

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัย เรื่องเซอร์โวไฮดรอลิกส์ควบคุมตำแหน่งแบบไฮบริดจ์นี้ มีเอกสารและงานวิจัยที่เกี่ยวข้องที่ผู้วิจัยได้ศึกษา ซึ่งจะเป็นประโยชน์และทำให้การวิจัยครั้งนี้บรรลุตามจุดมุ่งหมายที่ตั้งไว้ดังต่อไปนี้

1. ไมโครคอนโทรลเลอร์ (Microcontroller)
2. การเขียนโปรแกรมด้วยภาษาซี (C Language)
3. ระบบไฮดรอลิกส์ (Hydraulic System)
4. การควบคุมระบบเปิด (Open Loop control system)
5. การควบคุมระบบปิด (Closed Loop control system)
6. การควบคุมระบบ พี ไอ ดี (PID control system)
7. การควบคุมระบบฟัซซี่ (Fuzzy control system)

1. ไมโครคอนโทรลเลอร์ (Microcontroller)

เทคนิคในสมัยใหม่ส่วนใหญ่ที่นำเข้าสู่ท้องตลาด มักจะนำไปสู่ความเจริญก้าวหน้าทางวิทยาการอีกมุมมองหนึ่งอาจจะถูกวิจารณ์ในด้านลบหรือเกิดผลกระทบเมื่อนำเทคโนโลยีนั้น มาใช้งานอุปกรณ์ทางอิเล็กทรอนิกส์นับวันก็มีความซับซ้อนขึ้นเรื่อยๆ เนื่องจากการพัฒนาอุปกรณ์ทางอิเล็กทรอนิกส์อย่างต่อเนื่อง

1.1 ประวัติไมโครคอนโทรลเลอร์ ครั้งแรกที่ไมโครโปรเซสเซอร์ได้ถูกนำเข้าสู่ท้องตลาด และได้เป็นที่รู้จักอย่างรวดเร็ว โดยกลุ่มอุตสาหกรรม รมผู้ผลิตทำให้สามารถผลิตผลิตภัณฑ์ในต้นทุนที่ถูกลงอย่างไม่เคยมีมาก่อนเช่นเดียวกับมุมมองของงานทดลองและงานอดิเรกซึ่งพัฒนาผ่านลายวงจรทำให้ไมโครโปรเซสเซอร์ไม่มีความสามารถในการทำงานได้รวดเร็วเหมือนเดิม สาเหตุที่เป็นเช่นนี้เพราะในช่วงเวลาเริ่มแรกใช้จ่ายอุปกรณ์ ต้นทุนราคาที่สูง แต่ไม่ใช้ราคาของไมโครโปรเซสเซอร์อย่างเดียว ราคาของอุปกรณ์ที่เป็นส่วนประกอบอื่นๆ รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ การพัฒนาโปรแกรมรวมทั้งการทดสอบก็สูงเช่นกัน ด้วยเหตุนี้ไมโครคอนโทรลเลอร์ได้ถูกสร้างและพัฒนาขึ้นมา โดยราคาของอุปกรณ์ลดลงและเหมาะสม ที่จะนำมาใช้งาน ถ้าพูดถึงการผลิตไมโครโปรเซสเซอร์เป็นไมโครคอนโทรลเลอร์ คือ เพียงนำเอาไมโครโปรเซสเซอร์รวมกับกลุ่ม Peripherals ต่างๆ เข้า

ด้วยกัน ในปี ค.ศ.1976 บริษัท INTE ได้เริ่มผลิตและจำหน่ายไมโครโปรเซสเซอร์ตระกูล MCS-48 เพื่อใช้ในการควบคุม โดยเฉพาะและเนื่องจากไมโครโปรเซสเซอร์ตระกูลนี้มีจุดประสงค์หลักเพื่อใช้ในการควบคุมจึงมีวงจรพื้นฐานที่จำเป็นรวมอยู่ภายในแล้วทำให้สามารถทำงานได้โดยไม่ต้องพึ่งอุปกรณ์ภายนอกอีกเท่าใดนักไมโครโปรเซสเซอร์ตระกูลนี้จึงมีชื่อเรียกอีกอย่างหนึ่งว่า ไมโครคอนโทรลเลอร์หรือซิงเกิลชิพ (Single Chip)

1.2 ความหมายของไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์ คือ อุปกรณ์อิเล็กทรอนิกส์อย่างหนึ่งซึ่งภายในประกอบด้วยวงจรอื่นๆ หลายวงจรและทำงานร่วมกันเช่น หน่วยประมวลผลกลาง (CPU: Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic Logic Unit) วงจรออสซิลเลเตอร์ (Oscillator) หน่วยความจำ (Memory: ROM, RAM) วงจรรับสัญญาณอินพุตและขับสัญญาณเอาต์พุต (Input/output port) เป็นต้นด้วยเหตุนี้ ไมโครคอนโทรลเลอร์จึงสามารถนำไปประยุกต์ใช้งานควบคุมต่างๆ ได้ดีเนื่องจากสามารถเขียนโปรแกรมควบคุมได้อย่างอิสระแล้วแต่เราต้องการควบคุมอะไร

1.3 หน่วยประมวลผลกลาง (CPU) CPU เปรียบได้กับสมองของคนเรานั้นเองเพราะการคำนวณต่างๆ เกิดขึ้นที่ CPU ประกอบด้วยวงจรต่างๆ หลายวงจร เช่น วงจรควบคุมเวลาและระบบการทำงาน (Timing and Control Unit) ซึ่งจะทำหน้าที่จัดการทั้งหมดของวงจรทั้งประมวลผลและควบคุมตามคำสั่งที่ได้รับ การคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) โดยจะทำหน้าที่คำนวณและประมวลผลทางคณิตศาสตร์และระบบลอจิกวงจรถอดรหัสคำสั่ง (Instruction Decoder) จะทำหน้าที่แปลงคำสั่งทั้งหมดให้เป็นภาษาเครื่อง (Machine Language) วงจรควบคุมการทำงานของ Counter (Program Counter) วงจรควบคุมสัญญาณนาฬิกา (Oscillator) ตลอดจนหน่วยความจำภายใน

1.4 พอร์ตอินพุต/เอาต์พุต ไมโครคอนโทรลเลอร์จะมีพอร์ตสำหรับติดต่อสื่อสารกับอุปกรณ์ภายนอกแล้วแต่วัตถุประสงค์ในการใช้งานและคุณสมบัติของพอร์ตโดยสามารถติดต่อสื่อสารกับอุปกรณ์ภายนอกทำหน้าที่เป็นอินพุตและเอาต์พุตได้ เช่น Pushbutton, Keypzd, Sensor, LCD, Timer/Counter ตลอดจนการแปลงสัญญาณ Analog to Digital Converter เป็นต้น

1.5 ทฤษฎีพื้นฐาน

1) ระบบฐานเลข (Number System) ในการโปรแกรมกับภาษาชั้นสูงทั่วไปไม่สามารถจะเข้าใจกับโครงสร้างฮาร์ดแวร์เหมือนกับการที่ต้องห้ามไม่ให้รู้ภายใน PC ทำงานกับโปรแกรมอย่างไรในการแสดงตัวเลขเพื่อที่จะสามารถทำงานกับไมโครคอนโทรลเลอร์ได้จำเป็นที่จะต้องศึกษาความหมายพื้นฐานและการแสดงระบบจำนวน (ตัวเลข) ในทาง Digital Technic

2) ตัวเลขฐานสิบ (Decimal System) เป็นระบบฐานที่ใช้ในชีวิตประจำวันซึ่งรหัสอักษรแทน 10 ตัว (0 ถึง 9) ตัวอย่าง เช่น $1327 = 1 \times 1000 + 3 \times 100 + 2 \times 10 + 7 \times 1$

โดยอักษรตัวท้ายมีน้ำหนักเป็นหนึ่ง 1 ถัดมาเป็น 10 และ 100 ตามลำดับจากตัวอย่างมีน้ำหนักสูงสุดคือ 1000 น้ำหนักทุกตัวอักษรจะเพิ่มขึ้นด้วยแฟกเตอร์ 10

3) ตัวเลขฐานสอง (Binary System) ไมโครคอนโทรลเลอร์ไม่สามารถแสดงสถานะได้ถึง 10 สถานะ มันสามารถแสดงสถานะตามระดับตรรกศาสตร์ “สูงและต่ำ” เท่านั้น สถานะสูงสามารถกล่าวได้ว่า (ตามหลักการของแรงดันไฟฟ้า)

High, Low = แรงดันไฟฟ้า

U = 0.5 V

4) การแสดงตัวเลขในเครื่องคอมพิวเตอร์ ระบบคอมพิวเตอร์ทำงานกับระบบตัวเลขระบบฐานสองนั้น เพราะว่าโครงสร้างของหน่วยความจำข้อมูลในคอมพิวเตอร์จัดในระบบเลขฐานสองในทางปฏิบัติหน่วยความจำจะถูกเขียนและอ่านในลักษณะของไบนารี คือ 0 สามารถใช้งานได้ 8 บิต ต่อ 1 หน่วยพื้นที่ความจำต่อ 1 ไบนารี สามารถแสดงค่าตั้งแต่ 00000000 B (0 dec) ถึง 11111111 B (255 dec) ไมโครคอนโทรลเลอร์ตระกูล PIC ก็ทำงานกับข้อมูลขนาด 8 บิต ทุกๆ รีจิสเตอร์และหน่วยความจำจะมีความกว้าง 8 บิต เช่นเดียวกับบัสข้อมูลซึ่งหน่วยควบคุมใช้ในการแลกเปลี่ยนข้อมูลกับกลุ่มข้อมูลอื่นๆ สำหรับคอมพิวเตอร์ที่มีขนาดใหญ่ที่มีความสามารถสูง เช่น PC หรืออื่นๆ ที่ทำงานกับข้อมูลกว้างขนาด 16 หรือ 32 บิต จะได้เปรียบในเรื่องของการประมวลผลได้รวดเร็ว

5) Microcontroller PIC16F877 PIC16F877 เป็น CPU ที่มีคุณสมบัติและฟังก์ชันการทำงานต่างๆ ที่ค่อนข้างครบถ้วนและมีโครงสร้างที่ไม่ยุ่งยากมากนัก ซึ่งไมโครคอนโทรลเลอร์ในตระกูล PIC นี้จะมีโครงสร้างการทำงานใกล้เคียงกันซึ่งในเบอร์ 16F877 นี้จะเป็นแบบ 14 Bit Core คือ มีโครงสร้างการทำงานของคำสั่งเป็นแบบ 14 บิต และข้อดีของ CPU ในตระกูล PIC นี้ที่เห็นได้ชัด คือ เรื่องของการจ่ายและรับของกระแสสัญญาณ I/O พอร์ตต่างๆ ที่สามารถจ่ายและรับกระแสสูงสุดได้ถึง 25 Ma ทำให้สามารถนำไปขับโหลดต่างๆ ได้เป็นอย่างดีโดยไม่ต้องต่อบัฟเฟอร์หรือ วงจรขยายกระแสอื่นๆ แต่อย่างใด ไมโครคอนโทรลเลอร์ CP-PIC V3 PLUS EXP เป็นบอร์ดไมโครคอนโทรลเลอร์ PIC เบอร์ 16F877-20 โดยความถี่ในการใช้งานของบอร์ดนี้ คือ 10 MHz และเป็นบอร์ดที่ใช้โปรแกรมแบบแรงดันไฟสูง ทำให้สามารถใช้ขาสัญญาณ I/O ได้ครบทั้ง 33 ขาสัญญาณ

5.1 คุณสมบัติของ PIC16F877

- 1) 35 Instruction คำสั่ง
- 2) ในการปฏิบัติงานคำสั่งต่างๆ จะใช้ Cycle เดียว และ 2 Cycle

ในคำสั่งที่เป็นการกระโดด

- 3) ความถี่สูงสุดที่ทำงานได้ คือ 20 MHz (16F877-20/p)
- 4) การทำงานจะเป็นลักษณะ (Pipeline) ทำให้มีการทำงานที่เร็วขึ้น

- 5) หน่วยความจำโปรแกรม Flash program memory มีขนาด 8 k (14 bit word)
- 6) หน่วยความจำข้อมูล (RAM) 368 Byte
- 7) หน่วยความจำข้อมูล (EEPROM) 256 Byte
- 8) สามารถตอบสนองการอินเทอร์รัพได้ถึง 14 แหล่ง
- 9) STACK 8 ระดับ
- 10) เพาเวอร์อนรีเซต (POR) เพาเวอร์อัปไทมเมอร์ (PWRT) และ Oscillator

Start-Up Timer

- 11) Watchdog Timer
- 12) สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- 13) โหมดประหยัดพลังงาน (Sleep Mode)
- 14) เลือกโหมดของสัญญาณนาฬิกาได้หลายโหมด
- 15) สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- 16) ฟังก์ชันการโปรแกรมแบบ ICSP (In-Circuit Serial Programming)
- 17) ทำงานที่ไฟเลี้ยง 2.0 V ถึง 5.5 V
- 18) กระแสซิงก์และซอสของพอร์ต คือ 25 Ma
- 19) Timer/Counter จำนวน 3 ตัว คือ Timer0, Timer1, และ Timer2
- 20) โมดูล Capture/Compare/PWM จำนวน 2 ชุด
- 21) Analog to Digital Converter ความละเอียด 10 บิต 8 แชนแนล

ภายในตัว

- 22) มีโมดูลการสื่อสาร USART
- 23) มีโมดูลตรวจจับแรงดันไฟเลี้ยง Brown-out reset (BOR)
- 25) พอร์ต I/O มี 5 พอร์ต ประกอบด้วย A, B, C, D และ E แต่ละพอร์ต

จะมีจำนวนบิตไม่เท่ากันซึ่งรวมแล้วจะมี I/O จำนวน 33 บิต

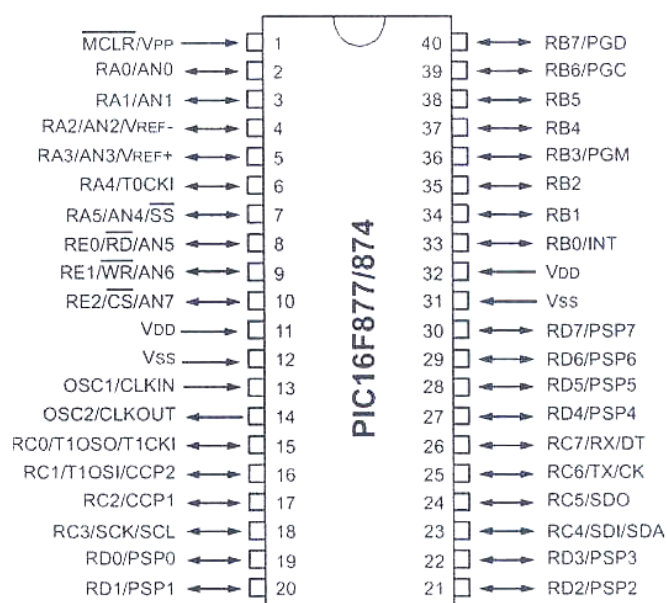
5.2 ขาสัญญาณของ Microcontroller PIC16F877 ขาสัญญาณของ PIC เบอร์ 16F877 นี้จะมีทั้งหมด 40 ขา จะประกอบไปด้วยขาที่ทำหน้าที่ต่างๆ โดยจะมีขาสัญญาณ I/O Ports ทั้งหมด 33 ขา โดยสามารถนำไปใช้เป็นอินพุต / เอาต์พุตได้ทั้งหมดทุกขา ยกเว้น RA4 ซึ่งโครงสร้างภายในเป็นแบบ Open Drain ดังนั้นหากต้องการนำไปใช้เป็นขาสัญญาณเอาต์พุตจะต้องต่อตัวต้านทานพูลอัพ (Pull-up) ไว้ด้วยส่วนขาที่เหลือใช้งานได้ตามปกตินอกจากขาสัญญาณ I/O แล้วยังประกอบด้วยขาสัญญาณอื่นๆ อีก คือ ขาไฟเลี้ยง, กราวด์, ขารีสเซต และขาออสซิลเลเตอร์ใน

ตัว CPU ของ Microcontroller PIC16F877 ขนาด 40 Pins มีขาสัญญาณที่ใช้ติดต่อ Input/output port มีจำนวนขาสัญญาณทั้งสิ้น 33 Pins ดังนี้

- 1) PORTA RA0_RA5 จำนวน 6 Pins
- 2) PORTB RB0_RB7 จำนวน 8 Pins
- 3) PORTC RC0_RC7 จำนวน 8 Pins
- 4) PORTD RD0_RD7 จำนวน 8 Pins
- 5) PORTE RE0_RE2 จำนวน 3 Pins



รูปที่ 2.1 Microcontroller PIC16F877



รูปที่ 2.2 ขาทั้ง 40 Pins ของ PIC16F877

5.3 การทำงานของขาสัญญาณของ Microcontroller PIC16F877

1) PORT RAO_RA5 พอร์ต A มีขาสัญญาณจำนวน 6 Pins โดยเป็นขาสัญญาณใช้ติดต่อ Input/output port และยังทำหน้าที่เป็น Input ของสัญญาณ ADC (Analog to Digital Converter) ด้วย

RA0/AN0	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ADC ช่อง 0
RA1/AN1	ขาสัญญาณติดต่อ Input สำหรับ Output Port ขาสัญญาณ Input สำหรับ ADC ช่อง 1
RA2/AN2	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ACC ช่อง 2
RA3/AN3	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ADC ช่อง 1
RA4/TOCK1	ขาสัญญาณนาฬิกา Input Timer 0 ขาสัญญาณ Output เปรียบเทียบ ADC ช่อง 1
RA5/AN4/SS	ขาสัญญาณ Input สำหรับ ADC ช่อง 4 ขาสัญญาณ Slave Select สื่อสาร Serial Port แบบ Synchronize ขาสัญญาณ Output เปรียบเทียบ ADC ช่อง 2

2) PORTB RBO_RB7 พอร์ต B มีขาสัญญาณจำนวน 8 pins ภายในมีวงจรพูลอัพ (R-Pull-Up) ใช้ติดต่อ Input/output port และยังทำหน้าที่เป็นสัญญาณ Input ในการ Interrupt จากภายนอกด้วย

RB0/INT	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input รับสัญญาณ Interrupt จากภายนอก
RB1	ขาสัญญาณติดต่อ Input/output port
RB2	ขาสัญญาณติดต่อ Input/output port
RB3/LVP	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input รับแรงดันต่ำ (5V) ถ้ามีการ Enable
RB4-RB7	ขาสัญญาณขาสัญญาณติดต่อ Input/output port ภายในมีวงจรพูลอัพ (R-Pull-Up) และ Interrupt Logicขาที่สามารถเกิด Interrupt ได้ถ้ามีการ Enable

3) PORTC RC0_RC7 พอร์ต C มีขาสัญญาณจำนวน 8 pins โดยพอร์ต C เป็นพอร์ตที่ติดต่อสัญญาณได้หลายรูปแบบ เช่น ติดต่อ Input/output port, Timer, 12C, PWM

PC0/T1OSO/TICK1	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Output วงจร Oscillator ของ Timer 1 ขาสัญญาณ Input ของสัญญาณ Clock Timer1
PC1/T1OSCCP2	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input วงจร Oscillator ของ Timer1 ขาสัญญาณ Output วงจรเปรียบเทียบสัญญาณของโมดูล CCP2 ขาสัญญาณ Output PWM RC2/CCP1 ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Output วงจรเปรียบเทียบ สัญญาณของโมดูล CCP1 ขาสัญญาณ Output PWM
RC3/SCK/SCL	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Clock ของวงจร SPI (SCK) ขาสัญญาณ Clock ของระบบบัส I²C (SCL)
RC4/SDI/SDA	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input/Serial Data ของระบบ SPI (SDI) ขาสัญญาณ Data ของ I²C (SCA)
RC5/SDO	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณส่ง Data/Serial Port (IxD) ขาสัญญาณ Clock/Synchronize (CK)
RC7/RxD/DT	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณรับ Data/Serial Port (RxD) ขาสัญญาณ Data/Synchronize (DT)

4) PORT RD0_RD7 พอร์ต D มีขาสัญญาณจำนวน 8 Pins

โดยเป็นขาสัญญาณใช้ติดต่อ Input/output port และยังทำหน้าที่ขยายพอร์ตแบบขนาน (PSP: Parallel Slave Port)

RD0/PSP0	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 0
RD1/PSP1	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 1
RD2/PSP2	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 2

RD3/PSP3	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 3
RD4/PSP4	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 4
RD5/PSP5	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 5
RD6/PSP6	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 6
RD7/PSP7	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณขยายพอร์ตแบบขนานบิต 7

5) PORTE PE0_RE2 พอร์ต E มีขาสัญญาณจำนวน 3 pins โดยเป็นขาสัญญาณใช้ติดต่อ Input/output port และยังทำหน้าที่เป็น Input ของสัญญาณ ADC (analog to Digital Converter) ด้วย

RE0/AN5/RD	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ADC ช่อง 5 ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการอ่าน(RD: Read)
RE1/AN6/WR	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ADC ช่อง 6 ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการ(WR: Write)
/AN7/CS	ขาสัญญาณติดต่อ Input/output port ขาสัญญาณ Input สำหรับ ADC ช่อง 7 ขาสัญญาณขยายพอร์ตแบบขนานควบคุมการเลือกอุปกรณ์

(CS: Chip Select)

6) ขาสัญญาณทั่วไป จะมีหน้าที่ในการติดต่อสัญญาณดังนี้

MCLR/Vpp	ขาสัญญาณสำหรับ Reset จะทำหน้าที่ลอจิก 0 (MasterClear Input) ขาสัญญาณรับแรงดันโปรแกรม (Programming-Voltage)
OSC1/CLKIN	ขาสัญญาณสำหรับต่อCrystal/รับ สัญญาณClock
OSC2/CLKOUT	ขาสัญญาณสำหรับต่อ Crystal/รับสัญญาณClock
Vdd	ขาต่อไปเลี้ยง (+5V)
Vss	ขาต่อกราวด์

สำหรับการควบคุมพอร์ตในการใช้งานจะต้องกำหนดที่ Register ว่าต้องการให้ขาสัญญาณของพอร์ตเป็น Input หรือ Output โดยกำหนด ดังนี้

PORT A SET_TRIS_A ()

PORT B SET_TRIS_B ()

PORT C SET_TRIS_C ()

PORT D SET_TRIS_D ()

PORT E SET_TRIS_E ()

การกำหนดให้พอร์ตไหนเป็น Output ให้เขียนข้อมูล 0 ที่บิตนั้น แต่ถ้าต้องการให้พอร์ตเป็น Input ให้เขียนข้อมูล 1 ที่บิตนั้น เช่น กำหนด PORT A ขนาด 6 บิต มีรายละเอียดดังนี้

ตารางที่ 2.1 ตัวอย่างการกำหนด PORT A ขนาด 6 บิต

Register	7 bit	6 bit	5 bit	4 bit	3 bit	2 bit	1 bit	0 bit
TRIS A	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
PORT A	-	-	RA5	RA4	RA3	RA2	RA1	RA0

PORT A มีทั้งหมด 6 บิต

ตัวอย่าง

SET_TRIS_A (0XFF) » A5, A4, A3, A2, A1, A0 เป็นขาอินพุต

SET_TRIS_A (0XF0) » A5, A4 เป็นขาอินพุต

» A3, A2, A1, A0, เป็นขาเอาต์พุต

SET_TRIS_A (0X0F) » A5, A4 เป็นขาเอาต์พุต

» A3, A2, A1, A0 เป็นขาอินพุต

SET_TRIS_A (0X00) » A5, A4, A3, A2, A1, A0 เป็นขาเอาต์พุต

5.4 การจัดหน่วยความจำ ของไมโครคอนโทรลเลอร์ PIC16F877 ในการเขียนโปรแกรมด้วยภาษาซีให้กับไมโครคอนโทรลเลอร์นั้นต้องคำนึงถึงชนิดของหน่วยความจำและวิธีการเข้าถึงด้วย ซึ่งต่างจากการเขียนบน PC ที่สนใจเพียงชนิดของตัวแปรว่าจะใช้เก็บข้อมูลประเภทใด สำหรับหน่วยความจำในระบบไมโครคอนโทรลเลอร์ PIC นั้น จะมีหน่วยความจำในการใช้งาน 3 ประเภท ดังนี้

1) หน่วยความจำโปรแกรมแฟลช หน่วยความจำแบบแฟลช ในปัจจุบันนี้ หน่วยความจำชนิดนี้ได้ถูกนำมาใช้กับ Microcontroller หลายบริษัทหลายรุ่นโดยมีคุณสมบัติในการ

เขียนโปรแกรมและลบโปรแกรมได้มากกว่า 100,000 ครั้ง ซึ่งการทำงานจะมีความเร็วสูงมากเหมาะกับการพัฒนางานที่มีขนาดใหญ่

2) หน่วยความจำโปรแกรม หน่วยความจำส่วนนี้มีไว้ใช้เก็บข้อมูลขณะประมวลผลโปรแกรม สามารถอ่านและเขียนข้อมูลได้ขณะมีไฟเลี้ยง แต่เมื่อไม่จ่ายไฟเลี้ยงข้อมูลต่างๆ จะสลายไปหากหน่วยความจำส่วนนี้ไม่พอใช้งาน จะต้องต่อหน่วยความจำแรมภายนอกเพิ่ม ปัจจุบันเทคโนโลยีก้าวหน้าขึ้นมากชิพบางตัวจะมีการบรรจุหน่วยความจำประเภท Data memory เข้าไปในชิพเลย

3) หน่วยความจำแบบอีอีพรอม หน่วยความจำแบบอีอีพรอม เป็นหน่วยความจำที่สามารถเขียนและลบโปรแกรมด้วยกระแสไฟฟ้า ในหน่วยความจำถาวร ของ PROM โดยภายในมีการพัฒนาให้ RAM ที่มีหน่วยความจำชั่วคราวไว้เก็บข้อมูลได้ถาวรแบบหน่วยความจำ โดยสามารถเขียนและลบโปรแกรมจำนวนหลายๆ ครั้งได้

5.5 ไมโครคอนโทรลเลอร์ PIC16F877 แบบ Flash Memory มีหน่วยความจำข้อมูล (Data Memory) แบบ RAM 368 bytes และ EEPROM 256 bytes โดย CPU จะมีการจัดสรรพื้นที่ในการใช้งานทั้งหมด 4 bank ในแต่ละ bank จะมีขนาดสูงสุด 128 bytes โดยในแต่ละ bank จะมีพื้นที่ในการทำงานที่แตกต่างกัน การจัดสรรพื้นที่มีรายละเอียด ดังนี้

พื้นที่ bank 0	Address (0x00-0x7F)
	Address (0x00-0x1F) เป็นพื้นที่ของ Register
	Address (0x20-0x7F) เป็นพื้นที่หน่วยความจำข้อมูลสำหรับใช้งานทั่วไปโดยมีขนาด 90 bytes
พื้นที่ bank 1	Address (0x80-0xFF)
	Address (0x80-0x9F) เป็นพื้นที่ของ Register

2. การเขียนโปรแกรมด้วยภาษาซี (C Language)

ภาษาซี คือภาษาคอมพิวเตอร์ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาโดย เดนิส ริสชี (Dennis Ritchie) เมื่อประมาณต้นปีค.ศ. 1970 เพื่อใช้งานบนระบบปฏิบัติการยูนิกซ์ ต่อมาถูกนำไปใช้ในระบบปฏิบัติการต่าง ๆ จนถูกใช้เป็นภาษาพื้นฐานสำหรับภาษาอื่น เช่น ภาษาจาวา (Java) ภาษาพีเอชพี (PHP) ภาษาซีชาร์ป (C#) ภาษาซีพลัสพลัส (C++) ภาษาเพิร์ล (Perl) ภาษาไพทอน (Python) หรือภาษารูบี้ (Ruby)

การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์นั้น สามารถใช้ได้หลายภาษา เช่น ภาษาแอสเซมบลี (Assembly Language) ภาษาซี (C Language) ภาษาเบสิก (Basic Language) เป็นต้น ในปัจจุบันใช้ภาษาซี เพราะมีความยืดหยุ่นในการใช้กับไมโครคอนโทรลเลอร์ตระกูลอื่น ตลอดจนสามารถเขียนโปรแกรมที่มีความซับซ้อน เช่น การวนลูป หรือฟังก์ชันคณิตศาสตร์ เป็นต้น

ข้อดีอีกประการ คือ โปรแกรมที่เขียนด้วยภาษาซี สามารถแก้ไขได้ง่ายกว่าภาษาแอสเซมบลี เพราะโครงสร้างง่ายต่อการสร้างโมดูลาร์ (Modular) เช่น ฟังก์ชัน Main , ฟังก์ชัน While ตลอดจนคำสั่ง if ,else เป็นต้น

2.1 องค์ประกอบพื้นฐานของภาษา C

อักขระ (characters) ค่าคงที่ (constants) ตัวแปร(variables) ตัวดำเนินการ(operators) นิพจน์(expressions)

Characters แบ่งเป็น ตัวเลข (digits) คือ ตัวเลข 0 ถึง 9 และเลขฐานสิบหก คือ A B C D E และ F

ตัวอักษร (letters) คือตัวอักษรในภาษาอังกฤษ เป็นตัวพิมพ์ใหญ่และตัวพิมพ์เล็ก คือ A-Z และ a-z

อักขระพิเศษ (special characters) คือ ! * + " < # (= | > %) ~ : / ^ - [; ? , & _] ' . space ในภาษาซี ถือว่า เครื่องหมายขีดเส้นใต้เป็นตัวอักษร ตัวหนึ่ง characters ต่าง ๆ นี้จะใช้เป็นค่าคงที่ ตัวแปร ตัวดำเนินการ

Constants เป็นค่าที่มีค่าไม่เปลี่ยนแปลง แบ่งเป็น ค่าคงที่ประเภทเลขจำนวนเต็ม (integer constant) คือเลขจำนวนเต็ม ที่อยู่ระหว่าง -32768 ถึง 32767 เช่น -25 หรือ 0 หรือ 236 ค่าเหล่านี้แต่ละค่าใช้หน่วยความจำในการเก็บ 2 ไบต์ ค่าเหล่านี้เขียนในรูปเลขฐานสิบ ฐานแปด(เขียนโดยใช้เลขศูนย์นำหน้า เช่น 045) เลขฐานสิบหก(เขียนโดยใช้เลขศูนย์และ x นำหน้า เช่น 0x28 0X2AF)

ค่าคงที่ชนิดตัวเลขทศนิยม(floating point constants) เป็นตัวเลขทศนิยม เช่น 1.0 1.6 E+09 ค่านี้ใช้หน่วยความจำในการเก็บค่าละ 4 ไบต์ และมีค่าอยู่ในช่วง 1.2 E - 38 ถึง 3.4E+38 โดย 3 ไบต์แรกเก็บค่าตัวทศนิยม ส่วนไบต์สุดท้ายเก็บเลขยกกำลัง

ค่าคงที่ชนิดตัวเลขทศนิยมความละเอียดสองเท่า (double floating point) เรียกสั้น ๆ ว่า double เก็บจำนวน 2.2E-308 ถึง 1.8E+308 เท่านั้น ใช้หน่วยความจำ 8 ไบต์ 7 ไบต์แรกเก็บเลขทศนิยม ไบต์สุดท้ายเก็บเลขยกกำลัง

ค่าคงที่ชนิดตัวอักขระ (single character constant) เก็บตัวอักขระ 1 ตัว โดยใช้เครื่องหมาย ' และ ' ล้อม 1 ตัวอักขระใช้ 1 ไบต์ เช่น 'E' 'X'

ค่าคงที่ชนิดข้อความ (string constant) ใช้เก็บข้อความ มีความยาวตั้งแต่ 1 ตัวอักขระ เก็บในรูปอาร์เรย์ แต่ละตัวใช้หน่วยความจำ 1 ไบต์ เรียงต่อกันโดยไบต์สุดท้ายจะต้องเก็บ \0 (null charactor) เพื่อบอกว่าจบข้อความแล้ว เช่น "Bodindecha" ใช้หน่วยความจำ 11ไบต์

2.2 ตัวแปร(variables)

เป็นชื่อที่ตั้งเพื่อใช้อ้างอิงถึงข้อมูลต่าง ๆ โดยตัวแปรจะมีการใช้เนื้อที่ในหน่วยความจำในปริมาณที่ต่างกันขึ้นกับชนิดของข้อมูล ข้อมูลพื้นฐานในภาษาซี มี 5 ชนิด คือ อักขระ (char) จำนวนเต็ม (int) จำนวนจริง(float) จำนวนจริงละเอียด2เท่า(double) ไม่ให้ค่าใด ๆ (void) นอกจากนี้เพื่อความสามารถในการทำงานจึงมีการเพิ่มชนิดของตัวแปรขึ้นมาดังตารางที่ 2.2

ตารางที่ 2.2 แสดงชนิดของตัวแปร

คำประกาศชนิดของ ตัวแปร	เครื่องหมาย	จำนวนไบต์ ที่ใช้	ค่าที่เป็นไปได้
char	คิดเครื่องหมาย	1	-128 ถึง 128
int	คิดเครื่องหมาย	2	-32768 ถึง 32767
short	คิดเครื่องหมาย	2	-32768 ถึง 32767
long	คิดเครื่องหมาย	4	-2147483648 ถึง 2147483647
unsigned char	ไม่คิดเครื่องหมาย	1	0 ถึง 255
unsigned int	ไม่คิดเครื่องหมาย	2	0 ถึง 65535
unsigned short	ไม่คิดเครื่องหมาย	2	0 ถึง 65535
unsigned long	ไม่คิดเครื่องหมาย	4	0 ถึง 4294967295
float	คิดเครื่องหมาย	4	3.4E-38 ถึง 3.4E+38
double	คิดเครื่องหมาย	8	1.7E-308 ถึง 1.7E+308
long double	คิดเครื่องหมาย	10	3.4E-4932 ถึง 1.1E+4932

2.3 การตั้งชื่อตัวแปร

2.3.1 ต้องขึ้นต้นด้วยตัวอักษรเท่านั้น ตัวต่อไปจะเป็นตัวเลขหรือตัวอักษรได้ ความยาวไม่เกิน 31 ตัว

2.3.2 ชื่อตัวแปรควรสื่อความหมายของตัวแปร เพื่อป้องกันความสับสนของการพิจารณาโปรแกรม

2.3.3 ห้ามตั้งชื่อตรงกับคำสงวนในภาษาซี ซึ่งมี 33 คำ ดังนี้

asm auto break case char const continue default do double else enum
extern float for goto if int long register return short signed sizeof static struct
switch typedef union unsigned void volatile while

2.4 การประกาศตัวแปร

ในการจะใช้งานตัวแปรจะต้องมีการประกาศชนิดและชื่อของตัวแปรนั้นก่อน โดยการประกาศตัวแปรใช้รูปแบบ คือ **ชนิดของตัวแปร ชื่อตัวแปร** โดยถ้ามีตัวแปรชนิดเดียวอาจประกาศพร้อมกันโดยใช้ คอมมา คั่นระหว่างชื่อของตัวแปร ถ้ามีการกำหนดค่าให้ **ชื่อเครื่องหมาย =** และใช้เครื่องหมายแสดงการจบคำสั่งเมื่อสิ้นสุด คำสั่ง ตัวอย่าง เช่น

```
char name, day = 'S' , surname[20] ="Kodedee";
```

```
int x=5 ,y,z[100];    float a=5.00 ,b,c;    double k=1.234567, m ;
```

2.5 ตัวดำเนินการ(operator)

ตัวดำเนินการมีหลายประเภทลักษณะการดำเนินการแตกต่างกันไป ในที่นี้จะกล่าวถึงบางประเภทเท่านั้น

ตารางที่ 2.3 ตัวดำเนินการทางคณิตศาสตร์ (mathematical operators)

สัญลักษณ์	การดำเนินการ	ตัวอย่าง
+	การบวก	2+5 ผลลัพธ์ 7
-	การลบ	7-4 ผลลัพธ์ 3
*	การคูณ	2*6 ผลลัพธ์ 12
/	การหาร	8/2 ผลลัพธ์ 4
%	การหารหาเศษ	9%4 ผลลัพธ์ 1

ตารางที่ 2.4 ตัวดำเนินการความสัมพันธ์หรือการเปรียบเทียบ (relational operators)

สัญลักษณ์	การดำเนินการ	ตัวอย่าง
<	น้อยกว่า	2<3 ผลลัพธ์ จริง(1)
>	มากกว่า	2>3 ผลลัพธ์ เท็จ(false)(0)
<=	น้อยกว่าหรือเท่ากับ	2<= 3 ผลลัพธ์ จริง(true)
>=	มากกว่าหรือเท่ากับ	2>= 3 ผลลัพธ์ เท็จ(false)
==	เท่ากับ	4==4 ผลลัพธ์ จริง(true)
!=	ไม่เท่ากับ	2!= 2 ผลลัพธ์ เท็จ(false)

ตารางที่ 2.5 ตัวดำเนินการตรรกะ (logical operators)

สัญลักษณ์	การดำเนินการ	ตัวอย่าง
&&	และ(AND)	(2<3)&&(3>1) ผลลัพธ์ จริง
	หรือ(OR)	(2>3) (4<1) ผลลัพธ์ เท็จ(false)
!	ไม่(NOT)	!(2> 3) ผลลัพธ์ จริง(true)

ตารางที่ 2.6 ตัวดำเนินการกำหนดค่า (assignment operators)

สัญลักษณ์	การดำเนินการ	ตัวอย่าง
=	กำหนดค่า(assignment)	a=2 ความหมายคือ กำหนดให้ a มีค่าเป็น 2
+=	การบวก(addition)	a+=b ความหมายคือ(a=a+b)
=	การคูณ(multiplication)	a=b ความหมายคือ(a=a*b)
-=	การลบ(subtraction)	a-=b ความหมายคือ(a=a-b)
/=	การหาร(division)	a/=b ความหมายคือ(a=a/b)
%=	การหารหาเศษ (remainder)	a%=b ความหมายคือ(a=a%b)
++	เพิ่มค่า(increment)	a++ หรือ ++a ความหมายคือ a=a+1
--	ลดค่า(decrement)	a-- หรือ --a ความหมายคือ a=a-1

ตารางที่ 2.7 ลำดับการดำเนินการของตัวดำเนินการ (operator precedence)

ลำดับที่	ตัวดำเนินการ	ลักษณะการทำงาน
1	() [] . ->	ซ้าย ไป ขวา
2	- ~ * &	ขวา ไป ซ้าย
3	++ --	ขวา ไป ซ้าย
4	* / %	ซ้าย ไป ขวา
5	+ -	ซ้าย ไป ขวา
6	<< >>	ซ้าย ไป ขวา
7	< > <= >=	ซ้าย ไป ขวา
8	== !=	ซ้าย ไป ขวา
9	&(bitwise AND)	ซ้าย ไป ขวา
10	^(bitwise XOR)	ซ้าย ไป ขวา
11	(bitwise OR)	ซ้าย ไป ขวา
12	&&	ซ้าย ไป ขวา
13		ซ้าย ไป ขวา
14	?:	ซ้าย ไป ขวา
15	= += -= /= %=	ขวา ไป ซ้าย
16	<<= >>=	ขวา ไป ซ้าย

โดยตัวดำเนินการที่มีลำดับน้อยกว่าจะดำเนินการก่อนตัวดำเนินการที่มีลำดับสูงกว่า เช่น

$X = 2 + 5 * 3$ จะมีลำดับการดำเนินการ คือ

ลำดับที่ 1 $5 * 3$ (เพราะ * มีลำดับเป็น 4 ส่วน + อยู่ลำดับ 5)

ลำดับที่ 2 $2 + 15$

ลำดับที่ 3 17 เป็นค่าของ X

2.6 นิพจน์ (expression)

เกิดจากการนำ ค่าคงที่ หรือ ตัวแปร และตัวดำเนินการมาประกอบกัน โดยนิพจน์มีหลายชนิด เช่นนิพจน์ทางคณิตศาสตร์ นิพจน์ทางตรรกะ นิพจน์ทางการเปรียบเทียบ นิพจน์เกี่ยวกับเงื่อนไข นิพจน์เกี่ยวกับข้อความ เช่น $2 + x * 5$ เป็นตัวอย่างของนิพจน์ทางคณิตศาสตร์ $(2 > 3) \&\& (5 >= 4)$ เป็นนิพจน์ทางตรรกะและการเปรียบเทียบ $!(a)$ เป็นนิพจน์ทางตรรกะ $if(x=y)$ เป็นนิพจน์เงื่อนไข

ตัวอย่าง การหาผลลัพธ์ของนิพจน์

กำหนด `int a = 10 , b = 12, c =3;` จงหาผลลัพธ์ของ $2 * a + b \% c$ ดำเนินการดังนี้ $(2 * a) + (b \% c)$ ผลลัพธ์ คือ $20 + 0$ คือ 20

จากข้อกำหนดด้านบน จงหาผลลัพธ์ ของ $(a > b) \&\& (c <= b)$ ซึ่งเราอาจพิจารณา ดังนี้ (เท็จ)&& (จริง) ดังนั้น ผลลัพธ์ คือ เท็จ

จากความรู้ที่เรียนมาในใบความรู้ที่ 1 และ 2 ควรจะเขียนโปรแกรมเป็นภาษาซีได้บ้างแล้ว

ตัวอย่าง กำหนดให้ x y z เป็นจำนวนเต็ม จงเขียนโปรแกรม หา ค่าของ z ที่มีความสัมพันธ์กับ x y เป็นสมการทางคณิตศาสตร์ ดังนี้ $z = 5x + 10y$ โดย x มีค่าเป็น 6 ส่วน y มีค่า 7 และแสดงผลทางจอภาพ โปรแกรม อาจเป็นทำนองนี้

```
/* program name 2_1_1.c */
```

```
#include <stdio.h>
```

```
int x = 6,y=7,z;
```

```
main()
```

```
{
```

```
    z = (5 * x) +( 10 * y);
```

```
    printf("value of x = %d \t value of y = %d \t value of z = %d",x,y,z);
```

```
}
```

หรือ

```
/* program name 2_1_1.c */
```

```
#include <stdio.h>
```

```
int x = 6,y=7,z;
```

```
main()
```

```
{
    printf("value of x = %d \t value of y = %d \t value of z = %d",x,y,5*x+10*y);
}
```

ระบบไฮดรอลิกส์ (Hydraulic System)

ระบบไฮดรอลิกส์มีใช้งานอย่างแพร่หลายในปัจจุบัน ระบบไฮดรอลิกส์สามารถควบคุมอุปกรณ์ขนาดเล็กขนาดกลาง หรือเครื่องจักรกลอัตโนมัติ ที่มีขนาดปานกลาง ไปจนถึงขนาดใหญ่ หรืองานที่ต้องการกำลังมาก (HeavyLoad) เช่น เครื่องปั๊มขึ้นงาน (Press Machine) เครื่องฉีดพลาสติก (Injection Molding) การประกอบชิ้นส่วนอัตโนมัติ (Automatic Assembly) หรือหุ่นยนต์อุตสาหกรรม (Industrial Robot) เป็นต้น โดยทั่วไปแล้วการทำงานของระบบไฮดรอลิกส์ จะเหมือนกับระบบนิวแมติกส์ คือจะมีการควบคุมทิศทางการทำงาน ส่วนอุปกรณ์และวงจรอื่นๆจะเหมือนกัน แต่จะแตกต่างตรงที่ระบบนิวแมติกส์ จะใช้ลมเป็นตัวกลางในการส่งถ่ายพลังงาน โดนปกติจะใช้ความดันที่ 6-8 bar และลมจะถูกปล่อยทิ้งสู่บรรยากาศ ส่วนระบบ ไฮดรอลิกส์ จะใช้น้ำมันเป็นตัวกลางในการส่งถ่ายพลังงานโดยปกติจะใช้ความดันที่ 30-400 bar แต่น้ำมันจะไหลกลับสู่ถังเก็บน้ำมันเหมือนเดิม ตัวกลางระหว่างลมกับน้ำมันนั้น น้ำมันจะมีการยุบตัวน้อยกว่าระบบลม จึงทำให้ระบบไฮดรอลิกส์ มีการทำงานที่ถูกต้อง และแม่นยำกว่าระบบ

1. ระบบไฮดรอลิกส์

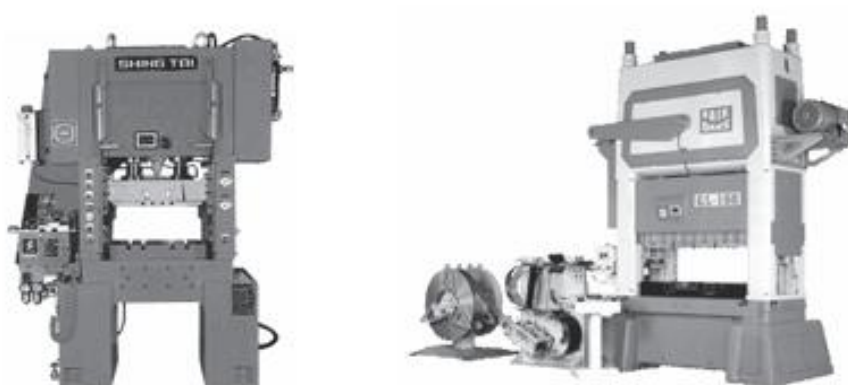
ระบบไฮดรอลิกส์ เป็นระบบที่ใช้อยู่ในเครื่องจักร และเครื่องทุ่นแรงหลายประเภท โดยทั่วไประบบไฮดรอลิกส์จะใช้กับงานที่มีลักษณะที่ต้องการแรงดึง แรงดัน แรงกด หรือแรงยก ที่ใช้แรงมาก ๆ ดังนั้นอุปกรณ์ที่ใช้ในระบบไฮดรอลิกส์จึงมีความหลากหลาย ใ้การใช้งานแต่ละประเภท ถึงแม้ว่าสัญลักษณ์การทำงานจะเหมือนกันแต่โครงสร้างในการทำงานจะแตกต่างกันไปตามความจำเป็นของงาน เครื่องจักรที่ใช้ระบบไฮดรอลิกส์สำหรับควบคุมการทำงาน สามารถแบ่งออกได้ดังนี้ คือ

1) เครื่องจักรที่อยู่กับที่ จะถูกออกแบบมาให้ประหยัดพลังงาน ดังนั้นจะเลือกใช้อุปกรณ์การทำงานของระบบไฮดรอลิกส์ ที่มีขนาดใหญ่ แต่ใช้แรงดันในระบบต่ำ จึงทำให้ประหยัดพลังงาน

2) เครื่องจักรที่เคลื่อนที่ จะถูกออกแบบให้มีน้ำหนักเบา เพื่อสะดวกในการเคลื่อนที่ ดังนั้นจะเลือกใช้อุปกรณ์การทำงานของระบบไฮดรอลิกส์ที่มีขนาดเล็ก แต่ใช้แรงดันในระบบสูง ทำให้ได้กำลังในการทำงานสูงด้วย

2. ลักษณะของระบบไฮดรอลิกส์ สำหรับระบบไฮดรอลิกส์ที่ใช้ในงานอุตสาหกรรมทั่วไปที่ใช้ในการควบคุมเครื่องจักรกลอัตโนมัติ สามารถแบ่งออกตามลักษณะการใช้งานดังนี้

2.1 ไฮดรอลิกส์อุตสาหกรรม (Industrial Hydraulic) ไฮดรอลิกส์อุตสาหกรรม คือ ไฮดรอลิกส์ที่ใช้ควบคุมเครื่องจักรในงานอุตสาหกรรม โดยทั่วไปใช้ในเครื่องจักรชนิดอยู่กับที่ เช่น เครื่องฉีดพลาสติก (Injection Molding) เครื่องปั๊มชิ้นงาน (Press Machine) เครื่องจักร CNC (CNC Machine) เป็นต้น



รูปที่ 2.3 เครื่องปั๊มชิ้นงาน (Press Machine)



รูปที่ 2.4 เครื่องพับ-ตัดชิ้นงาน (Cutting Machine)

2.2 ไฮดรอลิกส์โมบายล์ (Mobile Hydraulic) ไฮดรอลิกส์โมบายล์ คือ ไฮดรอลิกส์ที่ใช้ควบคุมเครื่องจักรในงานอุตสาหกรรม โดยทั่วไปใช้ในเครื่องจักรที่มีการเคลื่อนที่ โดยส่วนใหญ่จะมีการใช้งานในภาคสนาม เช่น รถไถ่ปูน รถเกรดถนน รถขุด รถบดถนน หรือรถเครนยกของบนตึกสูง (Tower Crane) เป็นต้น



รูปที่ 2.5 รถตักดิน



รูปที่ 2.6 รถยกของ

2.3 ไฮดรอลิกส์เทคโนโลยี (Technology Hydraulic) ไฮดรอลิกส์เทคโนโลยีคือ ไฮดรอลิกส์ที่ใช้ควบคุมเครื่องจักรในงานอุตสาหกรรมหนัก หรืออุตสาหกรรมที่ต้องการเทคโนโลยีขั้นสูง (Advanced Technology) ตลอดจนมีประสิทธิภาพและให้กำลังงานสูงด้วย เช่นเครื่องจักรขุดลอกแม่น้ำหรือลำคลอง (Dredge Industry) เครื่องจักรขุดเจาะน้ำมัน (Offshore Industry) หรือเครื่องจักรเทอร์ไบน์ (Turbine Industry) เป็นต้น



รูปที่ 2.7 เครื่องจักรขุดลอกแม่น้ำหรือลำคลอง (Dredge Industry)

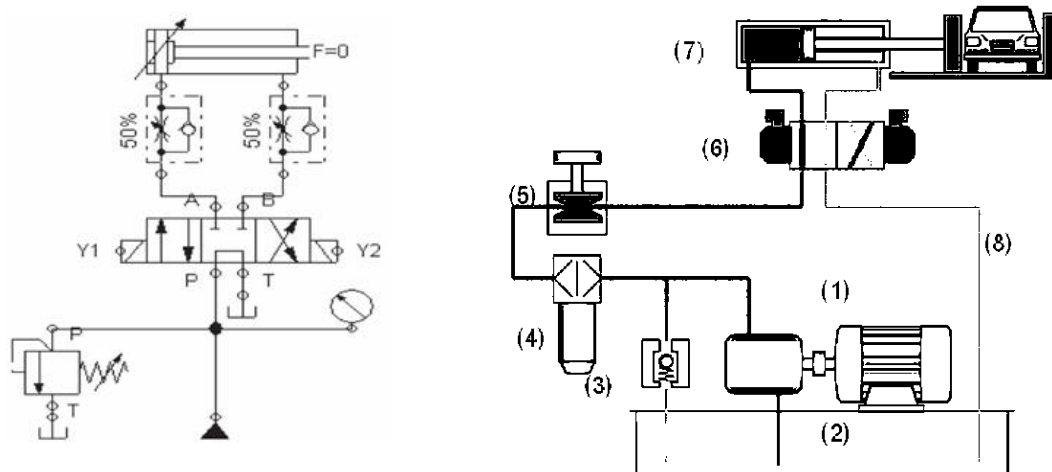


รูปที่ 2.8 เครื่องจักรขุดเจาะน้ำมัน (Offshore Industry)

3. ระบบไฮดรอลิกส์ในงานอุตสาหกรรม

ปัจจุบันระบบไฮดรอลิกส์ถูกนำมาใช้ในงานอุตสาหกรรมอย่างแพร่หลาย ซึ่งหลักการของไฮดรอลิกส์จะใช้น้ำมัน เป็นตัวกลางในการส่งผ่าน จากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยน้ำมันจะเปลี่ยนพลังงานการไหลให้เป็นพลังงานกล เช่น เครื่องปั๊มขึ้นงาน (Press machine) ในระบบของ

เครื่องปั๊มชิ้นงาน จะมีการส่งถ่ายกำลังการไหลของน้ำมันไฮดรอลิกส์ไปยังอุปกรณ์ ทำให้เครื่องจักรสามารถปั๊มชิ้นงานหรือขึ้นรูปชิ้นงาน



รูปที่ 2.9 วงจรไฮดรอลิกส์

ระบบไฮดรอลิกส์ สามารถแบ่งออกตามลักษณะการทำงานได้ดังนี้

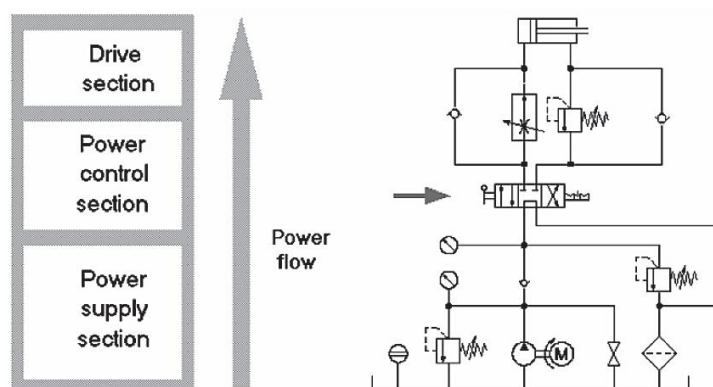
1. ปั๊มไฮดรอลิกส์ (Hydraulic Pump) มีหน้าที่ควบคุมการจ่ายน้ำมันและดูดน้ำมันในระบบ
2. ถังพักน้ำมัน (Tank) มีหน้าที่พักน้ำมันในระบบ ตลอดจนทำความสะอาดน้ำมันและระบายความร้อนน้ำมัน
3. วาล์วควบคุมน้ำมัน (Pressure relief valve) หรือวาล์วนิรภัย มีหน้าที่ควบคุมความดันในระบบ ขณะที่ระบบมีความดันสูงเพิ่มขึ้น วาล์วจะควบคุมให้น้ำมันไหลกลับสู่ถังน้ำมัน
4. ชุดกรองน้ำมัน (Filter) มีหน้าที่กรองสิ่งสกปรกไม่ให้เข้าไปในระบบ ซึ่งถ้ามีสิ่งสกปรกเข้าไปในระบบ อาจจะทำให้อุปกรณ์เสียหายได้
5. วาล์วปิด-เปิด (Shut off valve) มีหน้าที่ปิด-เปิด การไหลของน้ำมัน
6. โซลินอยด์วาล์ว (Solenoid valve) มีหน้าที่ควบคุมทิศทางการไหลของน้ำมัน
7. อุปกรณ์การทำงาน (Actuator) มีหน้าที่เปลี่ยนพลังงานการไหลให้เป็นพลังงานกล เช่น ลูกสูบ (piston) มอเตอร์ (motor) เป็นต้น
8. ท่อน้ำมัน (Tube) มีหน้าที่ส่งผ่านน้ำมันในระบบทั้งหมด

4. การควบคุมระบบไฮดรอลิกส์

การควบคุมระบบไฮดรอลิกส์มีการพัฒนาไปอย่างรวดเร็ว เพื่อให้ทันกับเทคโนโลยีที่ทันสมัย ในปัจจุบันอุตสาหกรรม ระบบไฮดรอลิกส์ได้มีการพัฒนานำเทคโนโลยีมาควบคุม เพื่อให้ได้ประสิทธิภาพสูงสุด ตลอดจนถูกต้องและแม่นยำ โดยแบ่งวงจรการควบคุมออกเป็น 2 ชนิด คือ วงจรกำลัง (Circuit Power) และวงจรควบคุม (Circuit Control) โดยวงจรกำลัง จะมีน้ำมันเป็นตัวกลางในการส่งกำลัง เพื่อควบคุมวงจรการทำงาน ส่วนวงจรควบคุม จะใช้ตัวควบคุมหลายอย่าง เช่น Electro, PLC, Proportional, Servo, Microcontroller เป็นต้น การควบคุมระบบไฮดรอลิกส์สามารถแบ่งออกตามลักษณะการใช้งานได้ดังนี้

4.1 การควบคุมระบบไฮดรอลิกส์ด้วยน้ำมัน ระบบไฮดรอลิกส์ใช้น้ำมัน

เป็นตัวกลางในการส่งถ่ายกำลัง โดยระบบของไฮดรอลิกส์ จะประกอบไปด้วยอุปกรณ์ต้นกำลัง (Motor, Hydraulic Pump) กรองน้ำมัน (Filter) วาล์วควบคุมความดัน (Pressure relief valve) เกจวัดความดัน (Pressure gauge) วาล์วควบคุมทิศทาง (Solenoid valve) ลูกสูบ (Piston) เป็นต้น



รูปที่ 2.10 วงจรควบคุมไฮดรอลิกส์ด้วยน้ำมัน

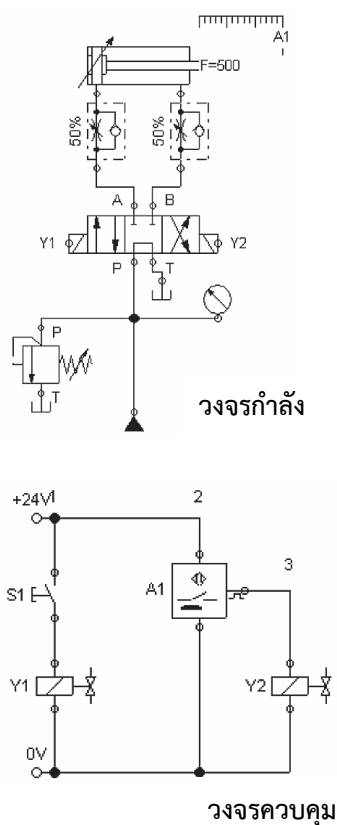
4.2 การควบคุมระบบไฮดรอลิกส์ด้วยไฟฟ้า การควบคุมระบบไฮดรอลิกส์ด้วย

ไฟฟ้า จะแบ่งวงจรการควบคุมออกเป็น 2 ชนิด คือ

1) วงจรกำลัง (Circuit Power) เป็นวงจรควบคุมการทำงานของระบบไฮดรอลิกส์ ประกอบด้วย อุปกรณ์ต้นกำลัง (Power source) ชุดกรองน้ำมัน (Filter) โซลีนอยด์วาล์ว (Solenoid valve) กระบอกสูบ (Cylinder) เป็นต้น

2) วงจรควบคุม (Circuit control) เป็นวงจรควบคุมการทำงานของ

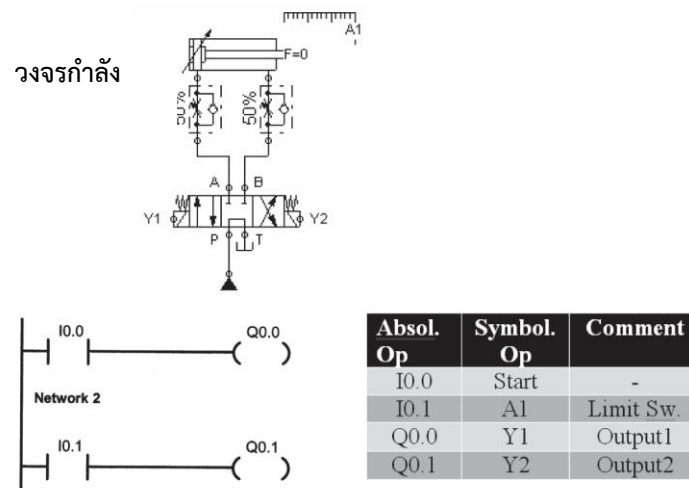
ระบบไฮดรอลิกส์ไฟฟ้า ประกอบด้วยสวิตช์ปุ่มกด (Pushbutton switch) รีเลย์ (Relay) ตัวตั้งเวลา (Timer relay) ตัวนับเวลา (Counter relay) เซ็นเซอร์ (Sensor) เป็นต้น



รูปที่ 2.11 วงจรควบคุมระบบไฮดรอลิกส์ด้วยไฟฟ้า

4.3 การควบคุมระบบไฮดรอลิกส์ ด้วยโปรแกรมเมเบิล ลอจิก คอนโทรลเลอร์ (PLC) ปัจจุบัน PLC (Programmable Logic Controller) ได้เข้ามามีบทบาทในการควบคุมระบบไฮดรอลิกส์ในงานอุตสาหกรรมต่าง ๆ เช่น อุตสาหกรรมรถยนต์ อุตสาหกรรมอาหาร หรืออุปกรณ์ต่าง ๆ ในงานอุตสาหกรรมโดยภายใน PLC จะมีฟังก์ชันและอุปกรณ์ควบคุมภายในมากมาย เช่น Relay, Timer, Counter เป็นต้น ทำให้การเรียกใช้หรือแก้ไขโปรแกรมสะดวกรวดเร็ว และลดค่าใช้จ่ายลงอีกด้วย สำหรับภาษาที่ใช้ในการเขียนโปรแกรม PLC มีให้เลือกใช้งานได้หลายภาษา ดังนี้

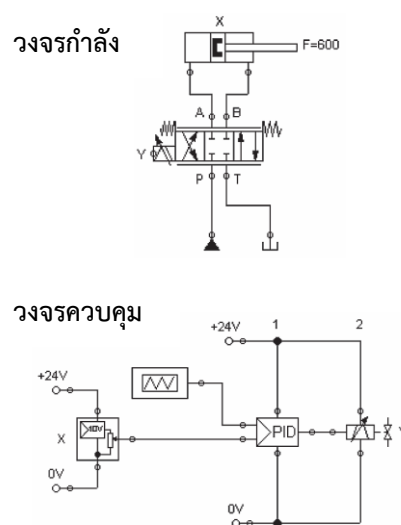
- 1) ภาษาบูลิเนียน (STL : Instruction List Boolean Logic Element)
- 2) ภาษาแลดเดอร์ (Ladder Diagram)
- 3) ภาษาบล็อก (Function Chart)



รูปที่ 2.12 วงจรควบคุมระบบไฮดรอลิกส์ด้วย PLC

4.4 การควบคุมระบบไฮดรอลิกส์ ด้วยพรอพอร์ชันนัลวาล์ว (Proportional valve) การควบคุมระบบไฮดรอลิกส์ด้วย Proportional valve เป็นระบบที่นิยมมากในปัจจุบัน เพราะสามารถควบคุมลูกสูบให้หยุดตามตำแหน่งที่ต้องการ โดยสามารถควบคุมให้หยุดทุกระยะของการทำงานได้พรอพอร์ชันนัลวาล์ว (Proportional valve) สำหรับควบคุมระบบไฮดรอลิกส์ มีหลายชนิดดังนี้

- 1) พรอพอร์ชันนัลวาล์วควบคุมตำแหน่ง (Directional control valve)
- 2) พรอพอร์ชันนัลวาล์วควบคุมแรงดัน (Pressure valve)
- 3) พรอพอร์ชันนัลวาล์วควบคุมอัตราการไหล (Flow control valve)

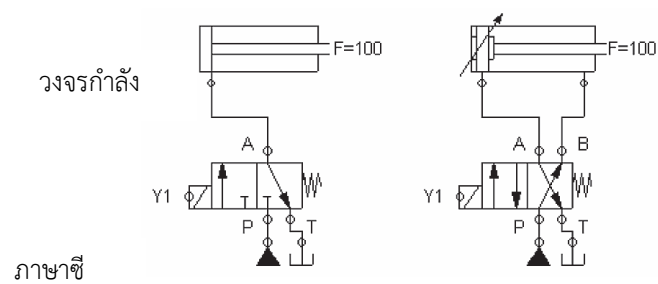


รูปที่ 2.13 วงจรควบคุมระบบไฮดรอลิกส์ด้วย พรอพอร์ชันนัลวาล์ว

4.5 การควบคุมระบบไฮดรอลิกส์ ด้วยไมโครคอนโทรลเลอร์ (Microcontroller)

การใช้ Microcontroller ควบคุมระบบไฮดรอลิกส์ในปัจจุบัน ยังไม่นิยมมากนัก (แต่นิยมใช้ควบคุมเครื่องจักร และระบบการผลิตในอุตสาหกรรม) เพราะต้องศึกษาหลายอย่าง และต้องรู้เกี่ยวกับขบวนการทำงานอย่างแท้จริง เช่น สถาปัตยกรรมของ Microcontroller โดยประกอบด้วยหน่วยประมวลผลกลาง หน่วยความจำอินพุต/เอาต์พุต โปรแกรมวงจรรีเลย์ทรอนิกส์ เป็นต้น โดยทั่วไป Microcontroller จะประกอบด้วย Timer, Counter, I/O, Serial port เป็นต้น ในปัจจุบันมีหลายบริษัทที่ผลิต Microcontroller เช่น Intel, Phillips, Motorola, Microchip เป็นต้น สำหรับภาษาที่ใช้ในการเขียนสั่งงานควบคุม Microcontroller สามารถเขียนได้หลายภาษา ดังนี้

- 1) ภาษาเบสิก (Basic Language)
- 2) ภาษาแอสเซมบลี (Assembly language)
- 3) ภาษาซี (C language)



```

1 #include "D:\data_pic\hydraulic_book\ex1.h"
2 #define start pin_a0
3 #define y1 pin_b7
4 //=====
5 void main()
6 {
7   setup_adc_ports(NO_ANALOGS);
8   setup_adc(ADC_OFF);
9   setup_psp(PSP_DISABLED);
10  setup_spi(SPI_SS_DISABLED);
11  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
12  setup_timer_1(T1_DISABLED);
13  setup_timer_2(T2_DISABLED,0,1);
14  output_high (start);
15  output_low (y1);
16  while(1)
17  {
18    if(!input(start))
19    {
20      delay_ms(100);
21      output_high(y1);
22      while(!input(start));
23    }
24    if(input(start))
25    {
26      delay_ms(100);
27      output_low(y1);
28      while(input(start)); }

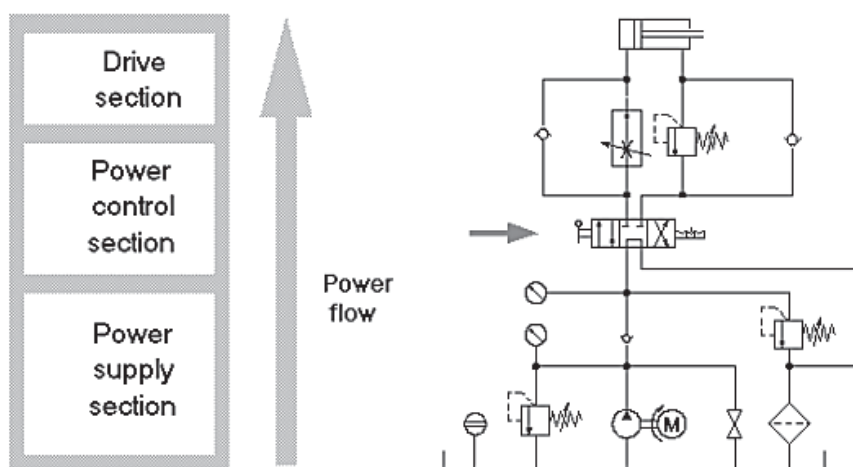
```

รูปที่ 2.14 วงจรควบคุมระบบไฮดรอลิกส์ด้วย ตัวไมโครคอนโทรลเลอร์

5. อุปกรณ์ทำงานในระบบไฮดรอลิกส์

ระบบไฮดรอลิกส์ (Hydraulic System) คือ การนำเอาอุปกรณ์ระบบไฮดรอลิกส์หลายส่วนมาทำงานร่วมกัน เช่น ส่วนที่ 1 ปั๊มไฮดรอลิกส์ (Hydraulic Pump) จะทำหน้าที่เปลี่ยนกำลังงานกลไปเป็นกำลังงานไฮดรอลิกส์ ส่วนที่ 2 วาล์วควบคุม (Control Valve) จะทำหน้าที่ควบคุมทิศทางควบคุมแรงดัน และควบคุมอัตราการไหลในระบบไฮดรอลิกส์ ส่วนที่ 3 กระบอกสูบ (Cylinder) จะทำหน้าที่เปลี่ยนกำลังงานไฮดรอลิกส์ให้เป็นกำลังงานกล โดยทั้ง 3 ส่วนจะทำงานและส่งถ่ายกำลังแบบต่อเนื่องหรือหยุดชั่วขณะได้อย่างมีประสิทธิภาพ โดยทั่วไปองค์ประกอบของระบบไฮดรอลิกส์ที่ใช้กันในปัจจุบันจะประกอบไปด้วย 2 ส่วน คือ ส่วนกำลัง และ ส่วนควบคุม ซึ่งจะขออธิบายหลักการของทั้ง 2 ส่วน ดังต่อไปนี้

1. ส่วนกำลัง (Power) ส่วนกำลังจะประกอบไปด้วยระบบกำลังหรือส่งถ่ายพลังงาน (Power Supply) ระบบควบคุมทิศทางการไหล (Direction Control) ระบบควบคุมอัตราการไหล (Flow Control) ระบบควบคุมแรงดัน (Pressure Control) ในการทำงานและระบบเปลี่ยนพลังงาน (Transferring Energy) เป็นต้น



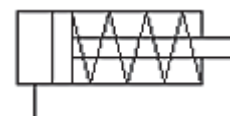
รูปที่ 2.15 ไดอะแกรมการทำงานของส่วนกำลัง

ส่วนกำลังนี้ประกอบด้วย 3 ส่วนด้วยกัน คือ

1.1 ส่วนการขับเคลื่อน (Drive section) โดยทั่วไปแล้วจุดการขับเคลื่อนในการส่งกำลังก็คือตัวทำงาน (Actuators) โดยสามารถแบ่งออกได้ หลายชนิด ดังนี้

1) กระบอกสูบทางเดียว (Single Acting Cylinder) กระบอกสูบทาง

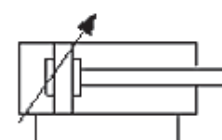
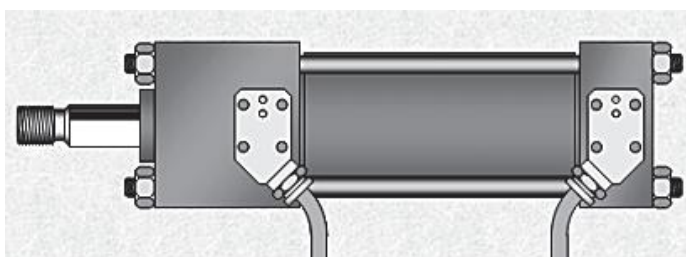
เดี่ยว (Single Acting Cylinder) จะมีตำแหน่งรูให้น้ำมันเข้าด้านเดียว เมื่อน้ำมันไหลเข้าไปภายใน กระบอกสูบจะเปลี่ยนพลังงานของไหลให้เป็นพลังงานกลหรือการขับเคลื่อน ซึ่งลูกสูบจะเคลื่อนที่เข้าหรือเร็วขึ้นอยู่กับอัตราการไหลของน้ำมันด้วย และลูกสูบจะเคลื่อนที่กลับด้วยแรงของสปริง (Spring)



สัญลักษณ์

รูปที่ 2.16 กระบอกสูบทำงานทางเดียว

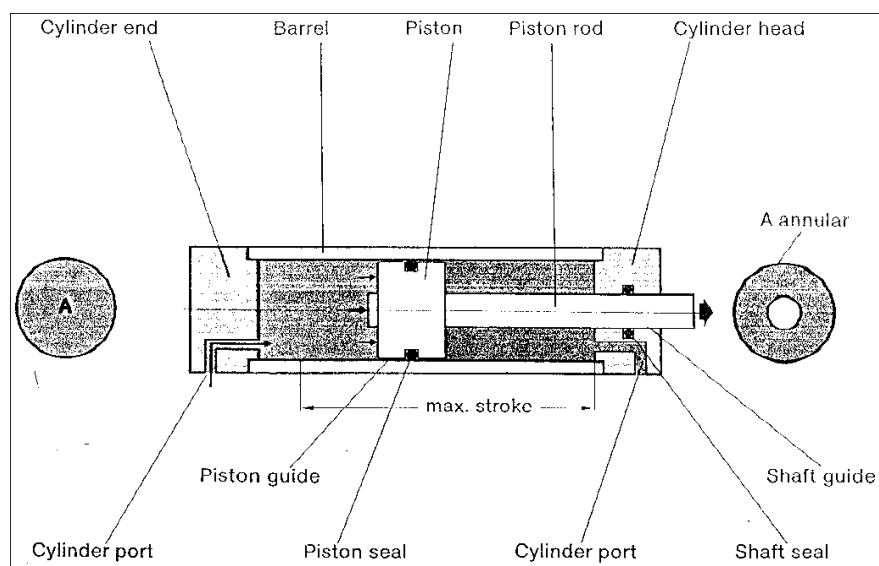
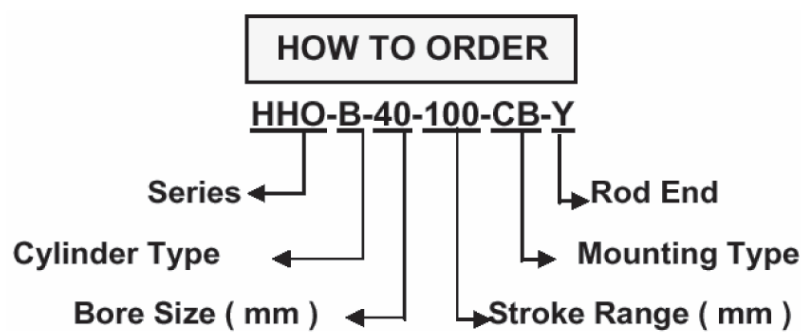
2) กระบอกสูบสองทาง (Double Acting Cylinder) กระบอกสูบสองทาง (Double Acting Cylinder) จะมีตำแหน่งรูให้น้ำมันเข้าสองด้านคือด้านลูกสูบ เคลื่อนที่ออกและด้านลูกสูบเคลื่อนที่เข้า การเคลื่อนที่ของลูกสูบจะอาศัยแรงดันน้ำมันในการเคลื่อนที่



สัญลักษณ์

รูปที่ 2.17 กระบอกสูบทำงานสองทาง

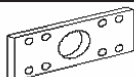
3) ขนาดของกระบอกสูบ (Dimension of Cylinder) โดยทั่วไปกระบอกสูบไฮดรอลิกส์จะมีความสัมพันธ์ระหว่างขนาดของกระบอกสูบ (Bore Size) กับขนาดช่วงชัก (Stroke Range) เพื่อให้มีความสมดุลและพอเพียงกับ ความดัน (Pressure) และอัตราการไหล (Flow rate) ที่ใช้งาน สำหรับในหน่วยนี้จะขอยกตัวอย่าง ขนาดของกระบอกสูบ ความดันและมาตรฐานของกระบอกสูบ

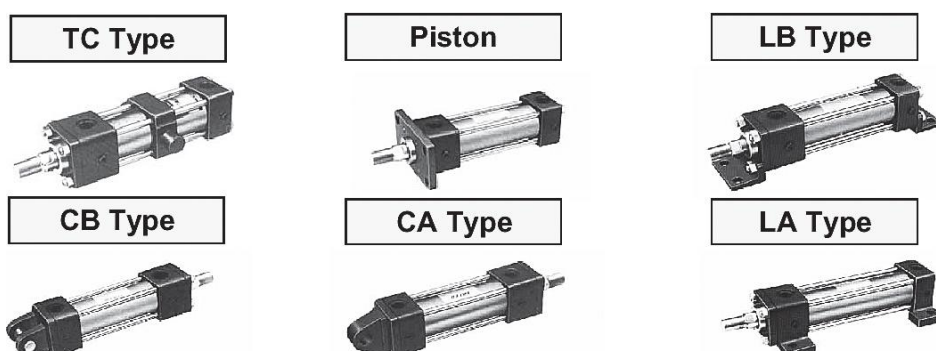


รูปที่ 2.18 รหัสและโครงสร้างของกระบอกสูบ

ตารางที่ 2.8 ขนาดของกระบอกสูบ

Series	Pressure (bar)	Bore Size (mm)	Stroke Range (mm)	Mounting Type	Rod End Type
MO	70	40,50,63		FA,FB,LB	Type I
HO	140	80,100,125	Max 3000	TC,CA,CB	Type Y
HHO	210	150,200,250			

Mounting Style & Rod End			
Type	Pic	Type	Pic
FA		CA	
FB		CB	
LB		Y	
TC		I	



รูปที่ 2.19 ชนิดของกระบอกสูบ

- พื้นฐานการคำนวณทางฟิสิกส์

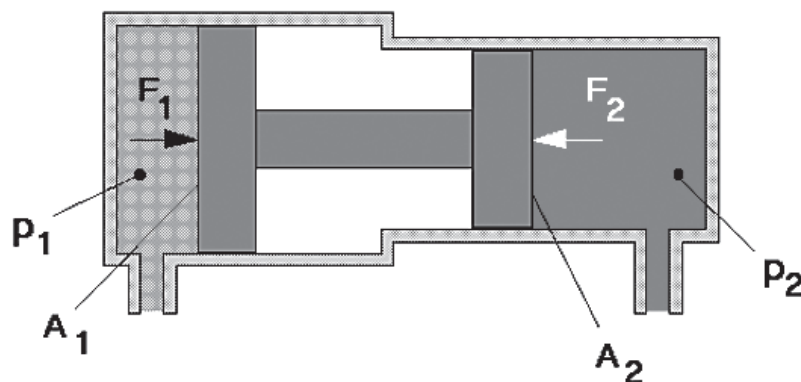
การคำนวณของไหลในระบบจะประกอบไปด้วย ความดัน อัตราการไหล พลังงาน แรง เป็นต้น โดยทั่วไปหน่วยวัดพื้นฐาน ได้แก่ ความยาว แรง และเวลา จะเป็นหน่วยอังกฤษ หรือเมตริก ซึ่งยังแยกหน่วยวัดตามการใช้งานจริงได้เป็นหน่วยวัดมวล เช่น ปอนด์มวล (lbm) และหน่วยวัดแรง (lbf) อีกด้วย ระบบเมตริกจะใช้ระบบสมบูรณ์

(Absolute) หรือระบบ C.G.S (Centimeter-gram-second) ซึ่งมีหน่วยวัดมวลเป็นกรัม และหน่วยวัดแรงเป็นไดน์ (dyne) ในระบบ SI (International System Units) มีหน่วยวัดมวลเป็น กิโลกรัม ความยาววัดเป็นเมตร และเวลาวัดเป็นวินาที หน่วยวัดแรงวัดเป็นนิวตัน (Newton)

$$1 \text{ N} = 1 \text{ kgm/sec} = 105 \text{ dynes}$$

โดยกำหนดให้

- หน่วยวัดความดันจะบอกเป็น N/m^2 หรือ (Pascal, Pa)
- หน่วยวัดพลังงาน (Energy) หรืองาน (Work) คือ จูล (Joule N-m)
- หน่วยวัดกำลัง (Power) คือ วัตต์ (Watts)
- หน่วยวัดความถี่ คือ เฮิร์ตซ (Hertz, Hz) ซึ่งมีค่าเป็นรอบต่อวินาที (cycle/second)



รูปที่ 2.20 ความสัมพันธ์ระหว่างความดัน แรง และพื้นที่

- การคำนวณหาแรงของกระบอกสูบ

จากกฎของปาสคาล (Pascal's Law) คือ ความดันที่กระทำต่อของไหลในภาชนะที่ปิดจะส่งต่อไปยังทุกส่วนของของไหล และผนังภาชนะที่บรรจุของไหล เช่น เครื่องยกไฮดรอลิกส์ ดังรูป เป็นตัวอย่างของการประยุกต์ใช้กฎของ Pascal's Law

สูตร $F = PA$

$$P_1 = \frac{F_1}{A_1} \quad \text{และ} \quad P_2 = \frac{F_2}{A_2}$$

ดังนั้น

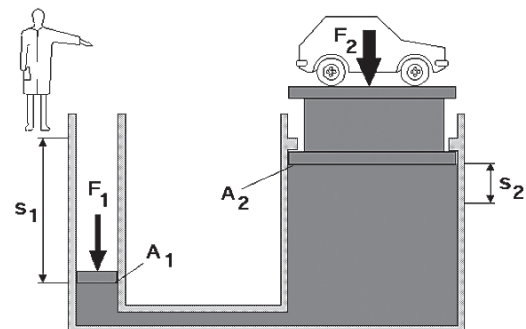
$$P_1 = P_2, \quad \frac{F_2}{F_1} = \frac{A_2}{A_1}$$

เนื่องจากการไหลไม่สามารถเพิ่มหรือลดได้ ดังนั้น ปริมาตรที่ลูกสูบเคลื่อนที่ลงด้านซ้ายจะเท่ากับ ปริมาตรที่ลูกสูบเคลื่อนที่ขึ้นทางขวานั้นเอง

$$S_1 A_1 = S_2 A_2$$

$$P_2 = P_1 * \frac{A_1}{A_2}$$

$$\frac{F_1}{F_2} = \frac{A_1}{A_2} = \frac{S_2}{S_1}$$



รูปที่ 2.21 หลักการประยุกต์การใช้กฎของ Pascal's Law

โดยกำหนดให้

F = แรงของกระบอกสูบ (lb,N)

P = ความดันของน้ำมัน (Psi, N/m²)

A = พื้นที่หน้าตัดของลูกสูบ (in²,m²)

S = ระยะทางลูกสูบ (in,m)

- การคำนวณหาอัตราการไหล

อัตราการไหล (Flow rate) คือ ปริมาตรของของเหลวที่มีการไหลผ่านพื้นที่หน้าตัดในหนึ่งหน่วยเวลา เช่นอัตราการไหล ลูกบาศก์เมตรต่อวินาที แกลลอนต่อนาทีหรือลิตรต่อนาที เป็นต้น

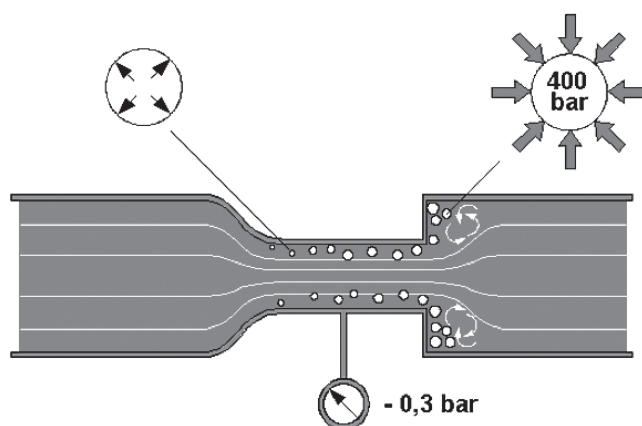
สูตร

$$Q = AV$$

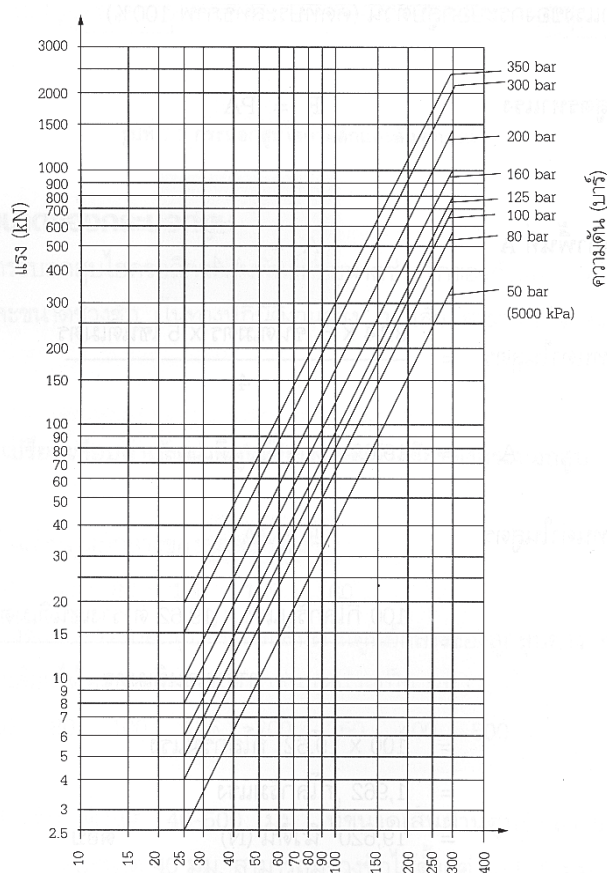
โดยกำหนดให้ Q = อัตราการไหลของน้ำมัน (ft³/sec, m³/sec)

V = ความเร็วในการไหล (ft/sec, m/sec)

A = พื้นที่หน้าตัดของลูกสูบ (ft², m²)



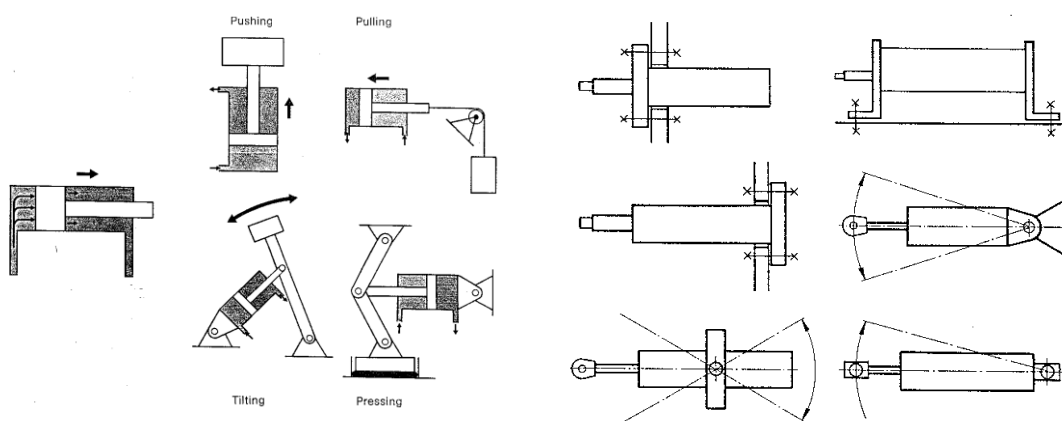
รูปที่ 2.22 แสดงการไหลในท่อ



รูปที่ 2.23 ตารางการหาขนาดเส้นผ่าศูนย์กลางการของลูกสูบ

- การติดตั้งกระบอกลูกสูบในงานอุตสาหกรรม

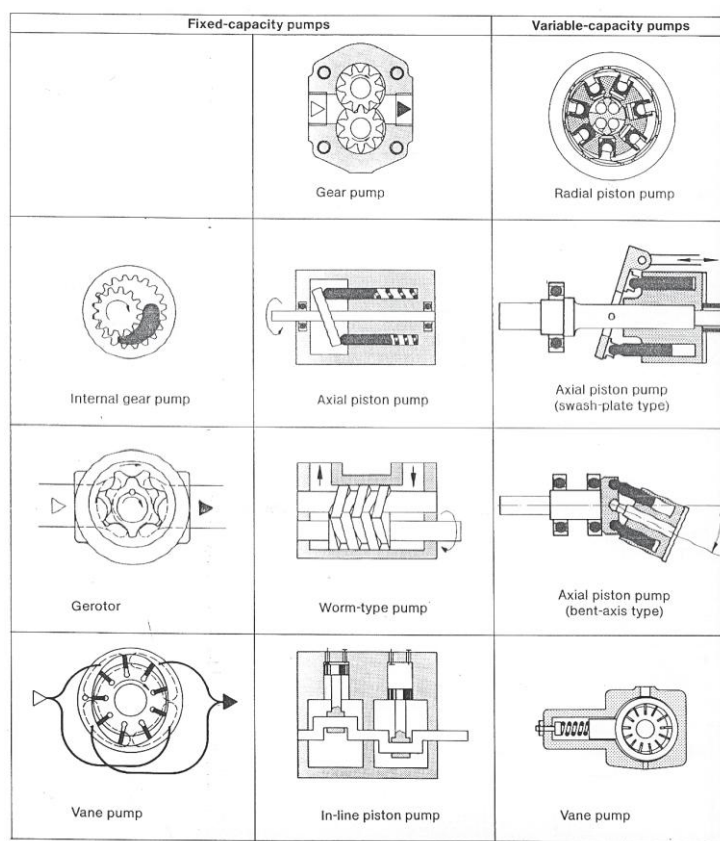
การติดตั้งกระบอกลูกสูบในงานอุตสาหกรรม สามารถติดตั้งได้หลายวิธีขึ้นอยู่กับลักษณะของงาน เช่น การติดตั้งกระบอกลูกสูบในแนวนอน แนวตั้ง แนวเอียง เป็นต้น



รูปที่ 2.24 การติดตั้งกระบอกลูกสูบในงานอุตสาหกรรม

- มอเตอร์ไฮดรอลิกส์ (Hydraulic Motor)

มอเตอร์ไฮดรอลิกส์ (Hydraulic Motor) จะมีหน้าที่เปลี่ยนพลังงานของไหลให้เป็นพลังงานกล คือเมื่อน้ำมันไหลเข้าทางด้านเข้า (suction) ของมอเตอร์และไหลออกทางด้านออก (Discharge) ของมอเตอร์เช่นกัน มีผลทำให้มีแรงดันสูงขึ้นในระบบ สำหรับแรงดันจะมากหรือน้อยขึ้นอยู่กับความเร็วของมอเตอร์ในการหมุนด้วยมอเตอร์ไฮดรอลิกส์ (Hydraulic Motor) และปั๊มไฮดรอลิกส์ (Hydraulic Pump) จะมีลักษณะโครงสร้างคล้าย ๆ กัน โดยสามารถแสดงได้ดังรูปที่ 2.25

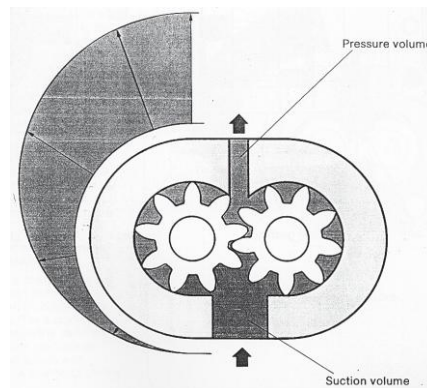
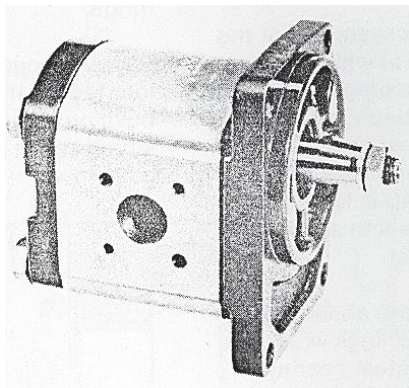


รูปที่ 2.25 โครงสร้างของมอเตอร์ไฮดรอลิกส์ชนิดต่าง ๆ

มอเตอร์ไฮดรอลิกส์สามารถแบ่งออกได้ตามลักษณะการใช้งานได้ดังนี้

- มอเตอร์ไฮดรอลิกส์แบบเฟือง (Gear Motor)

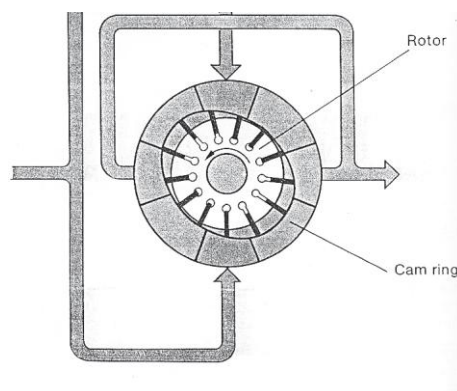
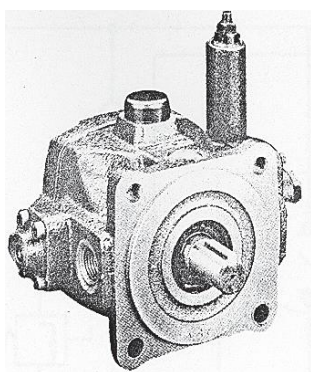
มอเตอร์ไฮดรอลิกส์แบบเฟืองจะประกอบไปด้วยเฟืองตรง 2 ตัว ขบกันเพื่อให้ น้ำมันไหลไปตามฟันเฟืองและไหลออกไปทางออก (Discharge) ลักษณะของฟันเฟืองจะมีช่องขบฟันทำให้เกิดสุญญากาศระบบจะดูดน้ำมันเข้ามาทางเข้า (Suction) อีกครั้ง การทำงานจะทำงานแบบนี้ไปเรื่อย ๆ ทุกรอบของการทำงาน



รูปที่ 2.26 มอเตอร์ไฮดรอลิกส์แบบเฟือง

- มอเตอร์ไฮดรอลิกส์แบบใบพัด

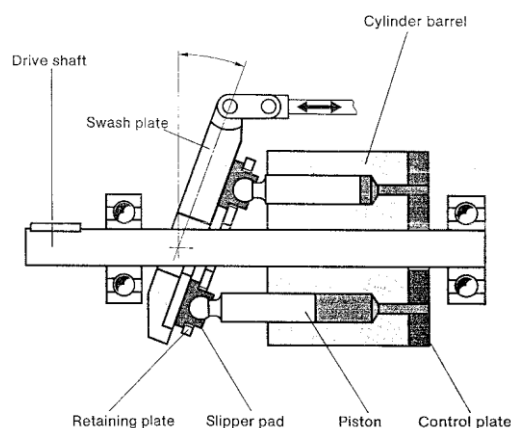
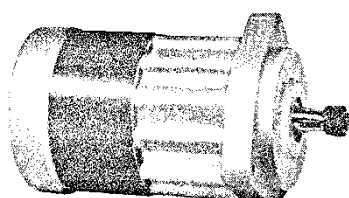
มอเตอร์ไฮดรอลิกส์แบบใบพัด สามารถหมุนกลับทิศทางได้ โดยจะประกอบไปด้วย โรเตอร์ (Rotor) ใบพัด (Vane) วงแหวนลูกเบี้ยว (Cam Ring) สปริง (Spring) ใบพัดจะมีขนาดไม่เท่ากันและโรเตอร์กับวงแหวนลูกเบี้ยวจะเอียงศูนย์กลางกัน เมื่อแกนเพลา (Shaft) หมุนจะทำให้มีช่องสุญญากาศดูดน้ำมันช่องทางเข้า (Suction) และใบพัดหมุนทำให้น้ำมันไหลออกช่องทางออก (Discharge)



รูปที่ 2.27 มอเตอร์ไฮดรอลิกส์แบบใบพัด

- มอเตอร์ไฮดรอลิกส์แบบลูกสูบ (Piston Motor)

มอเตอร์ไฮดรอลิกส์แบบลูกสูบ จะมีประสิทธิภาพในการทำงานสูงและสามารถทำงานที่ความเร็วรอบสูงได้โดยมอเตอร์ไฮดรอลิกส์แบบลูกสูบจะประกอบด้วย ลูกสูบ (Piston) เฟลาขับ (Drive Shaft) ชุดลูกสูบ (Cylinder Barrel) ชุดแผ่นเอียง (Swash Plate) เป็นต้น โดยปกติจะมีลูกสูบไม่น้อยกว่า 3 ลูก จะวางเอียงมุมกันเมื่อลูกสูบลูกที่ 1 เคลื่อนที่ลงมากระทบกับแผ่นเอียง จะเกิดแรงบิด (Torque) กับพื้นที่หน้าตัดทำให้เกิดการหมุน ขณะเดียวกันลูกสูบที่ 2 จะเคลื่อนที่ลงพร้อมกับลูกสูบที่ 1 จะเคลื่อนที่ขึ้น ตามลำดับ การทำงานจะเคลื่อนที่ขึ้นลงตลอดเวลา โดยเรียงลำดับลูกสูบ 1,2,3 ขึ้นลงตามลำดับ



รูปที่ 2.28 มอเตอร์ไฮดรอลิกส์แบบลูกสูบ

- การคำนวณหาแรงบิด (Torque)

แรงบิด (Torque) ของมอเตอร์ไฮดรอลิกส์ เกิดจากการหมุนของเฟลาทำให้เกิดงานเพื่อให้เกิดไหลเคลื่อนที่ด้วยความเร็วเพิ่มขึ้น

สูตร

$$T = F * r$$

เมื่อ

T = แรงบิดของเฟลา (in.lb, N.m)

F = น้ำหนักของชิ้นงาน (lb, N)

r = ระยะทางที่ไหลเคลื่อนที่ (in, m)

- การคำนวณหาแรงบิดจากมอเตอร์ไฮดรอลิกส์ (Hydraulic Motor Torque)

$$\begin{aligned}
 \text{สูตร} \quad T(lb.in) &= \frac{P(lb/in^2) * V_m(in^3/rev)}{2\pi} \\
 &= \frac{P(lb/in^2) * V_m(in^3/rev)}{6.28} \\
 T(N.m) &= \frac{P(bar) * V_m(cm^3/rev)}{2\pi * 10} \\
 &= \frac{P(bar) * V_m(cm^3/rev)}{62.8}
 \end{aligned}$$

เมื่อ

T = แรงบิดจากมอเตอร์ไฮดรอลิกส์ ($lb.in, N.m$)

P = ความดัน (psi, pa)

V_m = ปริมาตรของมอเตอร์ไฮดรอลิกส์ ($in^3/rev, cm^3/rev$)

การคำนวณหาอัตราการไหล (Flow rate)

$$\begin{aligned}
 \text{สูตร} \quad GPM &= \frac{V_m(in^3/rev) * RPM}{231 * \eta_m}
 \end{aligned}$$

เมื่อ

GPM = อัตราการไหลของน้ำมัน ($litre/min$)

V_m = ปริมาตรของมอเตอร์ไฮดรอลิกส์ (in^3/rev)

RPM = ความเร็วรอบในการหมุน (rev/min)

η_m = ประสิทธิภาพของมอเตอร์ไฮดรอลิกส์

$$\text{สูตร} \quad P(Psi, lb^2/in^2) = \frac{2\pi * T(lb.in)}{V_m(in^3/rev)}$$

1.2 ส่วนควบคุมระบบกำลัง (Power Control System) คือส่วนที่กำหนดหน้าที่ควบคุมตัวทำงาน (Actuator) ของภาคกำลัง โดยจะควบคุมการทำงานของส่วนส่งสัญญาณอินพุต (Input signal) และส่วนส่งสัญญาณเอาต์พุต (Output Signal) เป็นต้น ส่วนควบคุมที่ใช้ในระบบไฮดรอลิกส์สามารถแบ่งตามลักษณะการใช้งานได้ดังนี้

วาล์วสั่งงานด้วยมือกด (Manual valve) วาล์วสั่งงานชนิดนี้ จะทำหน้าที่ส่งสัญญาณตัวแรกของวงจรทำงาน เมื่อมีสัญญาณภายนอกเข้ามา จะทำให้วาล์วทำงานเปลี่ยนทิศทางหรือตำแหน่งในการทำงาน วาล์วชนิดนี้สามารถแบ่งออกได้หลายชนิด เช่น วาล์วสั่งงานแบบใช้กล้ามเนื้อหรือมือกด (Manually operated) วาล์วสั่งงานแบบใช้ลูกกลิ้ง (Cam-operated) หรือวาล์วสั่งงานแบบใช้เท้าเหยียบ (Pedal-operated) เป็นต้น



รูปที่ 2.29 วาล์วสั่งงานด้วยมือกดชนิดต่าง ๆ

วาล์วสั่งงานด้วยแรงดันไฮดรอลิกส์ (Hydraulic valve with pilot operated)

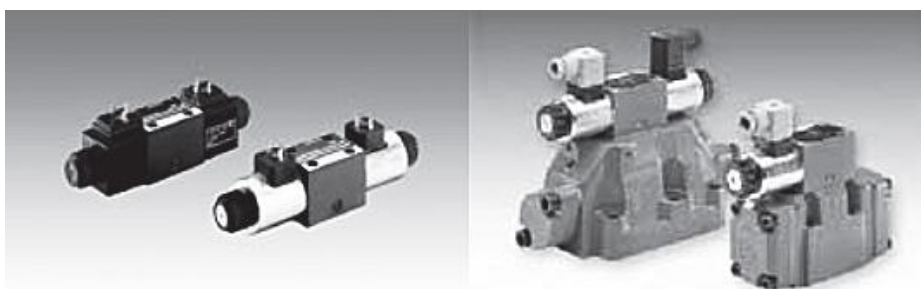
วาล์วสั่งงานชนิดนี้จะทำหน้าที่ส่งสัญญาณถัดมาหรือส่งสัญญาณตัวสุดท้ายของระบบ เพื่อเปลี่ยนตำแหน่งหรือทิศทางให้ลูกสูบเคลื่อนที่เข้าหรือออกนั่นเอง วาล์วชนิดนี้มีใช้งานอยู่หลายชนิด เช่น วาล์ว 2/2 , วาล์ว 3/2 , และวาล์ว 4/3 เป็นต้น



รูปที่ 2.30 วาล์วสั่งงานด้วยแรงดันไฮดรอลิกส์

วาล์วสั่งงานด้วยไฟฟ้า (Solenoid valve) วาล์วสั่งงานชนิดนี้ จะถูกควบคุมด้วยสัญญาณไฟฟ้า เพื่อเปลี่ยนตำแหน่งหรือทิศทางของวาล์ว และมีผลทำให้ลูกสูบเคลื่อนที่เข้าหรือออก หรือควบคุมการทำงานของมอเตอร์ให้หมุนตามทิศทางที่ต้องการ วาล์วสั่งงานด้วย

ไฟฟ้าที่ใช้อยู่ในปัจจุบันมีอยู่หลายชนิด เช่น 3/2, 4/2, 4/3, (Single or Double Solenoid valve) เป็นต้น

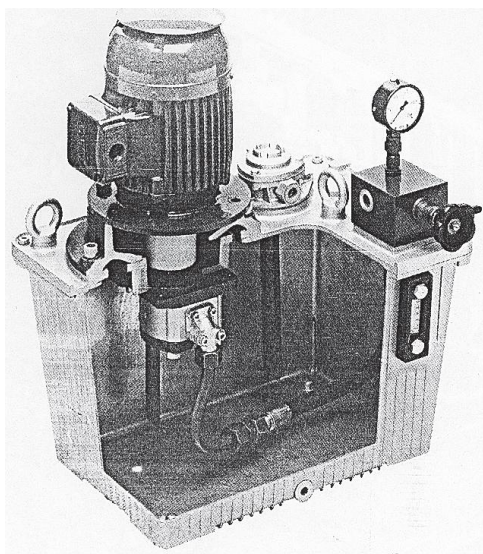


รูปที่ 2.31 วาล์วสั่งงานด้วยไฟฟ้า

1.3 ส่วนต้นกำลัง (Power Supply Section) ส่วนต้นกำลังเป็นส่วนที่สำคัญมากในระบบ โดยจะทำหน้าที่จ่ายพลังงานไฮดรอลิกส์ (Hydraulic Power) เพื่อขับเคลื่อนตัวทำงาน (Actuators) ให้เคลื่อนที่ ส่วนต้นกำลังประกอบด้วยหลาย ส่วน เช่น ปั๊มไฮดรอลิกส์ (Hydraulic Pump) ชุดเก็บน้ำมัน (Oil reservoirs) ชุดกรองน้ำมัน (Filter) เกจวัดแรงดัน (Pressure gauge) วาล์วกำจัดแรงดัน (Pressure-relief valve)

ปั๊มไฮดรอลิก (Hydraulic Pumps) ปั๊มไฮดรอลิก จะทำหน้าที่เปลี่ยนพลังงานกลให้เป็นพลังงานของไหล หรือทำหน้าที่ดูดน้ำมันทางขาเข้า(Suction) และเพิ่มความดันให้ระบบ โดยออกทางขาออก (Discharge) เพื่อให้ขับเคลื่อนตัวทำงาน (Actuators) ในระบบต่อไป จากที่ได้กล่าวมาแล้ว ปั๊มไฮดรอลิก (Hydraulic Pumps) จะทำหน้าที่คล้ายกับมอเตอร์ไฮดรอลิก (Hydraulic motor) ปั๊มไฮดรอลิกที่ใช้ในปัจจุบันสามารถแบ่งออกได้ตามลักษณะการใช้งานได้ดังนี้

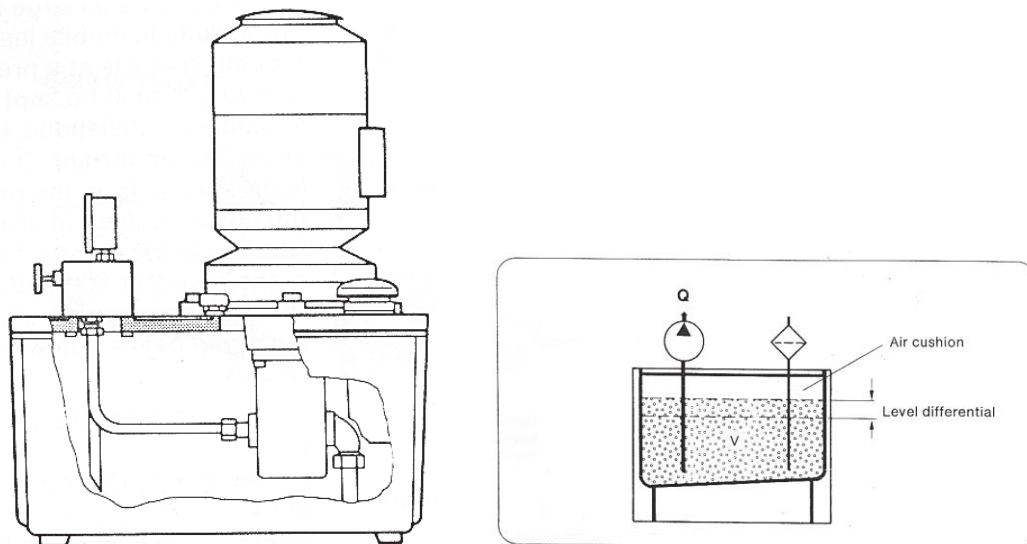
- ปั๊มไฮดรอลิกแบบเฟือง (Gear pump)
- ปั๊มไฮดรอลิกแบบสกรู (Screw pump)
- ปั๊มไฮดรอลิกแบบใบพัด (Vane pump)
- ปั๊มไฮดรอลิกแบบลูกสูบ (Piston pump)



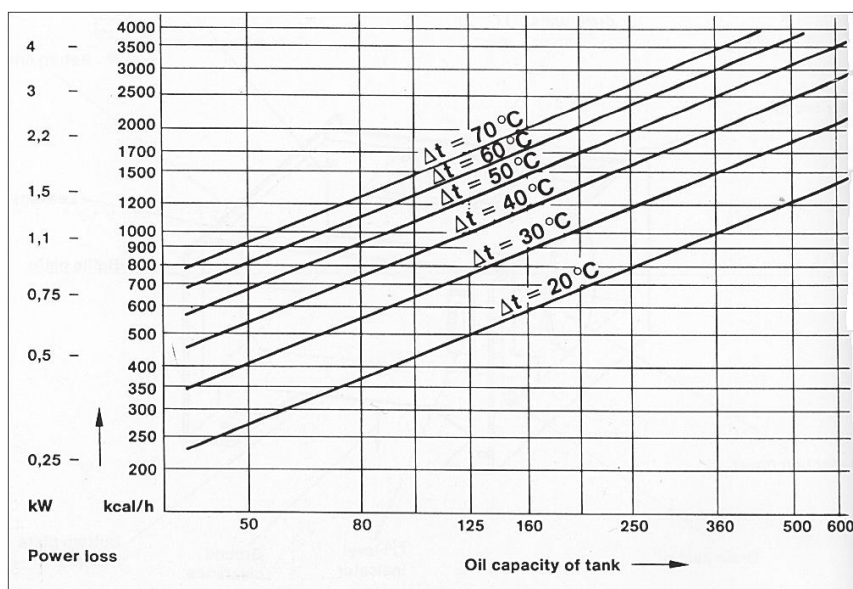
รูปที่ 2.32 ปั๊มไฮดรอลิก

ชุดเก็บน้ำมัน (Oil reservoirs) ถังเก็บน้ำมันมีความสำคัญมากในระบบไฮดรอลิก เพราะระบบไฮดรอลิก น้ำมันจะมีการไหลย้อนกลับ (Return) ตลอดเวลา ซึ่งจะต่างจากระบบนิวแมติกส์

(neumatic) จะระบายลมออกสู่บรรยากาศ ถังเก็บน้ำมันจะมีหน้าที่เก็บและพักน้ำมัน ระบายความร้อนของน้ำมัน ตลอดจนขจัดสิ่งสกปรกที่มากับน้ำมันอีกด้วย

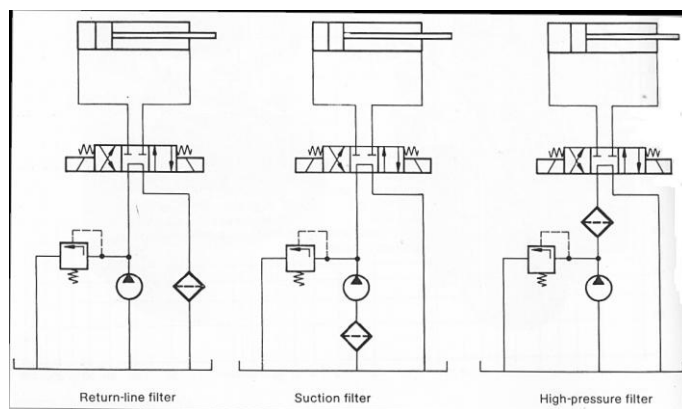
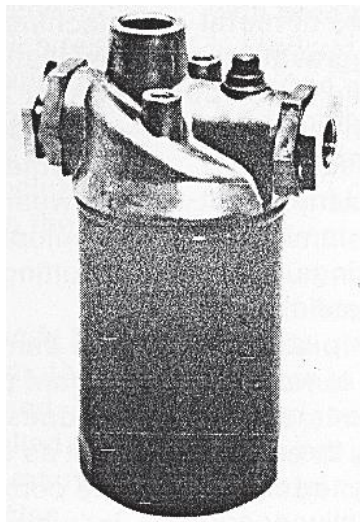


รูปที่ 2.33 ชุดเก็บน้ำมัน

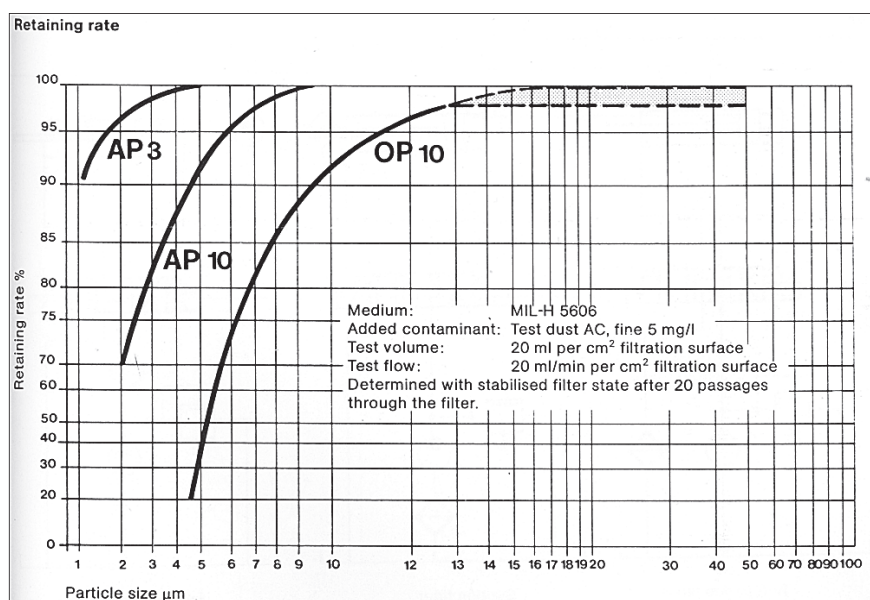


รูปที่ 2.34 กราฟแสดงความสัมพันธ์ อุณหภูมิน้ำมันของชุดเก็บน้ำมัน

ชุดกรองน้ำมัน (Filter) ชุดกรองน้ำมันจะมีหน้าที่กรองน้ำมันก่อนเข้าสู่ระบบ เพราะน้ำมันจะมีฝุ่นหรือสิ่งสกปรกปะปนมา จะทำให้เกิดความเสียหายแก่ตัวทำงาน (Actuators) และทำให้ลดประสิทธิภาพในการทำงานลงอีกด้วย

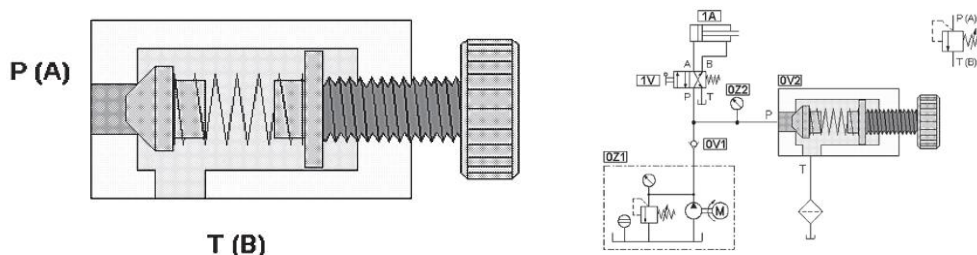


รูปที่ 2.35 ชุดกรองน้ำมันและตำแหน่งการติดตั้งชุดกรองน้ำมัน



รูปที่ 2.36 ประสิทธิภาพในการกรองอากาศ

วาล์วกำจัดแรงดัน (Pressure-relief valve) วาล์วกำจัดแรงดัน จะมีหน้าที่ควบคุมแรงดันในระบบให้อยู่ในสภาวะปกติ เพราะระบบไฮดรอลิกส์จะมีความดันสูง เมื่อในระบบมีความดันสูง จะทำให้สปริงเคลื่อนที่กลับ ทำให้รีลิวาล์วเปิด มีผลทำให้ความดันที่สูงลงกลับไปถึงน้ำมัน

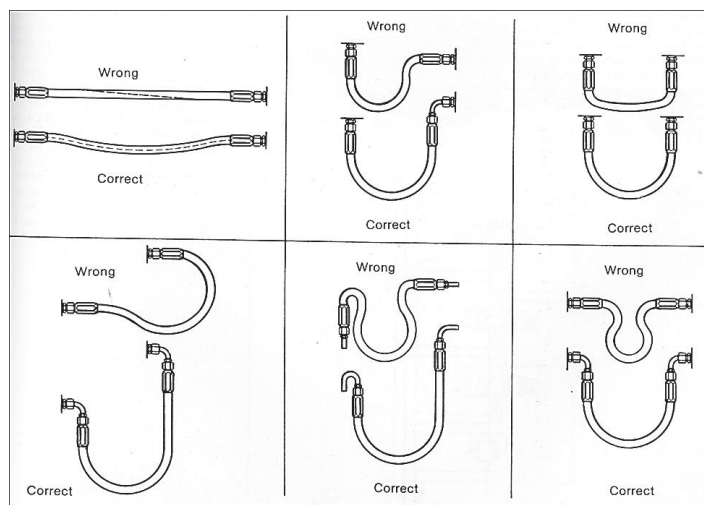


รูปที่ 2.37 วาล์วกำจัดความดัน

สายไฮดรอลิกส์และข้อต่อ (Pipes, Fitting and Connectors) ชุดสายไฮดรอลิกส์มีหน้าที่เชื่อมต่ออุปกรณ์ที่เคลื่อนที่ และอุปกรณ์ที่อยู่กับที่เพื่อให้ น้ำมันไหลผ่านไปยังจุดที่ต้องการ การเลือกชุดสายไฮดรอลิกส์ควรเลือกให้เหมาะสมกับความดันที่ใช้ในระบบ ดังแสดงในตารางที่ 2.9

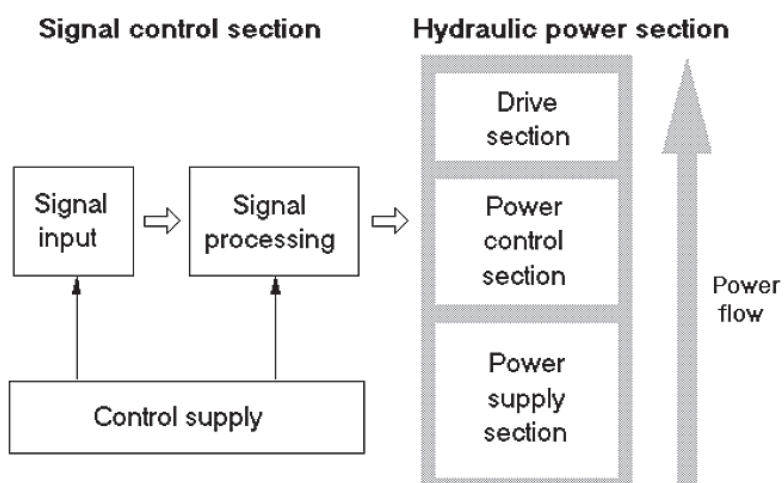
ตารางที่ 2.9 ขนาดของสายไฮดรอลิกส์และความดัน

OD mm	Wall Thick mm	Nominal Pressure (bar)		OD mm	Wall Thick mm	Nominal Pressure (bar)
4	1	600		18	3	365
5	1	400		20	2	193
6	1	300		20	3	313
6	2	1200		22	1.5	122
8	1	228		22	3	273
8	2	686		25	2	147
10	1	172		25	3	230
10	2	458		28	1.5	92
12	1	137		28	3	199
12	2	343		30	2.5	119
14	1	128		30	4	265
14	2	309		35	2	100
15	1.5	192		35	4	216
15	2.5	365		38	3	136
16	1.5	177		38	5	261
16	2.5	331		42	2	81
18	1.5	154				



รูปที่ 2.38 การติดตั้งชุดสายไฮดรอลิกส์

2. ส่วนควบคุม (Control System) ส่วนควบคุมจะมีองค์ประกอบของระบบต่างจากส่วนกำลัง โดยส่วนควบคุมจะควบคุมการทำงานของระบบเช่น สัญญาณควบคุม (Control signal) สัญญาณอินพุต (Input signal) การประมวลผลต่อสัญญาณ (Signal processing) และชุดจ่ายพลังงานในระบบ (Control supply) เป็นต้น



รูปที่ 2.39 ไตอะแกรมการทำงานของส่วนควบคุม

ส่วนควบคุมจะประกอบด้วย 3 ส่วน คือ

2.1 ชุดสัญญาณสั่งงาน (Signal Input) ชุดสัญญาณสั่งงานหรือสัญญาณอินพุต เป็นส่วนที่ส่งสัญญาณเข้าไปสู่ระบบประมวลผลเพื่อสั่งงานให้เครื่องจักรทำงาน ซึ่งสัญญาณสั่งงานดังกล่าว มักจะถูกส่งสัญญาณจากคน (Humans) หรือจากตัวตรวจจับ (Sensors) ในระบบการทำงานอุปกรณ์ที่ใช้ส่งสัญญาณ สามารถแบ่งออกได้ตามลักษณะการใช้งานดังนี้

1) อุปกรณ์ส่งสัญญาณไฟฟ้า อุปกรณ์ส่งสัญญาณไฟฟ้าที่ใช้ในโรงงานอุตสาหกรรมจะมีอยู่หลายชนิด โดยหลักการทำงานจะส่งสัญญาณ ON หรือ OFF ให้กับระบบ เช่น สวิตช์ปุ่มกด (Pushbutton) สวิตช์จำกัดระยะ (Limit switch) สวิตช์ทำงานแบบไม่สัมผัส (Proximity switch, Sensors) เป็นต้น



รูปที่ 2.40 อุปกรณ์ส่งสัญญาณไฟฟ้า

2) อุปกรณ์ส่งสัญญาณไฮดรอลิกส์แบบทางกล อุปกรณ์ส่งสัญญาณไฮดรอลิกส์ที่ใช้ในอุตสาหกรรม จะสั่งงานโดยอาศัยการทำงานทางกล ซึ่งอาจจะใช้ลูกสูบในการเคลื่อนที่สั่งงาน เช่น วาล์วแบบปุ่มกด (Pushbutton valve) วาล์วสั่งงานด้วยกลไก (Roller valve) เป็นต้น



รูปที่ 2.41 อุปกรณ์ส่งสัญญาณไฮดรอลิกส์แบบทางกล

3) อุปกรณ์ส่งสัญญาณไฮดรอลิกส์แบบสัมผัสอุปกรณ์ส่งสัญญาณชนิดนี้ จะส่งสัญญาณจากคนโดยตรงอาจจะใช้มือหรือเท้าในการส่งงาน เช่น วาล์วคันโยก (Manual valve) วาล์วแบบเท้าเหยียบ (Pedal operated) เป็นต้น



รูปที่ 2.42 อุปกรณ์ส่งสัญญาณไฮดรอลิกส์แบบสัมผัส

2.2 ชุดประมวลผลสัญญาณ (Signal Processing) ชุดประมวลผลสัญญาณหรือชุดจัดกระบวนการทำงาน ซึ่งจะได้รับสัญญาณมาจากอินพุท (Input) แล้วนำสัญญาณมาประมวลผลก่อนส่งสัญญาณเอาต์พุท (Output) ออกไปควบคุมการทำงานของชุดส่วนกำลัง (Hydraulic power section) ชุดประมวลผลสัญญาณสามารถแบ่งออกได้ดังนี้

1) การจัดการกระบวนการทำงานของระบบไฮดรอลิกส์ด้วยสัญญาณไฟฟ้า ซึ่งสามารถทำงานโดยใช้วงจรรีเลย์ (Relay) ควบคุมการทำงานของระบบพีแอลซี (Programmable Logic Controller) หรือไมโครคอนโทรลเลอร์ (Microcontroller)



รูปที่ 2.43 ชุดประมวลสัญญาณด้วยไฟฟ้า

2) การจัดการกระบวนการทำงานของระบบไฮดรอลิกส์ ด้วยระบบนิวแมติกส์ ซึ่งสามารถทำได้โดยการใช้วาล์วหน่วงเวลา (Timer,Counter) สเต็ปเปอร์โมดูล (Stepper Module) หรือชุดขยายสัญญาณ (Booster) เป็นต้น



รูปที่ 2.44 ชุดประมวลผลสัญญาณด้วยระบบนิวแมติกส์

3) การจัดการกระบวนการทำงานของระบบไฮดรอลิกส์ด้วยระบบ

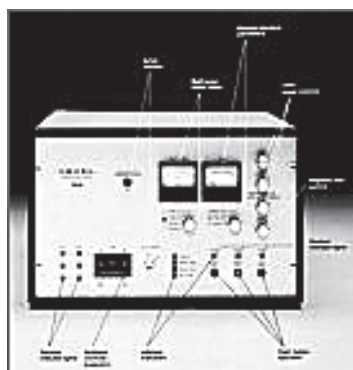
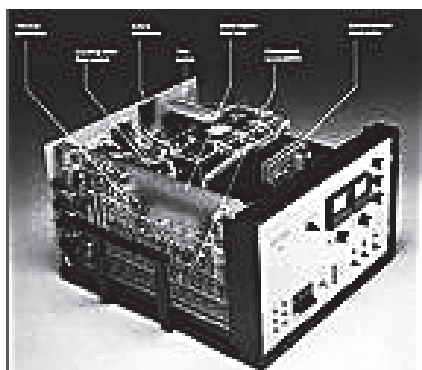
ไฮดรอลิกส์ ซึ่งสามารถทำได้โดยการใช้วาล์วปรับความดัน (Pressure sequence valve) วาล์วควบคุมอัตราการไหล (Flow control valve) หรือ เช็ควาล์ว (Check valve) เป็นต้น



รูปที่ 2.45 ชุดประมวลผลสัญญาณด้วยวาล์วระบบไฮดรอลิกส์

2.3 ชุดจ่ายพลังงานในระบบ (Control Supply) ส่วนจ่ายพลังงานนี้เป็นส่วนต้น

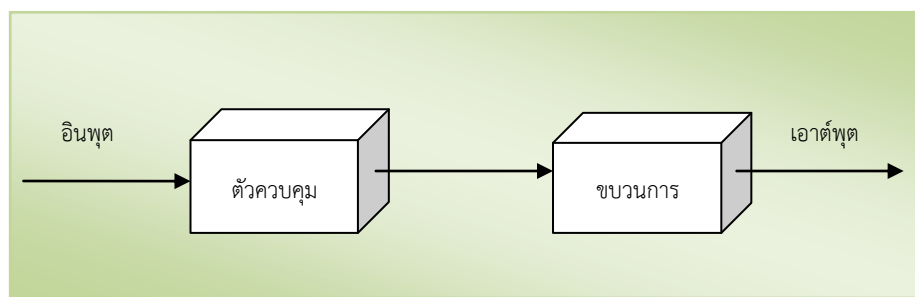
กำลังของระบบควบคุมทั้ง ระบบ โดยส่วนจ่ายพลังงานให้กับส่วนควบคุมนี้จะควบคุมด้วยสัญญาณไฟฟ้า โดยจะสามารถควบคุมการทำงานได้ทั้งระบบนิวแมติกส์ และระบบไฮดรอลิกส์



รูปที่ 2.46 ชุดจ่ายพลังงาน (Power supply)

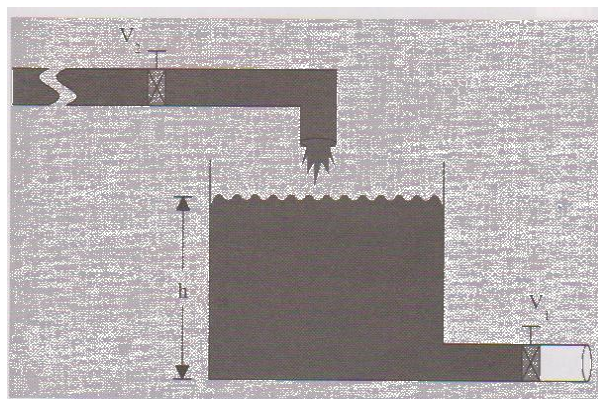
4. การควบคุมระบบเปิด (Open Loop control system)

ระบบควบคุมแบบเปิดหรือเรีย ยอีกอย่างหนึ่งว่า ระบบควบคุมแบบไม่มีการป้อนกลับ ระบบนี้เป็นระบบที่ง่ายที่สุดและมีอุปกรณ์ภายในที่ย่งยาก ระบบนี้ค่าเอาต์พุตที่ได้จะไม่มีการควบคุมขบวนการของระบบหมายความว่าไม่มีการนำเอาค่าเอาต์พุตที่ได้กลับมาเปรียบเทียบกับค่าอินพุต (อ้างอิง) ที่ป้อนให้กับระบบซึ่งสามารถแสดงลักษณะบล็อก ไดอะแกรมของระบบควบคุมแบบเปิด ดังรูป 2.47



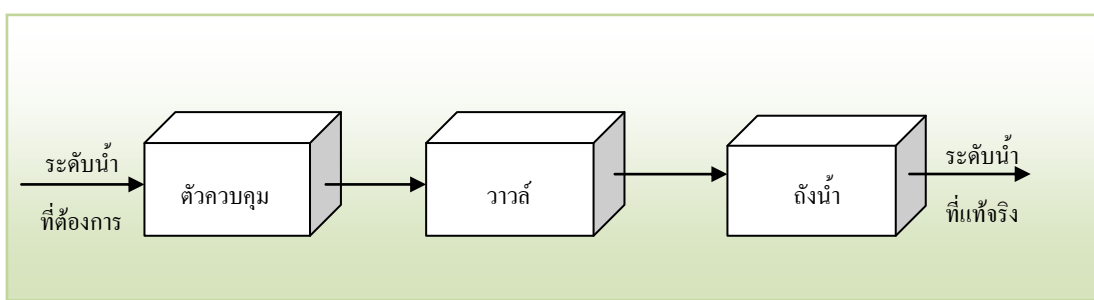
รูปที่ 2.47 บล็อกไดอะแกรมของระบบควบคุมแบบเปิด

ดังตัวอย่าง การควบคุมแบบเปิด (Open – loop control system) ซึ่งเป็นการควบคุมระบบของน้ำในถัง



รูปที่ 2.48 การควบคุมระดับน้ำในถัง

จากรูปที่ 2.48 เป็นระบบการควบคุมน้ำระดับ h ในถังโดยวาล์ว v_2 ที่เป็นวาล์วควบคุมการไหลเข้าของน้ำในถัง แต่ระบบนี้จะไม่ละเอียดเที่ยงตรง ถ้าวาล์ว v_1 มีการเปลี่ยนแปลงเกิดขึ้น เพราะไม่ได้มีการวัดอัตราการไหลของน้ำที่แน่นอนที่วาล์ว v_2 และ v_1 และจากความสัมพันธ์ต่าง ๆ จากรูปที่ 2.48 สามารถนำมาเขียนเป็นบล็อกไดอะแกรม ซึ่งจะได้ความสัมพันธ์ระหว่าง อินพุต กับเอาต์พุต ดังรูป 2.49



รูปที่ 2.49 บล็อกไดอะแกรมของระบบควบคุมน้ำในถัง

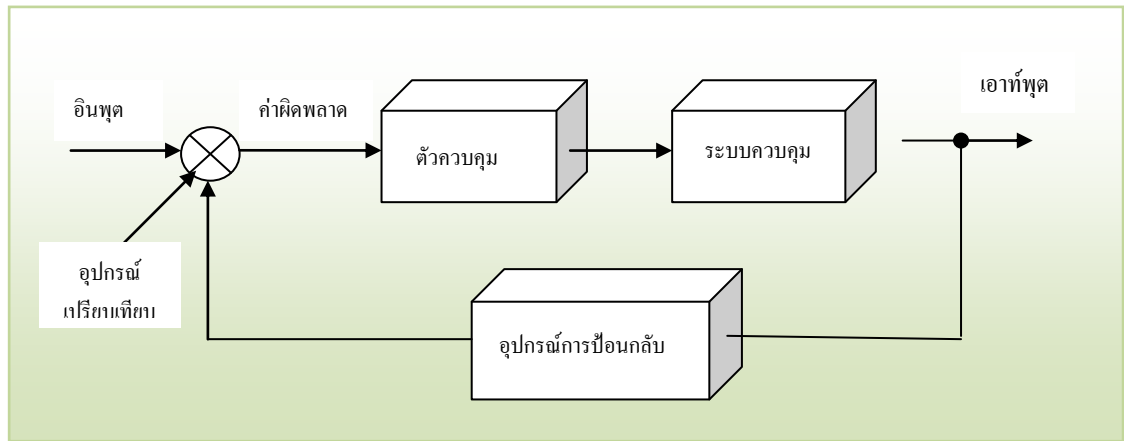
จากรูป 2.49 แสดงความสัมพันธ์ของระบบระหว่างอินพุต (ระดับน้ำที่ต้องการ) และเอาต์พุต (ระดับน้ำในถังที่แท้จริง) ทิศทางของลูกศรที่มีทิศทางพุ่งเข้าหาระบบควบคุมแสดงถึงอินพุต ส่วนลูกศรที่มีทิศทางพุ่งออกจากระบบควบคุมเอาต์พุต ซึ่งจากบล็อกไดอะแกรมจะเห็นว่าไม่มีสัญญาณใด ๆ ของระบบป้อนกลับไปยังอินพุตเลย ดังนั้นจึงเรียกว่าระบบควบคุมลักษณะนี้ว่า ระบบควบคุมแบบเปิด

ข้อสังเกตของการควบคุมระบบแบบเปิดคือ

- 1) ไม่มีการนำสัญญาณทางด้านเอาต์พุตป้อนกลับมาทางด้านอินพุต
- 2) ระบบไม่มีความซับซ้อน
- 3) ระบบใช้กับงานที่ไม่ต้องการความแม่นยำ
- 4) เป็นระบบที่ประหยัด

5. ระบบปิด (Closed - loop control system)

ระบบควบคุมแบบปิด เป็นการควบคุมที่มีการป้อนกลับค่าสัญญาณผลลัพธ์ที่เกิดขึ้นจริงมาที่ตัวควบคุม เพื่อให้ตัวควบคุมสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ทันที่ ดังแสดงโดยแผนผังการทำงานในรูปที่ 2.50



รูปที่ 2.50 บล็อกไดอะแกรมของระบบควบคุมแบบปิด

ดังนั้นโดยทั่วไปจะเรียกว่าการควบคุมวงจรมีการควบคุมป้อนกลับ (Feedback Control) และในที่นี้ก็เช่นเดียวกันอาจจะมีการเรียกชื่อทั้งสองนี้สลับไปมาแทนกัน ระบบควบคุมแบบปิด สามารถจำแนกส่วนประกอบเป็น 6 ส่วนหลัก ดังนี้

- 1) สัญญาณอ้างอิง (Reference Signal) ของการควบคุม ในที่นี้แทนด้วยตัวแปร $r(t)$ เป็นสัญญาณอินพุตของระบบควบคุมเพื่อกำหนดเป้าหมายของการควบคุม ตัวอย่างสัญญาณอ้างอิง เช่น การควบคุมให้มอเตอร์หมุนด้วยความเร็วรอบคงที่ 1,000 รอบต่อนาที จะกำหนดให้ $r(t) = 1,000$ รอบต่อนาที เป็นต้น
- 2) ตัวควบคุม (Controller) ทำหน้าที่ควบคุมระบบให้มีผลลัพธ์สอดคล้องกับค่าสัญญาณอ้างอิงหรือเป็นไปตามเป้าหมายของการควบคุม
- 3) สัญญาณควบคุม (Control Command) เป็นค่าเอาต์พุตของตัวควบคุมที่ใช้เป็นคำสั่งหรือสัญญาณอินพุตให้กับระบบที่ถูกควบคุม ในที่นี้แทนด้วยสัญลักษณ์ $u(t)$ โดยที่ค่าของสัญญาณจะถูกกำหนดโดยสมการกฎการควบคุม (Control Law)
- 4) กระบวนการหรือระบบที่ถูกควบคุม (Control System or Process)

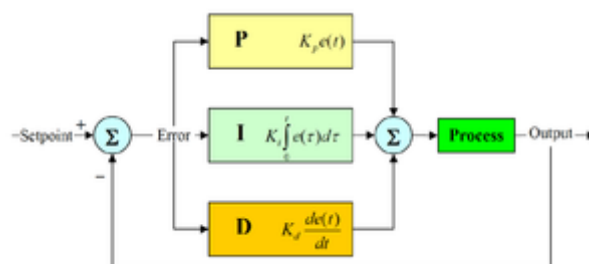
5) สัญญาณรบกวน (Disturbance) เป็นสัญญาณอินพุตที่ไม่พึงปรารถนาในการควบคุมระบบ ในที่นี้ แทนด้วยสัญลักษณ์ $d(t)$ ตัวอย่างเช่น ในการควบคุมความเร็วรอบของมอเตอร์ อาจจะมีแรงเสียดทานต่าง ๆ เป็นสัญญาณรบกวน เป็นต้น

6) ผลลัพธ์การควบคุม (Control Output) เป็นสัญญาณเอาต์พุตแสดงค่าตัวแปรที่ถูกควบคุมหรือผลตอบสนองของระบบควบคุม เช่น ค่าความเร็วรอบจริงมอเตอร์ โดยในที่นี้ค่าผลลัพธ์จะแทนด้วยสัญลักษณ์ $y(t)$ และโดยทั่วไปวัตถุประสงค์ของการควบคุมจะต้องการให้ค่าผลลัพธ์ $y(t)$ นี้เท่ากับค่าสัญญาณอ้างอิง $r(t)$ โดยที่ค่าผลต่าง $e(t) = r(t) - y(t)$ จะใช้เป็นตัวบ่งบอกว่าตัวควบคุมทำงานได้ดีบรรลุตามจุดประสงค์เพียงใด

ระบบควบคุมวงปิดเป็นระบบที่เป็นตามธรรมชาติที่สุดและมีอยู่ทั่วไปในชีวิตประจำวัน แต่เราอาจไม่ได้สังเกต ตัวอย่างเช่น การขับรถให้ได้ความเร็วที่ต้องการก็ถือว่าเป็นระบบควบคุมวงปิด ซึ่งตัวควบคุมในกรณีนี้คือคนขับรถโดยมีระบบสมองเป็นศูนย์กลางกำหนดกฎการควบคุม จะเห็นได้ได้ว่าในขณะที่ขับนั้นจะมีการป้อนกลับค่าผลลัพธ์ความเร็วของรถผ่านทางตาจาก การมองเห็นว่าปิดความเร็วแล้วส่งค่านั้นกลับไปที่สองคนขับ จากนั้นสมองจะออกคำสั่งควบคุมที่เหมาะสมกับค่าผลลัพธ์ที่เกิดขึ้นจริงเพื่อให้รถวิ่งได้ความเร็วที่ต้องการ โดยมีสัญญาณรบกวนซึ่งอาจจะเป็นความไม่สม่ำเสมอของลาดชันของพื้นถนน

6. ระบบ พี ไอ ดี (PID Control)

ระบบควบคุมแบบสัดส่วน-ปริพันธ์-อนุพันธ์ (อังกฤษ: PID controller) เป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของ PID ที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ



รูปที่ 2.51 บล็อกของการควบคุมแบบพีไอดี

วิธีคำนวณของ PID ขึ้นอยู่กับสามตัวแปรคือค่าสัดส่วน, ปริพันธ์ และ อนุพันธ์ ค่าสัดส่วน กำหนดจากผลของความผิดพลาดในปัจจุบัน, ค่าปริพันธ์กำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งพียงผ่านพ้นไป, และค่าอนุพันธ์กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการ

โดยการปรับค่าคงที่ใน PID ตัวควบคุมสามารถปรับรูปแบบการควบคุมให้เหมาะสมกับที่กระบวนการต้องการได้ การตอบสนองของตัวควบคุมจะอยู่ในรูปของการไหวตัวของตัวควบคุมจนถึงค่าความผิดพลาด ค่าโอเวอร์ชูต (overshoots) และ ค่าแกว่งของระบบ (oscillation) วิธี PID ไม่รับประกันได้ว่าจะเป็นระบบควบคุมที่เหมาะสมที่สุดหรือสามารถทำให้กระบวนการมีความเสถียรแน่นอน

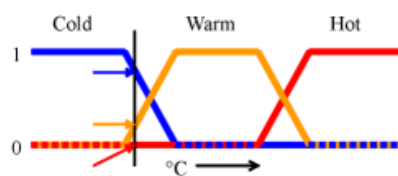
การประยุกต์ใช้งานบางครั้งอาจใช้เพียงหนึ่งถึงสองรูปแบบ ขึ้นอยู่กับกระบวนการเป็นสำคัญ ไฟโอดีบางครั้งจะถูกเรียกว่าการควบคุมแบบ PI, PD, P หรือ I ขึ้นอยู่กับว่าใช้รูปแบบใดบ้าง

7. ระบบฟัซซี่ (Fuzzy Control)

ฟัซซี่ลอจิก (fuzzy logic) พัฒนาจาก ทฤษฎีเซตวิภันชนัย โดยเป็นการใช้เหตุผลแบบประมาณ ซึ่งแตกต่างจากการใช้เหตุผลแบบเด็ดขาดในลักษณะ ญ ก/ผิด ใช่ /ไม่ใช่ ของ ตรรกศาสตร์แบบฉบับ (classical logic) ตรรกศาสตร์คลุมเครือนั้นสามารถถือเป็นการประยุกต์ใช้งานเซตวิภันชนัย เพื่อจำลองการตัดสินใจของผู้เชี่ยวชาญ ต่อปัญหาที่ซับซ้อนค่าระดับความจริง ในตรรกศาสตร์คลุมเครือ นั้นมักจะสับสนกับ ค่าความน่าจะเป็น ซึ่งมีแนวความคิดที่แตกต่างกัน ค่าระดับความจริงคลุมเครือ นั้นใช้ในการระบุ ค่าความเป็นสมาชิก ของเซต แต่ค่าความน่าจะเป็นนั้นระบุความเป็นไปได้ของสภาพการณ์แต่ละรูปแบบที่อาจจะเกิดขึ้น ตัวอย่างเช่น สมมติ นาย ก กำลังเดินเข้าบ้าน สถานะของ นาย ก ตามตรรกศาสตร์แบบฉบับ คือ "อยู่ในบ้าน" หรือ "อยู่นอกบ้าน" หากเขากำลังยืนอยู่ระหว่างช่องประตูละ? เราอาจพิจารณาว่าเขา "อยู่ในบ้านบางส่วน" ระดับของสถานะกึ่งนี้ จะระบุด้วยค่าความเป็นสมาชิกของเซตวิภันชนัย สมมติเขาเพิ่งจะก้าวปลายนิ้วเท้าผ่านข้ามธรณีประตูเข้าบ้าน เราอาจกล่าวว่า นาย ก นั้น 0.99 "อยู่นอกบ้าน" ซึ่งต่างจากความน่าจะเป็นของเหตุการณ์สุ่ม (เช่น ความน่าจะเป็นระบุผลลัพธ์ของการโยนเหรียญ แต่ผลลัพธ์จะออก หัว หรือ ก้อย) หากพิจารณาความน่าจะเป็นที่นาย ก "อยู่นอกบ้าน" และ "อยู่ในบ้าน" จะออกผลลัพธ์เป็น นาย ก อยู่นอกบ้าน หรือ ในบ้าน ไม่ได้จำลองสถานะกึ่ง คือ กำลังยืนอยู่ที่ประตู เซตวิภันชนัยนี้มีหลักการพื้นฐานจากเซตที่มีขอบเขต

คลุมเครือไม่ชัดเจน ไม่ได้มีพื้นฐานจากการสุ่มตรรกศาสตร์คลุมเครือนั้น สามารถระบุ ค่าความเป็นสมาชิกของเซต (set membership values) ด้วยค่าระหว่าง 0 และ 1 ทำให้เกิดระดับกึ่งในลักษณะของ สีเทา นอกจาก ขาว และ ดำ ซึ่งมีประโยชน์ในการจำลองระดับซึ่งสามารถระบุด้วยคำพูด "เล็กน้อย" "ค่อนข้าง" "มาก" โดยใช้ค่าความเป็นสมาชิกของเซตบางส่วน ตรรกศาสตร์คลุมเครือนี้มีความสัมพันธ์กับ เซตวิภังค์ (en:fuzzy set) และ ทฤษฎีความเป็นไปได้ (en:possibility theory) ซึ่งคิดค้นขึ้นในปี ค.ศ. 1965 โดยศาสตราจารย์ ลอตฟี ซาเดห์ แห่งมหาวิทยาลัยแห่งรัฐแคลิฟอร์เนีย เบิร์กลีย์ตรรกศาสตร์คลุมเครือ ถึงแม้ว่าจะได้รับการยอมรับค่อนข้างกว้างขวาง แต่ก็ยังถูกโต้แย้งจากบางกลุ่ม เช่น จาก วิศวกรระบบควบคุมในเรื่องของการอธิบายพฤติกรรมต่างๆ และจาก นักสถิติบางกลุ่ม ซึ่งเชื่อมั่นว่า ทฤษฎีความน่าจะเป็น เป็นวิธีทางคณิตศาสตร์ที่เคร่งครัดเพียงวิธีเดียว ในการจำลองความไม่แน่นอน (en:uncertainty) นอกจากนั้นแล้ว ก็ยังมีการวิเคราะหวิจารณ์ว่า เซตวิภังค์ นั้นไม่ได้เป็นซูเปอร์เซตของ ทฤษฎีเซตสามัญ เนื่องจาก ฟังก์ชันภาวะสมาชิก นั้นกำหนดในรูปของ เซตแบบดั้งเดิม

ระบบฟัซซี่สามารถนำไปประยุกต์ใช้งานในการควบคุม อุปกรณ์ต่างๆ เช่น ตัวอย่างของเครื่องใช้ในครัวเรือน เช่น เครื่องซักผ้า (โดยการวัดปริมาณผ้า และ ความเข้มข้นของน้ำยาซักผ้า และ ปรับวงจรการซักให้เหมาะสม) และ ตู้เย็น วิธีการใช้งานง่าย ๆ นั้นอาจใช้ในการจำลองช่วงย่อยๆ ของตัวแปรที่มีค่าต่อเนื่อง เช่น การวัดอุณหภูมิในระบบห้ามล้อแบบกันล้อตาย อาจมีฟังก์ชันความเป็นสมาชิกของเซตหลายฟังก์ชัน สำหรับอุณหภูมิซึ่งแบ่งเป็นหลายช่วง เพื่อควบคุมการห้ามล้อให้เหมาะสม โดยแต่ละฟังก์ชันจะทำการส่งค่าอุณหภูมิหนึ่งๆ ไปเป็นค่าความจริงในช่วง 0 ถึง 1 ซึ่งค่าความจริงเหล่านี้จะถูกนำไปใช้ในการควบคุมการห้ามล้อ



รูปที่ 2.52 ฟังก์ชันในการส่งค่าระดับอุณหภูมิ

ในภาพ cold (เย็น) warm (อุ่น) and hot (ร้อน) เป็นฟังก์ชันในการส่งค่าระดับอุณหภูมิ ที่แต่ละจุดของอุณหภูมิจะมีค่าความจริง 3 ค่า ซึ่งเป็นค่าของแต่ละฟังก์ชัน ซึ่งค่าความจริงทั้งสามนี้สามารถใช้ในความค่าอุณหภูมิใดๆ ว่า "ค่อนข้างเย็น" "อุ่นนิดๆ" "ไม่ร้อน"

ตัวอย่างของการใช้งานที่ซับซ้อนขึ้น ก็คือ การใช้งานตรรกศาสตร์คลุมเครือใน การแก้ความผิดพลาด (error correction) ประสิทธิภาพสูง เพื่อเพิ่มประสิทธิภาพของการรับข้อมูลผ่านช่องสัญญาณแบนด์วิดท์จำกัด ที่ข้อมูลถูกรบกวนด้วยสัญญาณรบกวน โดยการใช้ รหัสเทอร์โบ (en:turbo code) ส่วนต้นของภาคถอดรหัส จะวัดค่าความควรจะเป็น ของค่าบิตในสายบิตที่ส่งมาจากภาคส่ง (ว่าควรจะเป็น 0 หรือ 1) ค่าความควรจะเป็นอาจแบ่งเป็น 256 ระดับ ระหว่างระดับสูงสุดหมายถึง "ค่าควรจะเป็น 1 แน่แน่นอน" และต่ำสุดหมายถึง "ค่าควรจะเป็น 0 แน่แน่นอน" ตัวถอดรหัสสองตัวอาจวิเคราะห์ข้อมูลที่รับมาพร้อมกัน ได้เป็นค่าความควรจะเป็นที่ต่างกัน ตัวถอดรหัสแต่ละตัวสามารถใช้ค่าความควรจะเป็นที่ได้จากตัวถอดรหัสอื่นเข้าช่วยในการตีความ จนได้ข้อสรุปค่าที่ควรจะเป็นมากที่สุด

การประยุกต์ใช้งานตรรกศาสตร์คลุมเครือโดยทั่วไป จะใช้ในการจำลองความรู้หรือประสบการณ์ของผู้เชี่ยวชาญ โดยการใช้เหตุผล หรือ การตัดสินใจต่อสภาวะการณ์ต่างๆ ของมนุษย์นั้น สามารถเขียนอยู่ในรูปเชิงภาษาศาสตร์ของ ระบบกฎเกณฑ์ (rule-based system) คือ เงื่อนไข IF/THEN หรือ อยู่ในรูปอื่นที่เท่าเทียมกัน เช่น เมทริกซ์เปลี่ยนหมู่ฟัซซี (fuzzy associative matrices)

กฎเกณฑ์:

IF (ถ้า) <เงื่อนไข> THEN(แล้ว) <ผลที่ตามมา>

การใช้เหตุผล การตัดสินใจ หรือ การตอบสนองต่อเหตุการณ์ต่างๆ ของมนุษย์นั้น โดยปกติจะมีลักษณะที่คลุมเครือ เช่นการประเมินสภาวะการณ์ หรือ การระบุการตอบสนอง โดยไม่ได้ระบุเป็นค่าที่แน่นอนชัดเจน ดังนั้นจึงถูกจำลองไว้ในกฎเกณฑ์ด้วย เซตวิภันย์

ตัวอย่าง เช่น ระบบควบคุมอุณหภูมิโดยใช้พัดลม อาจมีกฎเกณฑ์ดังนี้:

IF อุณหภูมิ เย็นมาก THEN หยุดพัดลม
IF อุณหภูมิ เย็น THEN ปรับพัดลมให้ช้าลง
IF อุณหภูมิ ปานกลาง THEN รักษาระดับความเร็ว
IF อุณหภูมิ ร้อน THEN ปรับพัดลมให้เร็วขึ้น

สังเกตว่าไม่มีการใช้ **ELSE (ไม่เช่นนั้น)** ทุกเงื่อนไขจะต้องถูกนำมาพิจารณา เนื่องมาจากสภาพกำกวมในตรรกศาสตร์คลุมเครือ เช่น อุณหภูมิอาจเป็นสมาชิกของ ทั้งเซต "เย็น" และ "ปานกลาง" โดยอาจมีค่าระดับความเป็นสมาชิกของแต่ละเซตต่างกัน ซึ่งชุดของกฎเกณฑ์ดังกล่าวนี้เรียกว่า กฎเกณฑ์ฟัซซี (fuzzy rule) ฐานกฎเกณฑ์ฟัซซี (fuzzy rule base) หรือ ฐานความรู้ (knowledge base)

การดำเนินการทางตรรก เช่น AND(และ) OR(หรือ) NOT(ไม่) นั้นก็มีใช้สำหรับตรรกศาสตร์คลุมเครือ สำหรับการสร้างเงื่อนไขของเหตุการณ์ที่ซับซ้อน เช่นเดียวกับตรรกศาสตร์แบบฉบับ โดยปกติแล้วจะนิยาม AND=minimum OR=maximum และ NOT=complement ซึ่งการนิยามตามแบบนี้เรียกว่า *ตัวดำเนินการซาเดห์* (Zadeh operator) เนื่องจากเป็นไปตามลักษณะนิยามที่ ซาเดห์ใช้ในบทความดั้งเดิมของเขา ดังนั้นหากเรามีตัวแปรฟัซซี x และ y :

$$\text{NOT } x = (1 - \text{truth}(x))$$

$$x \text{ AND } y = \text{minimum}(\text{truth}(x), \text{truth}(y))$$

$$x \text{ OR } y = \text{maximum}(\text{truth}(x), \text{truth}(y))$$

ภาษาอังกฤษว่า *hedges* หรือ *linguistic hedges* (หมายถึง คำที่ทำให้คลุมเครือ) เช่น "มาก" "บ้าง" ซึ่งใช้ในการแปลงความหมายของเซตโดยใช้วิธีการทางคณิตศาสตร์

ในทางปฏิบัติ การนำกฎเกณฑ์ ความชำนาญนี้มาใช้งาน มักจะเกี่ยวข้องกับค่าที่แน่นอน เช่น ค่าที่ได้จากการวัด และค่าที่ส่งออกเพื่อใช้งานก็มักจะเป็นค่าที่แน่นอนเช่นเดียวกัน ส่วนที่ทำการแปลงค่ารับเข้าที่เป็นค่าแน่นอนนี้ ไปสู่เซตฟัซซี ด้วยฟังก์ชันภาวะสมาชิก เรียกว่า ตัวทำให้คลุมเครือ (fuzzifier) และส่วนที่ทำการส่งค่าผลลัพธ์จากเงื่อนไขไปสู่ค่าแน่นอนเพื่อส่งออกไปใช้งานจริง เรียกตัวกำจัดความคลุมเครือ (defuzzifier) นอกจากนั้นแล้วในการสร้างกฎเกณฑ์ฟัซซี ที่มีตัวแปรค่ารับเข้าเป็นจำนวนมาก ให้ครอบคลุมทุกเงื่อนไขนั้นเป็นไปได้ยากในทางปฏิบัติ บางครั้งจึง ต้องมีการผสมเงื่อนไขหลายเงื่อนไขในกฎเกณฑ์ฟัซซีเข้าด้วยกัน โดยมีส่วนตีความผลลัพธ์ร่วมจากการผสมเงื่อนไข เรียก เครื่องอนุมานฟัซซี (fuzzy inference engine) ซึ่งมีหลายชนิดด้วยกัน