื่องจักรกลเอ็นซีและซีเอ็นซี	5
2.2.1 การทำงานของเครื่องจักรกลเอ็นซี	5
2.2.2 เครื่องกลึงเอ็นซี (NC Latch Machines)	6
2.3 ระบบควบคุมเครื่องจักรกลด้วยตัวเลข	6
2.3.1 องค์ประกอบของเครื่องจักรกลที่ควบคุมได้	7
2.4 ระบบควบคุมซีเอ็นซี (CNC Control System)	19
2.4.1 หน้าที่การทำงานที่โปรแกรมได้ (Programmable Functions)	19
2.4.2 ชนิดของการควบคุม (Control modes)	20
2.4.3 การควบคุมหน้าที่การทำงานของเครื่องจักรกล	22
2.5 เรขาคณิตเบื้องต้นสำหรับการทำโปรแกรมเอ็นซี	23
2.5.1 จุดศูนย์และจุดอ้างอิง (Zero points and Reference points)	23
2.5.2 จุดอ้างอิง (Reference point , R)	24
2.5.3 จุดศูนย์ของชิ้นงาน (Workpiece zero point)	26
2.5.4 จุดอ้างอิงของเครื่องมือ (Tool reference points)	27
2.6 ทฤษฎีทางคณิตศาสตร์ สำหรับ CNC (Mathematic theory for CNC)	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

บทที่ 3 การออกแบบส่วนประกอบของเครื่องกลึงซีเอ็นซี 31

3.1 ส่วนของเครื่องกลึงซีเอ็นซี 31

3.2 ส่วนของระบบควบคุมเซอร์โวมอเตอร์ 34

บทที่ 4 การออกแบบโปรแกรม 43

บทที่ 5 ผลการทดลองและวิจารณ์ 50

บทที่ 6 สรุปและข้อเสนอแนะ 53

กิตติกรรมประกาศ

เอกสารอ้างอิง

ภาคผนวก ก.

ภาคผนวก ข.

ภาคผนวก ค.

ภาคผนวก ง.

ภาคผนวก จ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 เครื่องกลึง NC แบบ 2 แกนและ 3 แกน	6
รูปที่ 2.2 องค์ประกอบของเครื่องจักรกล CNC	7
รูปที่ 2.3 เครื่องกลึง CNC	7
รูปที่ 2.4 เครื่องกลึงซีอีนซีแบบ 2 แกน	8
รูปที่ 2.5 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด	9
รูปที่ 2.6 Diagram ระบบขับป้อน	10
รูปที่ 2.7 ส่วนประกอบของมอเตอร์กระแสตรง	10
รูปที่ 2.8 การเปรียบเทียบลักษณะสร้างและขนาดของมอเตอร์กระแสตรงกับ มอเตอร์กระแสสลับแบบ 3 เฟส	11
รูปที่ 2.9 การขับป้อนของโต๊ะงาน	12
รูปที่ 2.10 ลักษณะโครงสร้างภายในของชุดบอลล์สกรู	13
รูปที่ 2.11 การวัดตำแหน่งโดยตรง	14
รูปที่ 2.12 การวัดตำแหน่งทางอ้อม	14
รูปที่ 2.13 การวัดตำแหน่งแบบสัมบูรณ์	15
รูปที่ 2.14 การวัดตำแหน่งแบบต่อเนื่อง	16
รูปที่ 2.15 เพลางานของเครื่องกลึง	17
รูปที่ 2.16 หัวจับงานกลึง	17
รูปที่ 2.17 ชุดยันศูนย์ท้ายแท่นและแท่นประคองงานกลึง	18
รูปที่ 2.18 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง	18
รูปที่ 2.19 อินเสิร์ตแบบถอดเปลี่ยนได้รูปทรงต่าง ๆ	19
รูปที่ 2.20 ระบบ NC	19
รูปที่ 2.21 ระบบ CNC	20
รูปที่ 2.22 หน้าที่การทำงานของเครื่องจักรกล	22
รูปที่ 2.23 แบบชิ้นงานในระบบโคออดิเนทของเครื่องจักรกล CNC	23
รูปที่ 2.24 วัตถุประสงค์ของการใช้จุดอ้างอิง	24
รูปที่ 2.25 การเคลื่อนที่ไปยังตำแหน่งของจุดอ้างอิง	25
รูปที่ 2.26 การวางแบบชิ้นงานที่จุดใดๆ	26
รูปที่ 2.27 การวางแบบชิ้นงานโดยใช้แนวแกน	26
รูปที่ 2.28 การเคลื่อนที่แบบ Linear Interpolation	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 2.29 Contour approximation using line segments 29
รูปที่ 3.1 ตัวแท่นเครื่องกลึงต้นแบบซีเอ็นซี 32
รูปที่ 3.2 หัวจับชิ้นงาน แท่นวางใบมีด และดีซีเซอร์โวมอเตอร์แนวแกน X 32
รูปที่ 3.3 เพลางานของเครื่องกลึง และมอเตอร์ในแนวแกน Z 33
รูปที่ 3.4 บอลล์สกรู (Ball Screw) 33
รูปที่ 3.5 แสดงระบบควบคุมเซอร์โวมอเตอร์ 34
รูปที่ 3.6 แสดงการเชื่อมต่อระหว่าง Model 5650 กับ Driver 36
รูปที่ 3.7 การ์ดควบคุมการเคลื่อนที่หลัก Model 5650 37
รูปที่ 3.8 DC Servo Amplifier 37
รูปที่ 3.9 ดีซีเซอร์โวมอเตอร์ และ Encoder 38
รูปที่ 3.10 Flux Vector 38
รูปที่ 3.11 เอซีเซอร์โวมอเตอร์ 38
รูปที่ 4.1 Flowchart แสดงการทำงานของโปรแกรมหลัก (Main Program) 43
รูปที่ 4.2 เมนูหลัก 44
รูปที่ 4.3 Flowchart แสดงการทำงานของฟังก์ชันอัตโนมัติ (Auto Function) 44
รูปที่ 4.4 หน้าจอแสดงผลของฟังก์ชันอัตโนมัติ 44
รูปที่ 4.5 Flowchart แสดงการทำงานของ Manual Function 45
รูปที่ 4.6 หน้าจอแสดงผลของ Manual Function 45
รูปที่ 4.7 หน้าจอแสดงผลของฟังก์ชัน Load File 45
รูปที่ 4.8 หน้าจอแสดงผลของฟังก์ชัน Set Zero 46
รูปที่ 4.9 เมนูย่อยของฟังก์ชันควบคุม Spindle 46
รูปที่ 4.10 Flowchart แสดงการทำงานของ G code G01 47
รูปที่ 4.11 แนวการเคลื่อนที่ของ G92 48
รูปที่ 4.12 Flowchart แสดงการทำงานของ G code G03 49
รูปที่ 5.1 Drawing ของชิ้นงาน 50
รูปที่ 5.2-5.7 แสดงขั้นตอนการกลึงชิ้นงานตัวอย่าง 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 ตารางเปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วไป กับเครื่องจักรกลเอ็นซี 3

ตารางที่ 3.1 ข้อมูลตัวแปรต่างๆ เชิงกลของการคำนวณระบบเซอร์โวมอเตอร์ ของต้นแบบเครื่องกลึงซีเอ็นซี 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ปัจจุบันวงการอุตสาหกรรมในประเทศไทยได้เจริญเติบโตไปเป็นอย่างมาก เครื่องจักรในโรงงานต่าง ๆ มีการพัฒนากันอย่างต่อเนื่อง มีการนำเทคโนโลยีใหม่ ๆ เข้ามาจากต่างประเทศมากมาย ซึ่งเทคโนโลยีเหล่านี้มีราคาค่อนข้างแพง โรงงานที่มีขนาดไม่ใหญ่มาก มีเงินลงทุนไม่สูงนักอาจไม่สามารถนำเทคโนโลยีเหล่านี้มาใช้ได้เนื่องจากมีข้อจำกัดในด้านเงินทุน ดังนั้นเราจึงพยายามพัฒนาเทคโนโลยีให้เท่าเทียมกับต่างประเทศ ในขณะเดียวกันก็พยายามลดเงินทุนลงเพื่อให้สามารถใช้เทคโนโลยีนี้ได้อย่างแพร่หลาย

เครื่องกลึงตามโรงงานต่าง ๆ ในปัจจุบันส่วนใหญ่ยังใช้คนควบคุมการกลึงอยู่ ซึ่งทำให้เกิดข้อผิดพลาดจากการทำงานซึ่งมีอยู่พอสมควร โดยเฉพาะกับช่างที่ยังไม่ชำนาญ

ในการกลึงชิ้นงานด้วยเครื่องจักรกลโดยทั่วไปนั้น ช่างควบคุมเครื่องจะใช้มือหมุน เพื่อเลื่อนใบมีดหรือชิ้นงานให้เคลื่อนที่ไป ซึ่งเมื่อเสร็จสิ้นการทำงานแล้วจะได้ชิ้นงานที่มีรูปทรงและขนาดตามต้องการ ลักษณะเช่นนี้ช่างควบคุมเครื่องจะต้องคอยเฝ้าดูตำแหน่งของใบมีดที่สัมพันธ์กับรูปทรงของชิ้นงานที่กำลังกลึงอยู่ตลอดเวลา และในการเปลี่ยนตำแหน่งของใบมีด ช่างจะต้องหมุนมือหมุนเพื่อควบคุมการเคลื่อนที่ของใบมีด ช่างจะต้องคอยสังเกตคมมีดเพื่อตรวจสอบรูปทรงของชิ้นงาน และเมื่อได้รูปทรงที่ต้องการแล้ว ช่างจะหยุดหมุนมือหมุน คมมีดจะหยุดเคลื่อนที่ ในภาษาเทคนิคการทำงานเหล่านี้จะเรียกว่า **การควบคุม (Control)** และตำแหน่งของใบมีดก็จะถูกควบคุม (Controlled)

นอกเหนือจากการควบคุมตำแหน่งของใบมีดแล้ว ช่างยังต้องควบคุมอัตราป้อนซึ่งขึ้นอยู่กับชนิดของวัสดุชิ้นงาน วัสดุเครื่องมือตัด และตำแหน่งของคมมีดด้วย ซึ่งในบางครั้งช่างจะต้องลดอัตราป้อนลงเมื่อใกล้จะถึงตำแหน่งที่ต้องการ เพื่อให้ได้ตำแหน่งที่ต้องการจริง ๆ นอกจากนี้จะต้องคอยปรับความเร็วรอบของ Spindle และตำแหน่งของการหล่อเย็นให้ถูกต้องอีกด้วย เป็นต้น

เมื่อการใช้คนควบคุมเกิดปัญหามากมายเช่นนี้ จึงมีการเปลี่ยนมาใช้เครื่องจักรกล CNC ซึ่งเป็นเครื่องจักรกลสมัยใหม่ ที่มีประสิทธิภาพและการทำงานสูง แต่ในขณะเดียวกันราคาก็สูงตามไปด้วย

ในปี ค.ศ.1948 นักวิทยาศาสตร์ในสถาบัน MIT (Massachusetts Institute of Technology) ได้ริเริ่มทำโครงการพัฒนาเครื่องจักรกลที่ควบคุมด้วยระบบคอมพิวเตอร์ขึ้น โดยได้รับการสนับสนุนโครงการจากกองทัพอากาศของสหรัฐอเมริกา (U.S. Air Force)

เครื่องจักรระบบเอ็นซีเครื่องแรก คือ CINCINNATIC HYDROTEL VERTICAL-SPINDLE MACHINE และนำออกใช้งานในปี ค.ศ. 1957

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในประเทศไทยนั้น เริ่มต้นโดยการนำเข้าเครื่องเก่าจากต่างประเทศ ซึ่งมีราคาค่อนข้างต่ำเพื่อการเริ่มต้นศึกษาเรียนรู้ถึงความสามารถ เครื่องส่วนใหญ่ที่นำเข้าจะเป็นเครื่องที่ใช้ NC ในการควบคุม จึงมีขนาดใหญ่ในส่วนของคอมพิวเตอร์ในการควบคุม และในเวลาต่อมาก็เริ่มนำเข้าเครื่องจักรที่ควบคุมด้วย CNC จนถึงปัจจุบัน ในเวลาปัจจุบันเครื่องที่นำเข้าสมัยก่อนที่ควบคุมด้วย NC ก็เริ่มประสบปัญหาคือ ใช้งานไม่ได้ในส่วนของคอมพิวเตอร์หรือส่วนของอิเล็กทรอนิคส์ในการควบคุมเครื่องจักร ดังนั้น จึงมีอาจารย์และนักวิจัยหลายกลุ่มหลายสถาบันเริ่มให้ความสนใจในการวิจัยและพัฒนาส่วนควบคุมเครื่องจักรดังกล่าว

1.1 วัตถุประสงค์ของโครงงาน

จุดมุ่งหมายของปริญญานิพนธ์นี้ สามารถแบ่งออกเป็นหัวข้อได้ดังต่อไปนี้

1. เพื่อศึกษาการควบคุมตำแหน่งและความเร็วของดีซีเซอร์โวมอเตอร์ และอินดัคชั่นมอเตอร์

2. สร้างต้นแบบส่วนควบคุมเครื่องกลึงซีเอ็นซี บนพื้นฐานไมโครคอมพิวเตอร์ แล้วพัฒนาโปรแกรมการควบคุมให้สามารถง่ายต่อการใช้งานตามมาตรฐาน

3. สามารถนำไปประยุกต์ใช้กับเครื่องกลึงมือทั่วไปที่มีอยู่แล้วในโรงงานอุตสาหกรรมให้เป็นเครื่องกลึงซีเอ็นซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้นเกี่ยวกับเครื่องจักรกลซีเอ็นซี

2.1 พัฒนาการของเครื่องจักรกลเอ็นซีและซีเอ็นซี

2.1.1 ความหมายของเอ็นซีและซีเอ็นซี

เอ็นซี (NC) ย่อมาจากคำว่า Numerical Control หมายถึง การควบคุมเครื่องจักรกลด้วยระบบตัวเลขและตัวอักษร ซึ่งคำจำกัดความนี้ได้จากประเทศสหรัฐอเมริกา กล่าวคือ การเคลื่อนที่ต่างๆ ตลอดจนการทำงานอื่นๆ ของเครื่องจักรกล จะถูกควบคุมโดยรหัสคำสั่งที่ประกอบด้วยตัวเลข ตัวอักษร และสัญลักษณ์อื่นๆ ซึ่งจะถูกแปลงเป็นคลื่นสัญญาณ (pulse) ของกระแสไฟฟ้าหรือสัญญาณออกอื่นๆ ที่จะไปกระตุ้นมอเตอร์หรืออุปกรณ์อื่นๆ เพื่อทำให้เครื่องจักรกลทำงานตามขั้นตอนที่ต้องการ

ซีเอ็นซี (CNC) ย่อมาจากคำว่า Computerized Numerical Control ระบบควบคุมเอ็นซีแบบนี้จะมีคอมพิวเตอร์ที่มีความสามารถสูงเพิ่มเข้าไปภายในระบบ ทำให้สามารถจัดการกับข้อมูลที่ป้อนเข้าไปในระบบเอ็นซี และประมวลผลข้อมูลเพื่อนำผลลัพธ์ที่ได้ไปควบคุมการทำงานของเครื่องจักรกล

2.1.2 ความแตกต่างระหว่างเครื่องจักรกลเอ็นซีกับเครื่องจักรกลทั่วไป

ความแตกต่างในการใช้เครื่องจักรกลเอ็นซี เมื่อเปรียบเทียบกับเครื่องจักรกลที่ใช้ทั่วไปก็คือ การตัดสินใจในการกำหนดขั้นตอนการทำงานต่างๆ จะกระทำเพียงครั้งเดียว กล่าวคือ จะกระทำในขั้นตอนการวางแผนและสร้างโปรแกรมสำหรับควบคุมเครื่องจักรกลเท่านั้น ต่อจากนั้น โปรแกรมก็จะถูกนำไปใช้ในการควบคุมการทำงานของเครื่องจักรกล สำหรับการผลิตชิ้นงานที่ต้องการ โดยสามารถทำการผลิตซ้ำๆกันกี่ครั้งก็ได้ตามต้องการ

ตาราง 2.1 ตารางเปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วไปกับเครื่องจักรกลเอ็นซี

	รายละเอียด		เครื่องจักรกลทั่วไป	เครื่องจักรกลเอ็นซี
1	การป้อนโปรแกรม		ไม่มี	มี
2	การจับยึดชิ้นงาน	ขั้น	มือ	มือ
3	การจับยึดเครื่องมือตัด	เตรียม	มือ	มือ หรือชุดควบคุม
4	การตั้งจุดอ้างอิง	งาน	มือ	มือ
5	การตั้งความเร็วรอบ		มือ	ระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6	การเลื่อนแท่นเลื่อน	ขั้น	มือหมุน	ระบบควบคุม
7	การเปรียบเทียบระยะ	ตัด	สายตา	ระบบควบคุม
8	การตรวจสอบขนาด	เฉือน	เครื่องมือวัด	ใช้เวลาน้อยกว่า

ตาราง 2.1(ต่อ) ตารางเปรียบเทียบการทำงานระหว่างเครื่องจักรกลทั่วไปกับเครื่องจักรกลเอ็นซี

2.1.3 ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบซีเอ็นซีเป็นระบบที่พัฒนาต่อเนื่องมาจากระบบเอ็นซี ดังนั้น ความแตกต่างระหว่างระบบเอ็นซีกับระบบซีเอ็นซี ก็จะอยู่ที่ความสามารถของระบบควบคุม นั่นคือคอมพิวเตอร์ เมื่อนำระบบซีเอ็นซีไปควบคุมเครื่องจักรกล ความสามารถในการทำงานต่างๆ จะเพิ่มมากขึ้นเมื่อเปรียบเทียบกับเครื่องจักรกลเอ็นซีดังนี้

1. การแสดงภาพจำลอง (Simulation) การทำงานตามโปรแกรมที่ป้อนเข้าในระบบทางจอภาพ
2. ความจุของหน่วยความจำเพิ่มมากขึ้น สามารถเก็บข้อมูลโปรแกรมได้มาก
3. การแก้ไขและลบโปรแกรมสามารถกระทำได้ที่เครื่องจักรโดยตรง
4. สามารถส่งข้อมูลไปเก็บไว้ในหน่วยความจำภายนอกได้
5. ระบบความปลอดภัยเพิ่มมากขึ้น
6. มีการชดเชยความผิดพลาดที่เกิดจากการวัดและการส่งกำลัง
7. มีโปรแกรมสำเร็จสำหรับการคำนวณค่าต่างๆ เช่น ความเร็วรอบ อัตราป้อน เป็นต้น

2.1.4 ข้อดีและข้อเสียของเครื่องจักรกลซีเอ็นซี

ข้อดีของเครื่องจักรกล CNC เมื่อเปรียบเทียบกับเครื่องจักรกลอัตโนมัติประเภทอื่น ๆ พอจะสรุปได้ดังนี้

1. มีความยืดหยุ่นสูง การเปลี่ยนงานใหม่จะแก้ไขหรือเปลี่ยนแปลงเฉพาะโปรแกรมเท่านั้น
2. ความเที่ยงตรง (Accuracy) จะอยู่ระดับเดียวกันตลอดช่วงความเร็วรอบและอัตราป้อนที่ใช้ในการผลิต
3. ใช้เวลาในการผลิต (Production Time) สั้นกว่า
4. สามารถใช้ผลิตชิ้นงานที่มีรูปทรงซับซ้อนได้ง่าย
5. การปรับตั้งเครื่องจักรกระทำได้ง่าย ใช้เวลาน้อยกว่าการผลิตด้วยวิธีอื่นๆ
6. หลีกเลี่ยงความจำเป็นที่ต้องใช้ช่างควบคุมที่มีทักษะและประสบการณ์สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ช่างควบคุมเครื่องมีเวลาว่างจากการควบคุมเครื่อง สามารถที่จะจัดเตรียมงานอื่นๆไว้ล่วงหน้าได้

8. การตรวจสอบคุณภาพไม่จำเป็นต้องกระทำทุกขั้นตอนและทุกชิ้น

ส่วนข้อเสียของเครื่องจักรกล CNC มีดังนี้

1. ราคาของเครื่องจักรค่อนข้างสูง
2. การบำรุงรักษาค่อนข้างซับซ้อน
3. จำเป็นต้องใช้ช่างเขียนโปรแกรม (Part Programmer) ที่มีทักษะสูงและฝึกอบรมมาโดยเฉพาะ
4. การซ่อมบำรุงต้องใช้ช่างที่มีประสบการณ์สูงและผ่านการฝึกอบรมมาโดยเฉพาะ
6. ราคาของเครื่องมือต่าง ๆ ที่ใช้ในกระบวนการตัดเฉือน เช่น แกนเพลายึดใบมีดมีดกลึงแบบใช้อินเสิร์ต (Insert) เป็นต้น มีราคาสูง

2.2. เครื่องจักรกลเอ็นซีและซีเอ็นซี

2.2.1 การทำงานของเครื่องจักรกลเอ็นซี

หลักการทำงานของเครื่องจักรกลเอ็นซีหรือซีเอ็นซี จะคล้ายคลึงกับเครื่องจักรกลทั่วไป กล่าวคือโดยพื้นฐานเบื้องต้นแล้วเครื่องจักรกลเอ็นซีก็จะทำการผลิตชิ้นงานเหมือนกับเครื่องจักรกลทั่วไป เช่น เครื่องกลึงเอ็นซี ก็จะทำงานเหมือนกับเครื่องกลึงทั่วไป เพียงแต่ว่าระบบควบคุมเอ็นซีของเครื่องจะทำงานในขั้นตอนต่างๆ แทนช่างควบคุมเครื่อง อย่างไรก็ตาม ก่อนที่เครื่องจักรกลเอ็นซีจะสามารถทำงานได้นั้น ระบบควบคุมของเครื่องจะต้องได้รับการบอกกล่าวเสียก่อนว่าจะให้ทำอะไร และจะต้องบอกกล่าวเป็นภาษาที่ระบบควบคุมสามารถเข้าใจได้ นั่นคือ จะต้องป้อนโปรแกรมเข้าไปในระบบควบคุมของเครื่องผ่านแป้นพิมพ์ (keyboard) หรือเทปแม่เหล็ก (magnetic tape) ก็ได้

เมื่อระบบควบคุมอ่านโปรแกรมที่ป้อนเข้าไปแล้ว ก็จะนำไปควบคุมให้เครื่องจักรกลทำงาน แต่เนื่องจากเครื่องจักรกลเอ็นซีไม่มีมือสำหรับหมุนมือหมุนให้แท่นเลื่อนเคลื่อนที่ได้ ดังนั้น แท่นเลื่อนต่างๆ จะต้องมีมอเตอร์ป้อน (feed motor) ประกอบอยู่ เช่น เครื่องกลึงซีเอ็นซีจะมีการเคลื่อนที่ 2 แนวแกน ก็จะมีมอเตอร์ป้อน 2 ตัว

เมื่อระบบควบคุมอ่านโปรแกรมแล้ว ก็จะเปลี่ยนรหัสโปรแกรมนั้นให้เป็นสัญญาณทางไฟฟ้า เพื่อไปควบคุมให้มอเตอร์ทำงาน แต่เนื่องจากสัญญาณที่ออกจากระบบควบคุมนี้มีกำลังน้อย ไม่สามารถไปหมุนขับให้มอเตอร์ทำงานได้ ดังนั้น จึงต้องส่งสัญญาณนี้เข้าไปในภาคขยายสัญญาณของระบบขับ (drive amplified) และส่งต่อไปยังมอเตอร์ป้อนของแนวแกนที่ต้องการเคลื่อนที่

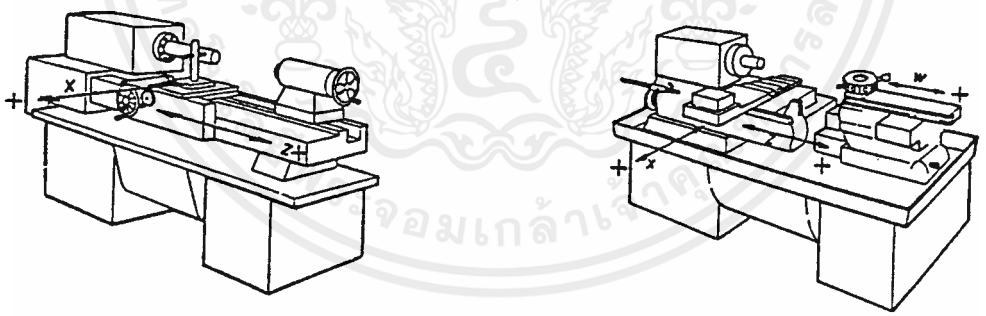
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วและระยะทางการเคลื่อนที่ของแท่นเลื่อน จะต้องกำหนดให้ระบบควบคุมรู้ ช่างควบคุมเครื่องอาศัยสายตามองดูตำแหน่งของคมตัดกับชิ้นงาน ก็จะรู้ว่าจะต้องเลื่อนแท่นเลื่อนไปอีกเป็นระยะทางเท่าใด แต่ระบบควบคุมเอ็นซีมองไม่ได้ ดังนั้น จึงต้องออกแบบอุปกรณ์หรือเครื่องมือที่สามารถจะบอกตำแหน่งของแท่นเลื่อนให้ระบบควบคุมรู้ได้ อุปกรณ์ชุดนี้เรียกว่า **ระบบวัดขนาด (Measuring System)** ซึ่งประกอบด้วยสเกลแนวตรง (Linear Scale) มีจำนวนเท่ากับจำนวนแนวแกนในการเคลื่อนที่ของเครื่องจักรกล ทำหน้าที่ส่งสัญญาณไฟฟ้าที่สัมพันธ์กับระยะทางที่แท่นเลื่อนเคลื่อนที่กลับไปยังระบบควบคุม ทำให้ระบบควบคุมรู้ว่าแท่นเลื่อนเคลื่อนที่ไปเป็นระยะทางเท่าใดแล้ว

จากหลักการควบคุมการทำงานดังกล่าว ทำให้เครื่องจักรกลซีเอ็นซีสามารถผลิตชิ้นงานให้มีรูปทรงและขนาดที่ต้องการได้ จากลักษณะสร้างและการทำงานที่เหนือกว่าเครื่องจักรกลทั่วไป ทำให้เครื่องจักรกลเอ็นซีและซีเอ็นซี เป็นปัจจัยหนึ่งที่มีความสำคัญมากในอุตสาหกรรมอัตโนมัติ และมีปริมาณความต้องการใช้เพิ่มมากขึ้นเรื่อยๆ

2.2.2 เครื่องกลึงเอ็นซี (NC Latch Machines)

เครื่องกลึงเอ็นซีส่วนใหญ่จะมีแนวการเคลื่อนที่เพียง 2 แนวแกนเท่านั้น ลักษณะการออกแบบส่วนใหญ่จะขึ้นอยู่กับชนิดของเครื่องกลึง ดังแสดงในรูปที่ 2.1



ก. แบบ 2 แกน ข. แบบ 3 แกน

รูปที่ 2.1 เครื่องกลึง NC แบบ 2 แกนและ 3 แกน

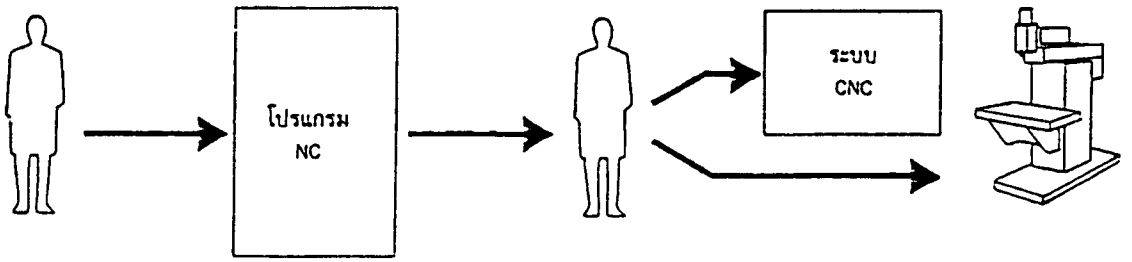
2.3 ระบบควบคุมเครื่องจักรกลด้วยตัวเลข

เครื่องจักรกลซีเอ็นซี จะประกอบด้วยองค์ประกอบใหญ่ๆ อยู่ 2 ส่วน คือ

1. เครื่องจักรกล เป็นส่วนที่ทำหน้าที่ตัดเฉือนชิ้นงานตามขั้นตอนการทำงานที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ระบบซีเอ็นซี เป็นส่วนที่ทำหน้าที่ควบคุมขั้นตอนการตัดเฉือนทั้งหมด

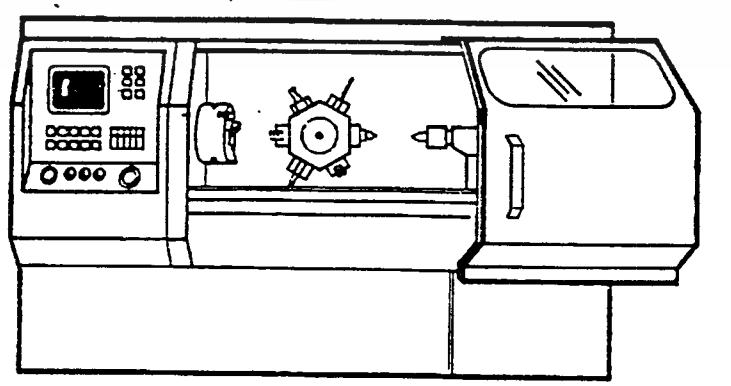


รูปที่ 2.2 องค์ประกอบของเครื่องจักรกล CNC

ข้อมูลเดิมที่อธิบายรายละเอียดของขั้นตอนที่ใช้ในการตัดเฉือนชิ้นงาน จะถูกป้อนเข้าไปในระบบควบคุมของเครื่องจักรกลก่อน ในรูปแบบของโปรแกรมเอ็นซี ซึ่งถูกจัดเตรียมโดยช่างเขียนโปรแกรม ช่างควบคุมเครื่องจะเป็นผู้ป้อนโปรแกรมเข้าไปในระบบควบคุม ซึ่งอาจป้อนด้วยมือผ่านแป้นพิมพ์โดยตรง หรือใช้แถบกระดาษเจาะรู (punched tape) ก็ได้ หลังจากนั้น ก็จะเดินเครื่องทดลองโปรแกรม และสังเกตสภาวะการตัดเฉือนชิ้นงานในแต่ละขั้นตอน บ่อยครั้งที่ช่างควบคุมเครื่องจะต้องจัดเตรียมโปรแกรม หรือเขียนโปรแกรมด้วยตนเอง หรือแก้ไขปรับปรุงโปรแกรมให้มีประสิทธิภาพในการตัดเฉือนสูงสุด ดังนั้น จึงเป็นสิ่งจำเป็นที่ช่างควบคุมเครื่องจะต้องมีความรู้ทั้งระบบควบคุมของเครื่องจักรกลและการเขียนโปรแกรมเอ็นซีด้วย

2.3.1 <u>องค์ประกอบของเครื่องจักรกลที่ควบคุมได้</u>

องค์ประกอบหรือชิ้นส่วนของเครื่องจักรกล ที่ทำหน้าที่เคลื่อนที่เข้าตัดเฉือนชิ้นงาน และองค์ประกอบอื่นๆ ที่ช่วยเสริมการทำงานตัดเฉือนให้สมบูรณ์ยิ่งขึ้น จะถูกควบคุมโดยโปรแกรมเอ็นซี ด้วยวิธีการควบคุมแบบต่าง ๆ กัน



รูปที่ 2.3 เครื่องกลึง CNC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่างชำนาญงานที่ทำหน้าที่ควบคุมการทำงานของเครื่องจักรกลเอ็นซีหรือซีเอ็นซี จะต้องมีความคุ้นเคยกับหน้าที่การทำงานและขีดจำกัดในการทำงานของ เครื่องจักรกลซีเอ็นซีนั้นเป็นอย่างดี ช่างจะใช้วิธีการทำงานแบบง่ายๆ โดยการจับยึดชิ้นงานเข้ากับโต๊ะงานและคาดว่าจะได้วิธีการตัดเฉือนที่ดีที่สุดไม่ได้ ในทางตรงข้ามช่างจะต้องจัดวางแผนขั้นตอนการทำงานไว้ล่วงหน้าเพื่อให้ได้ผลผลิตที่ดี ดังนั้นจึงเป็นสิ่งจำเป็นที่ช่างจะต้องรู้ว่าองค์ประกอบส่วนใดของเครื่องจักรกลซีเอ็นซีที่สามารถควบคุมได้และมีวิธีการควบคุมอย่างไร องค์ประกอบของเครื่องจักรกลเอ็นซีและซีเอ็นซีที่สามารถควบคุมได้ และจะกล่าวถึงในที่นี้ได้แก่

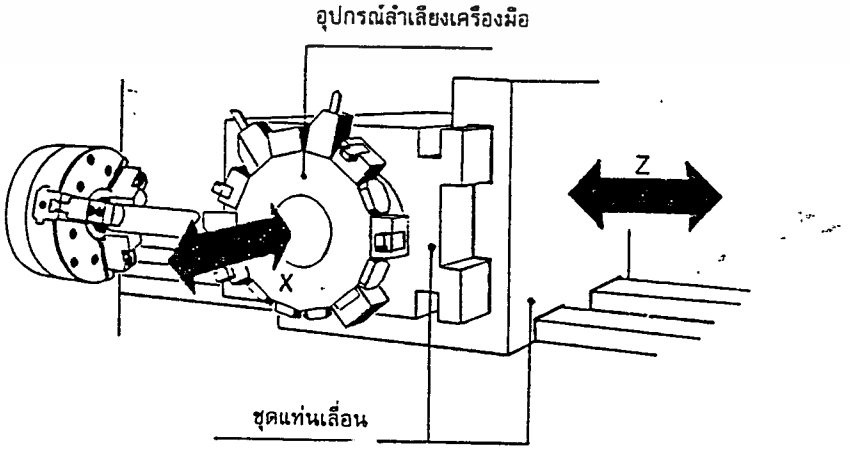
- แนวแกนป้อน (Feed axes)
- การขับป้อน (Feed drives)
- อุปกรณ์วัดขนาด (Measuring devices)
- เพลางาน (Work Spindle)
- อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

1) แนวแกนป้อน (Feed axes)

ในการกล่าวถึงเครื่องจักรกลซีเอ็นซีบ่อยครั้งที่เราจะได้ยินคำว่า แนวแกน (axes) ซึ่งหมายถึง แนวการเคลื่อนที่ขององค์ประกอบของเครื่องจักรกล เช่น โต๊ะงาน เพลาหัวเครื่อง อุปกรณ์ลำเลียงเครื่องมือ (Tool carriers) เป็นต้น

สำหรับเครื่องจักรกลทั่วไป การเคลื่อนที่ในแนวแกนต่างๆ จะเกิดจากการหมุนมือหมุนหรือโยกคันโยกป้อนอัตโนมัติ (Feed levers)

เครื่องจักรกลซีเอ็นซีมีแนวแกนป้อนรวมกันอยู่หลายแนวแกนทำให้สามารถตัดเฉือนชิ้นงานให้เป็นรูปทรงต่างๆ ที่ต้องการได้ การกำหนดแนวแกนต่างๆของเครื่องจักรกลซีเอ็นซีจะกำหนดตามมาตรฐานสากลภายใต้หัวเรื่อง Coordinate axes and direction of movement for numerically controlled machinery ซึ่งจะกำหนดแนวแกนเหล่านี้โดยใช้ตัวอักษร x,y และ z ดังแสดงในรูป 2.4



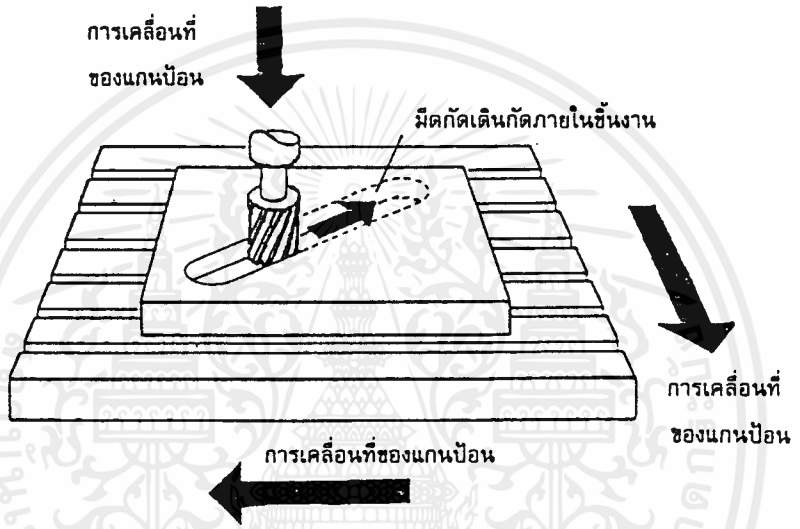
รูปที่ 2.4 เครื่องกลึงซีอีนซีแบบ 2 แกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องกลึงจะมีแนวการป้อนอยู่ 2 แนวแกน คือ แกน X และ แกน Z ทั้งสองแนวแกนจะอยู่ที่ชุดแท่นเลื่อน (Compound Slide) ซึ่งมีอุปกรณ์ลำเลียงเครื่องมือ (Tool Carrier) ติดตั้งอยู่ ลักษณะเช่นนี้ทำให้สามารถกลึงชิ้นงานที่มีรูปทรงต่าง ๆ กันได้ตามต้องการ

2) การขับป้อน (Feed drives)

การเคลื่อนที่เรียงลำดับกันหรือพร้อม ๆ กันอย่างต่อเนื่องของแนวแกนป้อน จะทำให้เกิดการตัดเฉือนของเครื่องมือในชิ้นงาน ดังรูป 2.5

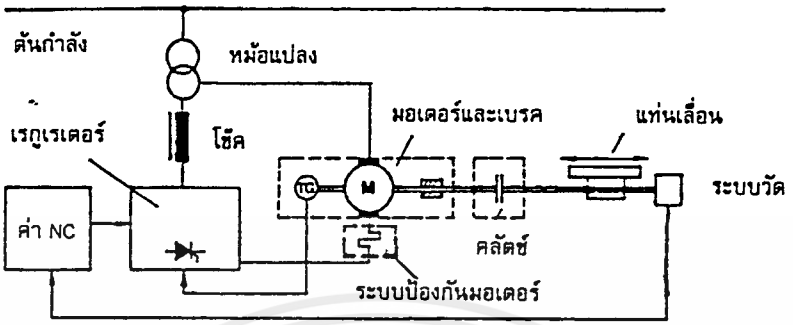


รูปที่ 2.5 การเคลื่อนที่ตัดเฉือนของเครื่องมือตัด

การขับป้อนจะทำให้เกิดการเคลื่อนที่ของแท่นเลื่อน ในขณะตัดเฉือนแท่นเลื่อนอาจพาให้ชิ้นงานเคลื่อนที่หรือคมตัดเคลื่อนที่ก็ได้ ระบบขับป้อนโดยทั่วไปจะใช้มอเตอร์กระแสตรงในการหมุนขับและควบคุมการทำงาน ด้วยวงจรอิเล็กทรอนิกส์จากภายนอก มอเตอร์ชนิดนี้จะสามารถหมุนและเบรคให้หยุดได้ทั้งสองทิศทางขณะตัดเฉือนชิ้นงาน การเคลื่อนที่ป้อนจะต้องเป็นไปอย่างสม่ำเสมอและสามารถต้านแรงกระทำจากภายนอกได้ เช่น แรงตัดเฉือน เป็นต้น ด้วยเหตุนี้ระบบขับป้อนจึงต้องได้รับการออกแบบให้มีความแข็งแกร่งสูง การเคลื่อนที่คงที่และสม่ำเสมอสามารถตอบสนองต่อการเปลี่ยนอัตราป้อนได้อย่างรวดเร็ว นอกจากนี้ในขณะทำงานคมตัดอาจทื่อหรือการเคลื่อนที่ของแท่นเลื่อนถูกกีดขวาง หรือการเร่งอัตราป้อนให้เคลื่อนที่เร็วและหยุดโดยทันทีทันใด สาเหตุเหล่านี้จะทำให้มอเตอร์รับภาระมากเกินไป (over loading) ซึ่งอาจทำให้มอเตอร์เสียหายได้ ดังนั้น จึงต้องมีการป้องกัน อุบัติเหตุเหล่านี้ โดยทั่วไปแล้วจะใช้คลัตช์แบบลูกกลิ้ง (Over running clutch) ร่วมกับวงจรอิเล็กทรอนิกส์ ปัจจัยหนึ่งที่จะทำให้ระบบขับป้อนสามารถทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

งานได้อย่างมีประสิทธิภาพก็คือ การเลือกใช้อุปกรณ์ในระบบขับป้อนให้เหมาะสมกับการทำงานของเครื่องจักรและการออกแบบวงจรควบคุมการทำงานที่มีประสิทธิภาพดังแสดงในรูป2.6

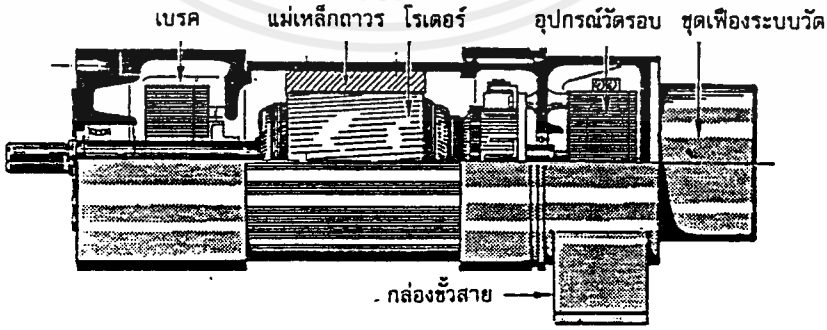


รูปที่ 2.6 Diagram ระบบขับป้อน

2.1) มอเตอร์

เครื่องจักรกลเอ็นซีสมัยใหม่จะออกแบบใช้ระบบขับป้อนแบบเซอโว (servo drives)ทำให้สามารถปรับอัตราป้อนและความเร็วรอบได้โดยไม่มีขีดจำกัดของขั้นความเร็วและอัตราป้อน มอเตอร์ที่ใช้ในระบบขับป้อนโดยทั่วไปจะมีอยู่ 3 ชนิดด้วยกันคือ

ก) <u>มอเตอร์กระแสตรง (DC motors)</u> ลักษณะสร้างของมอเตอร์กระแสตรงจะใช้เป็นแม่เหล็กถาวรที่มี 4 , 6 หรือ 8 ขั้ว ประกอบด้วยระบบเบรค (brake) แกนมอเตอร์ (Rotor) อุปกรณ์วัดรอบ (Tachogenerator) และอุปกรณ์วัด (Measuring box) ดังแสดงในรูป 2.7



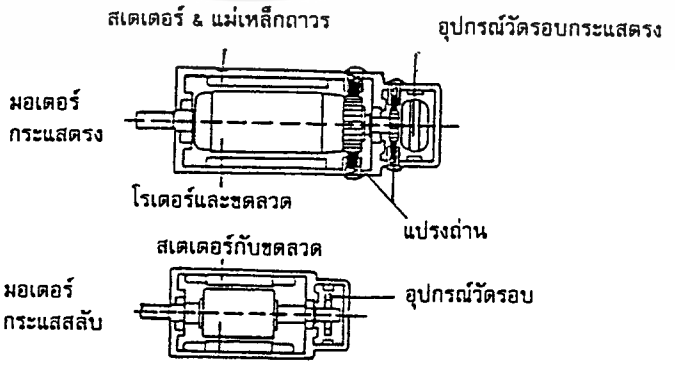
รูปที่ 2.7 ส่วนประกอบของมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้มอเตอร์กระแสตรง ทำให้สามารถปรับอัตราป้อนได้ละเอียดและมีวงจรควบคุมที่ไม่ซับซ้อน แต่ก็มีข้อเสียตรงที่มอเตอร์ชนิดนี้ต้องใช้แปรงถ่าน ซึ่งจะต้องคอยทำความสะอาดและเปลี่ยนเมื่อแปรงถ่านหมด นอกจากนี้ แปรงถ่านยังทำให้แกนมอเตอร์สึกหรออันเป็นผลทำให้กำลังมอเตอร์ลดลง ข้อเสียอีกประการหนึ่งก็คือ หากต้องการกำลังขับสูง มอเตอร์ก็จะมีขนาดใหญ่ด้วย และเมื่อใช้ความเร็วรอบสูงๆ จะทำให้แรงบิดลดลง ดังนั้น จึงมักใช้กับเครื่องจักรกลเอ็นซีขนาดเล็กและขนาดกลาง

ข) มอเตอร์แบบเป็นขั้น (Stepping motors)) เป็นมอเตอร์ที่ทำงานแบบต่อเนื่องโดยการแปลงคลื่นสัญญาณที่ป้อนเข้าไปในระบบให้เป็นการเคลื่อนที่เชิงมุม การหมุนในแต่ละมุมหรือขั้นที่เปลี่ยนไป 1 ขั้นจะเท่ากับ 1 คลื่นสัญญาณ ดังนั้น ตำแหน่งของเพลาจะถูกกำหนดโดยจำนวนคลื่นสัญญาณที่ป้อนเข้าไปในระบบ และความเร็วในการหมุนของเพลาจะวัดเป็นจำนวนขั้นต่อวินาที (Steps per second) ซึ่งจะเท่ากับความถี่ของคลื่นสัญญาณที่ป้อนเข้าในระบบที่วัดเป็นจำนวนคลื่นสัญญาณต่อวินาที (pulses per second) ความเที่ยงตรงของระบบจะขึ้นอยู่กับความสามารถของมอเตอร์ในการแบ่งขั้นการหมุนตามจำนวนคลื่นสัญญาณที่ป้อนเข้าในระบบ แรงบิดของมอเตอร์ชนิดนี้จะลดลงเมื่อความเร็วในการหมุนแบ่งเพิ่มขึ้น ดังนั้น จึงเหมาะสำหรับเครื่องจักรกลเล็กๆ ที่ไม่ต้องใช้กำลังขับมาก เช่น เครื่องพลอตเตอร์ (Plotter machine) เป็นต้น

ค) มอเตอร์กระแสสลับ (Alternate-current motor) ส่วนมากจะเป็นมอเตอร์แบบซิงโครนัส (Synchronous motor) ข้อดีของมอเตอร์ชนิดนี้คือ ไม่ต้องใช้แปรงถ่าน ทำให้สามารถลดงานบำรุงรักษาได้มาก และมอเตอร์ขนาดเดียวกันเมื่อเปรียบเทียบกับมอเตอร์กระแสตรง จะสามารถให้แรงบิดได้ดีกว่า และมีขนาดเล็กกว่าด้วย ดังแสดงในรูป 2.8 ส่วนข้อเสียของมอเตอร์แบบนี้คือ วงจรควบคุมจะมีความซับซ้อนมากกว่าวงจรควบคุมมอเตอร์กระแสตรง

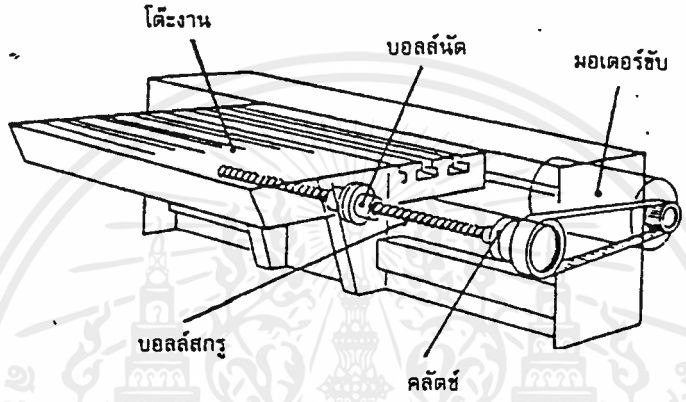


รูปที่ 2.8 การเปรียบเทียบลักษณะสร้างและขนาดของมอเตอร์กระแสตรงกับมอเตอร์กระแสสลับแบบ 3 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2) บอลล์สกรู (Ball screws)

หัวใจของระบบขับป้อนของเครื่องจักรกลซีเอ็นซี ก็คือ การส่งกำลังขับด้วย บอลล์สกรู ซึ่งจะมีลูกบอลล์ไหลหมุนเวียนอยู่ตลอดเวลา บอลล์สกรูจะประกอบด้วยสกรูกับนัตที่มีลักษณะเป็นเกลียวกลม ร่องเกลียวกลมบนสกรูและในนัตจะชุบแข็งและเจียระไนผิวเรียบมันเพื่อลดความฝืด และเพิ่มความเที่ยงในการเคลื่อนที่ ดังแสดงในรูป 2.9

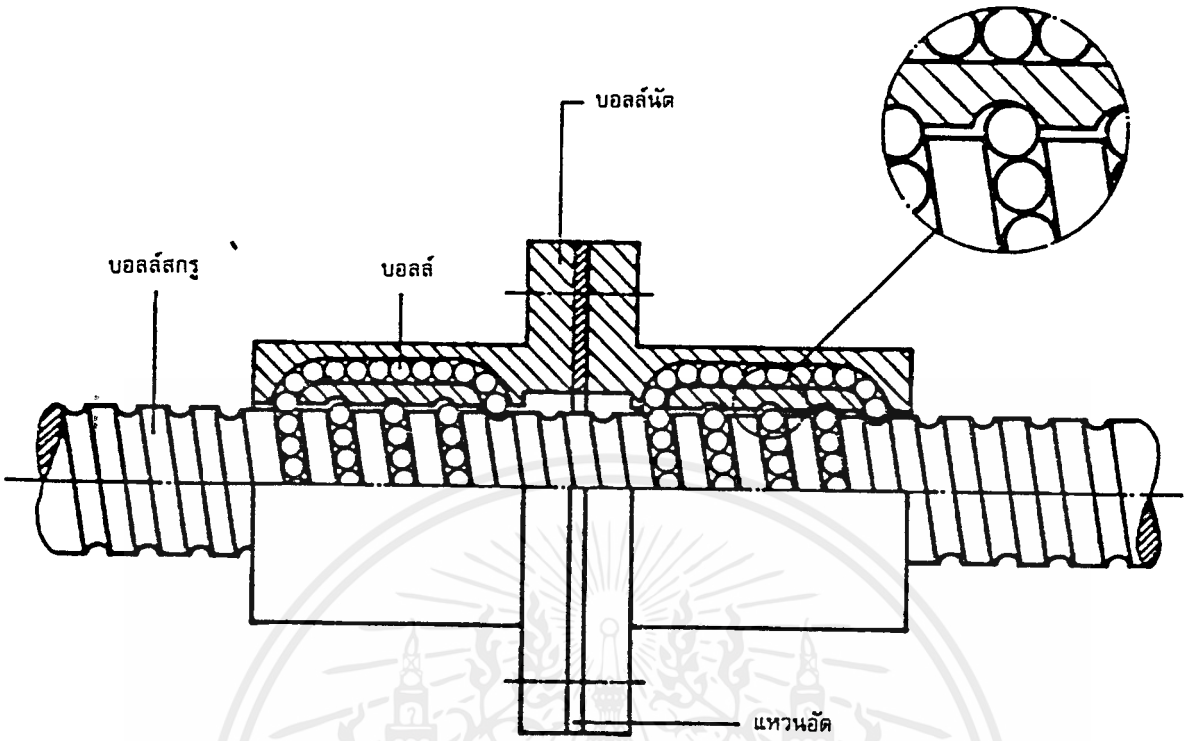


รูปที่ 2.9 การขับป้อนของโต๊ะงาน

เมื่อมอเตอร์หมุนขับสกรู นัตก็จะเคลื่อนที่ไปตลอดความยาวของสกรู พาให้แท่นเลื่อนและโต๊ะงานเคลื่อนที่ไปตามรางเลื่อน

ภายในของตัวนัตจะประกอบไปด้วยชุดของลูกบอลล์จำนวนมาก ดังแสดงในรูป2.10 ทำให้มั่นใจได้ว่าความเสียดทานในการส่งกำลังขับจากสกรูไปยังแท่นเลื่อนจะมีน้อยมาก นัตจะถูกแบ่งออกเป็นสองซีก และ ขันประกอบยึดเข้าด้วยกัน โดยมีการเตรียมอัดแรงไว้ก่อน (preloaded) ทำให้สามารถลดระยะคลอน (backlash) ให้เหลือน้อยที่สุดจนแทบจะไม่มีเลยได้ทำให้การเคลื่อนที่ของแท่นเลื่อนมีความเที่ยงตรงสูงสามารถทำงานซ้ำๆกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ลักษณะสร้างภายในของชุดบอลล์สกรู

การต่อกำลังระหว่างมอเตอร์กับบอลล์สกรู จะมีชุดคลัตช์ความฝืดเป็นตัวเชื่อม ซึ่งนอกจากจะมีหน้าที่ต่อกำลังขับแล้ว ยังมีหน้าที่ป้องกันอุบัติเหตุที่เกิดจากแท่นเลื่อนหรือโต๊ะงานชนหรือกระแทกกับสิ่งกีดขวางไม่ให้เครื่องจักรกลซีเอ็นซีเกิดความเสียหายมากเกินไป กล่าวคือ เมื่อมีการชน หรือกระแทกกันขึ้นจนแรงมากถึงค่าหนึ่ง ชุดคลัตช์ก็จะตัดระบบการส่งกำลังขับระหว่างมอเตอร์กับตัวบอลล์สกรูทันที

3) ระบบวัดขนาด (Measuring System)

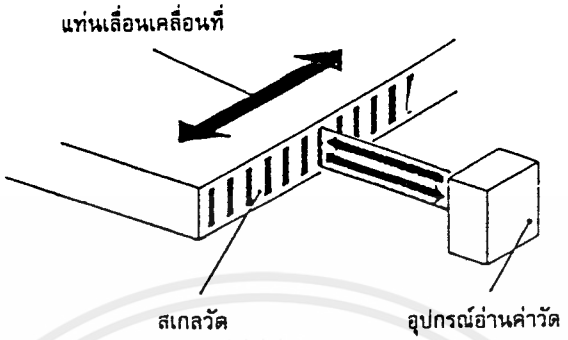
การเคลื่อนที่ไปที่ตำแหน่งต่างๆ ในแต่ละแนวแกนของแท่นเลื่อน จะถูกส่งไปยังระบบควบคุมโดยระบบวัดขนาด การวัดตำแหน่งของแท่นเลื่อนสามารถที่จะวัดได้ทั้ง โดยตรง (Direct Measurement) และโดยทางอ้อม (Indirect Measurment)

3.1) การวัดตำแหน่งโดยตรง

วิธีนี้จะใช้สเกลวัด (measuring scale) ยึดติดกับแท่นเลื่อนหรือโต๊ะงานโดยตรง ดังแสดงในรูป 2.11 ข้อดีของวิธีวัดแบบนี้ก็คือ ความไม่เที่ยงตรงของสกรูนำเลื่อน (leadscrew) ระบบขับจะไม่มีผลกระทบต่อค่าที่อ่านได้ อุปกรณ์อ่านค่าวัด(Measuring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

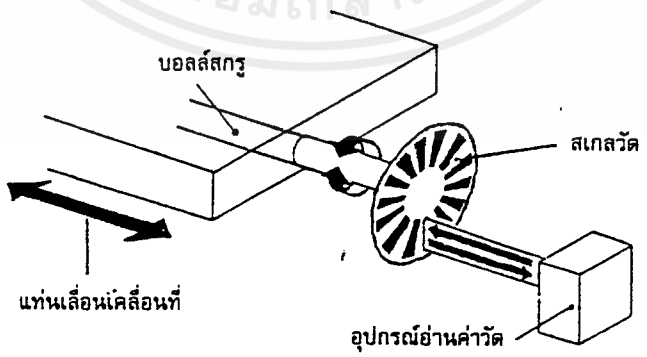
valve resolver) จะอ่านข้อมูลในการวัดจากขีดสเกลวัด (Measuring scale grid) และแปลงข้อมูลนี้เป็นสัญญาณไฟฟ้าและส่งกลับไปยังระบบควบคุม



รูปที่ 2.11 การวัดตำแหน่งโดยตรง

3.2) การวัดตำแหน่งทางอ้อม

การเคลื่อนที่ของแท่นเลื่อนจะได้รับกำลังขับมาจากการหมุนของบอลล์สกรู อุปกรณ์เปลี่ยนค่าวัด (Resolver) จะบันทึกการเคลื่อนที่หมุนของแผ่นจานสัญญาณ (pulse disc) ที่ต่อติดอยู่กับบอลล์สกรู และส่งต่อไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะใช้สัญญาณ ได้รับนี้ไปคำนวณหาระยะทางการเคลื่อนที่ของแท่นเลื่อนจากสัญญาณการหมุน (rotation pulses) ของแผ่นจานสัญญาณ ดังแสดงในรูป 2.12



รูปที่ 2.12 การวัดตำแหน่งทางอ้อม

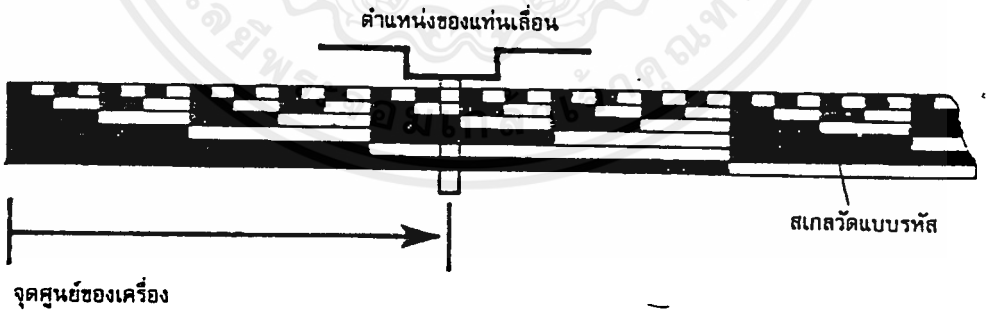
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากการวัดตำแหน่งทางตรงและทางอ้อมแล้ว ในระบบการวัดขนาดของเครื่องจักรกลเอ็นซีที่ต้องการให้การวัดตำแหน่งมีความเที่ยงตรงตลอดแนวแกนป้อน จะต้องต่อระบบขับป้อนเข้ากับอุปกรณ์วัดที่เหมาะสม อุปกรณ์วัดโดยทั่วไปจะประกอบด้วยสเกลกับอุปกรณ์อ่านค่าวัดที่สามารถอ่านสเกลได้ สเกลที่ใช้ในอุปกรณ์วัดมีอยู่ด้วยกัน 2 ชนิด คือ สเกลวัดแบบรหัส (coded measuring scale) กับสเกลวัดแบบช่อง (division grid) การใช้สเกลวัดทั้ง 2 ชนิดนี้จะขึ้นอยู่กับวิธีการวัดตำแหน่ง (position measurement) วิธีการวัดตำแหน่งที่นิยมใช้กันทั่วไปมีอยู่ 2 วิธีคือ การวัดตำแหน่งแบบสัมบูรณ์ absolute position measurement) กับการวัดตำแหน่งแบบต่อเนื่อง หรือแบบลูกโซ่ (incremental or chain position measurement) ซึ่งมีความแตกต่างกันดังรายละเอียดต่อไปนี้

3.3) การวัดตำแหน่งแบบสัมบูรณ์

คำว่า สัมบูรณ์ (absolute) ที่ใช้ร่วมกับการวัดตำแหน่งนี้จะหมายความว่า ค่าตำแหน่งต่างๆ สามารถวัดได้ตลอดเวลาและเป็นอิสระจากสถานะของเครื่องและระบบควบคุม ทั้งนี้เพราะค่าต่างๆเหล่านี้จะวัดอ้างอิงจากจุดศูนย์อ้างอิง (fixed zero datum) เสมอ

การวัดตำแหน่งแบบสัมบูรณ์ ดังแสดงในรูป 2.13 จะใช้สเกลวัดแบบรหัส (coded measuring scale) ซึ่งจะชี้ตำแหน่งของแท่นเลื่อนที่ถูกต้องตลอดเวลา โดยอ้างอิงจากตำแหน่งจุดศูนย์ของเครื่อง (Machine zero point) ซึ่งเป็นตำแหน่งศูนย์ที่มีจุดอ้างอิงที่แน่นอนและถาวรของเครื่องจักรกลเอ็นซี



รูปที่ 2.13 การวัดตำแหน่งแบบสัมบูรณ์

ข้อสำคัญของการใช้วิธีการวัดตำแหน่งแบบนี้ก็คือ ความยาวของช่วงอ่านค่าวัดของสเกลจะต้องยาวกว่าระยะเลื่อนทำงานของแท่นเลื่อนเพื่อให้ระบบควบคุมของ

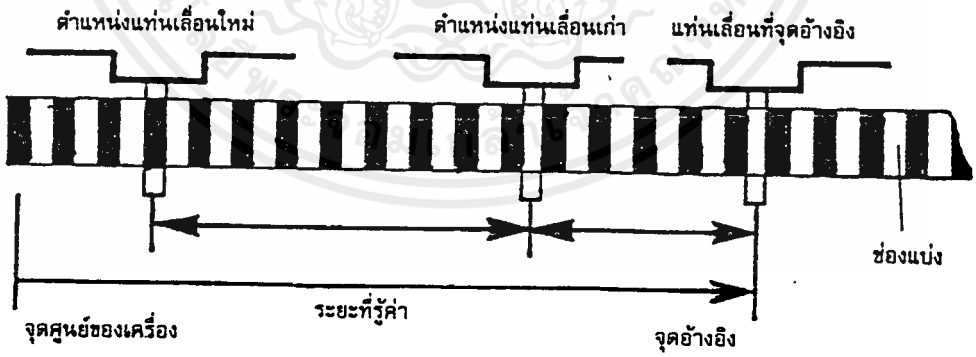
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องสามารถอ่านค่าวัดได้ทุกตำแหน่งสเกลนี้จะใช้รหัสเป็นระบบตัวเลขฐานสอง (Binary system)

3.4) การวัดตำแหน่งแบบต่อเนื่อง

คำว่า **ต่อเนื่อง** (incremental) แปลว่า ระยะเลื่อนสั้นๆ ตามความยาวที่กำหนดดังนั้นในการวัดตำแหน่งอาจจะเป็นการเพิ่มหรือลดขนาดความยาวในการเคลื่อนที่ที่วัดอยู่ก็ได้ ในระหว่างการเคลื่อนที่ของแท่นเลื่อนระบบควบคุมจะทำการนับจำนวนส่วนแบ่ง(divisions)ที่ตำแหน่งใหม่แตกต่างจากตำแหน่งก่อนหน้านี้เสมอ

วิธีการวัดตำแหน่งแบบต่อเนื่องที่แสดงในรูป 2.14 สเกลวัดจะแบ่งเป็นช่อง (grid) แบบง่ายๆ โดยที่แต่ละช่องจะมีพื้นที่สว่างกับมืดสลับกันไป เมื่อแท่นเลื่อนเคลื่อนที่ ช่องนี้ก็จะวิ่งผ่านอุปกรณ์เปลี่ยนค่าวัด (resolver) ซึ่งจะทำหน้าที่นับ จำนวนช่องพื้นที่สว่างและมืด จากนั้นก็จะส่งเป็นสัญญาณไฟฟ้าไปยังระบบควบคุมของเครื่อง ระบบควบคุมก็จะนำสัญญาณนี้มาคำนวณหาตำแหน่งสุดท้ายของแท่นเลื่อนที่แตกต่างจากตำแหน่งก่อนหน้านี้ ในทางปฏิบัติที่ต้องการให้วิธีการวัดแบบนี้ทำงานได้อย่างถูกต้อง เมื่อเริ่มเปิดสวิตช์ระบบควบคุมของเครื่องควรจะเลื่อนไปยังจุดที่ทราบค่าระยะห่างจากจุดศูนย์ของเครื่องจุดนี้จะเรียกว่า "จุดอ้างอิง" (Reference Point) หลังจากที่แท่นเลื่อนในแนวแกนต่างๆ เลื่อนไปยังจุดอ้างอิงแล้ว อุปกรณ์อ่านค่าวัดก็จะสามารถทำหน้าที่วัดตำแหน่งด้วยช่องสเกลได้

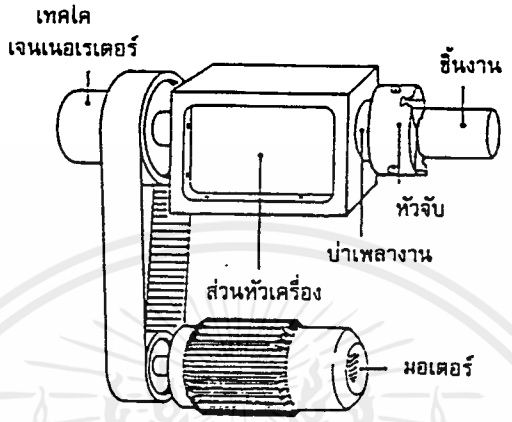


รูปที่ 2.14 การวัดตำแหน่งแบบต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เพลางาน (Work Spindle)

เพลางานเป็นชิ้นส่วนหรือองค์ประกอบของเครื่องจักรกลที่มีความสำคัญมาก มีหน้าที่หลักในการทำงาน คือจะทำหน้าที่ขับพาให้เครื่องมือ เช่น มีดกลึง ดอกสว่าน เป็นต้น

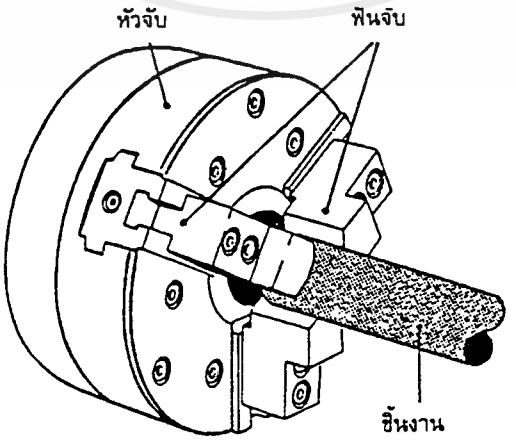


รูปที่ 2.15 เพลางานของเครื่องกลึง

5) อุปกรณ์จับยึดชิ้นงาน (Workpiece holding devices)

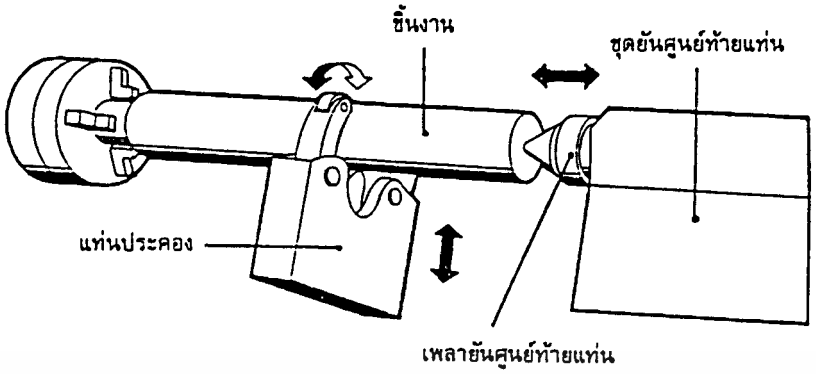
อุปกรณ์จับยึดชิ้นงานจะจัดเตรียมไว้สำหรับยึดชิ้นงานเข้ากับเพลางานในงานกลึง (Latch) สามารถเลือกใช้อุปกรณ์จับยึดชิ้นงานแบบต่างๆ กันได้หลายชนิด เช่น

- หัวจับแบบ 2,3 หรือ 4 ฟันจับ
- หน้าจานจับ สำหรับจับยึดชิ้นงานที่มีรูปทรงไม่สมมาตร
- ยันศูนย์สำหรับเพลางานและเพลาของชุดยันศูนย์ท้ายแท่น
- แท่นประคอง



รูปที่ 2.16 หัวจับงานกลึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



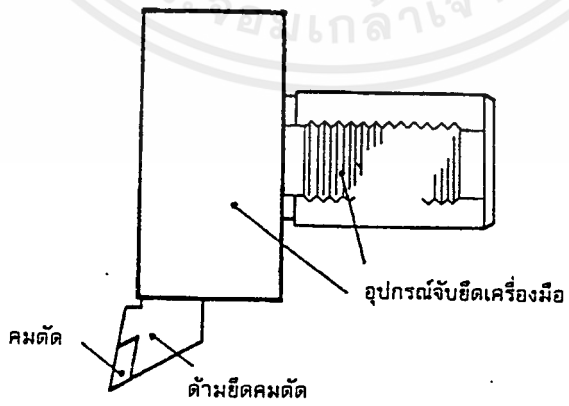
รูปที่ 2.17 ชุดยันศูนย์ท้ายแท่นและแท่นประคองงานกลึง

6) เครื่องมือ (Tools)

การทำงานของเครื่องจักรกลโดยทั่วไปจะต้องทำงานควบคู่กับเครื่องมือ (Tools) ซึ่งได้แก่ มีดกลึง ดอกสว่าน เป็นต้น เครื่องมือที่เหมาะสมจึงเป็นสิ่งสำคัญยิ่งและเป็นการช่วยเสริมการทำงานของเครื่องจักรกลซีเอ็นซีให้สามารถทำงานได้ประสิทธิภาพสูงสุด

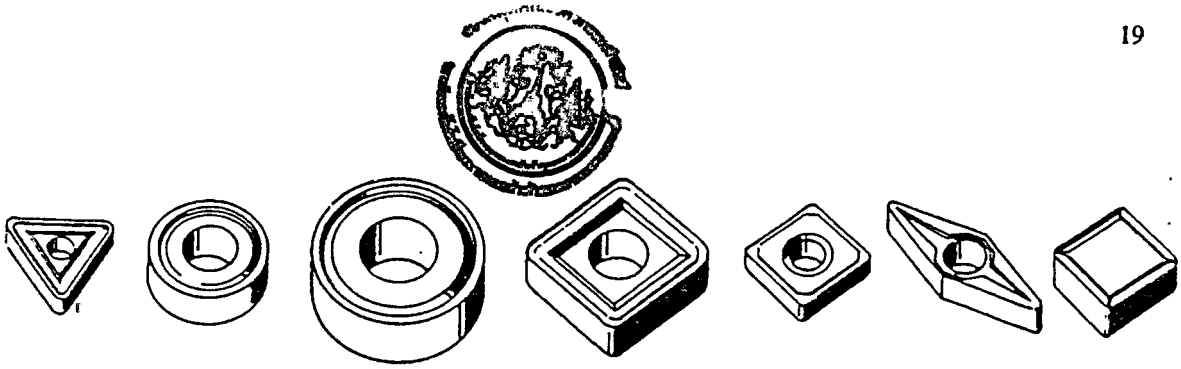
องค์ประกอบของเครื่องมือที่สมบูรณ์สำหรับเครื่องซีเอ็นซี ซึ่งจะประกอบด้วยส่วนต่าง ๆ ดังนี้

- อุปกรณ์จับยึดเครื่องมือ (Tool Holders)
- ด้ามยึดคมตัด (Tool Tip Carrier)
- คมตัด หรืออินเสิร์ต (Insert)



รูปที่ 2.18 องค์ประกอบของเครื่องมือที่ใช้ในงานกลึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



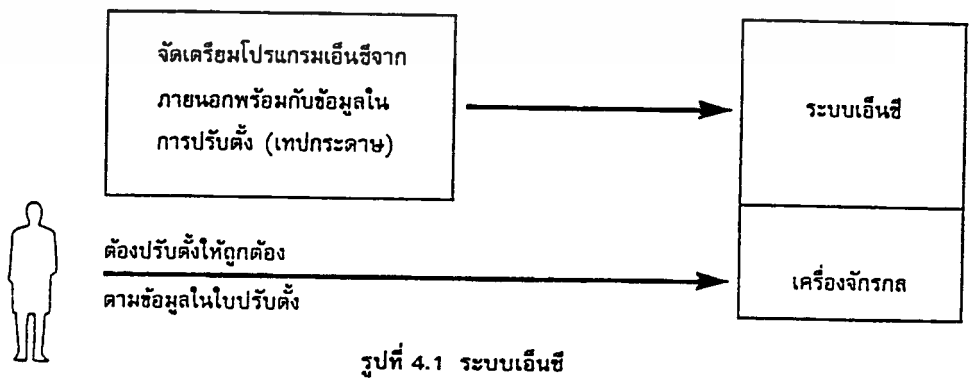
รูปที่ 2.19 อินเสิร์ตแบบถอดเปลี่ยนได้รูปทรงต่าง ๆ

2.4 ระบบควบคุมซีเอ็นซี (CNC Control System)

2.4.1 หน้าที่การทำงานที่โปรแกรมได้ (Programmable Functions)

ในปัจจุบัน ระบบควบคุมการทำงานของเครื่องจักรกลสมัยใหม่เกือบทั้งหมด จะควบคุมด้วยระบบซีเอ็นซี แต่เนื่องจากยังคงอ้างอิงถึงโปรแกรมเอ็นซี (NC Program) และเทคโนโลยีเอ็นซี (NC Technology) อยู่ ดังนั้น จึงเป็นเรื่องสำคัญที่จะต้องรู้ถึงความแตกต่างในการทำงานระหว่างระบบเอ็นซีกับระบบซีเอ็นซี

ระบบเอ็นซี (NC System) ดังแสดงในรูป 2.20 จะมีระบบควบคุมประกอบอยู่กับเครื่องจักรกลซึ่งจะต้องจัดเตรียมโปรแกรมเอ็นซีจากภายนอกก่อน แล้วจึงป้อนเข้าไปในระบบควบคุม โดยอาศัยสื่อข้อมูล (data carries) เช่น เทปกระดาษ (Punched tape)เป็นต้น โปรแกรมเอ็นซีที่ป้อนเข้าไปในระบบควบคุมของเครื่องจะถูกนำไปใช้เพื่อสั่งให้เครื่องเริ่มทำงานและหยุดชั่วคราวได้ แต่จะไม่สามารถแก้ไขโปรแกรมโดยช่างควบคุมเครื่องได้ ขนาดของเครื่องมือและอุปกรณ์จับยึดชิ้นงานจะถูกเลือกใช้ในขณะเขียนโปรแกรมไว้ก่อนและกำหนดไว้ในใบปรับตั้ง (set-up sheet) ซึ่งช่างควบคุมเครื่องจะต้องจัดเตรียมและประกอบยึดเครื่องมือ ตลอดจนอุปกรณ์จับยึดชิ้นงานให้ถูกต้องตามข้อมูลที่กำหนดไว้ในใบปรับตั้ง

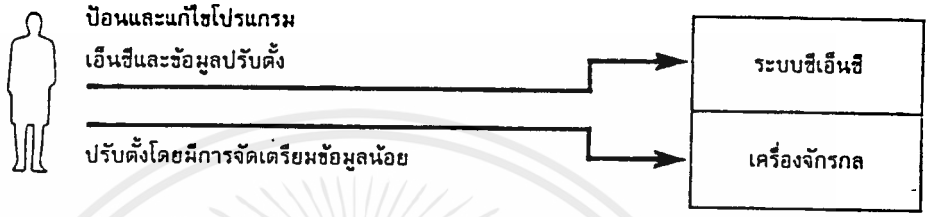


รูปที่ 4.1 ระบบเอ็นซี

รูปที่ 2.20 ระบบ NC

038044
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบซีเอ็นซี (CNC system) จะมีคอมพิวเตอร์ประกอบอยู่ด้วย ดังนั้น ช่างควบคุมเครื่องไม่เพียงแต่จะสามารถใช้โปรแกรมเอ็นซีสั่งให้เครื่องจักรทำงานได้เท่านั้น แต่จะยังสามารถเขียนและป้อนโปรแกรมด้วยตนเอง ตลอดจนการแก้ไขโปรแกรมได้หลังจากป้อนเข้าไปในระบบควบคุมของเครื่องแล้ว ดังแสดงในรูป 2.21



รูปที่ 2.21 ระบบ CNC

ขนาดต่างๆของเครื่องมือตัดและอุปกรณ์จับยึดชิ้นงาน สามารถที่จะเลือกใช้และป้อนเข้าไปในระบบควบคุมซีเอ็นซี ขณะทำการปรับตั้ง (setting-up) และเป็นอิสระจากตัวโปรแกรมเอ็นซี ขนาดต่างๆ ของเครื่องมือจะถูกนำไปใช้โดยอัตโนมัติในขณะทำการตัดเฉือน ด้วยเหตุนี้ช่างควบคุมเครื่องจึงไม่จำเป็นต้องมีข้อมูลในการปรับตั้งมากและสามารถที่จะเลือกใช้เครื่องมือและอุปกรณ์จับยึดชิ้นงานได้ด้วยตนเอง

หากพิจารณาถึงภาษาโปรแกรม (Programming language) และเทคโนโลยีทางด้านการตัดเฉือนของเครื่องจักรกลที่ใช้ในระบบเอ็นซีกับซีเอ็นซีแล้วจะไม่แตกต่างกัน

2.4.2 ชนิดของการควบคุม (Control modes)

ลักษณะการควบคุมการเคลื่อนที่ทำงานของแท่นเลื่อนต่าง ๆ ในเครื่องจักรกล เอ็นซีและซีเอ็นซี จะมีการเคลื่อนที่อยู่ 2 ลักษณะ คือ

การเคลื่อนที่ในแนวเส้นตรง(Linear Interpolation หรือ straight lineInterpolation) การเคลื่อนที่ลักษณะนี้ ระบบซีเอ็นซีจะคำนวณหาตำแหน่งของจุดต่างๆที่ต่อกันเป็นลูกโซ่ในแนวเส้นตรงระหว่างตำแหน่งของเครื่องมือ 2 ตำแหน่งในขณะที่เครื่องมือเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งนั้น ระบบควบคุมซีเอ็นซีจะตรวจสอบและแก้ไขแนวแกนในการเคลื่อนที่ให้ถูกต้องอยู่ตลอดเวลาทำให้การเคลื่อนที่ของเครื่องมือไม่ผิดพลาดหรือคลาดเคลื่อนออกจากจุดต่อของเส้นตรงมากกว่าค่าพิกัดความเผื่อของเครื่องที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนที่ในแนวเส้นโค้ง (Circular Interpolation) ระบบควบคุมซีเอ็นซี จะคำนวณหาตำแหน่งของจุดต่างๆ ที่ต่อกันเป็นเส้นโค้งตามขนาดรัศมีที่กำหนด ระหว่างตำแหน่งของเครื่องมือที่กำหนดไว้ 2 ตำแหน่ง ระบบควบคุมจะอาศัยจุดเหล่านี้ในการตรวจสอบและแก้ไขแนวการเคลื่อนที่ของเครื่องมือให้ถูกต้องและอยู่ภายใน พิกัดความเผื่อของเครื่องจักรกลที่กำหนด

ในระบบควบคุมซีเอ็นซีจะแบ่งการควบคุมการเคลื่อนที่ทั้งสองลักษณะตามลักษณะการเคลื่อนที่ป้อนออกเป็น 3 ชนิด คือ

1) การควบคุมจุดต่อจุด (Point to point control)

การควบคุมแบบนี้จะควบคุมการเคลื่อนที่ของเครื่องมือระหว่างจุดสองจุดที่โปรแกรมไว้ในลักษณะการเคลื่อนที่เร็ว (Rapid traverse) โดยที่เครื่องมือจะต้องไม่สัมผัสชิ้นงาน

แนวแกนในการเคลื่อนที่ขึ้นอยู่กับชนิดของระบบควบคุม กล่าวคือมอเตอร์ขับของระบบป้อนอาจจะเริ่มทำงานหลายๆ แนวแกนพร้อมกัน หรือทำงานทีละแนวแกน จนกว่าจะเคลื่อนที่ถึงตำแหน่งของเครื่องมือที่โปรแกรมไว้ ทำให้ไม่สามารถควบคุมทางเดินของเครื่องมือ (Tool path) ได้ การควบคุมแบบจุดต่อจุดมักจะใช้กับเครื่องเจาะ (drilling machine) เครื่องเชื่อมจุด (spot drilling) เป็นต้น

2. การควบคุมการตัดเฉือนแนวเส้นตรง (Straight-cut control)

การควบคุมชนิดนี้ นอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือแบบเคลื่อนที่เร็วได้แล้ว ยังสามารถควบคุมการเคลื่อนที่ของเครื่องมือในแนวขนานกับแนวแกนของเครื่องจักรกล ตามค่าอัตราป้อนที่ต้องการได้อีกด้วย แต่จะสามารถควบคุมการเคลื่อนที่ได้ครั้งละ 1 แนวแกนเท่านั้น การเคลื่อนที่ของเครื่องมือจะถูกควบคุมด้วยอัตราป้อนและความยาวในการเคลื่อนที่ ระบบการควบคุมการตัดเฉือนแนวเส้นตรงชนิดนี้จะใช้กับเครื่องกัดและเครื่องกลึงแบบง่ายๆ

3) การควบคุมตามเส้นขอบรูป (Contouring control)

การควบคุมแบบนี้จะสามารถควบคุมการเคลื่อนที่ทำงานได้ดังนี้

- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการแบบเคลื่อนที่เร็วได้
- ควบคุมเครื่องมือให้เคลื่อนที่ขนานกับแนวแกนไปยังตำแหน่งที่ต้องการตามค่าอัตราป้อนได้
- ควบคุมเครื่องมือให้เคลื่อนที่ไปยังตำแหน่งใดๆ บนชิ้นงานที่กำหนดในแนวเส้นตรงและ เส้นโค้งตามค่าอัตราป้อนได้

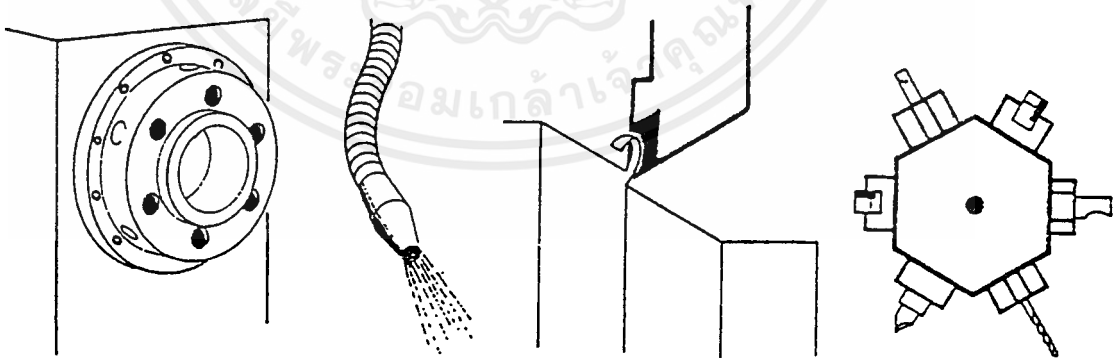
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 การควบคุมหน้าที่การทำงานของเครื่องจักรกล(Control of machine function)

กระบบควบคุมซีเอ็นซีนอกจากจะสามารถควบคุมการเคลื่อนที่ของเครื่องมือ ตามรูปทรงเรขาคณิตของชิ้นงานแล้ว ยังสามารถควบคุมหน้าที่การทำงานอื่นๆที่ช่วยเสริมการทำงานตัดเฉือนของเครื่องจักรกลให้เหมาะสมกับสภาวะการทำงานในขณะนั้นได้อีกด้วย จำนวนหน้าที่การทำงานและวิธีการควบคุมจะไม่ขึ้นอยู่กับตัวเครื่องจักรกลเพียงอย่างเดียว แต่ยังขึ้นอยู่กับชนิดระบบควบคุมอีกด้วย

ตัวอย่างหน้าที่การทำงานต่างๆ ที่จำเป็นจะต้องโปรแกรมเพื่อช่วยในการทำงาน ดังแสดงในรูป 2.22 มีดังนี้

- การเริ่มหมุนของเพลางาน ทิศทางการหมุนและการเปลี่ยนความเร็วรอบ
- การกำหนดตำแหน่งของเพลางาน
- การเปิดสารหล่อเย็น และความดันของสารหล่อเย็น
- การรักษาอัตราป้อนให้คงที่
- การเปลี่ยนตำแหน่งของเครื่องมือ
- การรักษาความเร็วตัดให้คงที่
- ชุดยันศูนย์ท้ายแท่น (Tail-stock)
- อุปกรณ์ใส่และถอดชิ้นงาน (loader and unloader)
- แท่นประคองศูนย์ (Steady rest)
- อุปกรณ์ลำเลียงเศษ (chip conveyor)
- Sorter



รูปที่ 2.22 หน้าที่การทำงานของเครื่องจักรกล

เครื่องจักรกลซีเอ็นซีที่สามารถใช้ระบบควบคุมสั่งการทำงานในหน้าที่ต่างๆ ได้ยิ่งมากเท่าใดก็จะเป็นระบบที่มีความเหมาะสมสำหรับการควบคุมแบบอัตโนมัติมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เรขาคณิตเบื้องต้นสำหรับการทำโปรแกรมเอ็นซี

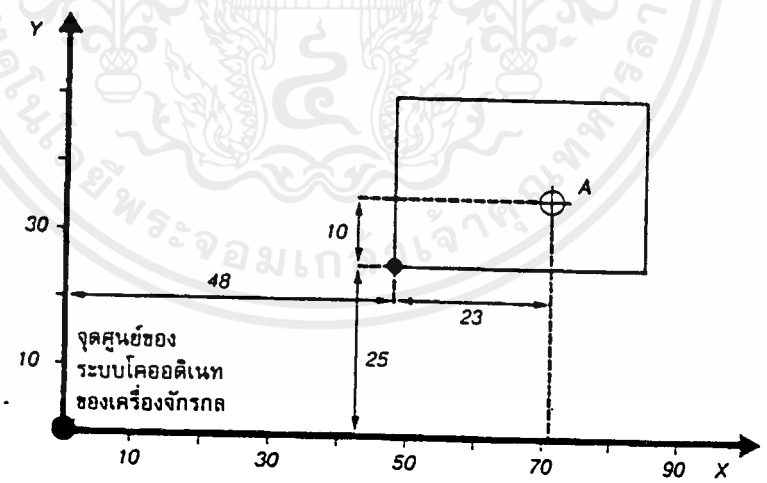
2.5.1 จุดศูนย์และจุดอ้างอิง (Zero points and Reference points)

การเคลื่อนที่ต่างๆ ของเครื่องจักรกลซีเอ็นซี จะถูกควบคุมด้วยระบบโคออดิเนท ตำแหน่งต่างๆที่ถูกต้องภายในพื้นที่ทำงานของเครื่องจักรกล จะวัดระยะมาจากจุดศูนย์ (Zero points) นอกจากจุดศูนย์แล้ว เครื่องจักรกลซีเอ็นซียังมีจุดอ้างอิง (Reference points)อื่นๆอีก เพื่อช่วยเสริมการทำงานและการทำโปรแกรม

1) จุดศูนย์ของเครื่อง (Machine zero point , M)

การจับยึดชิ้นงานเข้ากับเครื่องจักรกลซีเอ็นซีจะต้องให้สัมพันธ์กันระหว่างขนาดที่กำหนดในแบบงานกับระบบโคออดิเนทของเครื่องเพื่อให้สามารถเปรียบเทียบขนาดซึ่งกันและกันได้ เครื่องจักรกลซีเอ็นซีทุกเครื่องจะมีระบบโคออดิเนทประกอบอยู่ ระบบนี้จะกำหนดจากการเคลื่อนที่กับระบบวัดระยะการเคลื่อนที่ของเครื่องจักรกลที่มีอยู่

รูป 2.23 แสดงให้เห็นแบบชิ้นงานที่วางอยู่บนระบบโคออดิเนทของเครื่องจักรกล รู A ที่มีขนาดระยะกำหนดในแบบงาน คือ 23 และ 10 มม. จะมีค่าโคออดิเนทที่สัมพันธ์กับระบบโคออดิเนทของเครื่องจักรกล คือ โคออดิเนท X = 71 และ Y = 35 มม.



รูปที่ 2.23 แบบชิ้นงานในระบบโคออดิเนทของเครื่องจักรกล CNC

จุดเริ่มต้นของการกำหนดขนาดในแบบงาน (มุมซ้ายมือด้านล่าง) จะมีระยะเยื้องศูนย์จากระบบโคออดิเนทของเครื่องจักรกลซีเอ็นซี คือ X = 48 และ Y = 25 มม.

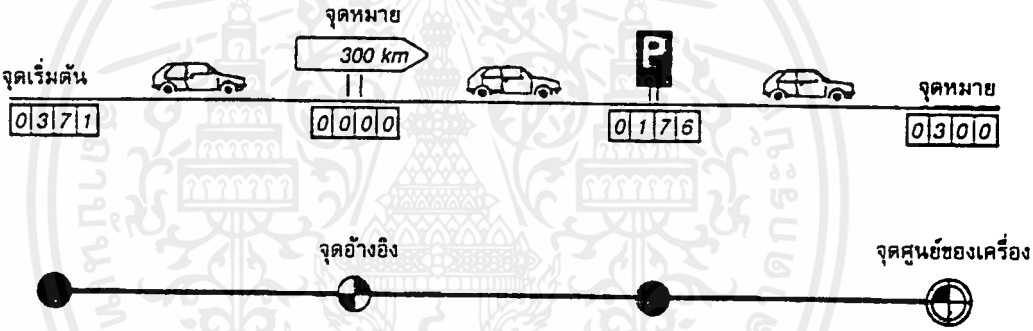
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดศูนย์ของเครื่อง (M) โดยทั่วไปจะใช้แทนด้วยสัญลักษณ์ ตำแหน่งจุดศูนย์ของเครื่องจะถูกกำหนดโดยบริษัทผู้ผลิตเครื่องจักรกลซีเอ็นซี จุดศูนย์ของเครื่องจะใช้เป็นจุดศูนย์ของระบบโคออดิเนทของเครื่องจักรกล และใช้เป็นจุดเริ่มต้นสำหรับระบบโคออดิเนทอื่นๆ และยังใช้เป็นจุดอ้างอิงในเครื่องจักรกลด้วย

สำหรับเครื่องกลึงจะมีตำแหน่งจุดศูนย์ของเครื่องที่แตกต่างกัน ซึ่งขึ้นอยู่กับบริษัทผู้ผลิตเครื่องกัด ดังนั้น ตำแหน่งจุดศูนย์ของเครื่องและทิศทางของแนวแกนที่ถูกต้องจึงต้องศึกษาจากคู่มือการปฏิบัติงานที่จัดเตรียมโดยบริษัทผู้ผลิตแต่ละบริษัท

2.52 จุดอ้างอิง (Reference point , R)

วัตถุประสงค์เบื้องต้นของการใช้จุดอ้างอิง สามารถอธิบายให้เข้าใจได้ง่ายโดยเปรียบเทียบกับการใช้หลักกิโลเมตรตามเส้นทางหลวงต่างๆ ดังแสดงในรูป 2.24



รูปที่ 2.24 วัตถุประสงค์ของการใช้จุดอ้างอิง

จากรูปที่ 2.24 สมมุติว่าขณะที่ท่านขับรถยนต์อยู่ มีเหตุการณ์ต่างๆ เกิดขึ้นตามลำดับขั้นตอนดังต่อไปนี้

2.1) ณ จุดเริ่มต้นการเดินทาง ท่านจะยังไม่ทราบว่า จุดหมายที่จะไปนั้นอยู่ห่างไกลออกไปเป็นระยะทางเท่าใด

2.2) ตลอดเส้นทาง ท่านจะคอยสังเกตดูหลักกิโลเมตร (จุดอ้างอิง) ซึ่งจะบอกว่าจุดหมายที่จะไปนั้นอยู่ห่างออกไปเท่าไร ท่านก็จะปรับตั้งมิเตอร์วัดระยะทางให้ตรงขีดศูนย์

2.3) จากนี้ไปท่านก็จะสามารถบอกได้ว่า ณ จุดต่างๆ ที่ท่านไปถึงนั้นอยู่ห่างจากจุดหมายของท่านเท่าไร หรืออยู่ห่างจากจุดอ้างอิงเท่าไร (หลักกิโลเมตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.24 จะมีสถานีอยู่ 3 แห่งด้วยกัน ซึ่งเมื่อเปรียบเทียบกับเครื่องจักรกลซีเอ็นซีแล้ว จะมีความหมาย

สถานีที่ 1 หมายถึง การเปิดสวิตช์เครื่อง

สถานีที่ 2 หมายถึง การเลื่อนไปยังจุดอ้างอิง และปรับตั้งระบบวัดระยะเลื่อนให้เริ่มต้นที่ศูนย์

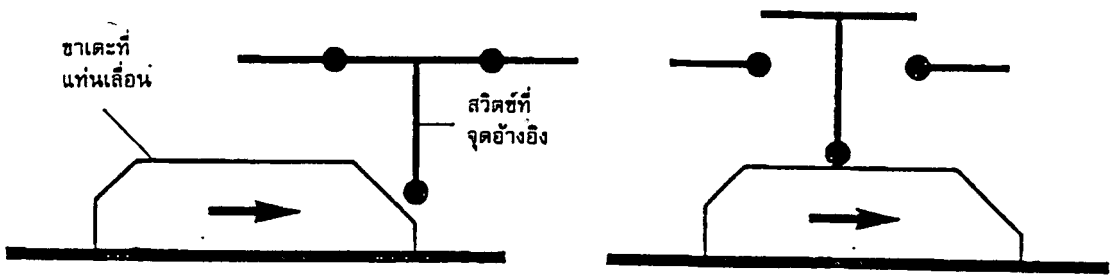
สถานีที่ 3 หมายถึง ตำแหน่งของเครื่องมือที่เคลื่อนที่ไปอย่างต่อเนื่อง

จุดอ้างอิง (R) มักจะใช้แทนด้วยสัญลักษณ์ เป็นจุดที่ใช้ช่วยในการปรับค่าและควบคุมระบบวัดขนาดระยะการเคลื่อนที่ของแท่นเลื่อน และเครื่องมือตำแหน่งของจุดอ้างอิงจะถูกกำหนดไว้ก่อนล่วงหน้าอย่างเที่ยงตรงในทุกแนวแกนของการเคลื่อนที่ด้วยสวิตช์จำกัดระยะหรือสวิตช์ขาเตะ (Limit switches หรือ Trip dogs) ดังนั้น ค่าโคออดิเนทของจุดอ้างอิงจะมีค่าเท่าเดิมเสมอ และรู้ค่าตัวเลขที่แน่นอนที่สัมพันธ์กับจุดศูนย์ของเครื่อง

ข้อสำคัญ : หลังจากเปิดสวิตช์ระบบควบคุมแล้ว แนวแกนทั้งหมดจะต้องเลื่อนไปยังจุดอ้างอิง ก่อนเสมอ เพื่อปรับค่าระบบวัดระยะการเคลื่อนที่

ถ้าเกิดเหตุขัดข้องขึ้นจนทำให้ข้อมูลของตำแหน่งแท่นเลื่อนและเครื่องมือในปัจจุบัน สูญหายไปจากระบบควบคุม ซึ่งอาจมีสาเหตุมาจากไฟฟ้าดับ เป็นต้น จะต้องเลื่อนแท่นเลื่อนต่างๆ กลับไปหาจุดอ้างอิงก่อนเริ่มทำงานใหม่เสมอ เพื่อปรับค่าของระบบวัดระยะการเคลื่อนที่ให้ถูกต้อง

เครื่องจักรกลซีเอ็นซีที่มีระบบวัดระยะการเคลื่อนที่แบบสัมบูรณ์(Absolute traverse measurement) อาจจะไม่ต้องใช้จุดอ้างอิง ทั้งนี้เพราะว่าสามารถอ่านค่าโคออดิเนทการเคลื่อนที่ของแนวแกนต่างๆ ได้โดยตรงและทุกเวลาที่ต้องการ อย่างไรก็ตาม สำหรับเครื่องจักรกลซีเอ็นซีที่ใช้ระบบวัดระยะการเคลื่อนที่แบบต่อเนื่อง (Incremental traverse measuring system) ยังต้องการใช้จุดอ้างอิงสำหรับการปรับค่าระบบวัดระยะการเคลื่อนที่



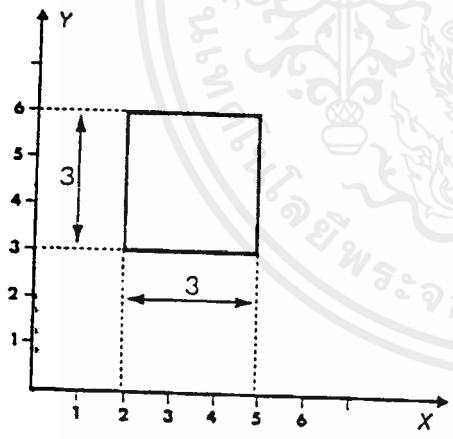
รูปที่ 2.25 การเคลื่อนที่ไปยังตำแหน่งของจุดอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

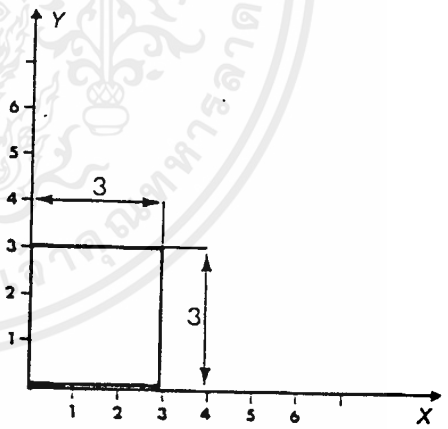
เมื่อขาเตะ (trip dog) เลื่อนสวิตช์ที่จุดอ้างอิงให้เปิดออกแล้ว ระบบวัดระยะการเคลื่อนที่ก็จะปรับค่าไปที่ศูนย์หรือค่าที่กำหนดไว้ล่วงหน้า เพื่อให้ได้ระดับความถูกต้องที่ต้องการ ในขณะที่ทำการปรับค่าของระบบวัดขนาดการเคลื่อนที่ของแท่นเลื่อนไปยังจุดอ้างอิงควรจะลดความเร็วลง และเลื่อนไปในทิศทางเดียวกันตลอด สำหรับเครื่องจักรบางแบบ เช่น เครื่องกัด ระบบวัดขนาดสามารถที่จะปรับค่าโดยการเลื่อนแท่นเลื่อนไปที่จุดศูนย์ของเครื่องก็ได้ อย่างไรก็ตาม ในกรณีทั่วไป การเคลื่อนที่ไปยังจุดศูนย์ของเครื่องจะไม่สามารถทำได้ เมื่อเครื่องมือและชิ้นงานอยู่ในตำแหน่งทำงานแล้ว ดังนั้น จึงต้องเพิ่มจุดอ้างอิงเข้าไปด้วย

2.53 จุดศูนย์ของชิ้นงาน (Workpiece zero point)

ในระบบโคออดิเนทสามารถที่จะเลือกข้อมูลโคออดิเนทได้ด้วยวิธีง่ายๆ โดยการกำหนดตำแหน่งที่จะวางชิ้นงานลงในระบบโคออดิเนทที่สะดวกต่อการอ่านค่าจุดโคออดิเนท รูปที่ 2.38 แสดงให้เห็นการวางแบบชิ้นงานสี่เหลี่ยมตรงจุดใดๆ ในระบบโคออดิเนท ส่วนรูปที่2.39 เป็นแบบชิ้นงานเดิมที่วางให้ขอบงานสองด้านซ้อนทับแกน X และแกน Y ซึ่งวิธีหลังนี้จะสามารถใช้ค่าขนาดที่กำหนดในแบบชิ้นงานเป็นค่าโคออดิเนทได้เลย และสามารถตรวจสอบค่าได้ง่ายกว่า และยังช่วยหลีกเลี่ยงการคำนวณหาค่าโคออดิเนทเพิ่มเติมได้



รูปที่ 2.26 การวางแบบชิ้นงานที่จุดใดๆ ในระบบโคออดิเนท



รูปที่ 2.27 การวางแบบชิ้นงานโดยใช้แนวแกนของระบบโคออดิเนทเป็นหลัก

จุดศูนย์ของชิ้นงาน (W) มักจะแสดงด้วยสัญลักษณ์ จะเป็นจุดที่ช่วยในการกำหนดระบบโคออดิเนทของชิ้นงานที่สัมพันธ์กับจุดศูนย์ของเครื่อง จุดศูนย์ของชิ้นงานจะถูกเลือกใช้โดยผู้เขียนโปรแกรม และป้อนเข้าไปในระบบซีเอ็นซีในขั้นตอนของการปรับตั้งตำแหน่งของจุดศูนย์ของชิ้นงานสามารถที่จะกำหนดเลือกใช้ได้อย่างอิสระโดยผู้เขียนโปรแกรม แต่ต้องอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในขอบเขตการทำงานของเครื่องจักรกล โดยมีหลักเกณฑ์ง่ายๆ คือ การกำหนดตำแหน่งจุดศูนย์ของชิ้นงานควรจะกำหนดไว้ในตำแหน่งที่เป็นจุดอ้างอิงต่างๆ ที่กำหนดไว้ในแบบชิ้นงานอยู่แล้ว กล่าวคือ เมื่อกำหนดตำแหน่งจุดศูนย์ของชิ้นงานแล้ว สามารถที่จะเปลี่ยนขนาดที่กำหนดในแบบชิ้นงานให้เป็นค่าโคออดิเนทได้โดยสะดวก และหลีกเลี่ยงการคำนวณค่าโคออดิเนทเพิ่มเติมได้

โดยทั่วไปแล้ว ตำแหน่งจุดศูนย์ของชิ้นงานมักนิยมใช้เป็นตำแหน่งเดียวกันกับจุดศูนย์ของโปรแกรม (Program zero point)

สรุป ตำแหน่งจุดศูนย์ของชิ้นงาน มีข้อพิจารณาในการเลือกใช้ดังนี้

3.1) สามารถกำหนดค่าโคออดิเนทจากขนาดกำหนดในแบบงานได้มากที่สุด

3.2) สามารถจับยึดชิ้นงาน ปรับตั้ง และตรวจสอบ ตลอดจนการควบคุมด้วยระบบวัดระยะการเคลื่อนที่ได้สะดวก

2.54 จุดอ้างอิงของเครื่องมือ (Tool reference points)

ในการเขียนโปรแกรมสำหรับการตัดเฉือนชิ้นงานตามเส้นขอบรูป สิ่งสำคัญที่จะต้องคำนึงถึง คือ การเคลื่อนที่ของขอบคมตัดของเครื่องมือที่ยึดอยู่กับชุดพาเครื่องมือ (Tool carrier) เช่น เพลางานของเครื่องกัด เป็นต้น จะต้องเคลื่อนที่ในลักษณะที่ทำให้ขอบคมตัดของมีดกัดเคลื่อนที่ตามเส้นขอบรูปของชิ้นงานอย่างถูกต้อง

การเขียนโปรแกรมขนาดความยาวของเครื่องมือ จะต้องตรงกับขนาดความยาวของเครื่องมือที่เป็นจริง ถ้าระบบควบคุมเริ่มทำงานด้วยขนาดความยาวของเครื่องมือที่ไม่ถูกต้องก็จะทำให้ไม่ได้ขนาดของเส้นขอบรูปที่ต้องการ ถ้าขนาดของเครื่องมือสั้นเกินไป) ก็จะตัดเฉือนเนื้อวัสดุออกไม่หมด และถ้าเครื่องมือยาวเกินไป ก็จะทำให้เกิดการตัดเฉือนเนื้อวัสดุชิ้นงานออกมากเกินไป

ดังนั้น ขนาดความยาวของเครื่องมือ จึงต้องทำการวัดขนาดก่อนที่เครื่องจักรกลจะเริ่มทำงานตามโปรแกรม และข้อมูลที่วัดได้จะถูกป้อนเข้าไปไว้ในระบบความจำส่วนที่ทำหน้าที่เก็บข้อมูลของเครื่องมือ (Tool data storage) ของระบบควบคุมซีเอ็นซี

ในการตัดเฉือนชิ้นงาน จุดสำคัญ คือ จะต้องสามารถควบคุมจุดปลายเครื่องมือหรือขอบคมตัดที่สัมพันธ์กับขนาดของเครื่องมือตลอดเส้นทางการเคลื่อนที่ตัดเฉือน (Machining path) เนื่องจากเครื่องมือมีรูปทรงและขนาดที่แตกต่างกัน ดังนั้น ขนาดของเครื่องมือที่ถูกต้องจะต้องหาให้ได้ก่อนและป้อนเข้าไปในระบบควบคุม ซึ่งขนาดของเครื่องมือจะต้องปรับตั้งให้สัมพันธ์กับจุดปรับตั้งเครื่องมือ (Tool setting point) ที่อยู่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ทฤษฎีทางคณิตศาสตร์ สำหรับ CNC (Mathematic theory for CNC)

ทฤษฎีทางการคำนวณทิศทางการเคลื่อนที่ (Feed rate speed)

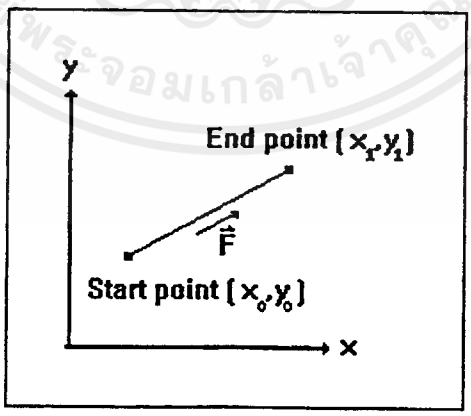
อัตราการป้อนของคำสั่ง G01,G02,G03 จะถูกกำหนดด้วยตัวเลขหลังคำสั่ง F____ และจะคงค่าเดิมตลอดจนกว่าจะมีการเปลี่ยนแปลงค่าใหม่ค่าอัตราการป้อนนี้เรียกว่า Tangential speed constant

1) Linear Interpolation

เมื่อมี Straight cut linear interpolation (G01) ส่วนควบคุมจะควบคุมแกนของเครื่องจักร 2 แกนหรือมากกว่านั้นพร้อมๆกัน ในส่วนการตัดเชิงมุมส่วนควบคุมจะใช้ข้อมูลที่ได้รับการโปรแกรมเข้ามาเก็บไว้แล้วเพื่อคำนวณ องศาหรือความชันของการตัดชิ้นส่วนของเส้นตรง ความยาวที่เปลี่ยนแปลงจากจุดเริ่มต้นจนถึงจุดสุดท้ายจะเป็นตัวกำหนด การแบ่งเส้นและความชันของแต่ละแกน เพื่อทำการควบคุมการเคลื่อนที่ของ Cutter ให้เคลื่อนที่เสมือนกับ การเคลื่อนที่ไปในทิศทางเดียว ในโครงงานนี้กำหนดให้ Linear interpolation เคลื่อนที่ในระนาบเดียวเท่านั้น

การใช้งานอย่างอื่นของ Linear interpolation คือใช้ในการประมาณเส้นโค้ง หรือวงกลม (Circular interpolation)ซึ่งจะกล่าวถึงวิธีการประมาณเส้นโค้งในช่วงต่อไป

การคำนวณใน Linear interpolation



รูปที่ 2.28 การเคลื่อนที่แบบ Linear Interpolation

F - อัตราป้อนในแนวการเคลื่อนที่

F_x - อัตราป้อนในแนวแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F_y- อัตราป้อนในแนวแกน Y

$$F_x = \frac{F(\text{ระยะทางที่เคลื่อนที่ในแนวแกน X})}{\text{ระยะทางที่เคลื่อนที่ทั้งหมด}} = \frac{F\cdot(x_1-x_0)}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}$$

$$F_y = \frac{F(\text{ระยะทางที่เคลื่อนที่ในแนวแกนy})}{\text{ระยะทางที่เคลื่อนที่ทั้งหมด}} = \frac{F\cdot(y_1-y_0)}{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}$$

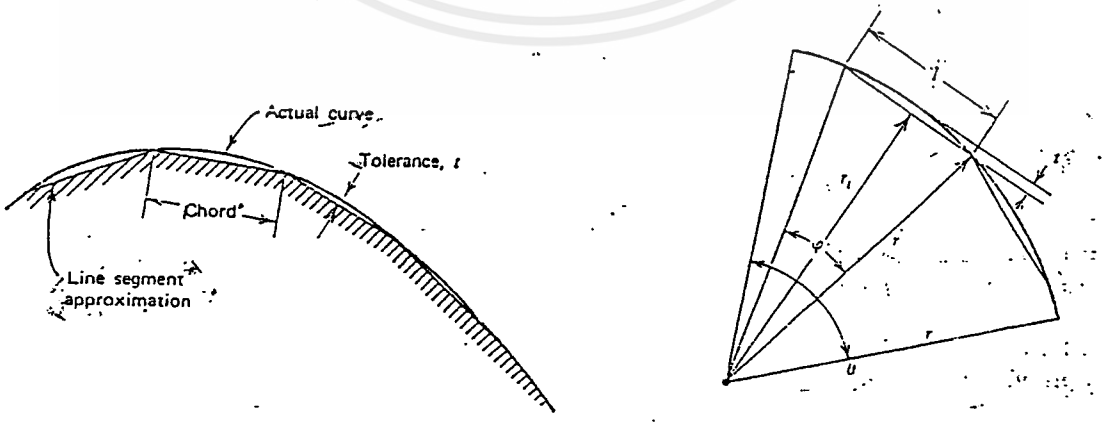
2) Circular Interpolation

Circular interpolation เป็นความสามารถในการตัดส่วนโค้งของวงกลม,จำนวนที่ตัดแปรผันตามขนาดของส่วนโค้งซึ่งขึ้นอยู่กับ รัศมีของส่วนโค้ง ส่วนควบคุมจะคำนวณทิศทางของ cutter จากข้อมูลที่โปรแกรมมา

ข้อมูลที่ต้องการได้แก่

1. ทิศทางของการเคลื่อนที่ (CW หรือ CCW) ตรวจสอบได้จาก Gcode ว่าเป็น G02 หรือ G03
2. จุดเริ่มต้นของส่วนโค้ง (arc) สามารถหาค่าได้จากการอ่านค่าตำแหน่งของแท่นใบมีดขณะนั้น
3. จุดสิ้นสุดของส่วนโค้ง (arc) ได้จากค่าที่ผู้ใช้ป้อนเข้ามา
4. รัศมีส่วนโค้ง (radius) ได้จากค่าที่ผู้ใช้ป้อนเข้ามา

หลังจากที่ทราบค่าต่างๆแล้วจะนำมาคำนวนโดยอาศัยหลักการของ Contour approximation using line segment ดังนี้



รูปที่ 2.29 Contour approximation using line segments

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการของ Contour approximation using line segment อาศัยหลักของการประมาณค่าส่วนโค้งด้วยเส้นตรง โดยมีขั้นตอนดังนี้

1. หาความยาวของส่วนโค้งทั้งหมด ($r\theta$)
2. กำหนดความละเอียดในการแบ่งส่วนโค้ง (n)
3. หามุมของเส้นตรงที่เกิดจากการแบ่งส่วนโค้ง

$$\varphi = \frac{\theta}{n}$$

เมื่อได้ค่าต่างๆที่ต้องการแล้วก็ทำการเดินแท่นใบมีดไปตามเส้นตรงนั้นๆ จะทำให้เกิดการเดินแบบกลึงโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบส่วนประกอบของเครื่องกลึงซีเอ็นซี

ในระบบของเครื่องกลึงซีเอ็นซีควบคุมโดยไมโครคอมพิวเตอร์นั้น จะมีส่วนประกอบหลักๆ อยู่ 2 ส่วนดังนี้

1. ส่วนของเครื่องกลึงซีเอ็นซี
2. ส่วนของระบบควบคุมเซอร์โวมอเตอร์

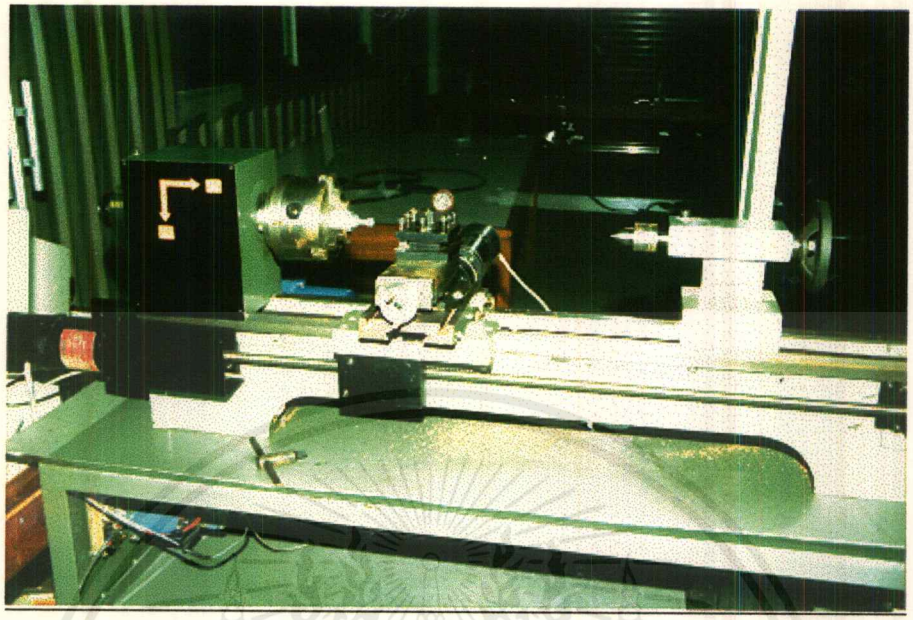
โดยในแต่ละส่วน ประกอบไปด้วยอุปกรณ์และเครื่องมือต่างๆ ต่อไปนี้

3.1 ส่วนของเครื่องกลึงซีเอ็นซี

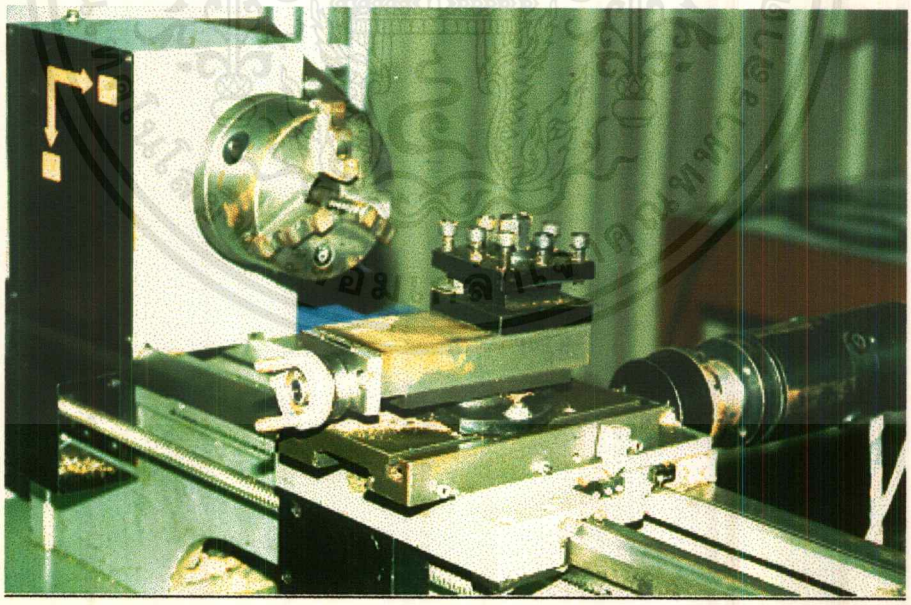
ในส่วนของตัวเครื่องกลึงเป็นตัวเครื่องกลึงที่มี 2 แกน ดัดแปลงมาจากเครื่องกลึงมือโดยทั่วไป โดยตัดเอาชิ้นส่วนย่อยออกไป เช่นมือหมุน มาตรวัดที่ติดกับแกน คงให้เหลือแต่หัวจับแท่นวางใบมีด ที่ยันศูนย์ ซึ่งมีรายละเอียดทางด้านเทคนิคดังนี้

- ฐานของเครื่องกลึงทำด้วยเหล็กหล่อ ขนาดกว้าง 580 มม. ยาว 1380 มม. สูง 830 มม.
- ระยะการเคลื่อนที่ของแนวแกน X = 150 มม.
- ระยะการเคลื่อนที่ของแนวแกน Z = 550 มม.
- Ball Screw ขนาดเส้นผ่าศูนย์กลาง 16 มม. ยาว 220 มม. ระยะ pitch 5 มม. ระยะ lead 5 มม ในแนวแกน X
- Ball Screw ขนาดเส้นผ่าศูนย์กลาง 16 มม. ยาว 1020 มม. ระยะ pitch 5 มม. ระยะ lead 5 มม ในแนวแกน Z
- หัวจับชิ้นงานเป็นแบบ 3 ฟันจับ
- ความเร็วการเคลื่อนที่ของแนวแกน ตั้งแต่ 1 - 5 เมตร/นาที
- ขนาดของชิ้นงานที่สามารถกลึงได้มีเส้นผ่าศูนย์กลาง 55 มม. ยาว 450 มม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

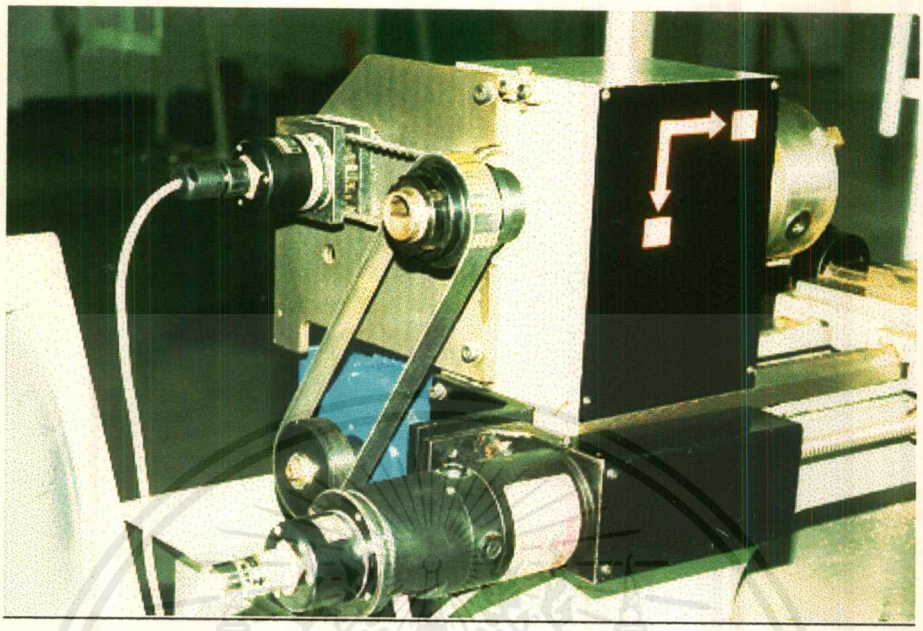


รูปที่ 3.1 ตัวแท่นเครื่องกลึงต้นแบบซีเอ็นซี



รูปที่ 3.2 หัวจับชิ้นงาน แท่นวางใบมีด และดีซีเซอร์โวมอเตอร์แนวแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



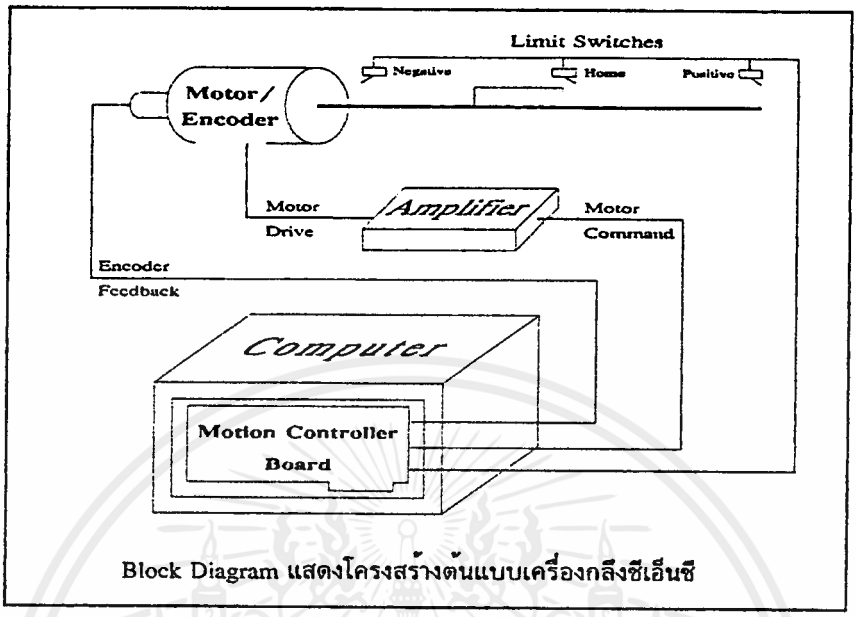
รูปที่ 3.3 เพลางานของเครื่องกลึง และมอเตอร์ในแนวแกน Z

รูปที่ 3.4 บอลล์สกรู (Ball Screw)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนของระบบควบคุมเซอร์โวมอเตอร์

ประกอบด้วยอุปกรณ์ต่าง ๆ ดังรูป 3.5



รูปที่ 3.5 แสดงระบบควบคุมเซอร์โวมอเตอร์

จากภาพจะพบว่าในระบบควบคุมมีอุปกรณ์ที่สำคัญๆ ดังนี้

3.2.1 การ์ดควบคุมสำเร็จรูป Model 5650

ลักษณะและความสามารถ

1.1 ใช้ชุดชิพของ PMD รุ่น MC1401

1.2 รีจิสเตอร์เก็บค่าตำแหน่ง,ความเร็ว,ความเร่ง และเจิร์ค ขนาด 32 บิต

1.3 สามารถรับค่าจากลิมิตสวิตช์ ป้องกันการเคลื่อนที่เกินขอบเขตได้ 2 ตัว ต่อ 1 แกน

1.4 มีรูปแบบการเคลื่อนที่ 3 แบบ คือ S-curve,Trapezoidal และ Velocity contouring

1.5 ใช้เวลา 100 µS ต่อการ update ข้อมูลหนึ่งครั้ง

1.6 มี Electronic Gearing

1.7 มีรูปแบบของ Digital Filter 2 แบบ คือ PID และ PIVFF

1.8 สามารถรับสัญญาณพัลซ์ที่มาจากเอนโคดเดอร์ได้สูงถึง 1 ล้านพัลซ์ต่อวินาที

1.9 มีรีจิสเตอร์ที่ใช้เก็บค่าตำแหน่ง Home และ Index ความเร็วสูง

1.10 มีสัญญาณควบคุม 2 แบบให้เลือก คือ 16-bit ± 10V DAC และ 10-bit PWM

1.11สามารถเปลี่ยนแปลงรูปแบบการเคลื่อนที่ และพารามิเตอร์ได้ขณะที่มีการทำงานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.12 มี counter แบบ 1/T สำหรับการเคลื่อนที่ที่ความเร็วต่ำๆ

1.13 มีการหยุดมอเตอร์โดยอัตโนมัติเมื่อมีข้อผิดพลาดเกิดขึ้น

1.14 ผู้ใช้สามารถกำหนดอินเตอร์รัพได้

1.15 ใช้พอร์ต I/O ขนาด 8 บิต

การใช้งานการ์ดควบคุมสำเร็จรูป Model 5650

<u>การติดตั้ง</u>

การ์ดควบคุม ต้องการใช้ไฟเลี้ยงขนาด +5 Vdc และ +12 Vdc จาก ISA busในเครื่องคอมพิวเตอร์ จึงต้องทำการเสียบการ์ดควบคุมลงในช่องเสียบของ ISA bus ที่มีอยู่บน Main Board ของคอมพิวเตอร์ แล้วทำการเซ็ต Address ของการ์ดควบคุมให้อยู่ในตำแหน่งที่ยังว่างอยู่ในที่นี้ เซ็ตค่า Address เป็น 300H สำหรับการเซ็ตค่า Address ของการ์ดควบคุมนั้นสามารถทำได้โดย การเซ็ต Jumper W2 เสียใหม่(ดูรูป ก.1,ก.2 ในภาคผนวก ก.)

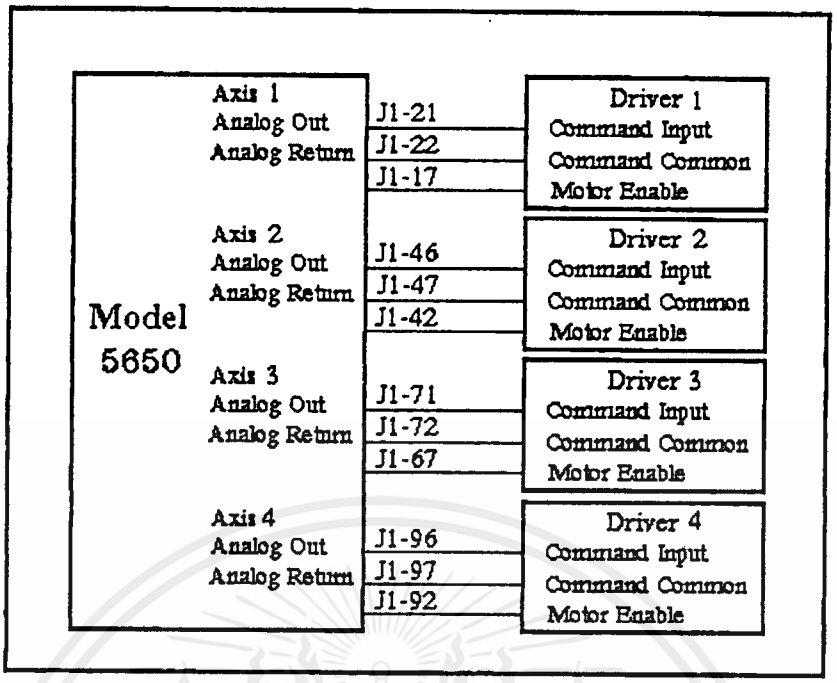
<u>การเชื่อมต่อ</u>

การเชื่อมต่อระหว่างการ์ดควบคุมกับอุปกรณ์ภายนอกต่างๆเช่น Driver, Encoder และ Limit Switchs สามารถทำได้โดย วิธีการต่อไปนี้

1. Driver - สัญญาณควบคุมที่จะส่งจากการ์ดควบคุมไปยัง Driver นั้นเราจะใช้สัญญาณไฟฟ้าขนาด 0 ถึง $\pm$ 10V ในการควบคุมความเร็วและทิศทางการหมุนของ Survo Motor ดังนั้นสายสัญญาณที่เชื่อมระหว่างการ์ดควบคุม และ Driver จะใช้ 2 เส้นต่อการควบคุม Survo Motor 1 แกน โดยเส้นหนึ่งเป็นสายของสัญญาณไฟฟ้าแบบ Analog ส่วนอีกเส้นหนึ่งจะเป็นสายสัญญาณอ้างอิง(Analog Return)สำหรับสายสัญญาณทั้ง Analog Output และ Analog Return นั้นจะต่อออกมาจากConnector J1 (ดูรูป ก.3, ก.4 ในภาคผนวก ก.)

สัญญาณอีกตัวหนึ่งที่ส่งจากการ์ดควบคุมไปยัง Driver ก็คือ Motor Enable Signal ซึ่งทำหน้าที่ในการยกเลิกการใช้งาน (Disable) Driver ชั่งคราวในช่วงที่การ์ดควบคุมเริ่มทำงาน (Power-up) เพื่อป้องกันไม่ให้เกิดความผิดพลาดขึ้นในระหว่างเริ่มเปิดเครื่องหลังจากที่การ์ดควบคุมพร้อมที่จะทำงานแล้วจึงทำการเรียกใช้(Enable) Driver ตามปกติ โดยในแต่ละแกนจะมีสัญญาณนี้ 1 เส้น โดยทำการต่อออกมา จาก Connector J1 (ดูรูป ก.3, ก.5 ในภาคผนวก ก.)ในกรณีที่ต้องการเปลี่ยนสัญญาณ Output ที่สายสัญญาณนี้ จากเดิมในสถานะ Power-up ($\pm$ 5 Vdc) และ สถานะพร้อมทำงาน (0 Vdc) ไปเป็นสถานะ Power-up(0 Vdc)และสถานะพร้อมทำงาน($\pm$ 5 Vdc) ก็สามารถทำได้โดยการเซ็ต Jumper W3 ใหม่(ดูรูป ก.6 ในภาคผนวก ก.) ในส่วนรายละเอียดเกี่ยวกับการควบคุมตำแหน่งและความเร็วของ Driver ตัวนี้จะอยู่ในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงการเชื่อมต่อระหว่าง Model 5650 กับ Driver

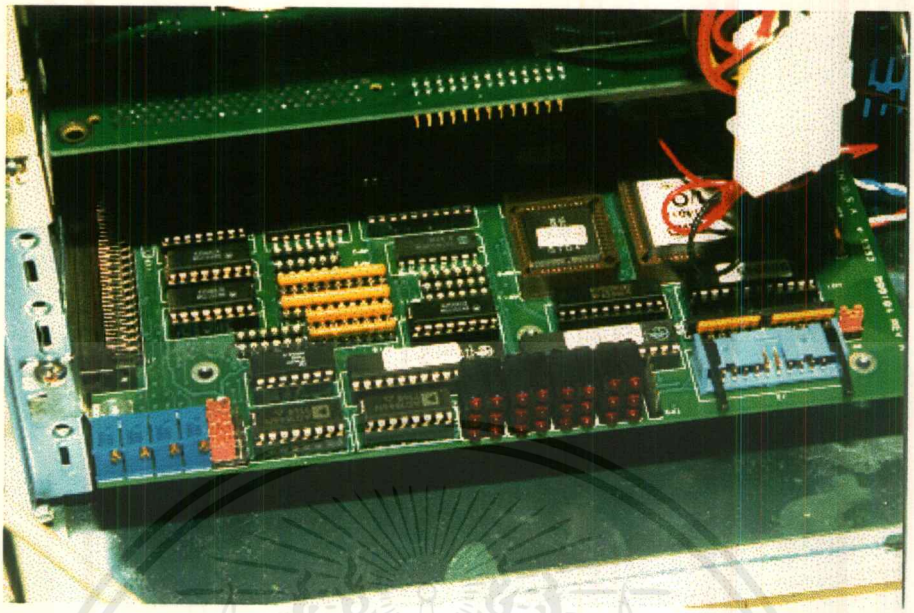
2. Encoder - สำหรับ Encoder ที่ใช้นั้นจะเป็นแบบ Differential Encoder ของ HEIDENHAIN Model ROD 426.0013-1024 (f = 300kHz) ซึ่งสัญญาณจะมีทั้งหมด 3-เฟสคือ เฟส A,เฟสB,และ Index แต่ละเฟสก็จะมีสายสัญญาณ 2 เส้นคือสายสัญญาณจริงหนึ่งเส้นและสัญญาณ Completement อีกหนึ่งเส้นรวม 6 เส้นอีก 2 เส้นจะเป็นสายไฟเลี้ยงที่จ่ายให้กับ Encoder รวมทั้งหมดจะมีสายสัญญาณอยู่ 8 เส้นต่อ Encoder 1 ตัว ในส่วนของการต่อเข้ากับการ์ดควบคุมนั้นสามารถทำได้โดยต่อสายเข้ากับConnector J1(ดูรูป ก.3, ก.7ในภาคผนวก ก.)

3. Limit Switchs - ในแต่ละแกนของ Servo Motor จะต้องใช้ Limit Switch จำนวน 3 ตัว โดยแบ่งหน้าที่เป็น

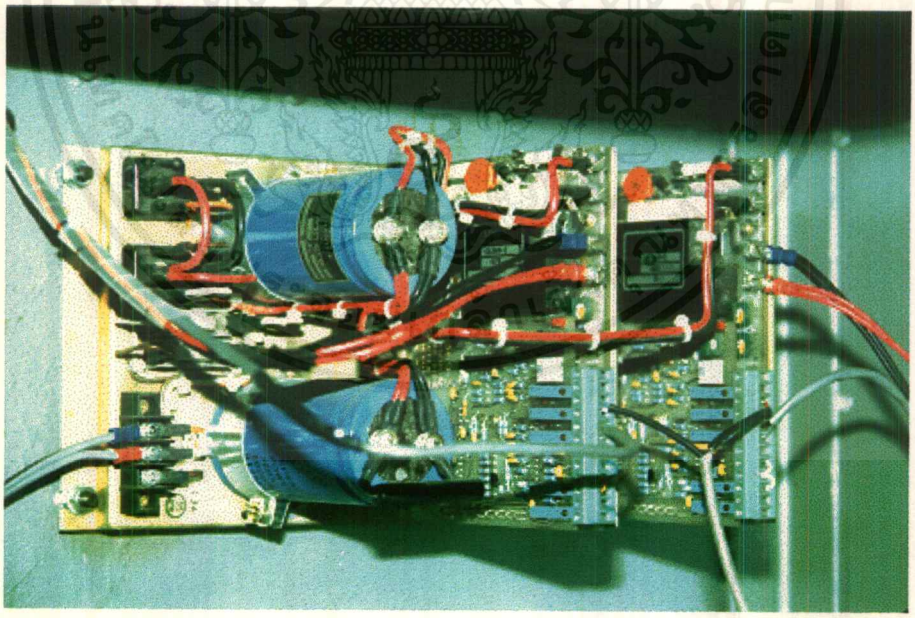
- Over-travel Limit Input จำนวน 2 ตัว ทำหน้าที่ในการส่งสัญญาณไปยังการ์ดควบคุมเพื่อบอกให้ทราบว่าขณะนั้นโต๊ะงานได้เคลื่อนที่ไปจนสุดขอบเขตของการเคลื่อนที่แล้ว
- Home Inpuut จำนวน 1 ตัวทำหน้าที่ในการส่งสัญญาณไปยังการ์ดควบคุมเพื่อเพิ่มหรือลดค่าใน Position Capture Register ซึ่งสามารถนำผลจากการเปลี่ยนแปลงนี้ไปประยุกต์ได้

สำหรับสัญญาณจาก Limit Switchที่จะต่อเข้าการ์ดควบคุมนั้น จะต้องต่อเข้ากับ Connector J1(ดูรูป ก.3, ก.8 ในภาคผนวก ก.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

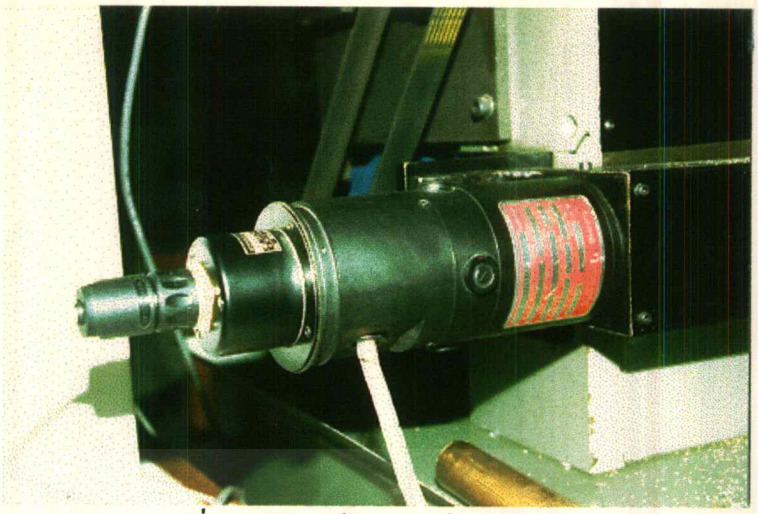


รูปที่ 3.7 การ์ดควบคุมการเคลื่อนที่หลัก Model 5650



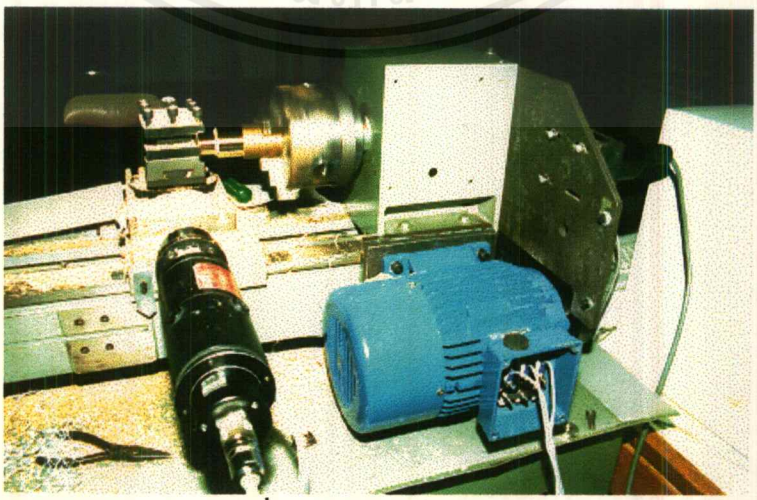
รูปที่ 3.8 DC Servo Amplifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ดีซีเซอร์โวมอเตอร์ และ Encoder

รูปที่ 3.10 Flux Vector



รูปที่ 3.11 เอซีเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุม

การควบคุมการทำงานของการ์ดควบคุมให้สามารถทำงานได้ตามที่เราต้องการนั้น สามารถทำได้โดยการเขียนโปรแกรมสั่งงานการ์ดผ่านทางคอมพิวเตอร์ โดยภาษาที่จะใช้การเขียนโปรแกรมนั้น ในที่นี้จะใช้ภาษา C เนื่องจาก เราจำเป็นจะต้องใช้ชุดคำสั่งที่ให้มาพร้อมกับการ์ดควบคุม ชุดคำสั่งนี้ปรากฏอยู่ในโปรแกรม host_io.c ซึ่งเขียนขึ้นด้วยภาษา C จึงเป็นการง่ายที่จะใช้ภาษา C ในการเขียนโปรแกรมควบคุม

ในส่วนรายละเอียดของโปรแกรมควบคุมที่จะเขียนนั้น ก็จะเป็นคำสั่งที่เรียกใช้งานชุดคำสั่งในโปรแกรม host_io.c เสียเป็นส่วนใหญ่ รายละเอียดของเนื้อหาเกี่ยวกับความหมายของคำสั่งแต่ละตัวในโปรแกรม host_io.c สามารถดูได้จากภาคผนวก ก.

3.2.2 การคำนวณและเลือกระบบเซอร์โวมอเตอร์

ในการเลือกระบบเซอร์โวมอเตอร์ของต้นแบบเครื่องกลึง ได้ทำการเลือกระบบดีซีเซอร์โวมอเตอร์สำหรับการขับเคลื่อนแนวแกน X , Z และระบบเอซีเซอร์โวมอเตอร์สำหรับการขับเคลื่อน Spindle

3.221 การคำนวณหาขนาดดีซีเซอร์โวมอเตอร์ในการขับเคลื่อนแนวแกน X และ Z

ในการคำนวณขนาดกำลังขับของดีซีเซอร์โวมอเตอร์ ในการขับเคลื่อนแนวแกน X และ Z ต้องทราบค่าตัวแปรต่างๆ ทางด้านเชิงกลดังตารางที่ 3.1

ตัวแปรเชิงกล	แนวแกน X	แนวแกน Z
- น้ำหนักของแท่นเลื่อนรวมน้ำหนักใบมีด (m)	15 ก.ก.	40 ก.ก.+ Motor_X
- ขนาดของ Ball Screw	16 มม.	16 มม.
- ความยาวของ Ball Screw	220 มม.	1020 มม.
- ระยะ pitch ของ Ball Screw	5 มม.	5 มม.
- ความเร็วสูงสุด (V_{max})	5 เมตร/นาที	5 เมตร/นาที
- แรงตัดเฉือน (F)	10 ก.ก.	10 ก.ก.
- อัตราทด (i)	1 : 1	1 : 1
- สัมประสิทธิ์แรงเสียดทาน (μ)	0.15	0.15

ตารางที่ 3.1 ข้อมูลตัวแปรต่างๆ เชิงกลของการคำนวณระบบเซอร์โวมอเตอร์ของต้นแบบเครื่องกลึงซีเอ็นซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณเขนาดกำลัง (Torque) ของระบบดีซีเซอร์โวมอเตอร์ สำหรับต้นแบบเครื่องกลึง CNC

$$TORQUE_{total} = T_{friction} + T_{weight} + T_{cuttiong} + T_{acceleration}$$

1.) $T_{friction} = [(TF_{table} + TF_{spindle}) \times i\ ]$

$$TF_{table} = \frac{m \times g \times \mu \times h \times i}{2\pi}$$

$$= \frac{45\ \text{kg} \times 9.81 \times 0.15 \times 0.005\ \text{m} \times 1}{6.28}$$

$$= 0.0527\ \text{Nm}$$

$$TF_{spindle} = \frac{DIA_{ball\ screw} \times F \times i}{1000}$$

$$= \frac{16\ \text{mm}}{1000} \times 10 \times 1$$

$$= 0.16\ \text{Nm}$$

$$T_{friction} = (0.0527\ \text{Nm} + 0.16\ \text{Nm}) \times 1$$

$$= 0.2127\ \text{Nm}$$

2.) $T_{weight} = \frac{m \times g \times \sin\alpha \times h \times i}{2\pi}$

$$T_{weight} = \frac{45\ \text{kg} \times 9.81 \times \sin 0 \times 0.005\ \text{m} \times 1}{6.28}$$

$$= 0\ \text{Nm}$$

3.) $T_{cutting} = \frac{F \times g \times h \times i}{2\pi}$

$$T_{cutting} = \frac{10\ \text{kg} \times 9.81 \times 0.005\ \text{m} \times 1}{6.28}$$

$$= 0.0731\ \text{Nm}$$

4.) $RPM_{motor} = V_{max} \times \text{Pitch} \times I$

$$RPM_{motor} = \frac{5\ \text{m/min}}{0.005\ \text{m/min}} \times 1$$

$$= 1000\ \text{REV/min}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.) $T_{static} = T_{friction} + T_{weight} + T_{cutting}$

$T_{static} = 0.2127 \text{ Nm} + 0 \text{ Nm} + 0.0731 \text{ Nm}$

$= 0.2858 \text{ Nm}$

6.) Preselection motor : $T_{static} < T_{motor}$

ดังนั้นในการเลือกใช้ดีซีเซอร์โวมอเตอร์จะต้องเลือกมอเตอร์ให้มีค่า Torque และความเร็วรอบมากกว่า Torque ของระบบที่คำนวณได้ สำหรับต้นแบบเครื่องกลึงนี้เลือกใช้ดีซีเซอร์โวมอเตอร์ของ Glentek Model GM3320-22 (ดูรายละเอียดในภาคผนวก ข.)

3.222 การคำนวณหาขนาดเอซีเซอร์โวมอเตอร์ในการขับเคลื่อน Spindle

ในการคำนวณหาขนาดของเอซีเซอร์โวมอเตอร์ในการขับเคลื่อน Spindle สามารถคำนวณได้จาก

$$HP_m = \frac{HP_c}{E}$$

โดยที่ HP_m = horsepower at spindle motor

HP_c = horsepower at cutter

$= UHP \cdot MRR \cdot C$

E = machine efficiency

UHP = unit horsepower (power required to remove 1 cubic inch of material per minute)

MRR = material removal rate (cubic inches per minute)

$= \frac{\pi \cdot (D^2 - d^2) \cdot f_r \cdot N}{4}$

C = feed correction factor

D = original(major) diameter of workpiece (inch)

d = diameter after cut (inch)

f_r = feed (IPR)

N = revolutionn per minute (RPM)

$= \frac{1000 \cdot CS}{\pi \cdot D}$

CS = surface cutting speed (m/min)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดสำหรับการคำนวณเพื่อหาค่าขนาดของเอซีเซอร์โวมอเตอร์ของต้นแบบเครื่องกลึงซีเอ็นซี มีดังนี้

1. ใช้ใบมีดแบบ carbide ซึ่งมี surface cutting speed (CS) = 180 m/min และอัตราป้อน (feed, f_r) = 0.01 IPR
2. ชิ้นงานเป็นทองเหลือง มีค่า UHP = 0.5 in^3/min ขนาดชิ้นงานที่ก่อนที่จะกลึง มี ϕ = 50 mm. หลังกลึงจะมี ϕ = 46 mm. และมีค่า C = 1
3. มอเตอร์มีประสิทธิภาพ 90 %

ดังนั้น $N = \dfrac{1000 \times 180 \text{ m/min}}{3.14 \times 50 \text{ mm.}}$

$= 1146 \text{ RPM}$

$MRR = \dfrac{3.14 \times [(1.97 \text{ in})^2 - (1.81 \text{ in})^2] \times 0.01 \text{ IRR} \times 1146 \text{ RPM}}{4}$

$= 5.45 \text{ in}^3/\text{min}$

$HP_c = 0.5 \text{ in}^3/\text{mim} \times 5.45 \text{ in}^3/\text{mim} \times 1$

$= 2.725 \text{ HP}$

$HP_m = \dfrac{2.725 \text{ HP}}{0.90}$

$= 3.0 \text{ HP}$

หมายเหตุ ค่าของ unit horsepower (UHP), surface cutting speed (CS), feed correction factor (C) และ feed (f_r) ของชิ้นงานและใบมีดแต่ละชนิดดูได้จากภาคผนวก ค.

ในส่วนการควบคุมความเร็วรอบของเอซีเซอร์โวมอเตอร์ จะควบคุมด้วย Flux Vector ของ DymamAx model DF400S ซึ่งจะควบคุมผ่านทางการ์ดควบคุม Model 5650

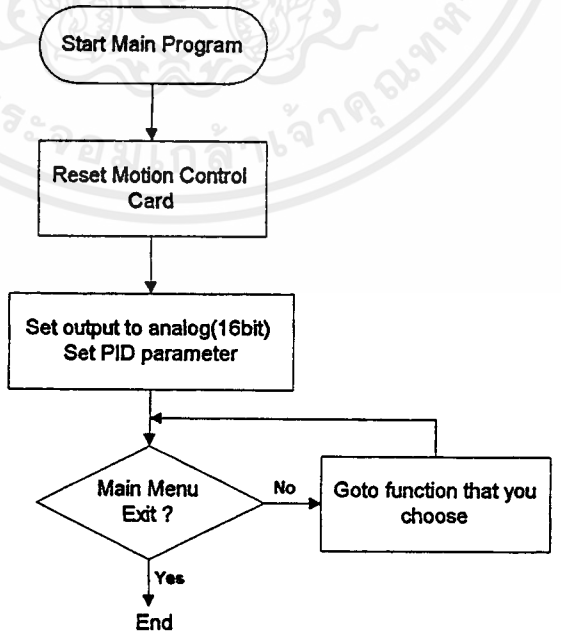
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบโปรแกรม

แนวทางการเขียนโปรแกรมเพื่อใช้ในการควบคุมการทำงานของเครื่องกลึง ในที่นี้ได้เลือกใช้ภาษา C ในการเขียนโปรแกรมเนื่องจากเป็นภาษาที่มีความยืดหยุ่นสูง สามารถติดต่อกับฮาร์ดแวร์ได้ดีในระดับหนึ่ง และคำสั่งพื้นฐานของการ์ด PMD 5650 เขียนมาในรูปของภาษา C

ในโปรแกมที่เขียนขึ้นมาจะประกอบด้วยส่วนต่าง ๆ ดังนี้

- การแปลงคำสั่ง G (G code) ให้เป็นการทำงานของเครื่องกลึง โดยใช้คำสั่งที่มีอยู่ในการ์ดมาเขียนเป็นการทำงานของเครื่อง
- การเลื่อนแท่น และใบมีด โดยใช้ Manual Control ในที่นี้เราได้ใช้ Arrow key ในการเลื่อนแท่นใบมีดและมีการป้อนอัตราความเร็วได้ตามต้องการ
- การ Set Zero เป็นการตั้งค่าที่แท่นใบมีดอยู่ขณะนั้น ให้เป็นตำแหน่ง 0 , 0
- การ Save, Load, Edit G code ในรูปของ File
- การป้อนค่า G code ทางหน้าจอ(Enter command) เพื่อทำงานตามคำสั่งที่ป้อนเข้าไปเมื่อป้อนแล้วสามารถสั่งให้ทำงาน โดยสั่งจากคำสั่งการทำงานอัตโนมัติ
- การเปิดและปิดการทำงานของเอซีเซอร์โวมอเตอร์ขับ Spindle
- การทำงานโดยอัตโนมัติ โดยทำงานตาม File G code ที่เขียนหรือ Load ขึ้นมา

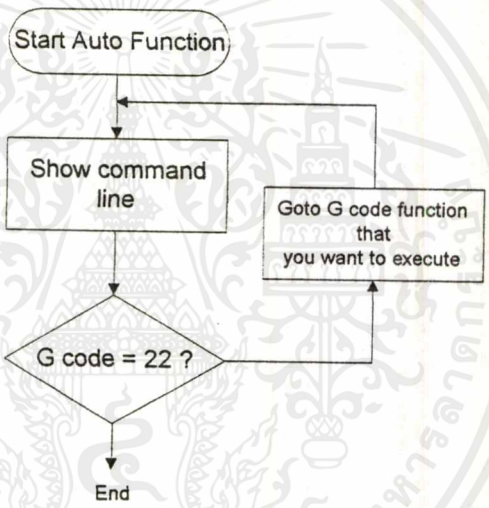


รูปที่ 4.1 Flowchart แสดงการทำงานของโปรแกรมหลัก (Main Program)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
********** KINGMONGKOUT INSTITUTE OF TECHNOLOGY LADKRABANG **********
**********          CNC LATHE MACHINE CONTROL          **********

SELECT MODE :                    1. MANUAL
                                 2. ENTER COMMAND
                                 3. LOAD
                                 4. SAVE
                                 5. SET ZERO
                                 6. ON/OFF SPINDLE
                                 7. GO
                                 8. EXIT
>> _
```

รูปที่ 4.2 เมนูหลัก

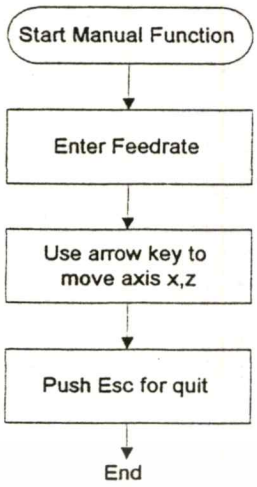


รูปที่ 4.3 Flowchart แสดงการทำงานของฟังก์ชันอัตโนมัติ (Auto Function)

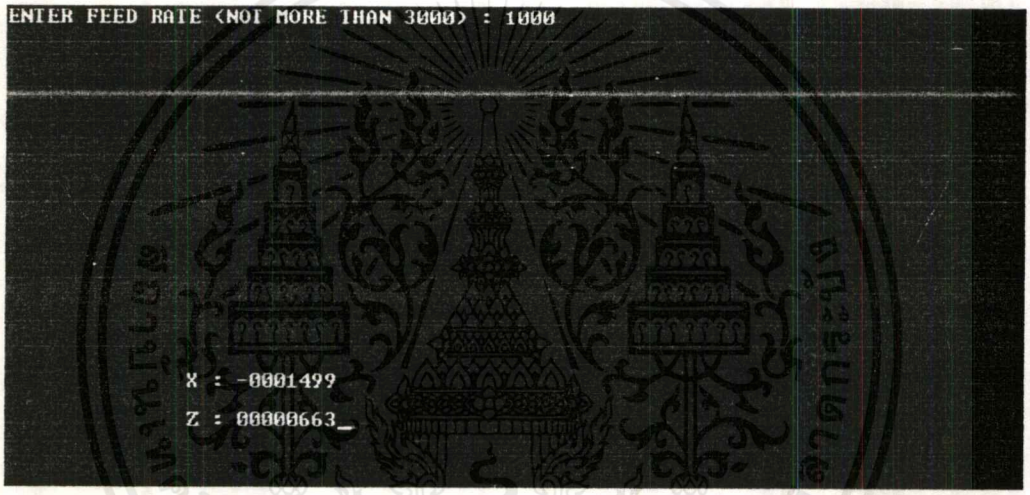
```
NOW OPERATING.....

COMMAND N1 G1 X-200 Z2000 Feed200

            X : -0000102
            Z : 00001026_
```

รูปที่ 4.4 หน้าจอแสดงผลของฟังก์ชันอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



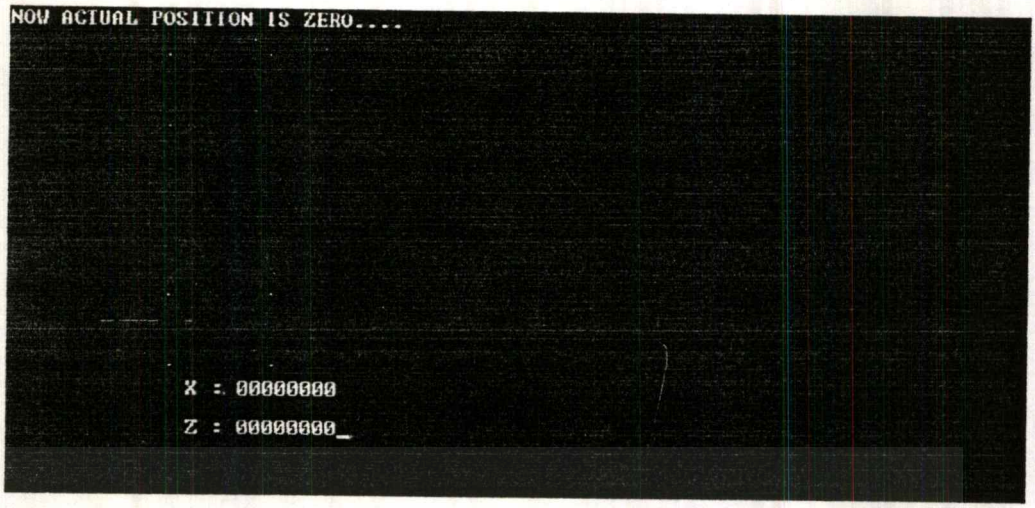
รูปที่ 4.5 Flowchart แสดงการทำงานของ Manual Function



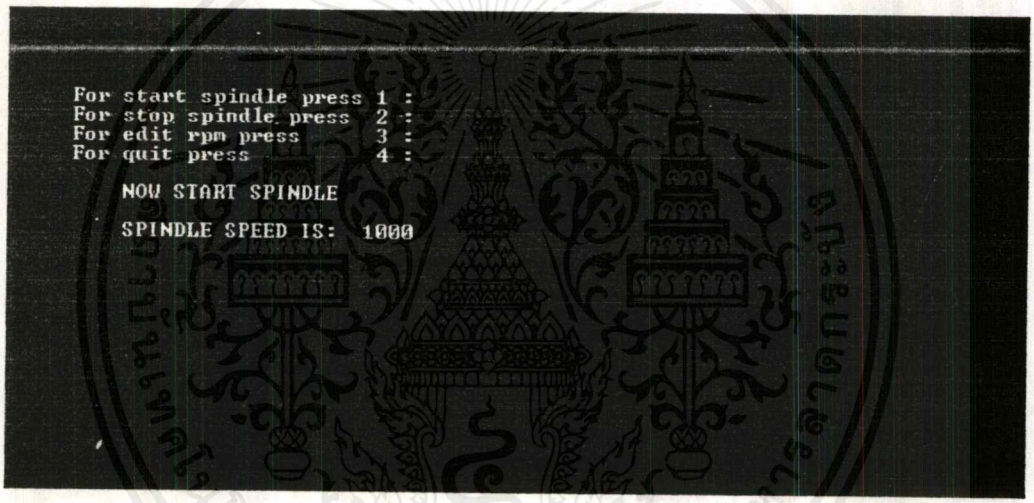
รูปที่ 4.6 หน้าจอแสดงผลของ Manual Function

รูปที่ 4.7 หน้าจอแสดงผลของฟังก์ชัน Load File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าจอแสดงผลของฟังก์ชัน Set Zero



รูปที่ 4.9 เมนูย่อยของฟังก์ชันควบคุม Spindle

ในส่วนของ G code ที่เขียนขึ้นเพื่อควบคุมการทำงานของเครื่องกลึงต้นแบบซีเอ็นซีได้แก่ (อ้างอิงตามเครื่องกลึงของ Fanuc)

Group 01

G00 (Positioning)

G01 (Linear interpolation)

G02 (Circular interpolation , CW (clockwise))

G03 (Circular interpolation , CCW (counterclockwise))

G92 (Simple thread cutting cycle)

G22 (end G code)

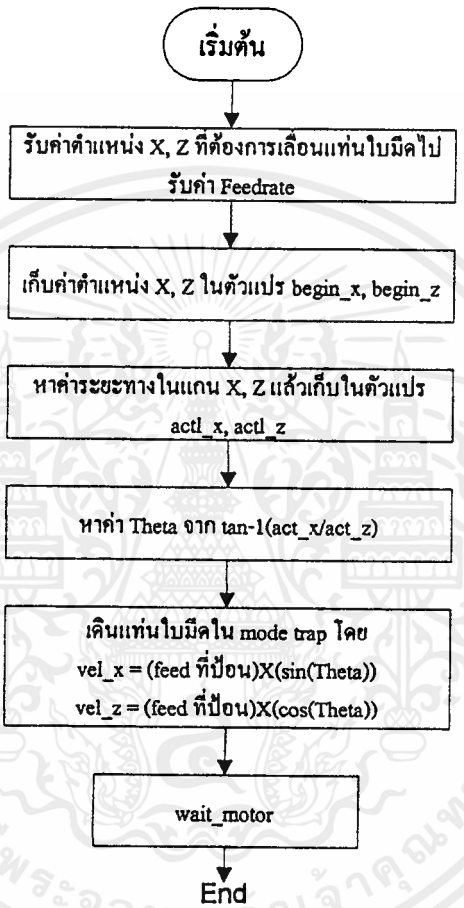
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Group 06

G20 (Data input in inch system)

G21 (Data input in metric system)

รายละเอียดการเขียนโปรแกรมเกี่ยวกับ G code



รูปที่ 4.10 Flowchart แสดงการทำงานของ G code G01

G01 (Moving the Cutting Tool Along a Straight Path At a Cutting Feedrate)

G01 เป็นการเลื่อนแท่นใบมีดในแนวเส้นตรง โดยแท่นใบมีดจะเลื่อนไปตามแกน x , z หรือเลื่อนพร้อมๆ กันทั้งสองแกน แล้วแต่ตำแหน่งที่ต้องการจะเลื่อนแท่นใบมีดไป สำหรับอัตราความเร็วในการเลื่อนแท่นใบมีดก็ขึ้นอยู่กับค่า feedrate ที่เราป้อนเข้าไป

หลักการ การเขียนโปรแกรมการเลื่อนแท่นใบมีดแบบ G01 ใช้หลักของ Linear interpolation โดยมีขั้นตอนดังนี้

- อ่านค่าตำแหน่งของแท่นใบมีดในขณะนั้น ในแต่ละแกนเข้ามาเก็บไว้ในตัวแปรที่เรากำหนดขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

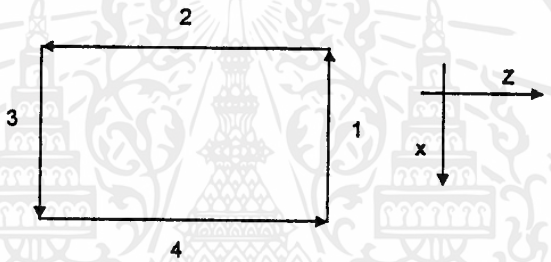
- อ่านค่าตำแหน่งที่ต้องการจะเลื่อนแท่นใบมีดไป เข้ามาเก็บไว้ในตัวแปรที่เรากำหนดขึ้น
- หาค่าระยะทางระหว่างจุดสองจุด ในแต่ละแนวแกน
- หาค่ามุมของเส้นตรงซึ่งวัดจากแกน Z ในทิศทางทวนเข็ม โดยหาจาก tangent inverse
- ใช้หลักของ Linear interpolation แตกเวคเตอร์ เข้ามายังแกน X, Z
- เขียนคำสั่งให้การ์ดทำการสั่งให้แท่นเลื่อน ให้เลื่อนไปยังตำแหน่งที่ต้องการ โดยมีที่ค่าที่ต้องบอกให้การ์ดทราบ คือ 1). ตำแหน่งที่ต้องการไป 2). ค่าอัตราเร็วในแกน X 3). ค่าอัตราเร็วในแกน Z 4). ค่าอัตราเร่ง

G00 (Positioning the Cutting Tool at a Rapid Traverse Rate)

G00 ใช้หลักการเขียนโปรแกรมเช่นเดียวกับ G01 แต่ไม่มีการรับค่า feedrate จากผู้ใช้ ค่า feedrate ที่ป้อนให้การ์ดจะถูกกำหนดให้จากโปรแกรมที่เขียนขึ้นมา โดยมีความเร็วพอสมควร

G92 (Continuous Thread Cutting and Tread Cutting Cycle)

G92 มีลักษณะการทำงานดังรูป



รูปที่ 4.11 แนวการเคลื่อนที่ของ G92

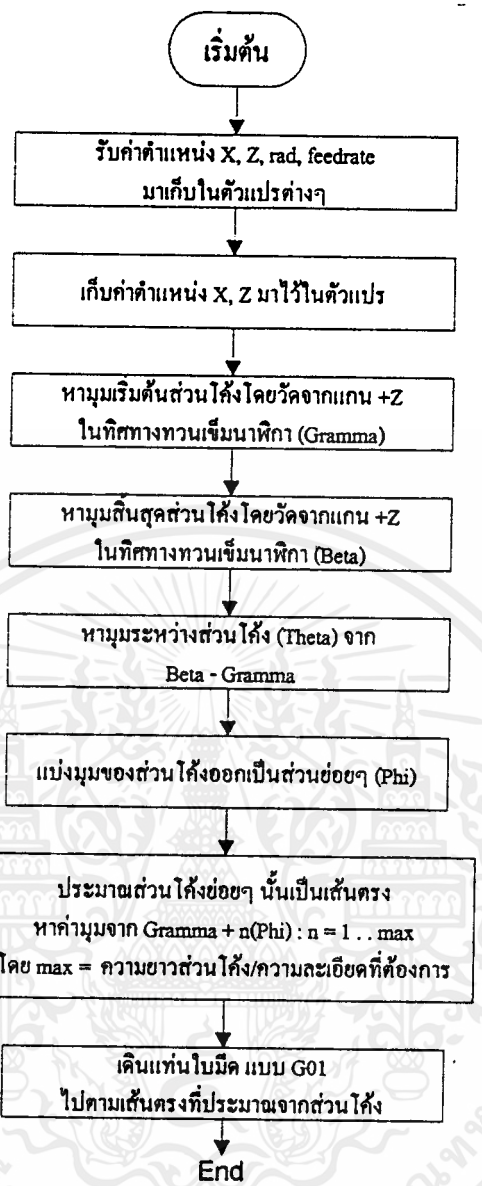
หลักการ

1.) แท่นใบมีดเคลื่อนที่แบบ G01 โดยให้ค่าในแกน Z คงที่ เคลื่อนที่ในแนวแกน X ไปยังค่า x ที่ป้อนเข้ามา
2.) แท่นใบมีดเคลื่อนที่แบบ G01 โดยให้ค่าในแกน X คงที่ เคลื่อนที่ในแนวแกน Z ไปยังค่า z ที่ป้อนเข้ามา
3.) แท่นใบมีดเคลื่อนที่แบบ G01 โดยให้ค่าในแกน Z คงที่ เคลื่อนที่ในแนวแกน X ไปยังค่า x เดิมก่อนที่จะเริ่มการทำงานของ G92
4.) แท่นใบมีดเคลื่อนที่แบบ G00 โดยให้ค่าในแกน X คงที่ เคลื่อนที่ในแนวแกน Z ไปยังค่า z เดิมก่อนที่จะเริ่มการทำงานของ G92

G02, G03 (Moving the Cutting Tool Along Arcs at a Cutting Feedrate)

G02, G03 มีการทำงานคล้ายๆ กันเพียงแต่กลับทิศกัน การเขียนโปรแกรมใช้หลักการของ Contour approximation using line segments โดยการประมาณส่วนโค้งให้เป็นเส้นตรงสั้นๆ ต่อกัน หลังจากนั้นก็เดินแท่นใบมีดไปตามเส้นตรงสั้นๆนั้น จนครบส่วนโค้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



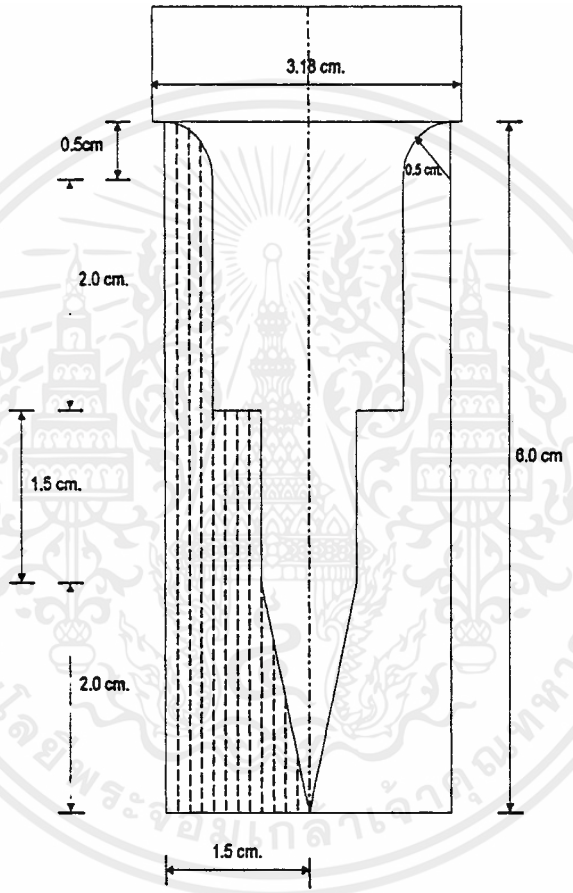
รูปที่ 4.12 Flowchart แสดงการทำงานของ G code G03

สำหรับ Source Code ของโปรแกรมดูรายละเอียดได้ในภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการทดลองและวิจารณ์

จากการทำปริญญานิพนธ์ในเรื่องนี้ เราได้ทำการทดลองกลึงชิ้นงานตามแบบที่เขียนขึ้นมา โดยก่อนการป้อนค่าให้แก่ตัวโปรแกรมต้องมีการแปลงจากแบบที่เขียนขึ้นให้เป็น G code ตามมาตราฐาน หลังจากนั้นก็ทำการกลึงโดยกลึงในโหมดอัตโนมัติ ดังแสดงในรูปข้างล่างนี้



รูปที่ 5.1 Drawing ของชิ้นงาน

G code List

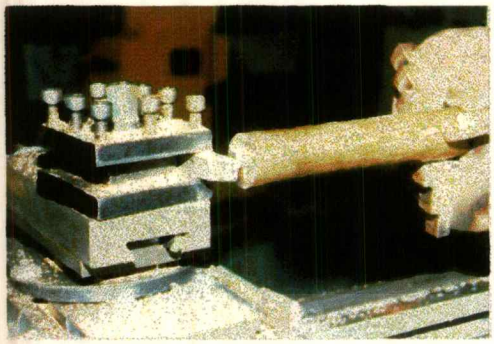
```
N1 G01 X1590 Z0       F40
N2 G92 X1550 Z-6000 F40
N3 G92 X1500 Z-6000 F40
N4 G92 X1450 Z-5950 F40
N5 G92 X1400 Z-5900 F40
N6 G92 X1350 Z-5850 F40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

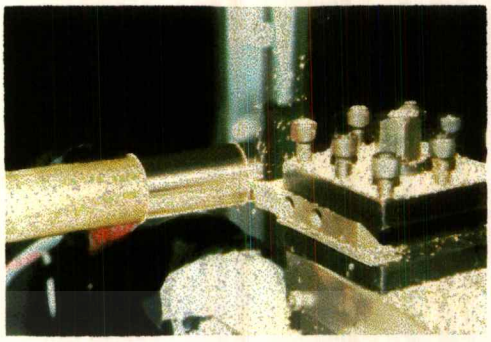
```
N7 G92 X1300 Z-5800 F40
N8 G92 X1250 Z-5750 F40
N9 G92 X1200 Z-5700 F40
N10 G92 X1150 Z-5650 F40
N11 G92 X1100 Z-5600 F40
N12 G92 X1150 Z-5550 F40
N13 G92 X1000 Z-5500 F40
N14 G92 X950  Z-3500 F40
N15 G92 X900  Z-3500 F40
N16 G92 X850  Z-3500 F40
N17 G92 X800  Z-3500 F40
N18 G92 X750  Z-3500 F40
N19 G92 X700  Z-3500 F40
N20 G92 X650  Z-3500 F40
N21 G92 X600  Z-3500 F40
N22 G92 X550  Z-3500 F40
N23 G92 X500  Z-3500 F40
N24 G92 X450  Z-1800 F40
N25 G92 X400  Z-1600 F40
N26 G92 X350  Z-1400 F40
N27 G92 X300  Z-1200 F40
N28 G92 X250  Z-1000 F40
N29 G92 X200  Z-800 F40
N30 G92 X150  Z-600 F40
N31 G92 X100  Z-400 F40
N32 G92 X50   Z-200 F40
N33 G01 X0    Z0 F40
N34 G01 X500  Z-2000 F40
N35 G01 X500  Z-3500 F40
N36 G01 X1000  Z-3500 F40
N37 G01 X1000  Z-5500 F40
N38 G03 X1500  Z-6000 F40
N39 G22 X0    Z0 F0
```

ผลจากการกลึงอัตโนมัติได้ชิ้นงานออกมาตามแบบที่กำหนด โดยเกิดค่าความผิดพลาดอยู่ในขั้นที่ยอมรับได้ คือ ประมาณ 0.01 - 0.03 mm.

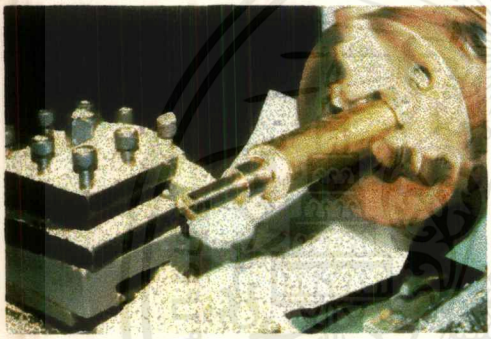
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



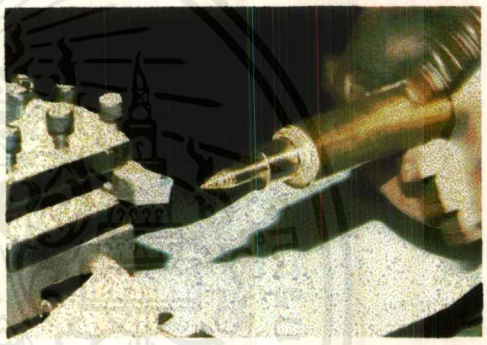
รูปที่ 5.2



รูปที่ 5.3



รูปที่ 5.4



รูปที่ 5.5

รูปที่ 5.6

รูปที่ 5.7

รูปที่ 5.2-5.7 แสดงขั้นตอนการกลึงชิ้นงานตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและข้อเสนอแนะ

ปริญญานิพนธ์ฉบับนี้เป็นการสร้างต้นแบบคอมพิวเตอร์ควบคุมเครื่องกลึง เพื่อสามารถนำไปประยุกต์ใช้กับเครื่องกลึงมือธรรมดาทั่วไปที่มีอยู่แล้วในโรงงานให้เป็นเครื่องกลึงอัตโนมัติ โดยติดตั้งส่วนประกอบต่าง ๆ เพิ่มเข้าไป เช่น DC Servo Motor, Driver, Encoder. Card Motion Control, PC Computer,Ball Screw และเขียนโปรแกรมควบคุม เป็นต้น ในส่วนของการสร้างและพัฒนาโปรแกรมจะอ้างอิงกับการใช้งานที่สะดวกและคำสั่งมาตรฐานสากล

เครื่องกลึงมือเหล่านี้เมื่อพัฒนาแล้วจะกลายเป็นเครื่องกลึง CNC ซึ่งมีราคาต่ำกว่าที่สั่งเข้ามาจากต่างประเทศ และสามารถดูแลรักษาได้ง่าย ในอนาคตเราจะปรับปรุงโปรแกรม ให้สามารถใช้งานง่ายและสะดวกมากขึ้น ซึ่งก่อให้เกิดประโยชน์ต่อภาคอุตสาหกรรม

ข้อเสนอแนะ

การดำเนินงานในขั้นนี้กระทำเพื่อให้ได้เครื่องกลึงที่มีประสิทธิภาพ สามารถใช้ในงานอุตสาหกรรมโดยทั่วไปได้อย่างสะดวก

1. จากการทดลองกลึงชิ้นงานได้ผลเป็นที่น่าพอใจ แต่ยังไม่มีความสะดวกในการใช้งานผ่านทาง Menu ซึ่งเขียนขึ้นจากภาษา C ดังนั้นในการพัฒนาขั้นต่อไปควรที่จะเขียน Menu ให้ใช้งานได้ง่ายและสะดวกขึ้น โดยได้ทำแนวทางในการเขียน Menu โดยการใช้ Visual C++ ในการเขียนดังที่แสดงไว้ในภาคผนวก จ.
2. จากการทดลองควรระวังในการ Set Zero เพราะถ้ามีการ Set Zero ผิดผลาด ทำให้ขนาดชิ้นงานผิดผลาดได้ เนื่องจากการอ้างอิง Coordinate เป็นแบบ Absolute
3. การวัดขนาดชิ้นงานในการหาข้อผิดผลาดควรวัดด้วยเครื่องมือที่มีความเที่ยงตรงสูง เช่น เครื่อง CMM (Coordinate measuring machine) เพื่อที่จะได้ทราบ ข้อผิดผลาดของขนาดชิ้นงานที่เกิดชิ้นอย่างถูกต้อง
4. ปัจจุบันอุปกรณ์ต่างๆ ยังไม่ได้จัดวางอย่างมีระเบียบ ในการพัฒนาต่อไปน่าที่จะจัดวางไว้ใต้แท่นกลึงและทำฝาปิดให้มิดชิดเพื่อป้องกันอุบัติเหตุที่อาจเกิดขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.ดร.โยธิน เปรมปราณีรัชต์ และอาจารย์กวิน สนธิเพิ่มพูน ที่ให้คำปรึกษาและสนับสนุนในด้านต่างๆ ตลอดจนช่วยจัดหาอุปกรณ์ที่ใช้ในการทำปริญญานิพนธ์นี้

ขอขอบพระคุณอาจารย์ทุกท่านทั้งภาควิชาระบบควบคุม ภาควิชาเครื่องกลและภาควิชาอิเล็กทรอนิคที่ให้คำแนะนำและ

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจและความเห็นใจ จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

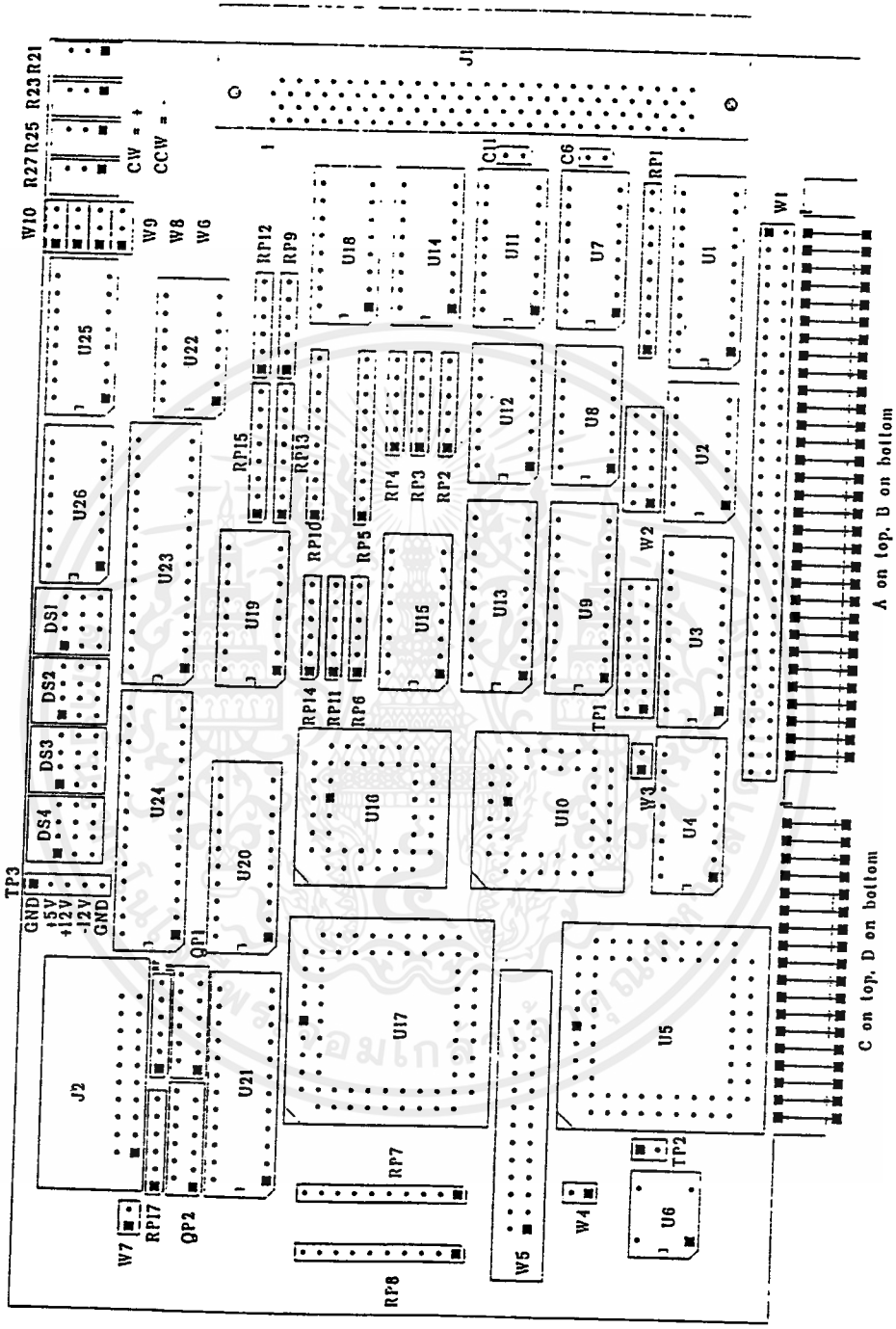
เอกสารอ้างอิง

1. กวิน สนธิเพิ่มพูน, รายงานการวิจัยสมบูรณ์ เรื่อง การวิจัยและพัฒนาการสร้างส่วนควบคุมเครื่องกัดแนวดิ่งซีเอ็นซี ระยะที่ 2, ศูนย์เทคโนโลยีอิเล็คโทรนิกส์และคอมพิวเตอร์แห่งชาติ, 2537
2. Larry Horath, Computer Numerical control Programmig of Machines, 1993
3. MORI SEIKI CO., LTD., SL-15,15MC Applicable Model Programming Manual I , 1995
4. S.C. Jonathan Lin, Computer Numerical Control,1993
5. David Gibbs, Thomas M. Crandell, An Introduction to CNC Machineing and Programming, 1994
6. Roger S. Pressman, John E. Williams, Numerical Control and Computer-Aided Manufacturing, John Willey & Sons, 1991
7. PMD , 5650 Board Technical Reference Version 0.2 , Technology 80 Inc. , 1994
8. GLENTEK , Operation And Service Manual For Model GA369 , GLENTEK. , 1991
9. ชาลี ตระการกูล, เทคโนโลยีซีเอ็นซี , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก.1 แสดงส่วนประกอบต่าง ๆ ของการ์ดควบคุม Model 5650

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.9 Comprehensive Addressing Jumpering Map (Table 5-7A & B)

Base Addr In Hex	Address Bit/ W2 Jumper Position					
	SA9 W2 1-2	SA8 W2 3-4	SA7 W2 5-6	SA6 W2 7-8	SA5 W2 9-10	SA4 W2 11-12
100	S	O	S	S	S	S
110	S	O	S	S	S	O
120	S	O	S	S	O	S
130	S	O	S	S	O	O
140	S	O	S	O	S	S
150	S	O	S	O	S	O
160	S	O	S	O	O	S
170	S	O	S	O	O	O
180	S	O	O	S	S	S
190	S	O	O	S	S	O
1A0	S	O	O	S	O	S
1B0	S	O	O	S	O	O
1C0	S	O	O	O	S	S
1D0	S	O	O	O	S	O
1E0	S	O	O	O	O	S
1F0	S	O	O	O	O	O

Table 5-7A. Complete Address Table with SA9 Low (W2 1-2 shorted)

Note:
Addresses below 100H are used by the motherboard and are therefore invalid.

S = Shorted
O = Open

รูป ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Base Addr In Hex	Address Bit/ W2 Jumper Position					
	SA9 W2 1-2	SA8 W2 3-4	SA7 W2 5-6	SA6 W2 7-8	SA5 W2 9-10	SA4 W2 11-12
200	O	S	S	S	S	S
210	O	S	S	S	S	O
220	O	S	S	S	O	S
230	O	S	S	S	O	O
240	O	S	S	O	S	S
250	O	S	S	O	S	O
260	O	S	S	O	O	S
270	O	S	S	O	O	O
280	O	S	O	S	S	S
290	O	S	O	S	S	O
2A0	O	S	O	S	O	S
2B0	O	S	O	S	O	O
2C0	O	S	O	O	S	S
2D0	O	S	O	O	S	O
2E0	O	S	O	O	O	S
2F0	O	S	O	O	O	O
300	O	O	S	S	S	S
310	O	O	S	S	S	O
320	O	O	S	S	O	S
330	O	O	S	S	O	O
340	O	O	S	O	S	S
350	O	O	S	O	S	O
360	O	O	S	O	O	S
370	O	O	S	O	O	O
380	O	O	O	S	S	S
390	O	O	O	S	S	O
3A0	O	O	O	S	O	S
3B0	O	O	O	S	O	O
3C0	O	O	O	O	S	S
3D0	O	O	O	O	S	O
3E0	O	O	O	O	O	S
3F0	O	O	O	O	O	O

Table 5-7B. Complete Address Table with SA9 High (W2 1-2 open)

รูป ก.2 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.1 J1 Connector Pinout

Table 5-1 details the signals available at J1

Pin Numbers By Axis					
#1	#2	#3	#4	Name	Description
1	26	51	76	Encoder +5Vdc	Encoder Power Output
2	27	52	77	Encoder Gnd	
3	28	53	78	Phase A	Encoder Phase A and Compliment
4	29	54	79	Phase /A	
5	30	55	80	Phase B	Encoder Phase B and Compliment
6	31	56	81	Phase /B	
7	32	57	82	Phase I	Encoder Index and Compliment
8	33	58	83	Phase /I	
9	34	59	84	+5Vdc	Power Out
10	35	60	85	Gnd	
11	36	61	86	CW Limit	Directional Limit Inputs
12	37	62	87	CCW Limit	
13	38	63	88	Home	Home Position Capture Input
14	39	64	89	Open	Not Used - Floating
15	40	65	90	+5Vdc	Power Out
16	41	66	91	Gnd	
17	42	67	92	Motor Enable	Motor Enable Output (Open-Collector)
18	43	68	93	Open	Not Used - Floating
19	44	69	94	Reserved	Do Not Connect
20	45	70	95	PWM Mag	PWM Magnitude Motor Output
21	46	71	96	Analog Out / PWM Dir	Analog Motor Output or PWM Direction Output
22	47	72	97	Analog Ret	Analog Return
23	48	73	98	+12Vdc	Reference Voltages used for Analog Output
24	49	74	99	-12Vdc	
25	50	75	100	Gnd	

Note: If the Technology 80 100-conductor to two 50-conductor cable is used the Axis 3 and 4 pin numbers are duplicates of the Axis 1 and 2 pin numbers.

Table 5-1. J1 Connector Pinout

รูป ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal	Axis 1	Axis 2	Axis 3	Axis 4
Analog Out	J1-21	J1-46	J1-71	J1-96
Analog Ret	J1-22	J1-47	J1-72	J1-97

Table 3-1. J1 Analog Output Pins

รูป ก.4

Signal	Axis 1/2	Axis 3/4
Motor Enable 1/2	J1-17/J1-42	
Motor Enable 3/4		J1-67/J1-92

Table 3-6. J1 Motor Enable Pins

รูป ก.5

W3	Active State	Power-Up State
Open	High (+5Vdc)	Low (0Vdc)
Shorted	Low (0Vdc)	High (+5Vdc)

Table 2-4. Motor Enable Polarity (W3)
Note: Shaded sections are default.

รูป ก.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal	Axis 1	Axis 2	Axis 3	Axis 4
Enc Pwr	J1-1	J1-26	J1-51	J1-76
Enc Gnd	J1-2	J1-27	J1-52	J1-77
Phase A	J1-3	J1-28	J1-53	J1-78
Phase /A	J1-4	J1-29	J1-54	J1-79
Phase B	J1-5	J1-30	J1-55	J1-80
Phase /B	J1-6	J1-31	J1-56	J1-81
Index	J1-7	J1-32	J1-57	J1-82
/Index	J1-8	J1-33	J1-58	J1-83

Table 3-4. J1 Differential Encoder Input Pins

รูป ก.7

Signal	Axis 1	Axis 2	Axis 3	Axis 4
Negative Limit	J1-11	J1-36	J1-61	J1-86
Positive Limit	J1-12	J1-37	J1-62	J1-87
Home	J1-13	J1-38	J1-63	J1-88

Table 3-5. J1 Limit/Home Input Pins

รูป ก.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งในโปรแกรม host_io.c

ชุดคำสั่ง

แบ่งตามลักษณะการใช้งานได้ 8 กลุ่ม ดังนี้

1) **ควบคุมแกน (Axis Control)**

SET_1 เปลี่ยนแกนปัจจุบันไปยังแกน 1

คำอธิบาย SET_1 จะเปลี่ยนแกนปัจจุบันไปยังแกน 1 คำสั่งทั้งหมดที่ตามมาภายหลังจะมีผลต่อแกน 1 ทั้งหมด ดูคำสั่ง GET_STATUS ประกอบ

SET_2 เปลี่ยนแกนปัจจุบันไปยังแกน 2

คำอธิบาย SET_2 จะเปลี่ยนแกนปัจจุบันไปยังแกน 2 คำสั่งทั้งหมดที่ตามมาภายหลังจะมีผลต่อแกน 2 ทั้งหมด ดูคำสั่ง GET_STATUS ประกอบ

SET_3 เปลี่ยนแกนปัจจุบันไปยังแกน 3

คำอธิบาย SET_3 จะเปลี่ยนแกนปัจจุบันไปยังแกน 3 คำสั่งทั้งหมดที่ตามมาภายหลังจะมีผลต่อแกน 3 ทั้งหมด ดูคำสั่ง GET_STATUS ประกอบ

SET_4 เปลี่ยนแกนปัจจุบันไปยังแกน 4

คำอธิบาย SET_4 จะเปลี่ยนแกนปัจจุบันไปยังแกน 4 คำสั่งทั้งหมดที่ตามมาภายหลังจะมีผลต่อแกน 4 ทั้งหมด ดูคำสั่ง GET_STATUS ประกอบ

2) **ควบคุมรูปแบบการเคลื่อนที่**

SET_PRFL_TRAP เปลี่ยนรูปแบบการเคลื่อนที่เป็นแบบ trapezoidal point-to-point

คำอธิบาย SET_PRFL_TRAP จะเปลี่ยนรูปแบบการเคลื่อนที่เป็นแบบ trapezoidal point-to-point ในโหมดนี้จะต้องมีการระบุตำแหน่งที่จะเคลื่อนที่ไป (คำสั่ง SET_POS) , ความเร็วสูงสุด(คำสั่ง SET_VEL) และความเร่ง (คำสั่ง SET_ACC) ที่ใช้ โดยสามารถจะทำการเปลี่ยนค่าตำแหน่งและความเร็วได้ตลอดเวลาแม้ในขณะที่มีการเคลื่อนที่ แต่ความเร่งจะเปลี่ยนไม่ได้

แกนใดที่ถูกกำหนดให้มีโหมดการเคลื่อนที่เช่นนี้ ก็จะอยู่ในโหมดนี้ตลอดไปจนกว่าจะมีการกำหนดโหมดการเคลื่อนที่ใหม่ให้ สำหรับในการเปลี่ยนโหมดการเคลื่อนที่จากโหมดอื่นมาเป็นโหมดนี้นั้น แกนที่ควบคุมอยู่จะต้องหยุดนิ่งเสียก่อน

SET_PRFL_VEL เปลี่ยนรูปแบบการเคลื่อนที่เป็นแบบ velocity contouring

คำอธิบาย SET_PRFL_VEL จะเปลี่ยนรูปแบบการเคลื่อนที่เป็นแบบ velocity contouring ในโหมดนี้จะต้องมีการกำหนดความเร่ง (คำสั่ง SET_ACC) และความเร็วสูงสุด (คำสั่งSET_VEL) ที่ใช้ โดยสามารถทำการเปลี่ยนค่าความเร่งและความเร็วสูงสุดได้ตลอดเวลา แม้ในขณะกำลังเคลื่อนที่ แกนใดที่ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้มีโหมดการเคลื่อนที่เช่นนี้ก็จะอยู่ในโหมดนี้ตลอดไป จนกว่าจะมีการกำหนดโหมดการเคลื่อนที่ใหม่ให้ นอกเหนือจากนั้น ยังสามารถที่จะเปลี่ยนโหมดการเคลื่อนที่จากโหมดอื่นมาเป็นโหมดนี้ได้ทุกขณะ

SET_POS กำหนดค่าตำแหน่ง

คำอธิบาย SET_POS จะกำหนดตำแหน่งปลายทางที่จะเคลื่อนที่ไป ซึ่งจะอยู่ในรูปจำนวนของ counts ซึ่งมีค่าในช่วง -1,073,741,824 ถึง +1,073,741,823 ค่าตำแหน่งที่กำหนดให้จะยังไม่ถูกนำไปใช้จนกว่าจะมีการใช้คำสั่ง UPDATE

SET_VEL กำหนดความเร็ว

คำอธิบาย SET_VEL จะกำหนดขนาดความเร็วสูงสุดที่จะมีได้ ซึ่งจะอยู่ในรูปจำนวน counts/sample มีค่าอยู่ระหว่าง 0 ถึง +16,383 ค่าความเร็วที่กำหนดให้จะยังไม่ถูกนำไปใช้จนกว่าจะมีการสั่ง UPDATE

SET_ACC กำหนดความเร่ง

คำอธิบาย SET_ACC จะกำหนดขนาดความเร่งที่จะใช้ ซึ่งจะอยู่ในรูปจำนวน counts/sample/sample โดยถ้าอยู่ในรูปแบบการเคลื่อนที่แบบ trapezoidal point-to-point จะมีค่าอยู่ระหว่าง 0 ถึง +8,191 แต่ถ้าอยู่ในรูปแบบการเคลื่อนที่แบบ velocity contouring นั้นจะมีค่าอยู่ระหว่าง -8,191 ถึง +8,191 ค่าความเร่งที่กำหนดให้จะยังไม่ถูกนำไปใช้จนกว่าจะมีการสั่ง UPDATE คำสั่งนี้จะใช้ได้เมื่ออยู่ในโหมด trapezoidal point-to-point หรือ velocity contouring เท่านั้น

CLR_PRFL กำหนดค่าความเร็ว ความเร่ง และตำแหน่งให้เป็น 0

คำอธิบาย CLR_PRFL กำหนดค่าความเร็ว ความเร่ง และตำแหน่งให้เป็นศูนย์ ในแต่ละรูปแบบของการเคลื่อนที่ ฟังก์ชันนี้จะยังไม่ทำงานจนกระทั่งมีการสั่ง UPDATE คำสั่งนี้มีประโยชน์ในการหยุดมอเตอร์อย่างกระทันหัน

SYNCH_PRFL กำหนด target position ให้มีค่าเท่ากับ actual position

คำอธิบาย SYNCH_PRFL จะทำการกำหนด target position ให้มีค่าเท่ากับ actual position ขณะนั้น คำสั่งนี้ใช้ได้กับรูปแบบการเคลื่อนที่ทุกชนิด มักใช้บ่อยๆ หลังจากเกิดความผิดพลาดในการเคลื่อนที่ เพื่อที่จะทำให้ระยะเป้าหมายที่ตั้งไว้กับระยะที่เคลื่อนที่ไปจริงมีค่าเท่ากันอีกครั้ง ฟังก์ชันนี้จะไม่ทำงานจนกระทั่งมีคำสั่ง UPDATE

ZERO_POS กำหนดค่า actual position และ target position ให้เป็นศูนย์

คำอธิบาย ZERO_POS จะทำการกำหนดค่า actual position และ target position ในแกนนั้นๆ ให้เป็นศูนย์ คำสั่งนี้จะไม่ทำงานจนกระทั่งมีคำสั่ง UPDATE และสามารถใช้ร่วมกับคำสั่ง SYNCH_PRFL ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GET_POS อ่านค่าตำแหน่งจากการ์ดควบคุม

คำอธิบาย GET_POS จะทำการอ่านค่าตำแหน่งปลายทางที่ถูกกำหนดโดยคำสั่ง SET_POS ค่าที่ส่งกลับมาจะอยู่ในหน่วยของจำนวน counts ซึ่งจะเป็นเลขจำนวนเต็มไม่มีเครื่องหมายขนาด 32 bit

GET_VEL อ่านค่าความเร็วจากการ์ดควบคุม

คำอธิบาย GET_VEL จะทำการอ่านค่าความเร็วที่ถูกกำหนดโดยคำสั่ง SET_VEL ค่าที่ส่งกลับมาจะอยู่ในหน่วยของ counts/sample ซึ่งจะเป็นเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 32 bit

GET_ACC อ่านค่าความเร่งจากการ์ดควบคุม

คำอธิบาย GET_ACC จะทำการอ่านค่าความเร่งที่ถูกกำหนดโดยคำสั่ง SET_ACC ค่าที่ส่งกลับมาจะอยู่ในหน่วยของ counts/sample/sample ซึ่งจะเป็นเลขจำนวนเต็ม อาจจะมีเครื่องหมายหรือไม่มีเครื่องหมาย ขึ้นอยู่กับรูปแบบของการเคลื่อนที่ที่ใช้

GET_TRGT_POS อ่านค่า target position

คำอธิบาย GET_TRGT_POS จะทำการอ่านค่าตำแหน่งที่จะเคลื่อนที่ไปซึ่งคำนวณโดยการ์ดควบคุม ค่านี้แสดงถึง target position สำหรับแกนนั้นๆที่จะเคลื่อนที่ไปในช่วงเวลาหนึ่งๆ คำสั่งนี้สามารถทำงานในทุกๆ รูปแบบของการเคลื่อนที่ โดยจะทำการส่งค่ากลับมาอยู่ในหน่วยของ counts ซึ่งเป็นเลขจำนวนเต็มมีค่าอยู่ระหว่าง -1,073,741,824 ถึง +1,073,741,823 คำสั่งนี้มีประโยชน์ในการดูค่าตำแหน่งที่ได้จากการคำนวณของการ์ดควบคุม และประสิทธิภาพของระบบเซอร์โว

GET_ACTL_POS อ่านค่า actual position

คำอธิบาย GET_ACTL_POS จะทำการอ่านค่าตำแหน่งที่เคลื่อนที่ไปได้จริงของมอเตอร์ ค่าที่ส่งกลับมาจะอยู่ในหน่วยของ counts ซึ่งจะเป็นเลขจำนวนเต็มมีเครื่องหมายขนาด 32 bit

GET_TRGT_VEL อ่านค่า target velocity

คำอธิบาย GET_TRGT_VEL จะทำการอ่านค่าความเร็วที่คำนวณโดยการ์ดควบคุม ค่านี้จะแสดงถึง target velocity สำหรับแกนนั้นๆ ที่จะเคลื่อนที่ไปในช่วงเวลาหนึ่งๆ ค่าที่ส่งกลับมาจะอยู่ในหน่วยของ counts/sample ซึ่งเป็นเลขจำนวนเต็มมีเครื่องหมายขนาด 32 bit ในช่วง-16,384 ถึง +16,383 คำสั่งนี้มีประโยชน์ในการดูค่าความเร็วที่คำนวณโดยการ์ดควบคุม และประสิทธิภาพของระบบเซอร์โว

3) **ควบคุม Digital Filter**

SET_FLTR_PID เปลี่ยนโหมดของ digital filter เป็นแบบ PID

คำอธิบาย SET_FLTR_PID จะทำการเปลี่ยนโหมดของ digital filter เป็นแบบ PID ในโหมดนี้ผู้ใช้จะต้องกำหนดค่า Propotional gain (คำสั่ง SET_KP) , Derivative gain (คำสั่งSET_KD) , Integral gain (คำสั่ง SET_KI) และ Integration limit (คำสั่ง SET_I_LM) พารามิเตอร์ทั้งหมดเหล่านี้สามารถจะเปลี่ยนในขณะที่มอเตอร์ทำงานอยู่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SET_KP กำหนดค่า Propotional gain

คำอธิบาย SET_KP ใช้กำหนดค่า Propotional gain สำหรับ Digital Filter โดยจะสามารถมีค่าได้ตั้งแต่ 0 ถึง 32767 ค่าที่กำหนดจะยังไม่ถูกนำไปใช้งานจนกว่าจะสั่ง UPDATE

SET_KD กำหนดค่า Derivative gain

คำอธิบาย SET_KD ใช้กำหนดค่า Derivative gain สำหรับ Digital Filter โดยจะสามารถมีค่าได้ตั้งแต่ 0 ถึง 32767 ค่าที่กำหนดจะยังไม่ถูกนำไปใช้งานจนกว่าจะสั่ง UPDATE

SET_KI กำหนดค่า Integral gain

คำอธิบาย SET_KI ใช้กำหนดค่า Integral gain สำหรับ Digital Filter โดยจะสามารถมีค่าได้ตั้งแต่ 0 ถึง 32767 ค่าที่กำหนดจะยังไม่ถูกนำไปใช้งานจนกว่าจะสั่ง UPDATE

SET_I_LM กำหนดค่า Integration limit

คำอธิบาย SET_I_LM ใช้กำหนดค่า Integration limit สำหรับ Digital Filter โดยจะสามารถมีค่าได้ตั้งแต่ 0 ถึง 32767 ค่าที่กำหนดจะยังไม่ถูกนำไปใช้งานจนกว่าจะสั่ง UPDATE

SET_POS_ERR กำหนดค่า position error สูงสุด

คำอธิบาย SET_POS_ERR ใช้กำหนดค่า position error สูงสุดสำหรับ Digital Filter โดยมีค่าอยู่ระหว่าง 0 ถึง 32767 ในแต่ละรอบของ servo loop ขนาดของ position error ที่คำนวณโดย Digital Filter จะถูกนำมาเปรียบเทียบกับค่า position error สูงสุดที่กำหนดไว้ ถ้าค่า position error มีค่าเกินกว่าที่กำหนดไว้ มอเตอร์จะหยุดหมุน ค่า position error ที่ถูกกำหนดโดยคำสั่งนี้สามารถนำไปใช้ได้เลยโดยไม่ต้องสั่ง UPDATE

GET_KP อ่านค่า Propotional gain

คำอธิบาย GET_KP จะทำการอ่านค่า Propotional gain ที่ถูกกำหนดโดยคำสั่ง SET_KP ค่าที่ส่งกลับมาจะอยู่ในรูปเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 16 bit

GET_KD อ่านค่า Derivative gain

คำอธิบาย GET_KD จะทำการอ่านค่า Derivative gain ที่ถูกกำหนดโดยคำสั่ง SET_KD ค่าที่ส่งกลับมาจะอยู่ในรูปเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 16 bit

GET_KI อ่านค่า Integral gain

คำอธิบาย GET_KI จะทำการอ่านค่า Integral gain ที่ถูกกำหนดโดยคำสั่ง SET_KI ค่าที่ส่งกลับมาจะอยู่ในรูปเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 16 bit

GET_I_LM อ่านค่า Integration limit

คำอธิบาย GET_I_LM จะทำการอ่านค่า Integration limit ที่ถูกกำหนดโดยคำสั่ง SET_I_LM ค่าที่ส่งกลับมาจะอยู่ในรูปเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 16 bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GET_POS_ERR อ่านค่า Position error สูงสุด

คำอธิบาย GET_POS_ERR จะทำการอ่านค่า position error สูงสุด ที่ถูกกำหนดโดยคำสั่ง SET_POS_ERR ค่าที่ส่งกลับมาจะอยู่ในรูปเลขจำนวนเต็มไม่มีเครื่องหมาย ขนาด 16 bit

GET_ACTL_POS_ERR อ่านค่า Position error ที่เกิดขึ้น

คำอธิบาย GET_ACTL_POS_ERR จะทำการอ่านค่า position error ที่เกิดขึ้นในขณะนั้น โดยค่าที่ส่งกลับมาจะเป็นค่าผลต่างระหว่าง target position และ actual position (actual position ลบด้วย target position) ซึ่งเป็นตัวเลขมีเครื่องหมาย ขนาด 16 bit ในช่วงระหว่าง
-32768 ถึง +32767 คำสั่งนี้ใช้ประโยชน์ในการดูและวิเคราะห์การติดตามค่าความผิดพลาดของระบบ

SET_AUTO_STOP_ON อนุญาตให้มอเตอร์หยุดโดยอัตโนมัติ

คำอธิบาย SET_AUTO_STOP_ON จะอนุญาตให้มอเตอร์หยุดหมุนในทันทีที่มีการเคลื่อนที่ผิดพลาดขึ้น ในโหมดนี้ มอเตอร์จะหยุดหมุน (เหมือนในคำสั่ง MTR_OFF) เมื่อเกิดการเคลื่อนที่ผิดลาดขึ้น (ดูคำสั่ง SET_POS_ERR) สามารถทำให้กลับเป็นปกติ (มอเตอร์ทำงานได้ตามปกติ) ได้โดยใช้คำสั่ง MYR_ON

SET_AUTO_STOP_OFF ไม่อนุญาตให้มอเตอร์หยุดโดยอัตโนมัติ

คำอธิบาย SET_AUTO_STOP_OFF จะไม่อนุญาตให้มอเตอร์หยุดหมุนซึ่งมีสาเหตุมาจากการเคลื่อนที่ที่ผิดพลาด ในโหมดนี้ มอเตอร์จะไม่หยุดหมุนเมื่อเกิดความผิดพลาดขึ้น

4) **ควบคุมการเปลี่ยนค่าของพารามิเตอร์ต่างๆ**

UPDATE ทำการเปลี่ยนค่าพารามิเตอร์ทันที

คำอธิบาย UPDATE จะทำการเปลี่ยนค่าพารามิเตอร์ใหม่ โดยใช้ค่าที่กำหนดให้ในคำสั่งที่มาก่อนหน้าคำสั่งนี้

MULTI_UPDATE ทำการเปลี่ยนค่าพารามิเตอร์ของแกนต่างๆ ที่กำหนดพร้อมกันทันที

คำอธิบาย MULTI_UPDATE จะทำการเปลี่ยนค่าพารามิเตอร์ต่างๆ ที่ได้กำหนดให้กับแต่ละแกนพร้อมกัน

5) **ควบคุมสถานะ / โหมด**

CLR_STATUS เคลียร์บิตแสดงสถานะทุกตัวเฉพาะแกนที่กำหนด

คำอธิบาย CLR_STATUS จะทำการรีเซ็ตบิตแสดงสถานะของแกนที่กำหนดให้ (bit - 6 ของ status word) คำสั่งนี้มีประโยชน์สำหรับการเคลียร์บิตสถานะทุกตัวระหว่างที่เริ่มเปิดเครื่องและระหว่างที่ใช้งานอยู่ สำหรับรายละเอียดเกี่ยวกับ status word นั้น ดูได้จากคำสั่ง GET_STATUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GET_STATUS อ่านค่าของ Status word ของแกนที่กำหนด

คำอธิบาย GET_STATUS จะส่งค่าสถานะของแกนที่ต้องการทราบกลับมา โดยความหมายของแต่ละ bit ใน status word ปรากฏอยู่ในตาราง ก.1

Bit	Event
15-14	Unused (set to 0)
13-12	Current axis Bit 13 Bit 12 Axis 0 0 1 0 1 2 1 0 3 1 1 4
11-10	Reserved (may be 0 or 1)
9	Axis on/off status (1 indicates on)
8	Motor on/off status (1 indicates on)
7	Unusud (set to 0)
6	Negative limit switch (1 indicates negative limit triped)
5	Positive limit switch (1 indicates positive limit tripped)
4	Motion error (1 indicates motion error)
3	Position capture received (1 indicates posiion capture has occured)
2	Update breakpoint reached (1 indicates breakpoint reached)
1	Position wrap-around (1 indicates wrap)
0	Motion complete (1 indicates complete)

ตาราง ก.1 แสดงความหมายของแต่ละบิตในAxis Status

6) **ควบคุม Encoder**

SET_CAPT_INDEX กำหนดให้ค่าในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูงเปลี่ยนแปลงเมื่อได้รับสัญญาณ index

คำอธิบาย SET_CAPT_INDEX จะกำหนดให้ค่าในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูงเปลี่ยนแปลงไป เมื่อสัญญาณ index เข้ามา คำสั่งนี้จะใช้เมื่อโหมดของการรับค่าตำแหน่งเป็นแบบ incremental เท่านั้น

SET_CAPT_HOME กำหนดให้ค่าในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูงเปลี่ยนแปลงเมื่อได้รับสัญญาณ home

คำอธิบาย SET_CAPT_HOME จะกำหนดให้ค่าในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูงเปลี่ยนแปลงไป เมื่อสัญญาณ home เข้ามา คำสั่งนี้จะใช้เมื่อโหมดของการรับค่าตำแหน่งเป็นแบบ incremental เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GET_CAPT อ่านค่าที่เก็บไว้ในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูง

คำอธิบาย GET_CAPT จะส่งกลับค่าที่เก็บอยู่ในรีจิสเตอร์เก็บค่าตำแหน่งความเร็วสูง โดยจะอยู่ในรูปเลขจำนวนเต็มมีเครื่องหมายขนาด 32 bit ในหน่วยของ counts

7) **ควบคุมมอเตอร์**

SET-OUTPUT-DAC16 กำหนดให้สัญญาณที่จะส่งไปควบคุมมอเตอร์อยู่ในรูปAnalog

คำอธิบาย SET_OUTPUT_DAC16 จะกำหนดโหมดสัญญาณที่จะส่งไปควบคุมมอเตอร์อยู่ในโหมด 16-bit-DAC คำสั่งนี้จะมีผลต่อทุกๆแกน

MTR-ON ยอมให้มีสัญญาณออกไปควบคุมมอเตอร์ได้

คำอธิบาย MTR_ON จะอนุญาตให้ สัญญาณมอเตอร์สามารถออกไปได้เฉพาะ แกนที่กำหนดเท่านั้น

MTR-OFF ไม่ยอมให้มีสัญญาณออกไปควบคุมมอเตอร์ได้

คำอธิบาย MTR_OFF จะห้ามไม่ให้สัญญาณควบคุมมอเตอร์สามารถออกไปได้เฉพาะแกนที่กำหนด แต่จะส่งสัญญาณซึ่งมีค่า 0 ออกไปแทนคำสั่งนี้มีประโยชน์ในการยกเลิกการทำงานอย่างกะทันหัน

SET-MTR-CMD กำหนดค่าสัญญาณควบคุม ไปยังมอเตอร์โดยตรง

คำอธิบาย SET_MTR_CMD ใช้กำหนดค่าของสัญญาณควบคุมที่จะส่งไปมอเตอร์ โดยสามารถกำหนดได้โดยตรง ซึ่งมีค่าอยู่ระหว่าง-32767 ถึง +32676 แสดงถึงค่าสูงสุดของความเร็วที่หมุนตามเข็มนาฬิกาของมอเตอร์ ค่า 0 แสดงถึงมอเตอร์หยุดนิ่งค่า 32767 แสดงถึงค่าสูงสุดของความเร็วที่หมุนทวนเข็มนาฬิกาของมอเตอร์ ก่อนที่จะใช้คำสั่งนี้ต้องใช้คำสั่ง MTR_OFF นำก่อน

8) **ควบคุมส่วนปลีกย่อย**

AXIS-ON ยอมให้มีการส่งสัญญาณควบคุมไปยังแกนที่กำหนด(เปิดมอเตอร์)

คำอธิบาย AXIS_ON จะยอมให้มีการส่งสัญญาณควบคุมไปยังแกนที่กำหนด ซึ่งสามารถที่จะสั่งให้ทำการเปิดมอเตอร์ได้ทุกขณะ แต่ไม่ควรทำเพราะอาจทำให้การเคลื่อนที่เกิดการผิดพลาดขึ้น

AXIS-OFF ไม่ยอมให้มีการส่สัญญาณควบคุมไปยังแกนที่กำหนด

คำอธิบาย AXIS_OFF จะไม่ยอมให้มีการส่งสัญญาณควบคุมไปยังแกนที่กำหนด สัญญาณที่ส่งออกจะมีค่าเป็น 0 เราสามารถที่จะสั่งให้ทำการเปิดหรือปิดมอเตอร์ได้ทุกขณะ แต่ไม่ควรทำ เพราะอาจทำให้การเคลื่อนที่เกิดการผิดพลาดขึ้น

SET-LMT-SENSE กำหนดรูปแบบการรับรู้สัญญาณจาก Limit Switch ของการ์ดควบคุม

คำอธิบาย SET_LMT_SENSE เป็นคำสั่งที่ใช้กำหนดรูปแบบการรับรู้สัญญาณของการ์ดควบคุมที่ส่งมาจาก Limit Switch โดยรูปแบบการรับรู้สัญญาณจะมี 2 แบบคือ จะทำงานที่ขาขึ้นของสัญญาณ(Active high) หรือ จะให้ทำงานที่ขาลงของสัญญาณ(Active low)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit	Description
15-8	Not used (set to 0)
7	Axis 4 negative limit switch (0 = active high)
6	Axis 4 positive limit switch (0 = active high)
5	Axis 3 negative limit switch (0 = active high)
4	Axis 3 positive limit switch (0 = active high)
3	Axis 2 negative limit switch (0 = active high)
2	Axis 2 positive limit switch (0 = active high)
1	Axis 1 negative limit switch (0 = active high)
0	Axis 1 positive limit switch (0 = active high)

ตาราง ก.2 แสดงค่าและความหมายในแต่ละบิตของ Limit Switch Register

จาก ตารางจะเห็นได้ว่าถ้าเราต้องการเซ็ตค่าของ Limit Switch ทั้งทางด้านบวกและด้านลบของแกนใดๆให้ทำงาน(Trig) ที่ขาขึ้นของสัญญาณก็ทำได้โดยกำหนดให้บิตนั้นมีค่าเป็น 0 หรือในทางตรงกันข้ามก็กำหนดให้เป็น 1 แล้วแต่เราจะต้องการขึ้นอยู่กับระบบของเรา

GET-LMT-SWITCH อ่านค่าสถานะของ Register แสดงสถานะของ Limit Swtch

คำอธิบาย GET_LMT_SWITCH จะส่งค่าสถานะของ Register แสดงสภาวะของ Limit Switch แต่ละตัวในระบบ ซึ่งค่าที่ส่งกลับมาจะเป็นเลขจำนวนเต็มที่มีรูปแบบดังนี้

Bit	Description
15-8	Not used (set to 0)
7	Axis 4 negative limit switch (1 = tripped)
6	Axis 4 positive limit switch (1 = tripped)
5	Axis 3 negative limit switch (1 = tripped)
4	Axis 3 positive limit switch (1 = tripped)
3	Axis 2 negative limit switch (1 = tripped)
2	Axis 2 positive limit switch (1 = tripped)
1	Axis 1 negative limit switch (1 = tripped)
0	Axis 1 positive limit switch (1 = tripped)

ตาราง ก.3 แสดงค่าและความหมายใน Limit Switch Triggered Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LMTS-ON กำหนดให้มีการรับสัญญาณจาก Limit Switch

LMTS-OFF กำหนดให้ไม่มีการสัญญาณจาก Limit Switch

คำอธิบาย LMTS_OFF จะทำให้ค่าในRegister แสดงสภาวะของ Limit Switch ไม่มีการเปลี่ยนแปลงจึงส่งผลต่อคำสั่ง GET_LMT_SWITCH ด้วย

RESET ทำการ Reset ชิพของการ์ดควบคุม

คำอธิบาย Reset จะทำการReset ชิพของการ์ดควบคุม หลังจากที่มีการ Reset แล้วค่าต่างๆภายในตัวการ์ดควบคุมจะเป็นค่าดั้งเดิม (Default)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

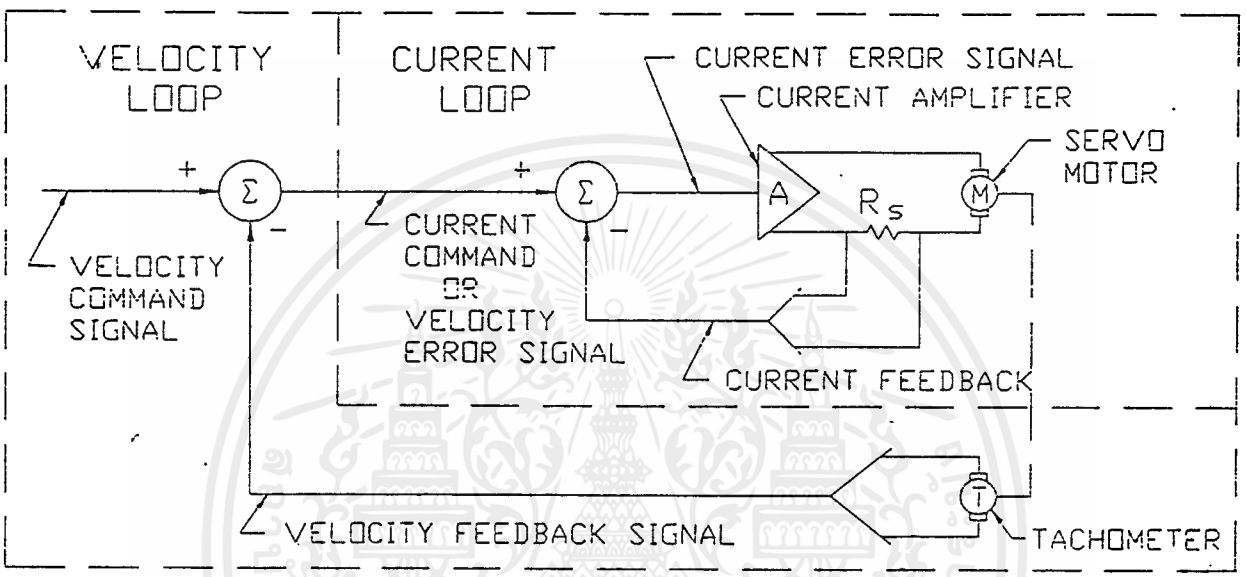
ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GLENTEK SERVO MOTOR CONTROLLERS MODEL GA369

INTRODUCTION TO THEORY OF OPERATION:

A Velocity Mode servo amplifier is essentially comprised of two control loops (see Fig.1)



The inside control loop is referred to as the Current Loop and the outside loop is referred to as the Velocity Loop. Before we begin our analysis of the Current Loop, let us review some basic concepts which will help you to better understand the amplifier's operation.

SPECIFICATIONS:

1. OUTPUT POWER

OUTPUT CURRENT (PEAK) :	±12 AMPS
OUTPUT CURRENT (RMS) :	± 6 AMPS
DC BUSS VOLTAGE :	+35 VDC TO +70 VDC

(Higher and lower Buss Voltages available, consult Glentek)

2. INPUT POWER :

INPUT POWER FOR DC BUSS: A fused single or three phase full wave rectifier and filter capacitor are provided on the base plate. The AC input to this circuit is supplied by a separately mounter power transformer unique to the application. The power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

transformer and motor output inductors are not standard parts of the amplifier package. Glentek has power transformers and inductors instock for your system requirements.

3. SIGNAL INPUTS:

VOLTAGE,MAXIMUM	
Aux. Input	±13 Volts
Sig. Input	±70 Volts
tach. Input	±90 Volts
Input Impedance, Minimum	10K Ohm
GAIN,MAXIMUM	
Sig. And Aux. Inputs	15,000 Amp/Volt
Tachometer	7,000 Amp/Vlot
Drift (Ref. To Input), Maximum	0.01 mV/ C
Frequency Response, Minimum	750 Hz
Dead band	None
Form Factor	1.01

4. OTHER INPUTS AND OUTPUTS

+ and - Limits (Type "A")	Activated by a 3ma. Contact closure to common
Total Inhibit (Type "A")	Activated by a 3ma. Contact closure to common
Gnd. To reset	Activated by a 3ma. Contact closure to common
Fault Monitor	4.7k lhm pull-up to +5VDC or +15VDC,will sink 50 ma.

5. MECHANICAL

Mounting	Any Position
Height	7 inches
Colling	50 C Ambient Max.
Baseplate	
GA369-2A-2	6.25 in. x 8.8 in.
Weight	
GA369-2A-2	1.14 lbs.

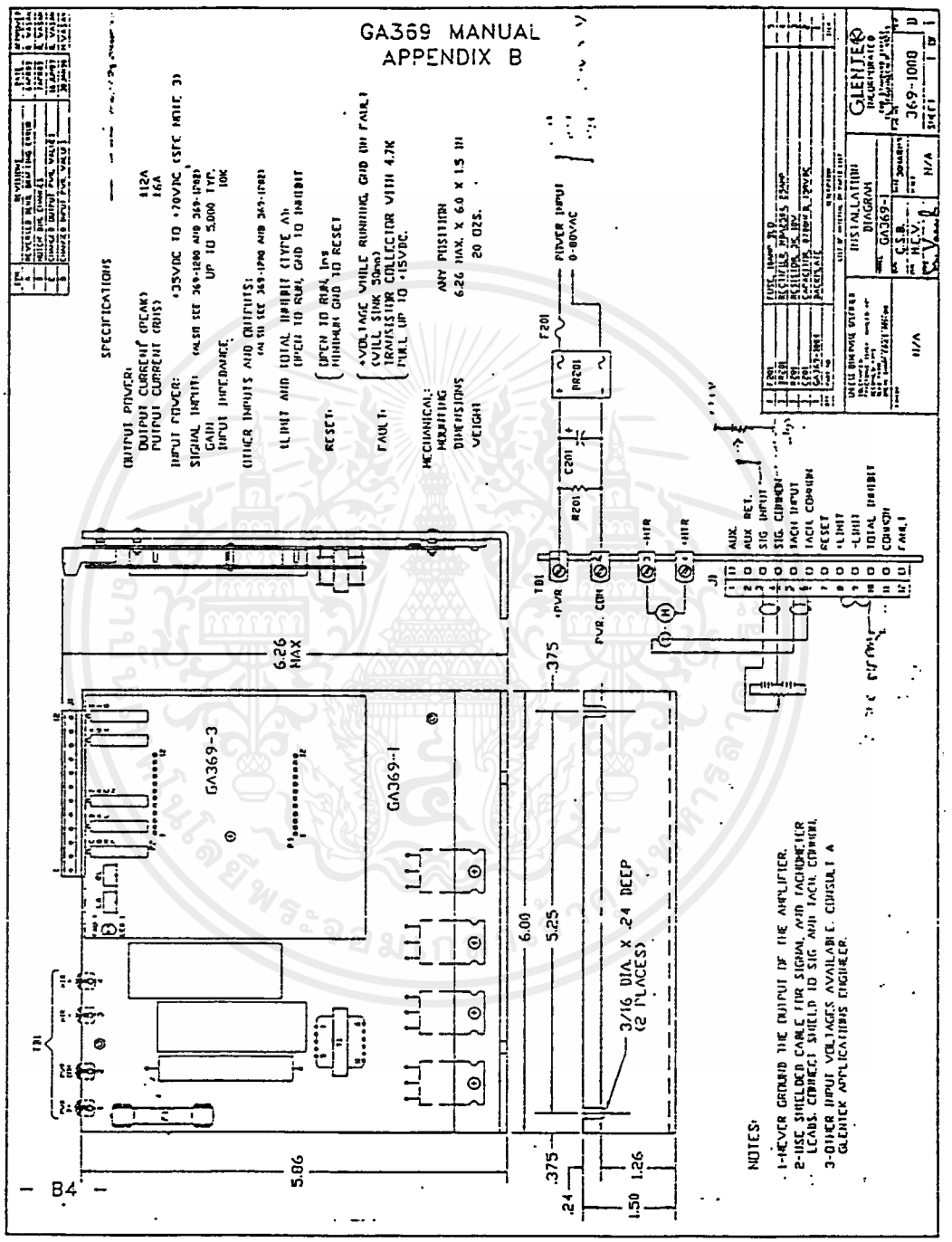
6. TYPICAL FACTORY SETTINGS

Sig. and Aux. Gain	5V(Sig.)/7V(Tech)
Tach. Gain	50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Loop Gain	CCW(OFF)
Comp.	CW(Min. Bandwidth)
LS/ECB	6A@0.5 sec.
HS/ECB	29A@ 10 micro sec.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 แสดงวงจรการเชื่อมต่อของ GLENTEK Model GA369

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC SERVO MOTOR GLENTEK MODEL GM3320

MOTOR DATA

MOTOR CONSTANTS (Units)	GM3320-22	GM3320-30
Max. Operating Speed (RPM)	3500	3200
Continuous Stall Torque at 40°C Ambient (LB-IN) / [Nm]	7.5 / [0.85]	7.5 / [0.85]
Continuous Stall Current (AMPS)	4.4	3.0
Peak Stall Torque (LB-IN) / [Nm]	51.8 / [5.85]	51.8 / [5.85]
Peak Current (AMPS)	30.5	20.7
Torque Sensitivity (Kt) (LB-IN/AMP) / [Nm/AMP]	1.7 / [0.19]	2.50 / [0.282]
Back EMF-Line to Line (Ke) (Volts/Krpm)	22	30
Terminal Resistance (OHMS)	2.1	6.9
Inductance (MILLIHENRYS)	4.9	20
Armature Inertia (LB-IN-SEC^2) / (KG M^2)	.000125 / [.000014]	.000125 / [.000014]
Static Friction (LB-IN) / [Nm]		
Motor Weight (LBS) / (Kg)	7 / [3.17]	7 / [3.17]

TACHOMETER DATA

TACHOMETER CONSTANTS (Units)			
Voltage Gradient (VOLTS/1000RPM)	3	7	10
Ripple (PERCENT) AVG. TO PEAK	1	1	1
Resistance (OHMS) ±12.5%	26	70	112
Inductance (MILLIHENRYS) ±30%	4.5	12	26
Minimum Load Resistance (OHMS)	3000	7000	10000

▲ Tolerance Plus or Minus 10%.

TECHNICAL DATA FOR BRAKE

Minimum Holding Torque: 60 IN. LB.
Coil Voltage: 24 VDC
Coil Current: 0.52 Amps
Coil Resistance: 46.5 Ohms
Release Voltage: 16 VDC Max.
Backlash: ≤ 50 Arc Min.
Engagement Time: 100 mSec.
Disengagement Time: 250 mSec.

DIMENSIONS ARE IN INCHES UNLESS ALL SHARP EDGES SURFACE ROUGHNESS	MODEL GM3320 SERIES	GLENTEK Incorporated	2035 STANDARD STREET EL SEGUNDO, CALIF
TOLERANCES ON DECIMAL .XX ±.005 .XXX ±.010 ANGLES ±2°	DRAWN H.O.D. 10/5/94	TITLE D.C. SERVO MTR-TACH-ENCODER RESOLVER-BRAKE DATA	
	CHECK S.S.V. 10/5/94		
	APPD'D M.E.V. 10/5/94		
MATERIAL	FINISH BLACK P...	SIZE C	DWG NO 3300-1043 REV -

รูปที่ ข.2 Datasheet ของ DC Servomotor GLENTEK Model GM3320

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UNIT HORSEPOWER (UHP), SURFACE CUTTING SPEED (CS), FEED CORRECTION FACTOR (C), FEED (f)

(a) Ferrous Metals and Alloys

Materials	BHN 150–175	176–200	201–250	251–300	301–350	351–400
	Unit Horsepower (UHP)					
AISI						
1010–1025	0.58	0.67				
1030–1055	0.58	0.67	0.80	0.96		
1060–1095			0.75	0.88	1.00	
1112–1120	0.50					
1314–1340	0.42	0.46	0.50			
1330–1350		0.67	0.75	0.92	1.10	
2015–2115	0.67					
2315–2335	0.54	0.58	0.62	0.75	0.92	
2340–2350		0.50	0.58	0.70	0.83	1.00
2512–2515	0.50	0.55	0.67	0.80	0.92	
3115–3130	0.50	0.58	0.70	0.83	1.00	
3160–3450		0.50	0.62	0.75	0.87	1.00
4130–4345		0.46	0.58	0.70	0.83	1.00
4615–4820	0.46	0.50	0.58	0.70	0.83	0.87
5120–5150	0.46	0.50	0.62	0.75	0.87	1.00
52100		0.58	0.67	0.83	1.00	
6115–6140	0.46	0.54	0.67	0.83	1.00	
6145–6195		0.70	0.83	1.00	1.20	1.30
Plain cast iron	0.30	0.33	0.42	0.50		
Alloy cast iron	0.30	0.42	0.52			
Malleable cast iron	0.42					
Cast steel	0.62	0.67	0.80			

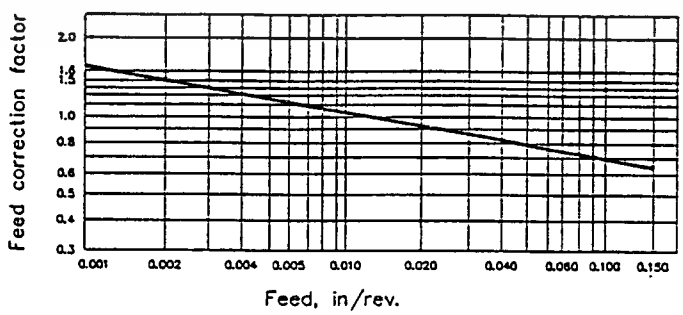
(b) High–Temperature Alloys

Material	BHN	UHP	Material	BHN	UHP	Material	BHN	UHP
A286	165	0.82	Hastelloy B	230	1.10	Ti–150A	340	0.65
A286	285	0.93	Inco 700	300	1.12	U–500	375	1.10
Chromoloy	200	0.78	Inco 702	230	1.10	4340	200	0.78
Chromoloy	310	1.18	M–252	230	1.10	4340	340	0.93
			M–252	310	1.20			

(c) Nonferrous Metals and Alloys

Material	UHP	Material	UHP	Material	UHP
Brass		Bronze		Aluminum	
Hard	0.83	Hard	0.83	Cast	0.25
Medium	0.50	Med.	0.50	Hard	0.33
Soft	0.33	Soft	0.33	Monel	1.00
Free–machining	0.25	Copper	0.90	Zinc(die cast)	0.25

ตารางที่ ค.1 Unit horsepower values for turning processes. (Source: Application Data Manual. Courtesy of Carboloy Inc., a Seco Tools Company.)



รูปที่ ค.1 Feed correction factor. (Source: Application Data Manual. Courtesy of Carboloy Inc., a Seco Tools Company.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cutting data for turning (meters per minute)
(Reproduced by kind permission of Anderson Strathclyde PLC)

MATERIAL	SPEED FT/MIN M/MIN			FEED INS/REV MM/REV			DEPTH INS MM			GRADE	
	ROUGH		FINISH	ROUGH		FINISH	ROUGH		FINISH	ROUGH	FINISH
ALUMINUM ALLOYS	800	1600	2500	.04	.02	.008	.25	.18	.01	CG	CF
	250	500	750	1.	5	.2	6.5	4 5	25		
ALUMINUM CASTINGS	800	1600	2500	.04	.02	.008	.25	.18	.01	CG	CF
	250	500	750	1.	5	.2	6.5	4 5	25		
ALUMINUM CASTINGS, HT. TREATED	300	600	1600	.04	.02	.008	.25	.18	.01	CG	CF
	90	180	500	1.	.5	2	6.5	4.5	.25		
BRASS	600	750	1000	.04	.02	.008	.25	.18	.01	CG	CF
	180	230	300	1	5	.2	6.5	4 5	.25		
BRONZE, PHOSPHOR	300	600	800	.04	.02	.008	.25	.18	.01	CG	CF
	90	180	250	1.	.5	.2	6.5	4 5	25		
CAST IRON, ALLOY	150	350	500	.04	.02	.008	.25	.18	.01	CR	CG
	45	105	150	1.	5	.2	6.5	4 5	25		
CAST IRON, CHILLED 4008	30	60	100	.04	.02	.008	.25	.18	.01	CG	CG
	9	18	30	1.	.5	.2	6.5	4 5	.25		
CAST IRON, CHILLED 6008	25	50	60	.04	.02	.008	.25	.18	.01	CG	CG
	8	15	18	1.	.5	.2	6.5	4.5	25		
CAST IRON, GREY	250	550	650	.04	.02	.008	.3	.2	.01	CR	CG
	75	165	190	1.	.5	.2	7.5	5.	.25		
CAST IRON, NODULAR, FERRITIC	150	300	500	.04	.02	.008	.2	.1	.01	CR	CG
	45	90	150	1.	.5	.2	5.	2.5	.25		
CAST IRON, NODULAR, PEARLITIC	150	300	450	.04	.02	.008	.2	.1	.01	CR	CG
	45	90	135	1.	.5	.2	5.	2.5	.25		
COPPER	600	1100	2000	.04	.02	.008	.25	.18	.01	CR	CG
	180	330	600	1.	.5	.2	6.5	4.5	.25		
FIBRE	300	500	700	.04	.02	.008	.25	.18	.01	CG	CF
	90	150	210	1.	.5	.2	6.5	4.5	.25		
HARD RUBBER, ASBESTOS	600	800	1000	.04	.02	.008	.3	.2	.01	CF	CF
	180	250	300	1.	.5	.2	7.5	5.	.25		
LEAD BRONZE, ALLOY	750	1000	1500	04	.02	.008	.25	.18	.01	CW	CG
	230	300	450	1.	.5	.2	6.5	4.5	.25		

ตารางที่ ค.2 Cutting data for turning (Reproduced by kind permission of Anderson Strathclyde PLC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MATERIAL	SPEED FT/MIN M MIN			FEED INS/REV MM/REV			DEPTH INS MM			GRADE	
	ROUGH		FINISH	ROUGH		FINISH	ROUGH		FINISH	ROUGH	FINISH
MALLEABLE IRON, LONG CHIP	150	500	650	.04	.02	.008	.3	.2	.01	M1	M1
	45	150	190	1	5	2	7 5	5	25		
MALLEABLE IRON, SHORT CHIP	250	400	600	.04	.02	.008	.3	.2	.01	M1/1	M1/1
	75	120	180	1.	.5	2	7 5	5.	25		
PORCELAIN	50	60	80	.04	.02	.008	.1	.01	.003	CF	CF
	15	18	25	1.	5	2	2.5	25	08		
RIGID PLASTICS, WOOD	600	800	1300	.04	.02	.008	.3	.2	.01	CW	CF
	180	250	390	1	.5	2	7 5	5	25		
STEEL, ALLOY, ANNEALED	300	500	650	.03	.01	.004	.5	.25	.015	SG	SF
	90	150	190	.75	.25	1	13.	6.5	.4		
STEEL, ALLOY, HARDENED 250B	250	400	600	.03	.01	.004	.3	.2	.01	SG	SF
	75	120	180	.75	.25	.1	7 5	5.	.25		
STEEL, ALLOY, HARDENED 300B	200	300	500	.03	.01	.004	.25	.18	.01	SG	SF
	60	90	150	.75	.25	1	6.5	4 5	25		
STEEL, ALLOY, HARDENED 400B	150	250	350	.03	.01	.004	.25	.18	.01	SG	SF
	45	75	100	.75	25	1	6 5	4 5	25		
STEEL, CARBON NORMALISED 125B	600	800	1100	.03	.01	.004	.5	.25	.015	SG	SF
	180	250	330	.75	25	1	13.	6.5	.4		
STEEL, CARBON NORMALISED 150B	400	650	1000	.03	.01	.004	.5	.25	.015	SG	SF
	120	190	300	.75	25	1	13.	6.5	.4		
STEEL, CARBON NORMALISED 250B	300	500	650	.03	.01	.004	.3	.2	.01	SG	SF
	90	150	190	.75	25	1	7 5	5.	.25		
STEEL, CAST 150B	200	300	500	.05	.01	.006	.5	.25	.015	SG	SF
	60	90	150	1.25	25	15	13.	6.5	.4		
STEEL, CAST 250B	150	250	350	.05	.01	.006	.3	.2	.01	SG	SF
	45	75	100	1.25	25	15	7.5	5.	.25		
STEEL, MANGANESE	60	100	200	.04	.02	.008	.3	.2	.01	CW	CG
	18	30	60	1.	5	2	7 5	5.	.25		
STEEL, STAINLESS, AUSTENTIC	300	400	500	.08	.015	.008	.25	.18	.01	CW	CW
	90	120	150	2.	.4	.2	6.5	4.5	.25		
STEEL, STAINLESS, MARTENSITIC	300	400	600	.08	.015	.008	.25	.18	.01	SG	SF
	90	120	180	2.	4	.2	6.5	4.5	.25		
STEEL, TOOL, HARDENED	30	60	100	.04	.02	.008	.2	.1	.01	SG	SF
	9	18	30	1.	.5	.2	5.	2.5	.25		
STONE, HARD GRANITE	25	35	50	.04	.02	.008	.2	.1	.01	CR	CR
	8	10	15	1.	.5	2	5.	2.5	.25		
STONE, SOFT MARBLE	150	200	250	.04	.02	.008	.2	.1	.01	CF	CF
	45	60	75	1.	.5	.2	5.	2.5	.25		

ตารางที่ ค.2 (ต่อ) Cutting data for turning (Reproduced by kind permission of Anderson Strathclyde PLC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEED (F) CONVERSION:

$$F\ (mm/min) = N\ (rev/min) \times f_r\ (mm/rev)$$

F (mm/min) = Feed in mm per minute

N (rev/min) = Speed of main spindle

f_r (mm/rev) = Feed in mm per revolution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ของโปรแกรมต้นแบบคอมพิวเตอร์ควบคุมเครื่องกลึง

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <conio.h>
#include <dos.h>
#include "mc1401.h"
#include "hrdwr.h"
#include "host_io.h"
#define KBD_PORT 0X60
#define NEWLINE 0x0a
#define SPACE 0x20
#define acc 1000000
#define pi 3.1415926
#define multi 1

int total,c;
long a,x,z,i,vel,speed,keystroke,finish,targetx,targetz;
long stat_x,stat_z,g48_x,g48_z;
volatile long choice,count;
float actl_x,actl_z,rad;
char fi_name[80],cmdst[1000][20];

struct
{
        int n;
        int g;
        long x;
        long z;
        float rad;
        long f;
}cmd[50] = {0,22,0,0,0,0};

void change_axis(int axis);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void trap_x();
void trap_z();
void show_x();
void show_z();
void show_pos();
void wait_mtr();
void set_x();
void set_z();
void set_spin();
void vel_x();
void vel_z();
void g01();
void g02();
void g03();
long convt();
void load_cmd();
void save_cmd();
void enter_cmd();
void start_spin();
void start_manual();
void start_auto();
void main()
{
        clrscr();
        reset(a);
        update(a);
        set_x();
        set_z();
        set_spin();
        /*CNC CONTROL PANEL*/
        do
 {
        clrscr();
        gotoxy(1,1);
        printf("\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("************ KINGMONGKOUT INSTITUTE OF TECHNOLOGY LADKRABANG
**************\n");
        printf("************************        CNC LATHE MACHINE CONTROL
**********************\n");
        printf("\n");
        printf("\n");
        printf("\n");
        printf(" SELECT MODE :          1. MANUAL\n");
        printf("                2. ENTER COMMAND\n");
        printf("                3. LOAD\n");
        printf("                4. SAVE\n");
        printf("                5. SET ZERO\n");
        printf("                        6. ON/OFF SPINDLE\n");
        printf("                7. GO\n");
        printf("                8. EXIT\n");
        printf(" >> ");
        choice = getch();
        switch (choice)
{
                case '1' :
        clrscr();
        start_manual();
        break;

                case '2' :
        enter_cmd();
        break;

                case '3' :
        clrscr();
        printf("Enter File Name : ");
        scanf("%s",&fi_name);
        load_cmd(fi_name);
        break;

                case '4' :
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            clrscr();
            printf("Enter File Name : ");
            scanf("%s",&fi_name);
            save_cmd(fi_name);
            break;

                    case '5' :
            clrscr();
            set_x();
            set_z();
            printf("NOW ACTUAL POSITION IS ZERO....\n");
            show_x();
            show_z();
            getch();
            break;

                    case '6' :
            clrscr();
            start_spin();
            break;

                    case '7' :
            start_auto();
            break;

            } /*SWITCH CHOICE*/

 }while (choice != '8');

}/*END MAIN*/

void   change_axis(int axis)
{  switch(axis)
   {
            case 0:
           set_1(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
                 case 1:
                set_2(0);
                break;
                 case 2:
                set_3(0);
                break;
                 case 3:
                set_4(0);
                break;
    default:
                break;
    }
}

void trap_x(void)
{
            set_1(a);
            mtr_on(a);
            set_prfl_trap(a);
            set_lmt_sense(0L);
            lmts_on(a);
            set_auto_stop_on(a);
            update(a);
            finish = get_lmt_swtch(a);
            if( finish == 0x4 ) clr_status(a);
}

void trap_z(void)
{
            set_2(a);
            mtr_on(a);
            set_prfl_trap(a);
            set_lmt_sense(0L);
            lmts_on(a);
            set_auto_stop_on(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        update(a);
        finish = get_lmt_swtch(a);
        if( finish == 0x4 ) clr_status(a);
}

void show_x()
{
 long targetx;
 set_1(a);
 targetx = get_trgt_pos(a);
 gotoxy(15,20);
 cprintf("X : %.8ld",(long)(targetx/8.15));
}

void show_z()
{
 long targetz;
 set_2(a);
 targetz = get_trgt_pos(a);
 gotoxy(15,22);
 cprintf("Z : %.8ld",(long)(targetz/8.15));
}

void show_pos()
{
 long targetx,targetz;
 set_1(a);
 targetx = get_trgt_pos(a);
 set_2(a);
 targetz = get_trgt_pos(a);
 gotoxy(15,20);
 cprintf("X : %.8ld",(long)(targetx/8.15));
 gotoxy(15,22);
 cprintf("Z : %.8ld",(long)(targetz/8.15));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void wait_mtr(void) /*WAIT MOTOR*/
{
  do
  {
  set_1(a);
  stat_x = (get_status(0))&0x0001;
  set_2(a);
  stat_z = (get_status(0))&0x001;
  show_pos();
  }while((stat_x!=1)|(stat_z!=1));
}

void set_x(void)    /*SET AXIS#1*/
{
        set_1(a);
        axis_on(a);
        mtr_on(a);
        set_output_dac16(a);
        set_prfl_trap(a);
        set_fltr_pid(a);
        set_kp(18L);
        set_ki(15L);
        set_kd(4L);
        zero_pos(a);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
        update(a);
}

void set_z(void)    /*SET AXIS#2*/
{
        set_2(a);
        axis_on(a);
        mtr_on(a);
        set_output_dac16(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        set_prfl_trap(a);
        set_fltr_pid(a);
        set_kp(18L);
        set_ki(15L);
        set_kd(4L);
        zero_pos(a);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
        update(a);
}
void set_spin(void)/*SET AXIS#3*/
{
        set_3(a);
        axis_on(a);
        mtr_on(a);
        set_output_dac16(a);
        set_prfl_trap(a);
        set_fltr_pid(a);
        set_kp(18L);
        set_ki(15L);
        set_kd(4L);
        zero_pos(a);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
        update(a);
}
void vel_x(void)
{
        set_1(a);
        mtr_on(a);
        set_prfl_vel(a);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        set_auto_stop_on(a);
        set_pos_err(1000);
        update(a);
        finish = get_lmt_swtch(a);
        if( finish == 0x2 ) clr_status(a);
}
void vel_z(void)
{
        set_2(a);
        mtr_on(a);
        set_prfl_vel(a);
        synch_prfl(a);
        set_lmt_sense(0L);
        lmts_on(a);
        set_auto_stop_on(a);
        set_pos_err(1000);
        update(a);
        finish = get_lmt_swtch(a);
        if( finish == 0x4 ) clr_status(a);
}
void g01(long x,long z,long feed)
{
        float rat_x,rat_z,theta;
        long begin_x,begin_z;
        set_1(a);
        clr_status(a);
        begin_x = (long)(get_trgt_pos(a)/8.15);
        set_2(a);
        clr_status(a);
        begin_z = (long)(get_trgt_pos(a)/8.15);
        actl_x = fabs(x-begin_x);
        actl_z = fabs(z-begin_z);
        if ((actl_x != 0)&(actl_z != 0))
        {
          theta = atan(actl_x/actl_z);
          rat_x = sin(theta);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            rat_z = cos(theta);
          }
          else
          { rat_x=1; rat_z=1;}

          trap_x();
          set_pos((long)(x*8.15));
          set_vel((long)(feed*354*rat_x));
          set_acc(5000);
          update(a);

          trap_z();
          set_pos((long)(z*8.15));
          set_vel((long)(feed*354*rat_z));
          set_acc(5000);
          update(a);

          wait_mtr();
}
void g02(long xxx,long zzz,long rad,long feed)
{
          float theta,phi,theta_r,gram,beta,arc_l,angle,sub,sub1;
          float xx,zz;
          long  num,x,z,n,begin_x,begin_z,begin_xx,begin_zz,new_x,new_z;
          float ab_xx,ab_zz;

          set_1(a);
          clr_status(a);
          begin_xx =(long)(get_trgt_pos(a)/8.15);

          set_2(a);
          clr_status(a);
          begin_zz =(long)(get_trgt_pos(a)/8.15);

          begin_x = fabs(begin_xx);
          begin_z = fabs(begin_zz);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ( begin_xx <= 0 && begin_zz >= 0)
{
        gram = atan(begin_x/begin_z);
}
else
{
        if (begin_xx <= 0 && begin_zz <= 0)
        gram = pi - atan(begin_x/begin_z);
        else
        {
                if (begin_xx >= 0 && begin_zz <= 0)
                gram = pi + atan(begin_x/begin_z);
                else
                {
                        if (begin_xx >= 0 && begin_zz >= 0)
                        gram = 2*pi - atan(begin_x/begin_z);
                        else
                        printf("error a\n");
                }
        }
}
xx = xxx/8.15;
zz = zzz/8.15;
ab_xx = fabs(xx);
ab_zz = fabs(zz);

if (xx <= 0 && zz >= 0)
{
        beta = atan(ab_xx/ab_zz);
}
else
{
        if (xx <= 0 && zz <= 0)
        beta = pi - atan(xx/zz);
        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                    if (xx >= 0 && zz <= 0)
                    beta = pi + atan(xx/zz);
                    else
                    {
                            if (xx >= 0 && zz >= 0)
                            beta = 2*pi - atan(xx/zz);
                            else
                            printf("error b\n");
                    }
            }
    }
    if ( beta < gram )
    {
    theta = gram - beta;
    arc_l = theta*(rad/8.15);
    num = (arc_l/multi)+1;
    phi = theta/num;
    for ( n = 1; n < num; n++)
    {
    theta_r = gram + n*phi;
    if (theta_r <= pi/2 && theta_r >= 0)
    {
            angle = theta_r;
            new_x = -rad*sin(angle);
            new_z = rad*cos(angle);
    }
    else
    {
            if ((theta_r >= pi/2) && (theta_r <= pi))
            {
                    angle = pi - theta_r;
                    new_x = -rad*sin(angle);
                    new_z = -rad*cos(angle);
            }
            else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        if ((theta_r >= pi) && (theta_r <= 3*pi/2))
                {
                        angle = theta_r - pi;
                        new_x = rad*sin(angle);
                        new_z = -rad*cos(angle);
                }
                        else
                        {
                                if (theta_r >= 3*pi/2 && theta_r <= 2*pi)
                        {
                                angle = 2*pi - theta_r;
                                new_x = rad*sin(angle);
                                new_z = rad*cos(angle);
                        }
                                else
                                printf("error c/n");
                        }
                }
        }

        g01(new_x,new_z,feed);

        }
        }
        else
        printf("error d");
}
void g03(long xxx,long zzz,long rad,long feed)
{
        float theta,phi,theta_r,gram,beta,arc_l,angle,sub,sub1;
        float xx,zz;
        long  num,x,z,n,begin_x,begin_z,begin_xx,begin_zz,new_x,new_z;
        float ab_xx,ab_zz;

        set_l(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clr_status(a);
begin_xx =(long)(get_trgt_pos(a)/8.15);

set_2(a);
clr_status(a);
begin_zz =(long)(get_trgt_pos(a)/8.15);

begin_x = fabs(begin_xx);
begin_z = fabs(begin_zz);

if ( begin_xx <= 0 && begin_zz >= 0)
{
        gram = atan(begin_x/begin_z);
}
else
{
        if (begin_xx <= 0 && begin_zz <= 0)
        gram = pi - atan(begin_x/begin_z);
        else
        {
                if (begin_xx >= 0 && begin_zz <= 0)
                gram = pi + atan(begin_x/begin_z);
                else
                {
                        if (begin_xx >= 0 && begin_zz >= 0)
                        gram = 2*pi - atan(begin_x/begin_z);
                        else
                        printf("error1\n");
                }
        }
}
xx = xxx/8.15;
zz = zzz/8.15;
ab_xx = fabs(xx);
ab_zz = fabs(zz);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (xx <= 0 && zz >= 0)
{
        beta = atan(ab_xx/ab_zz);
}
else
{
        if (xx <= 0 && zz <= 0)
        beta = pi - atan(xx/zz);
        else
        {
                if (xx >= 0 && zz <= 0)
                beta = pi + atan(xx/zz);
                else
                {
                        if (xx >= 0 && zz >= 0)
                        beta = 2*pi - atan(xx/zz);
                        else
                        printf("error2\n");
                }
        }
}
if ( beta > gram )
{
theta = beta - gram;
arc_l = theta*(rad/8.15);
num = (arc_l/multi)+1;
phi = theta/num;
for ( n = 1; n < num; n++)
{
theta_r = gram + n*phi;
if (theta_r <= pi/2 && theta_r >= 0)
{
        angle = theta_r;
        new_x = -rad*sin(angle);
        new_z = rad*cos(angle);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
{
        if ((theta_r >= pi/2) && (theta_r <= pi))
        {
                angle = pi - theta_r;
                new_x = -rad*sin(angle);
                new_z = -rad*cos(angle);
        }
        else
        {
                if ((theta_r >= pi) && (theta_r <= 3*pi/2))
        {
                angle = theta_r - pi;
                new_x = rad*sin(angle);
                new_z = -rad*cos(angle);
        }
                else
                {
                        if (theta_r >= 3*pi/2 && theta_r <= 2*pi)
                {
                        angle = 2*pi - theta_r;
                        new_x = rad*sin(angle);
                        new_z = rad*cos(angle);
                }
                        else
                        printf("error3/n");
                }
        }
}

g01(new_x,new_z,feed);

}
}
else
printf("error4");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

long convt(char cmdst[50])
{
        long longout;
        longout = atol(cmdst);
        return longout;
}
void load_cmd(char cmdname[80])    /*LOAD G-CODE FILE*/
{
        FILE *cmd_file;
        int c,t,n;
        char ch,st[2];

  if    ((cmd_file = fopen(cmdname,"rt+"))==NULL)
  {
        fprintf(stderr,"No file...\n");
        exit(0);
        }
        c = 1; n=0;

  while(!feof(cmd_file))
  {
        ch = fgetc(cmd_file);
        if      (ch>=SPACE)
        {
                cmdst[c][n]=ch;
                n++;
        }
        else
        if      (ch==NEWLINE)
        {
                n = 0;
                c++;
        }
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        total = c-1;
        t = 1; c=1;

    while(t<=total)
    {
                cmd[c].n = convt(cmdst[t]);
                printf("N%d ",cmd[c].n);
                cmd[c].g = convt(cmdst[t+1]);
                printf("G%d ",cmd[c].g);
                cmd[c].x = convt(cmdst[t+2]);
                printf("X%ld ",cmd[c].x);
                cmd[c].z = convt(cmdst[t+3]);
                printf("Z%ld ",cmd[c].z);
                if( (cmd[c].g == 3) || (cmd[c].g == 2))
                {
                cmd[c].rad = convt(cmdst[t+4]);
                printf("Rad%ld ",cmd[c].rad);
                cmd[c].f = convt(cmdst[t+5]);
                printf("F%ld \n",cmd[c].f);
                c++;
                t+=6;
                }else
                {
                cmd[c].f = convt(cmdst[t+4]);
                printf("F%ld \n",cmd[c].f);
                c++;
                t+=5;
                }
    }

        printf("SUCCESSFULL....\n");
        fclose(cmd_file);
        getch();
}
void save_cmd(char cmdname[80]) /* SAVE G-CODE FILE */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        FILE *cmd_file;
        int c;
        if((cmd_file = fopen(cmdname,"wt+"))==NULL)
        {
                fprintf(stderr,"No file...\n");
                exit(0);
        }
        c = 0;
        do
        {
                c++;
                sprintf(cmdst[c],"%d\n",cmd[c].n);
                fputs(cmdst[c],cmd_file);

                sprintf(cmdst[c+1],"%d\n",cmd[c].g);
                fputs(cmdst[c+1],cmd_file);

                sprintf(cmdst[c+2],"%ld\n",cmd[c].x);
                fputs(cmdst[c+2],cmd_file);

                sprintf(cmdst[c+3],"%ld\n",cmd[c].z);
                fputs(cmdst[c+3],cmd_file);

                sprintf(cmdst[c+4],"%ld\n",cmd[c].f);
                fputs(cmdst[c+4],cmd_file);

    }
          while (cmd[c].g != 22);
        fclose(cmd_file);
        printf("SUCCEFULL....\n");
}

void enter_cmd()
{
        c = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        do
        {
        clrscr();
        set_1(a);
        x = (long)(get_actl_pos(a)/8.15);
        set_2(a);
        z = (long)(get_actl_pos(a)/8.15);
        printf("NOW ACTUAL POSITION IS X = %ld",x);
        printf(" Z = %ld\n\n\n",z);
        printf(" WARNING!! before quit g-code must = G22\n\n");
        gotoxy(0,10);
        printf("PLEASE ENTER NUMBER OF G COMMAND : ");
        cmd[c].n = c;
        scanf("%d",&cmd[c].g);
        printf("ENTER X VALUE : ");
        scanf("%ld",&cmd[c].x);
        printf("ENTER Z VALUE : ");
        scanf("%ld",&cmd[c].z);
        if ((cmd[c].g == 2) || (cmd[c].g == 3))
        {
                printf("ENTER RADIUS : ");
                scanf("%f",&cmd[c].rad);
        }
        if (cmd[c].g != 0)
        {
                printf("ENTER FEED VALUE : ");
                scanf("%ld",&cmd[c].f);
        }
        printf("Type 'q' for quit");
        count = getch();
        c++;
        }
        while (count != 'q');
}

void start_spin()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        int spin,speed,vel_spin;
        gotoxy(10,10);
        printf(" Enter rpm : ");
        scanf("%d",&speed);
        do
        {
        clrscr();
        gotoxy(5,5);
        printf(" For start spindle press 1 : \n");
        printf("    For stop spindle press  2 : \n");
        printf("    For edit rpm press      3 : \n");
        printf("    For quit press          4 : \n");
        spin = getch();
        switch (spin)
                {
                case '1': /* for on spindle*/
                set_3(a);
                mtr_off(a);
                vel_spin = 16.667*speed;
                set_mtr_cmd(vel_spin);
                gotoxy(10,10);
                printf("NOW START SPINDLE\n\n");
                printf("        SPINDLE SPEED IS:  %d",speed);
                getch();
                break;

                case '2': /*for stop spindle*/
                set_3(a);
                mtr_off(a);
                set_mtr_cmd(0L);
                gotoxy(10,10);
                printf("NOW STOP SPINDLE");
                getch();
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case '3':
                    gotoxy(10,10);
                    printf ("Enter new rpm :");
                    scanf("%d",&speed);
                    }
                } while( spin != '4' );
}

void start_manual() /* CONTROL CNC BY MANUAL(ARROW KEY) */
{
        show_pos();
        gotoxy(1,1);
        printf("ENTER FEED RATE (NOT MORE THAN 3000) : ");
        scanf("%ld",&vel);
        speed = vel*354;

                do{
                keystroke = inportb(KBD_PORT);
                switch( keystroke ){
                        case 0x3b  : clrscr();
                                        show_pos();
                                        gotoxy(1,1);
                                        printf("ENTER FEED RATE (NOT MORE THAN
3000) : ");
                                        scanf("%ld",&vel);
                                        speed = vel*354;
                                        break;

                        case 0x50:

                                vel_x();
                do{ keystroke = inportb(KBD_PORT);
                set_vel(speed);
                set_acc(5000L);
                update(a);
                show_x();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                } while( keystroke != 0xaa );
                set_vel(0L);
                update(a);
                break;

                        case 0x48:

                          vel_x();
                          set_acc(-5000L);
                          do{ keystroke = inportb(KBD_PORT);
                          set_vel(speed);
                          show_x();
                          update(a);
                          }while( keystroke != 0xaa );
                          set_vel(0L);
                          update(a);
                          break;

case 0x4b:
        vel_z();
 do{ keystroke = inportb(KBD_PORT);
    set_acc(-5000L);
    set_vel(speed);
    update(a);
    show_z();
  }while( keystroke != 0xaa );
 set_vel(0L);
 update(a);
 break;

case 0x4d:
 vel_z();
 set_acc(5000L);
 do{ keystroke = inportb(KBD_PORT);
    set_vel(speed);
            update(a);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                show_z();
               }while( keystroke != 0xaa );
              set_vel(0L);
              update(a);
              break;
             default:break;}
    clr_prfl(a);
    update(a);
    synch_prfl(a);
    update(a);
    mtr_on(a);
    update(a);
          keystroke = inportb(KBD_PORT);
          count = getch();
   }while ( count != 0x1b );

 set_1(a);
 mtr_on(a);
 synch_prfl(a);
 update(a);
 set_2(a);
 mtr_on(a);
 synch_prfl(a);
 update(a);
}

void start_auto()
{
          int c;
          double rad;

          clrscr();
          printf("NOW OPERATING.....\n");
          c = 1;
          while(cmd[c].g != 22)
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

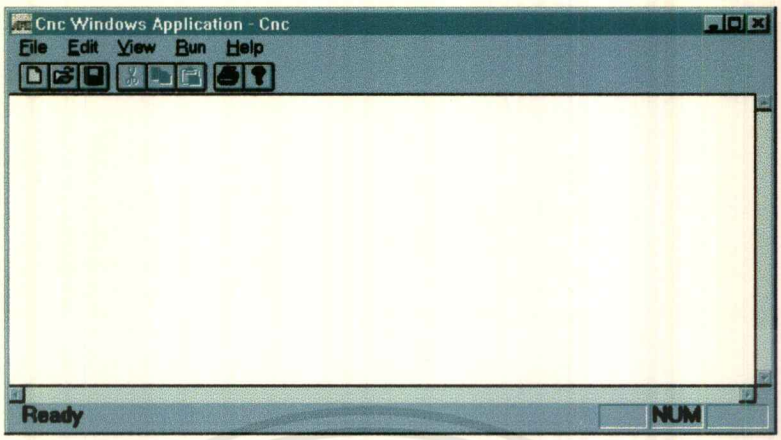
```
                gotoxy(1,9); printf("                                    ");
                gotoxy(1,9);
                if( (cmd[c].g == 3) || (cmd[c].g == 2))
                printf("COMMAND N%d G%d X%ld Z%ld Rad%.2f
Feed%ld\n",cmd[c].n,cmd[c].g,cmd[c].x,cmd[c].z,cmd[c].rad,cmd[c].f);
                else
                printf("COMMAND N%d G%d X%ld Z%ld
Feed%ld\n",cmd[c].n,cmd[c].g,cmd[c].x,cmd[c].z,cmd[c].f);
        switch(cmd[c].g)
        {
                case 0  :       g01(cmd[c].x,cmd[c].z,800);
                                break;
                case 1  :       g01(cmd[c].x,cmd[c].z,cmd[c].f);
                                break;
                case 92 :       set_1(a);
                                g48_x = (long)(get_trgt_pos(a)/8.15);
                                set_2(a);
                                g48_z = (long)(get_trgt_pos(a)/8.15);
                                g01(cmd[c].x,g48_z,cmd[c].f);
                                g01(cmd[c].x,cmd[c].z,cmd[c].f);
                                g01(g48_x,cmd[c].z,cmd[c].f);
                                g01(g48_x,g48_z,800);
                                break;
                case 2  :
                                g02(cmd[c].x,cmd[c].z,cmd[c].rad,cmd[c].f);
                                break;
                case 3  :
                                g03(cmd[c].x,cmd[c].z,cmd[c].rad,cmd[c].f);
                                break;

         default : printf("HEY! NO SUCH COMMAND"); break;
        }
  c++;
 }
 getch();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

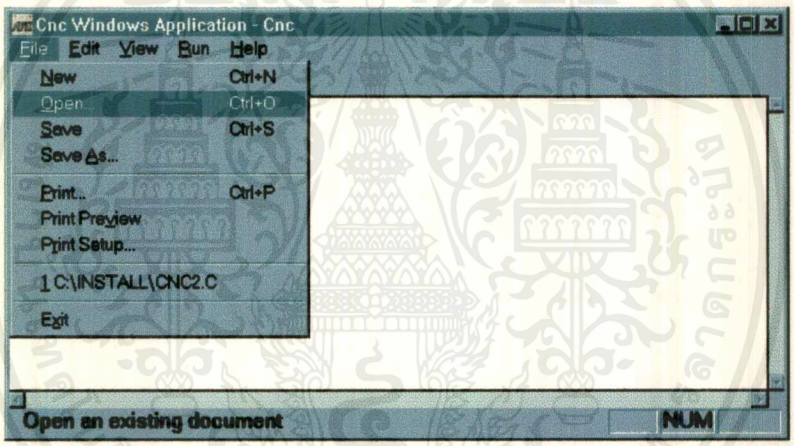
ภาคผนวก จ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

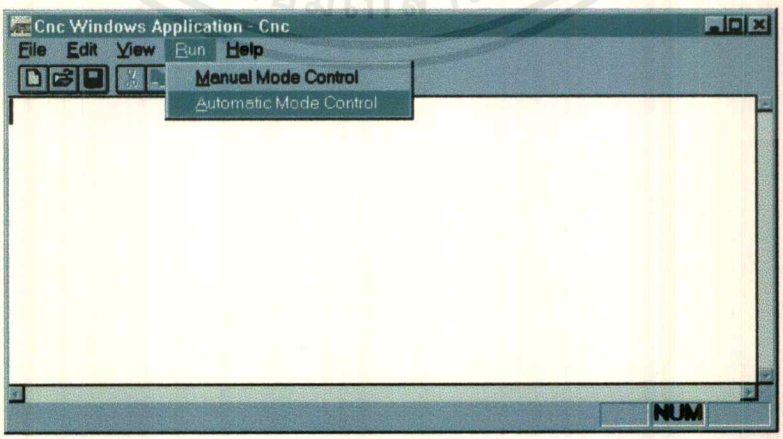
การเขียนส่วน Menu บนการทำงานของวินโดว์ โดยใช้ Visual C++



รูปที่ จ.1 แสดงส่วน Main windows

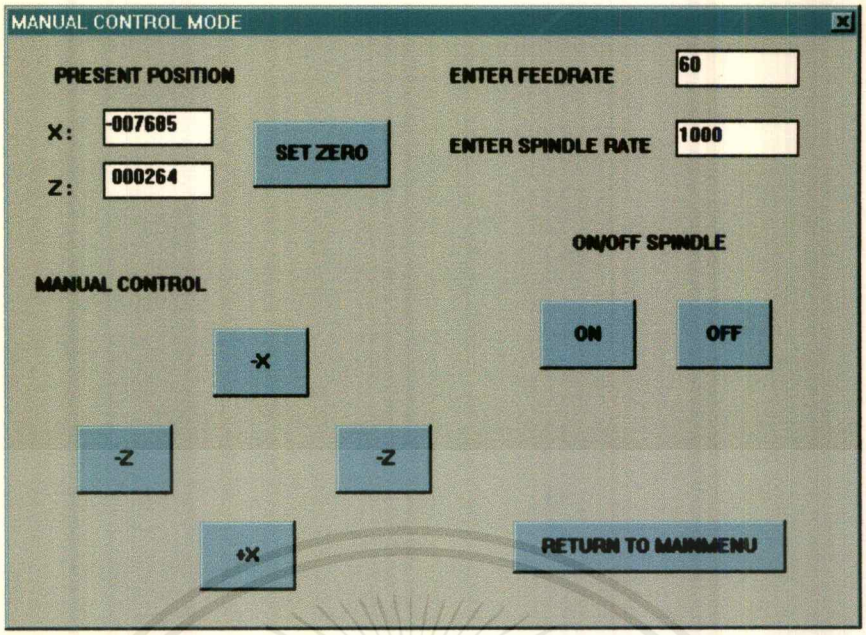


รูปที่ จ.2 แสดงส่วน File Menu

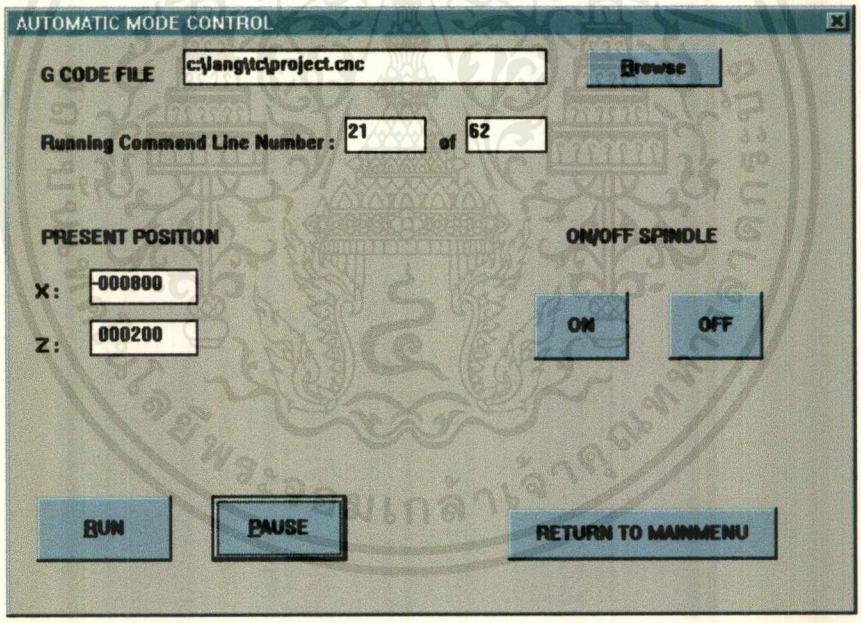


รูปที่ จ.3 แสดงส่วน Run Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.4 แสดงส่วน Manual Control Mode windows



รูปที่ จ.5 แสดงส่วน Auto Control Mode windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code ส่วนโปรแกรม Menu ที่เขียนบนพื้นฐานของ Windows

File cnc.cpp

```
// cnc.cpp : Defines the class behaviors for the application.
//
#include "stdafx.h"
#include "cnc.h"
#include "mainfrm.h"
#include "cncdoc.h"
#include "cncview.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// CCncApp
BEGIN_MESSAGE_MAP(CCncApp, CWinApp)
        //{{AFX_MSG_MAP(CCncApp)
        ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
                // NOTE - the ClassWizard will add and remove mapping macros here.
                //    DO NOT EDIT what you see in these blocks of generated code!
        //}}AFX_MSG_MAP
        // Standard file based document commands
        ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
        ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
        // Standard print setup command
        ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////////////////
// CCncApp construction
CCncApp::CCncApp()
{
        // TODO: add construction code here,
        // Place all significant initialization in InitInstance
}
/////////////////////////////////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// The one and only CCncApp object
CCncApp NEAR theApp;

/////////////////////////////////////////////////////////////////////////
// CCncApp initialization
BOOL CCncApp::InitInstance()
{
        // Standard initialization
        // If you are not using these features and wish to reduce the size
        //  of your final executable, you should remove from the following
        //  the specific initialization routines you do not need.
        SetDialogBkColor();      // Set dialog background color to gray
        LoadStdProfileSettings();  // Load standard INI file options (including MRU)
        // Register the application's document templates.  Document templates
        //  serve as the connection between documents, frame windows and views.
        CSingleDocTemplate* pDocTemplate;
        pDocTemplate = new CSingleDocTemplate(
                IDR_MAINFRAME,
                RUNTIME_CLASS(CCncDoc),
                RUNTIME_CLASS(CMainFrame),    // main SDI frame window
                RUNTIME_CLASS(CCncView));
        AddDocTemplate(pDocTemplate);
        // create a new (empty) document
        OnFileNew();
        if (m_lpCmdLine[0] != '\0')
        {
                // TODO: add command line processing here
        }
        return TRUE;
}
/////////////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About
class CAboutDlg : public CDialog
{
public:
        CAboutDlg();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Dialog Data
        //{{AFX_DATA(CAboutDlg)
        enum { IDD = IDD_ABOUTBOX };
        //}}AFX_DATA
// Implementation
protected:
        virtual void DoDataExchange(CDataExchange* pDX);   // DDX/DDV support
        //{{AFX_MSG(CAboutDlg)
                // No message handlers
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
        //{{AFX_DATA_INIT(CAboutDlg)
        //}}AFX_DATA_INIT
}
void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(CAboutDlg)
        //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
        //{{AFX_MSG_MAP(CAboutDlg)
                // No message handlers
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
// App command to run the dialog
void CCncApp::OnAppAbout()
{
        CAboutDlg aboutDlg;
        aboutDlg.DoModal();
}
/////////////////////////////////////////////////////////////////////////////
// CCncApp commands
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File mainfrm.cpp

```
// mainfrm.cpp : implementation of the CMainFrame class
#include "stdafx.h"
#include "cnc.h"
#include "mainfrm.h"
#include "manual.h"
#include "auto.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// CMainFrame
IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)
BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
        //{{AFX_MSG_MAP(CMainFrame)
        ON_WM_CREATE()
        ON_COMMAND(ID_RUN_AUTO, OnRunAuto)
        ON_COMMAND(ID_RUN_MANUAL, OnRunManual)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////////////////
// arrays of IDs used to initialize control bars
// toolbar buttons - IDs are command buttons
static UINT BASED_CODE buttons[] =
{
        // same order as in the bitmap 'toolbar.bmp'
        ID_FILE_NEW,
        ID_FILE_OPEN,
        ID_FILE_SAVE,
                ID_SEPARATOR,
        ID_EDIT_CUT,
        ID_EDIT_COPY,
        ID_EDIT_PASTE,
                ID_SEPARATOR,
        ID_FILE_PRINT,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ID_APP_ABOUT,
};
static UINT BASED_CODE indicators[] =
{
        ID_SEPARATOR,          // status line indicator
        ID_INDICATOR_CAPS,
        ID_INDICATOR_NUM,
        ID_INDICATOR_SCRL,
};
//////////////////////////////////////////////////////////////////////
// CMainFrame construction/destruction
CMainFrame::CMainFrame()
{
        // TODO: add member initialization code here
}
CMainFrame::~CMainFrame()
{
}
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
        if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
                return -1;
        if (!m_wndToolBar.Create(this) ||
                !m_wndToolBar.LoadBitmap(IDR_MAINFRAME) ||
                !m_wndToolBar.SetButtons(buttons,
                  sizeof(buttons)/sizeof(UINT)))
        {
                TRACE("Failed to create toolbar\n");
                return -1;      // fail to create
        }
        if (!m_wndStatusBar.Create(this) ||
                !m_wndStatusBar.SetIndicators(indicators,
                  sizeof(indicators)/sizeof(UINT)))
        {
                TRACE("Failed to create status bar\n");
                return -1;      // fail to create
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        return 0;
}
////////////////////////////////////////////////////////////////////
// CMainFrame diagnostics
#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
        CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const
{
        CFrameWnd::Dump(dc);
}
#endif //_DEBUG
////////////////////////////////////////////////////////////////////
// CMainFrame message handlers
void CMainFrame::OnRunAuto()
{
        // TODO: Add your command handler code here
        Auto dlg;
        dlg.DoModal();
}
void CMainFrame::OnRunManual()
{
        // TODO: Add your command handler code here
        Manual dlg;
        dlg.DoModal();
}
```

File cncdoc.cpp

```
// cncdoc.cpp : implementation of the CCncDoc class
//
#include "stdafx.h"
#include "cnc.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "cncdoc.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// CCncDoc
IMPLEMENT_DYNCREATE(CCncDoc, CDocument)
BEGIN_MESSAGE_MAP(CCncDoc, CDocument)
        //{{AFX_MSG_MAP(CCncDoc)
                // NOTE - the ClassWizard will add and remove mapping macros here.
                //    DO NOT EDIT what you see in these blocks of generated code!
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
/////////////////////////////////////////////////////////////////////////////
// CCncDoc construction/destruction
CCncDoc::CCncDoc()
{
        // TODO: add one-time construction code here
}
CCncDoc::~CCncDoc()
{
}
BOOL CCncDoc::OnNewDocument()
{
        if (!CDocument::OnNewDocument())
                return FALSE;
        // TODO: add reinitialization code here
        // (SDI documents will reuse this document)
        return TRUE;
}
/////////////////////////////////////////////////////////////////////////////
// CCncDoc serialization
void CCncDoc::Serialize(CArchive& ar)
{
        if (ar.IsStoring())
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                // TODO: add storing code here
        }
        else
        {
                // TODO: add loading code here
        }
}
/////////////////////////////////////////////////////////////////////////////
// CCncDoc diagnostics
#ifdef _DEBUG
void CCncDoc::AssertValid() const
{
        CDocument::AssertValid();
}
void CCncDoc::Dump(CDumpContext& dc) const
{
        CDocument::Dump(dc);
}
#endif //_DEBUG
/////////////////////////////////////////////////////////////////////////////
// CCncDoc commands
```

File auto.cpp

```
// auto.cpp : implementation file
//
#include "stdafx.h"
#include "cnc.h"
#include "auto.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// Auto dialog
Auto::Auto(CWnd* pParent /*=NULL*/)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        : CDialog(Auto::IDD, pParent)
{
        //{{AFX_DATA_INIT(Auto)
                // NOTE: the ClassWizard will add member initialization here
        //}}AFX_DATA_INIT
}
void Auto::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(Auto)
                // NOTE: the ClassWizard will add DDX and DDV calls here
        //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(Auto, CDialog)
        //{{AFX_MSG_MAP(Auto)
        ON_BN_CLICKED(IDC_BROWSE, OnBrowse)
        ON_BN_CLICKED(IDC_OFFSPINAUTO, OnOffspinauto)
        ON_BN_CLICKED(IDC_ONSPINAUTO, OnOnspinauto)
        ON_BN_CLICKED(IDC_PAUSE, OnPause)
        ON_BN_CLICKED(IDC_RETURNAUTO, OnReturnauto)
        ON_BN_CLICKED(IDC_RUN, OnRun)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////////////////////////////////////////
// Auto message handlers
void Auto::OnBrowse()
{
        // TODO: Add your control notification handler code here

}
void Auto::OnOffspinauto()
{
        // TODO: Add your control notification handler code here

}
void Auto::OnOnspinauto()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        // TODO: Add your control notification handler code here

}
void Auto::OnPause()
{
        // TODO: Add your control notification handler code here
}
void Auto::OnReturnauto()
{
        // TODO: Add your control notification handler code here
        Auto::OnOK();
}
void Auto::OnRun()
{
        // TODO: Add your control notification handler code here
}
```

File manual.cpp

```
// manual.cpp : implementation file
//
#include "stdafx.h"
#include "cnc.h"
#include "manual.h"
#ifdef _DEBUG
#undef THIS_FILE
static char BASED_CODE THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// Manual dialog
Manual::Manual(CWnd* pParent /*=NULL*/)
        : CDialog(Manual::IDD, pParent)
{
        //{{AFX_DATA_INIT(Manual)
                // NOTE: the ClassWizard will add member initialization here
        //}}AFX_DATA_INIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void Manual::DoDataExchange(CDataExchange* pDX)
{
        CDialog::DoDataExchange(pDX);
        //{{AFX_DATA_MAP(Manual)
                // NOTE: the ClassWizard will add DDX and DDV calls here
        //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(Manual, CDialog)
        //{{AFX_MSG_MAP(Manual)
        ON_BN_CLICKED(IDC_OFFSPIN, OnOffspin)
        ON_BN_CLICKED(IDC_ONSPIN, OnOnspin)
        ON_BN_CLICKED(IDC_RETURN, OnReturn)
        ON_BN_CLICKED(IDC_XNEGATIVE, OnXnegative)
        ON_BN_CLICKED(IDC_XPOSITIVE, OnXpositive)
        ON_BN_CLICKED(IDC_ZERO, OnZero)
        ON_BN_CLICKED(IDC_ZNEGATIVE, OnZnegative)
        ON_BN_CLICKED(IDC_ZPOSITIVE, OnZpositive)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////////////////////////////////////////
// Manual message handlers
void Manual::OnOffspin()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnOnspin()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnReturn()
{
        // TODO: Add your control notification handler code here
        Manual::OnOK();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void Manual::OnXnegative()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnXpositive()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnZero()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnZnegative()
{
        // TODO: Add your control notification handler code here

}
void Manual::OnZpositive()
{
        // TODO: Add your control notification handler code here

}
```

File stdafx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
//  stdafx.pch will be the pre-compiled header
//  stdafx.obj will contain the pre-compiled type information
#include "stdafx.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้