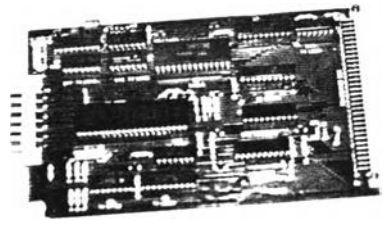
บทที่ 6

การสร้างเครื่องควบคุมและการทดสอบ

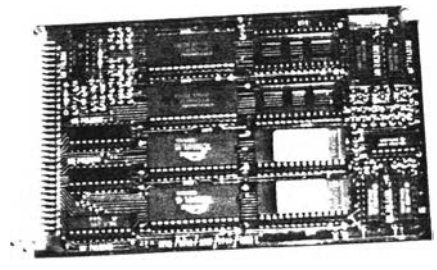
6.1 การสร้างเครื่องควบคุม

(1) **บอร์ดหลักของระบบ** (System main board) ใช้ card system SDA-88 [12] มีลักษณะของฮาร์ดแวร์เป็น rack โดยจะแบ่งวงจรออกเป็นบอร์ดตามหน้าที่การทำงาน ซึ่งแต่ละบอร์ดจะเชื่อมโยงกันผ่านบัส UROCON ประกอบด้วยบอร์ด 3 บอร์ด ดังนี้

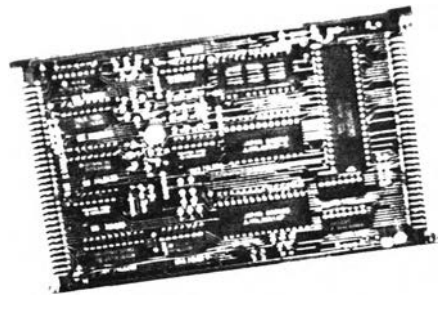
- CPU Board (รูปที่ 6.1)
- Memory Board (รูปที่ 6.2)
- Timer & Communication (รูปที่ 6.3)



รูปที่ 6.1 แผ่นวงจรซีพียู

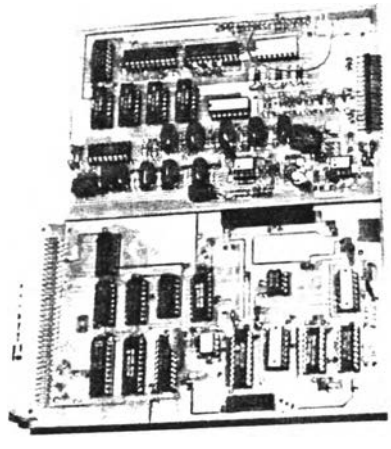


รูปที่ 6.2 แผ่นวงจรหน่วยความจำ



รูปที่ 6.3 แผ่นวงจร Timer & Communication

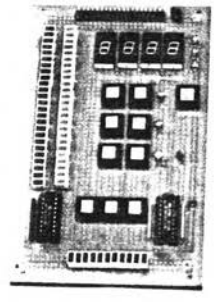
(2) **วงจรอินพุทและเอาท์พุท** อินเตอร์เฟสกับสัญญาณภายนอกทั้งแบบอนาลอค และดิจิตอล ได้ออกแบบเป็นแผ่นวงจรนิมน์ ดังรูปที่ 6.4



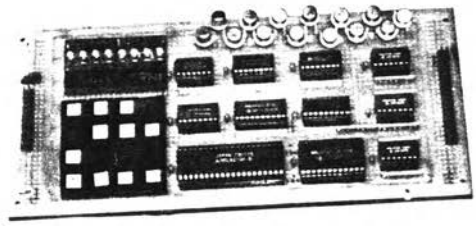
รูปที่ 6.4 แผ่นวงจรอินพุทและเอาท์พุทของเครื่องควบคุมเชิงเลข

(3) **วงจรแป้นนิมน์และแสดงผล** ออกแบบและต่อวงจรด้วยการเดินสาย Wire Wrap แบ่งเป็น 2 บอร์ด คือ

- บอร์ดแสดงผลและแป้นนิมน์ด้านหน้า (รูปที่ 6.5)
- บอร์ดแสดงผลและแป้นนิมน์ด้านข้าง (รูปที่ 6.6)



รูปที่ 6.5 แผ่นวงจรส่วนแสดงผลและแป้นพิมพ์ด้านหน้า



รูปที่ 6.6 แผ่นวงจรแสดงผลและแป้นพิมพ์ด้านข้าง

6.2 การทดสอบซอฟแวร์ของเครื่องควบคุม

การพัฒนาซอฟแวร์มีขั้นตอน ดังนี้

(1) เขียนโปรแกรมควบคุมระบบโดยใช้ภาษา PLM-86 [13] และ ASSEMBLY [14] ซึ่งจะได้โปรแกรม file.plm และ file.asm ตามลำดับ การเขียนโปรแกรมจะแบ่งเขียนเป็นโมดูล เพื่อง่ายต่อการทดสอบและแก้ไขโปรแกรม

(2) ใช้ Compiler PLM86 แปลง file.plm เป็น file.obj

ใช้ Compiler ASM86 แปลง file.asm เป็น file.obj

(3) การแปลง file.obj เป็น file.h (Intel Hex 16 บิท) ใช้ iAPX 86,88 Family Utilities [15] ดังนี้

- LINK86 รวมหลายๆ file.obj เป็น 1 โมดูล ได้ file.lnk
- LOC86 กำหนดตำแหน่งของ Segment Register ต่างๆของโปรแกรม โดยเปลี่ยน file.lnk เป็น file.loc
- OH86 เปลี่ยน file.loc เป็น file.h

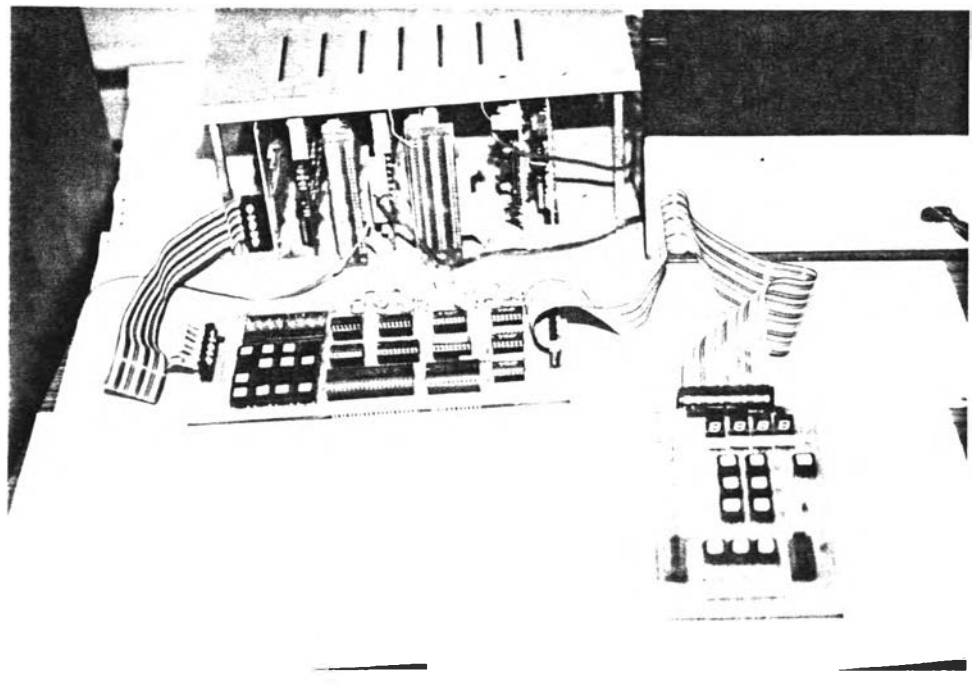
(4) ใช้โปรแกรม INTELH ซึ่งเขียนขึ้นเองเพื่อเปลี่ยน file.h เป็น file.hex (Intel Hex 8 บิท) เนื่องจากการเขียนโปรแกรมลงบน ROM โดยใช้ EPROM Programmer CLK3000 และ การโหลด file.hex ลงบนโปรแกรม DEBUG ต้องใช้ไฟล์ที่มี ฟอร์แมท Intel Hex 8 บิท

(5) การทดสอบซอฟแวร์ขณะพัฒนาโปรแกรม แบ่งการทดสอบเป็น 2 ลักษณะ ดังนี้

- ทดสอบบนไมโครคอมพิวเตอร์ โดยไม่ต้องใช้ฮาร์ดแวร์ของเครื่องควบคุม เช่น โปรแกรมการบวก ลบ คูณ หาร เลขทศนิยม การทดสอบทำโดยใช้โปรแกรม DEBUG อ่าน file.hex มาทดสอบ
- ทดสอบบนฮาร์ดแวร์ของเครื่องควบคุม บางโปรแกรมจำเป็นต้องใช้ฮาร์ดแวร์ เช่น โปรแกรมสแกนอนาลอคอินพุท เป็นต้น การทดสอบทำโดยใช้โปรแกรม DLFAST ซึ่งเขียนขึ้นเอง โหลด file.hex จากไมโครคอมพิวเตอร์ผ่าน Serial Comm. ลงใน RAM ของเครื่องควบคุม และทำการทดสอบโปรแกรม

6.3 การทดสอบเครื่องควบคุม

เครื่องควบคุมเชิงเลขชนิดโปรแกรมได้ ที่ใช้ทดสอบแสดงดังรูปที่ 6.7



รูปที่ 6.7 เครื่องควบคุมเชิงเลขชนิดโปรแกรมได้ใช้ในการทดสอบ

6.3.1 การทดสอบฟังก์ชันการทำงาน

ทดสอบโดยโหลดแต่ละโมดูลของฟังก์ชันลงในหน่วยความจำของเครื่องควบคุม โดยใช้โปรแกรม DLFAST และสั่งให้โปรแกรมฟังก์ชันนั้นทำงานและตรวจสอบผลการทดสอบจากหน่วยความจำ

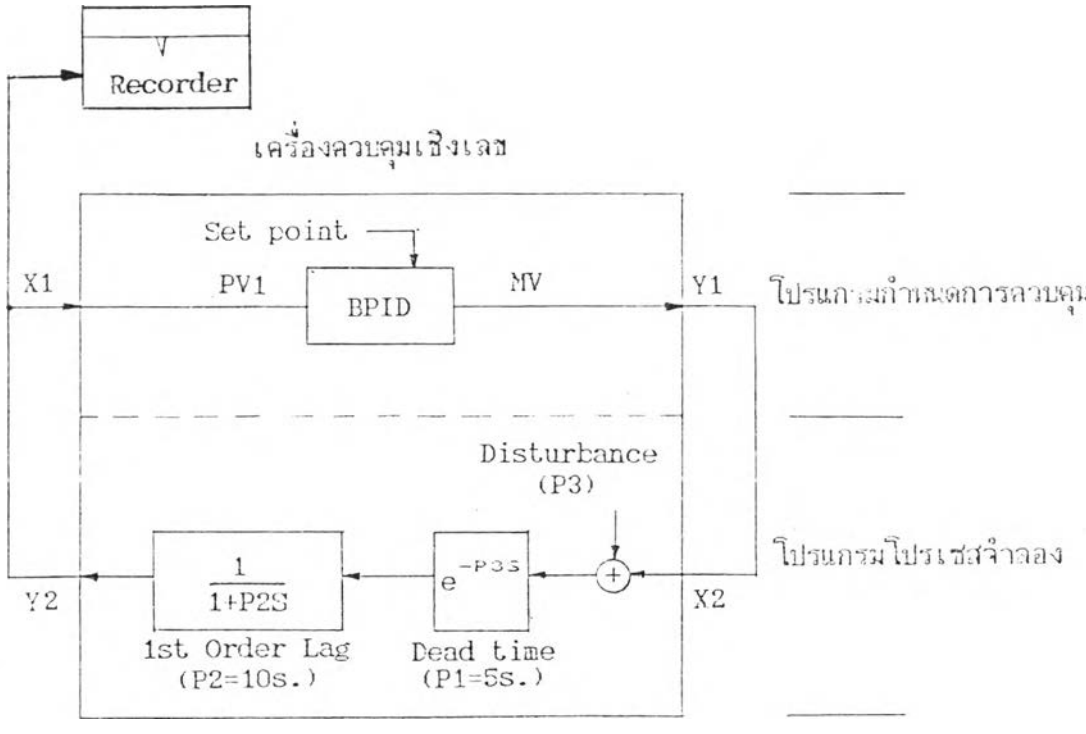
ผลการทดลอง ฟังก์ชันที่เขียนขึ้นสามารถทำงานได้ถูกต้องตามต้องการ

6.3.2 การทดสอบกับโปรเซส

แบ่งการทดสอบออกเป็น 2 ส่วน คือ

(1) **ทดสอบกับโปรเซสจำลองที่ถูกสร้างโดยโปรแกรม** (Process Simulation Program)

ไดอะแกรมที่ใช้ในการทดสอบแสดงดังรูปที่ 6.8



รูปที่ 6.8 ไดอะแกรมของโปรเซสจำลองที่สร้างจากโปรแกรมเพื่อใช้ทดสอบ

การทดสอบทำได้โดยใช้ฟังก์ชันภายในเครื่องควบคุมเขียนโปรแกรมสร้างโปรเซสจำลองที่มี dead time = 5 sec., lag time = 10 sec. โปรแกรมกำหนดรูปแบบการควบคุมและโปรแกรมสร้างโปรเซสจำลองเขียนได้ ดังรูปที่ 6.9 และเขียนเป็นภาษา Assembly ได้ดังรูปที่ 6.10 นำมาแปลงเป็น file.hex เขียนลง ROM โดยใช้ EPROM Programmer CLK3000 เพื่อควบคุมการทำงาน

```
-------------------------------------------------------------
Control Configuration
                LD    X1      ; read PV1
                BPID          ; PID Control
                ST    Y1      ; out control signal
-------------------------------------------------------------
Process Simulation Program
                LD    X2      ; read PV2
                LD    P3      ; read disturbance
                ADD           ; add disturbance
                LD    P1      ; read dead time
                DED1          ; dead time function
                LD    P2      ; read lag time
                LAG1          ; 1st order lag function
                ST    Y2      ; output channel 2
-------------------------------------------------------------
```

รูปที่ 6.9 โปรแกรม mnemonic กำหนดรูปแบบการควบคุมและสร้างโปรเซสจำลอง

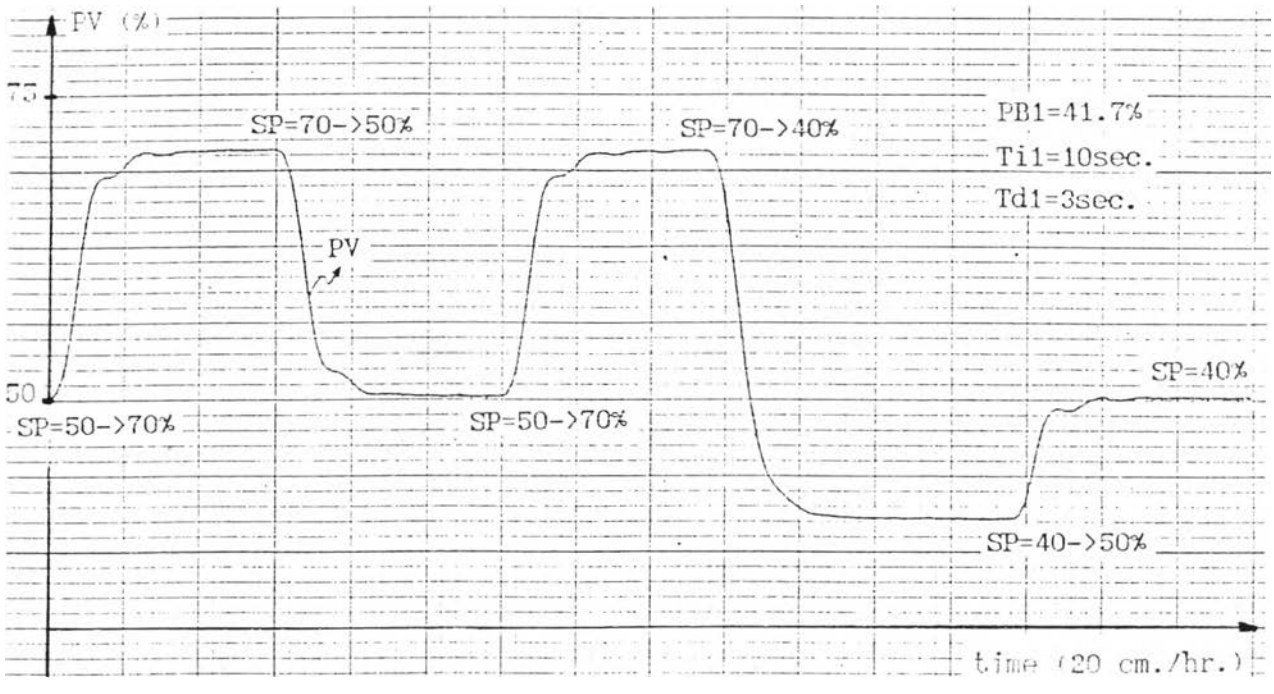
```
NAME  user_prg
ASSUME CS:CODE,DS:DATA
DATA  SEGMENT WORD PUBLIC 'DATA'
DATA  ENDS

CODE  SEGMENT WORD PUBLIC 'CODE'
userp proc far
      push bp
      push ax
      push bx
      mov  al,6h
      push ax
      mov  bx,785Ah
      call bx           ; ld x1
      mov  bx,82cah
      call bx           ; bpid
      mov  al,8h
      push ax
      mov  bx,7850h
      call bx           ; st y1
      mov  al,1h
      push ax
      mov  bx,785ah
      call bx           ; ld x2
      mov  ax,21h
      push ax
      mov  bx,7884h
      call bx           ; ld p3
      mov  bx,7faah
      call bx           ; add
      mov  ax,1bh
      push ax
      mov  bx,7884h
      call bx           ; ld p1
      mov  ax,1b8h
      push ax
      mov  ax,1baah
      push ax
      mov  bx,7c6ch
      call bx           ; decl
      mov  ax,1eh
      push ax
      mov  bx,7884h
      call bx           ; ld p2
      mov  ax,13ch
      push ax
      mov  bx,7a1eh
      call bx           ; dagl
      mov  al,1h
      push ax
      mov  bx,7850h     ; st y2
      call bx
      pop  bx
      pop  ax
      pop  bp
      ret
userp ENDP
CODE  ENDS
      END
```

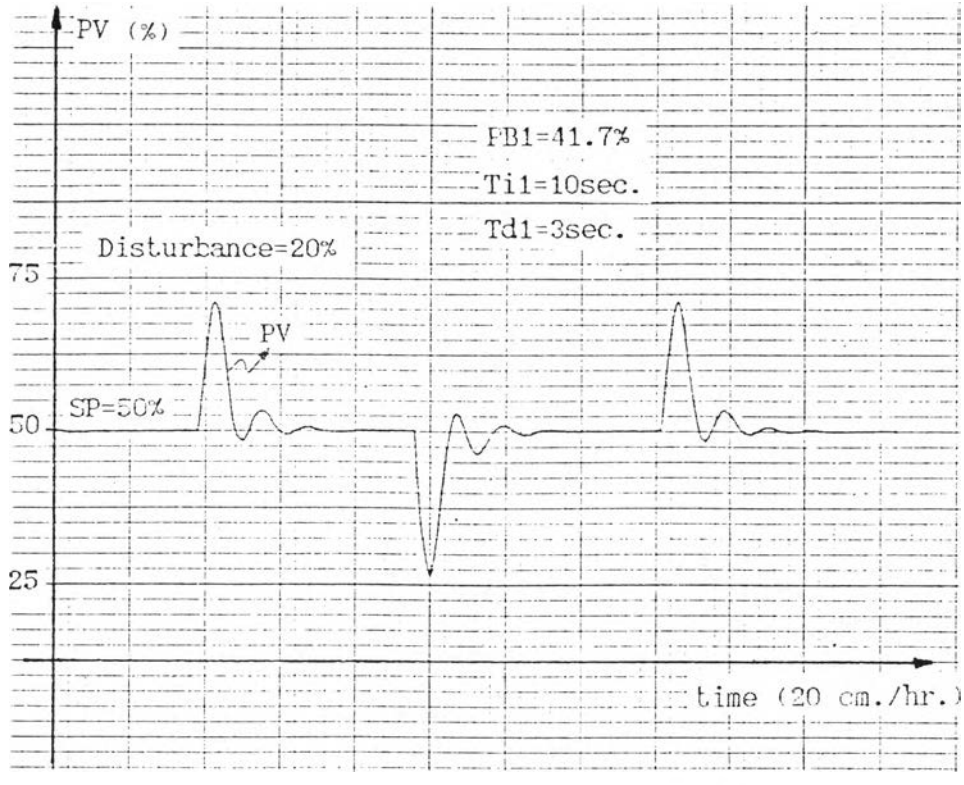
รูปที่ 6.10 โปรแกรม assembly กำหนดรูปแบบการควบคุมและสร้างโปรเซสจำลอง

ขั้นตอนการทดลองมีดังนี้

- Power On เครื่องควบคุมจะอยู่ในโหมด Manual โปรแกรมควบคุมระบบและโปรแกรมโปรเซสจำลองยังไม่ทำงาน
- ตั้งค่า PB1 = 41.7 %, Ti1 = 10 s., Td1 = 3 s.
- เปลี่ยนโหมดเป็น Auto โดยกดแป้นพิมพ์ A ด้านหน้าเครื่อง โปรแกรมควบคุมและโปรเซสจำลองเริ่มทำงาน
- เปลี่ยนค่า Set point โดยกดแป้นพิมพ์ SV เพื่อเปลี่ยนค่า Set point ได้กราฟผลการทดสอบดังรูปที่ 6.11
- เพิ่มค่า Disturbance โดยเปลี่ยนค่าพารามิเตอร์ P3 จากแป้นพิมพ์ด้านข้าง ผลการทดสอบดังรูปที่ 6.12



รูปที่ 6.11 กราฟผลตอบสนองของโปรเซสเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย



รูปที่ 6.12 กราฟผลตอบสนองของโปรเซสเมื่อมีสิ่งรบกวนระบบ

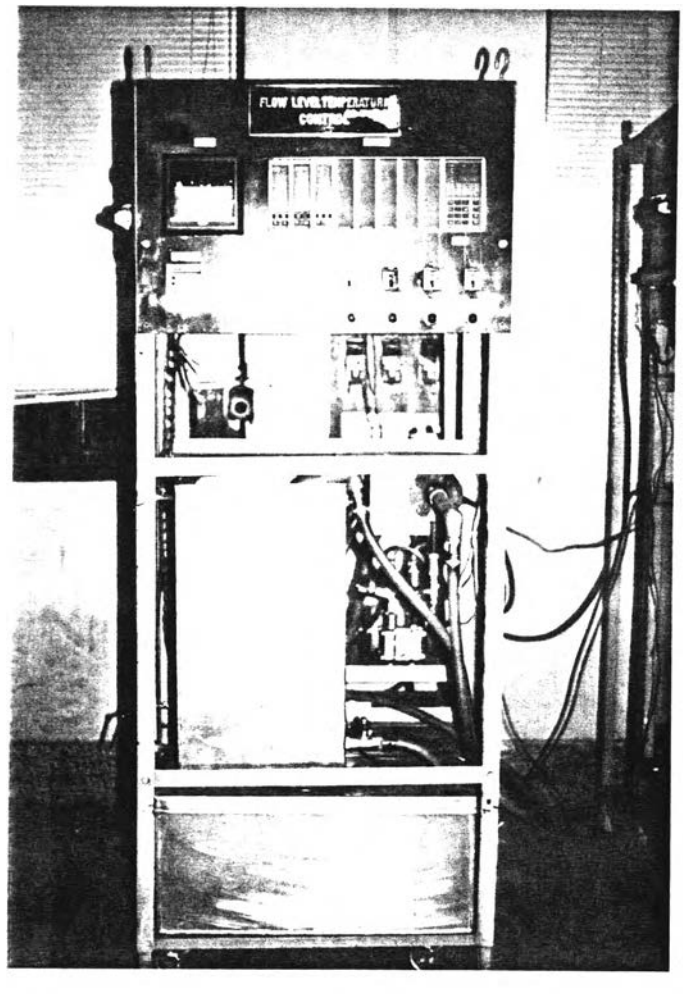
ผลการทดสอบ

การทดสอบกับโปรเซสจำลองที่สร้างจากโปรแกรม เครื่องควบคุมสามารถควบคุมการทำงานได้ เมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย หรือมีสิ่งรบกวนระบบ

(2) การทดสอบกับระบบจำลองทางอุตสาหกรรม

ทดสอบกับระบบจำลองของการไหล ระดับ และอุณหภูมิ (ดูรูปที่ 6.13) ซึ่งส่วนประกอบสำคัญที่ใช้ในการทดสอบคือ

- Ultrasonic Level Sensor
- Level Transmitter
- Control Valve
- Differential Pressure Transmitter
- Orifice
- Pump
- Tank

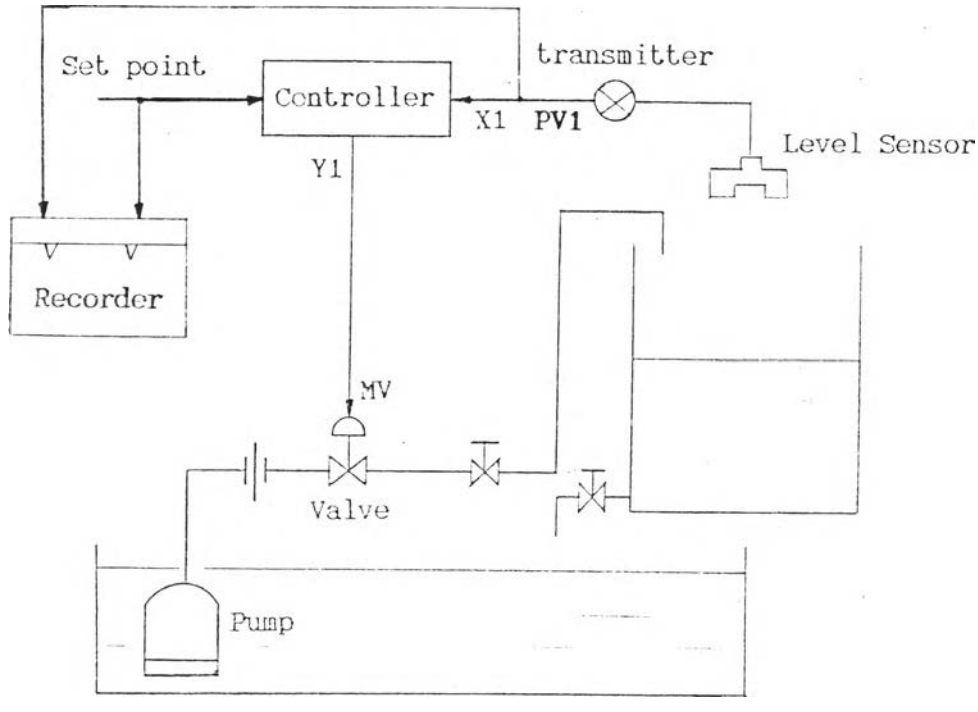


รูปที่ 6.13 ระบบจำลองการไหล ระดับ และอุณหภูมิ

การทดสอบจะใช้เครื่องควบคุมเชิงเลข ควบคุมระดับน้ำในถังของระบบจำลองทางอุตสาหกรรม โดยใช้รูปแบบวิธีการควบคุม 2 แบบ ดังนี้

(ก) การควบคุมแบบป้อนกลับแบบง่ายๆ

มีไดอะแกรมที่ใช้ในการทดลองดังรูปที่ 6.14



รูปที่ 6.14 ไดอะแกรมที่ใช้ในการทดสอบการควบคุมแบบง่ายๆ

ใช้ฟังก์ชันภายในเครื่องควบคุมเขียนโปรแกรมบนไมโครคอมพิวเตอร์ เพื่อกำหนดรูปแบบการควบคุมได้ดังรูปที่ 6.15 และเขียนเป็นภาษา Assembly ได้ดังรูปที่ 6.16 นำมาแปลงเป็น file.hex เขียนลง ROM

```
LD      SV1     ; read set point
ST      Y2      ; out to recorder
LD      X1      ; read process variable
BPID            ; PID control
ST      Y1      ; out control signal
```

รูปที่ 6.15 โปรแกรม mnemonic กำหนดรูปแบบการควบคุม

```
NAME user_rom
ASSUME CS:CODE,DS:DATA
DATA  SEGMENT  WORD  PUBLIC  'DATA'
DATA  ENDS

CODE  SEGMENT  WORD  PUBLIC  'CODE'
userp  proc  far
       push  bp
       push  ax
       push  bx
       mov   ax,7eh
       push  ax
       mov   bx,7884h
       call  bx          ; ld sv1
       mov   al,1h
       push  ax
       mov   bx,78f0h
       call  bx          ; st y2
       mov   al,0h
       push  ax
       mov   bx,785ah
       call  bx          ; ld x1
       mov   bx,82cah
       call  bx          ; bpid
       mov   al,0h
       push  ax
       mov   bx,78f0h
       call  bx          ; st y1
       pop   bx
       pop   ax
       pop   bp
       ret
userp  ENDP
CODE  ENDS
       END
```

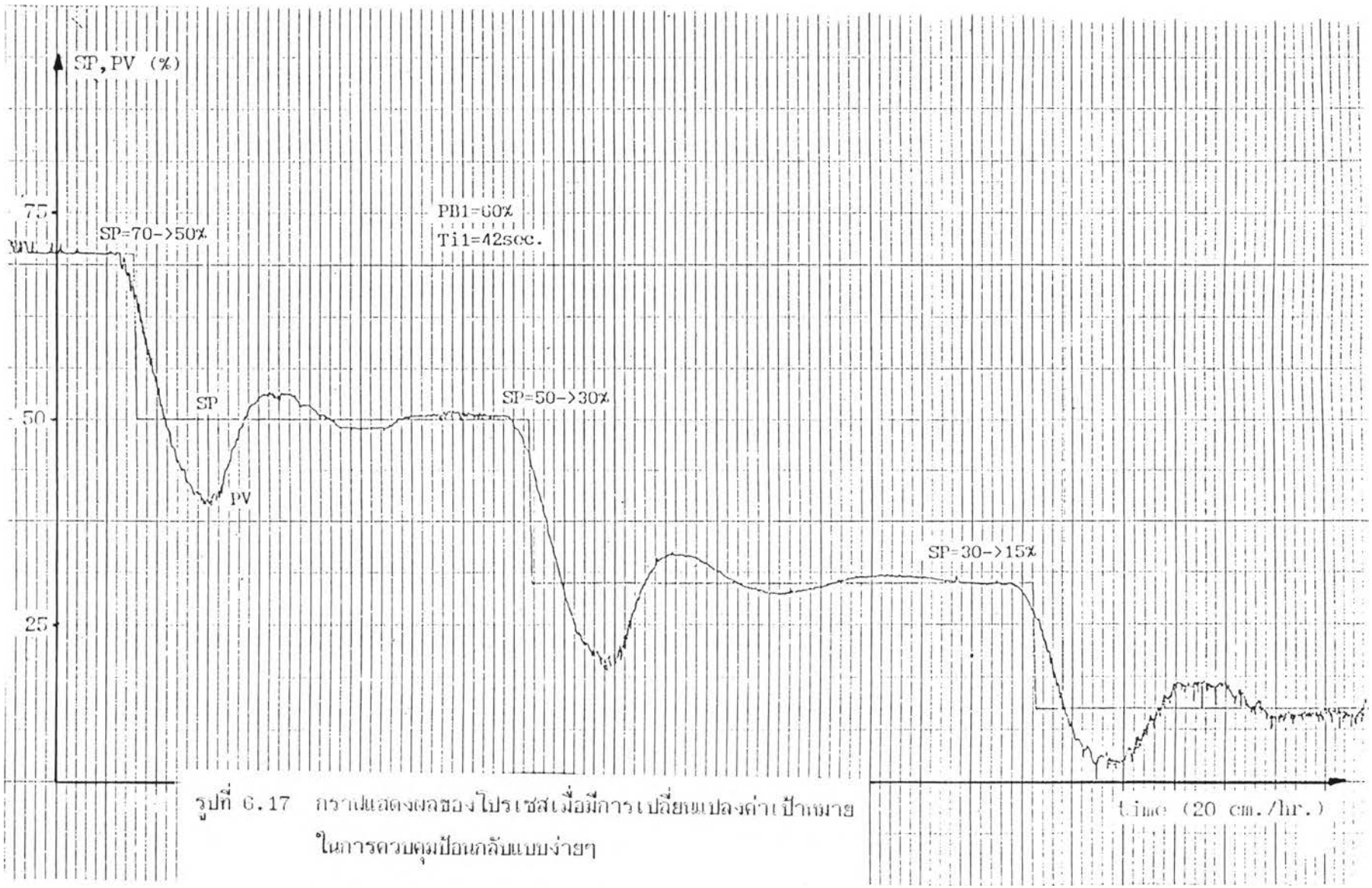
รูปที่ 6.16 โปรแกรม assembly กำหนดรูปแบบการควบคุม

ขั้นตอนการทดลอง

- Power On เครื่องควบคุมจะอยู่ในโหมด Manual
- ตั้งค่า PB1 = 60 %, Ti1 = 42 sec.
- เปลี่ยนโหมดเป็น Auto โดยกดแป้นพิมพ์ A ด้านหน้าเครื่อง
- เปลี่ยนค่า Set point โดยกดแป้นพิมพ์ SV เพื่อตรวจสอบผลการตอบสนองของเครื่องควบคุมเมื่อมีการเปลี่ยนแปลงค่า Set point

ผลการทดลอง

เครื่องควบคุมเชิงเลขสามารถควบคุมระดับน้ำให้เท่ากับค่าเป้าหมายของระดับน้ำที่ต้องการได้ถูกต้อง ดังรูปที่ 6.17

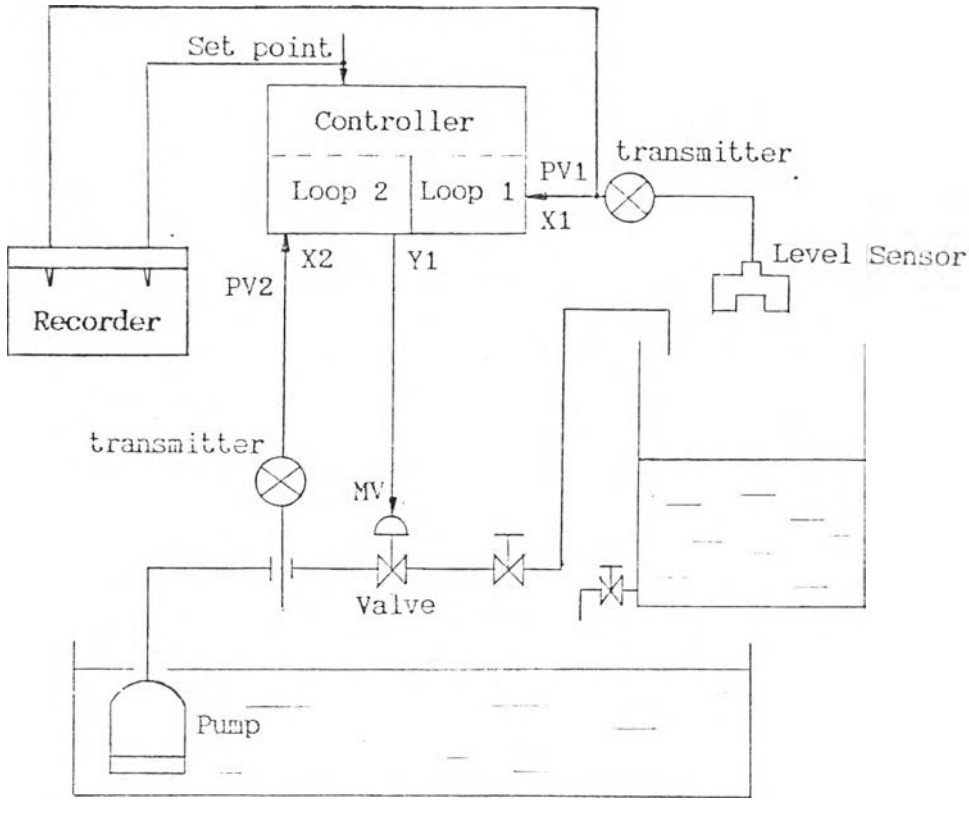


รูปที่ 6.17 กราฟแสดงผลของโปรเซสเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย ในการควบคุมป้อนกลับแบบง่ายๆ

รูปที่ 6.17 กราฟแสดงผลของโปรเซสเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย ในการควบคุมป้อนกลับแบบง่ายๆ (ต่อ)

(ข) การควบคุมแบบ Cascade

มีไดอะแกรมที่ใช้ในการทดลองดังรูปที่ 6.18



รูปที่ 6.18 ไดอะแกรมที่ใช้ในการทดสอบการควบคุมแบบ Cascade

ใช้ฟังก์ชันภายในเครื่องควบคุมเขียนโปรแกรมบนไมโครคอมพิวเตอร์ เพื่อกำหนดรูปแบบการควบคุมได้ดังรูปที่ 6.19 และเขียนเป็นภาษา Assembly ได้ดังรูปที่ 6.20 นำมาแปลงเป็น file.hex เขียนลง ROM

```
LD      SV1     ; read set point
ST      Y2      ; out to recorder
LD      X2      ; read flow PV2
SQR             ; square root function
LD      X1      ; read level PV1
CPID            ; Cascade control
ST      Y1      ; out control signal
```

รูปที่ 6.19 โปรแกรม mnemonic กำหนดรูปแบบการควบคุม

```
        NAME user_rom
ASSUME CS:CODE,DS:DATA
DATA    SEGMENT  WORD  PUBLIC  'DATA'
DATA    ENDS
CODE    SEGMENT  WORD  PUBLIC  'CODE'
userp   proc  far
        push  bp
        push  ax
        push  bx
        mov   ax,7eh
        push  ax
        mov   bx,7884h      ; LD SV1
        call  bx
        mov   al,1h
        push  ax
        mov   bx,78f0h
        call  bx            ; ST Y2
        mov   al,1h
        push  ax
        mov   bx,785Ah
        call  bx            ; LD X2
        mov   bx,80A6h
        call  bx            ; SQR
        mov   al,0h
        push  ax
        mov   bx,785ah
        call  bx            ; LD X1
        mov   bx,8550h
        call  bx            ; CPID
        mov   al,0h
        push  ax
        mov   bx,78f0h
        call  bx            ; ST Y1
        pop   bx
        pop   ax
        pop   bp
        ret
userp   ENDP
CODE    ENDS
        END
```

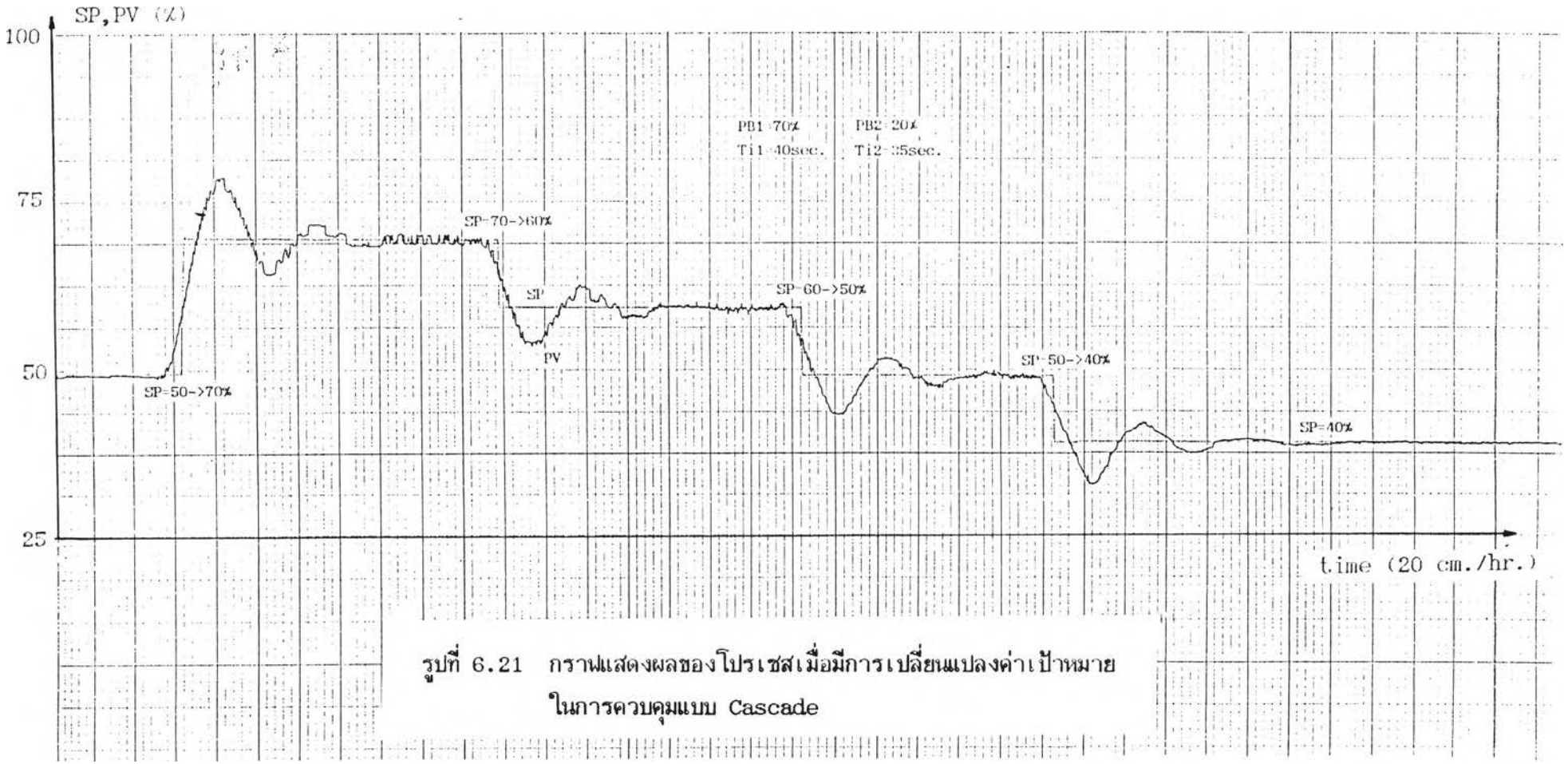
รูปที่ 6.20 โปรแกรม assembly กำหนดรูปแบบการควบคุม

ขั้นตอนการทดสอบ

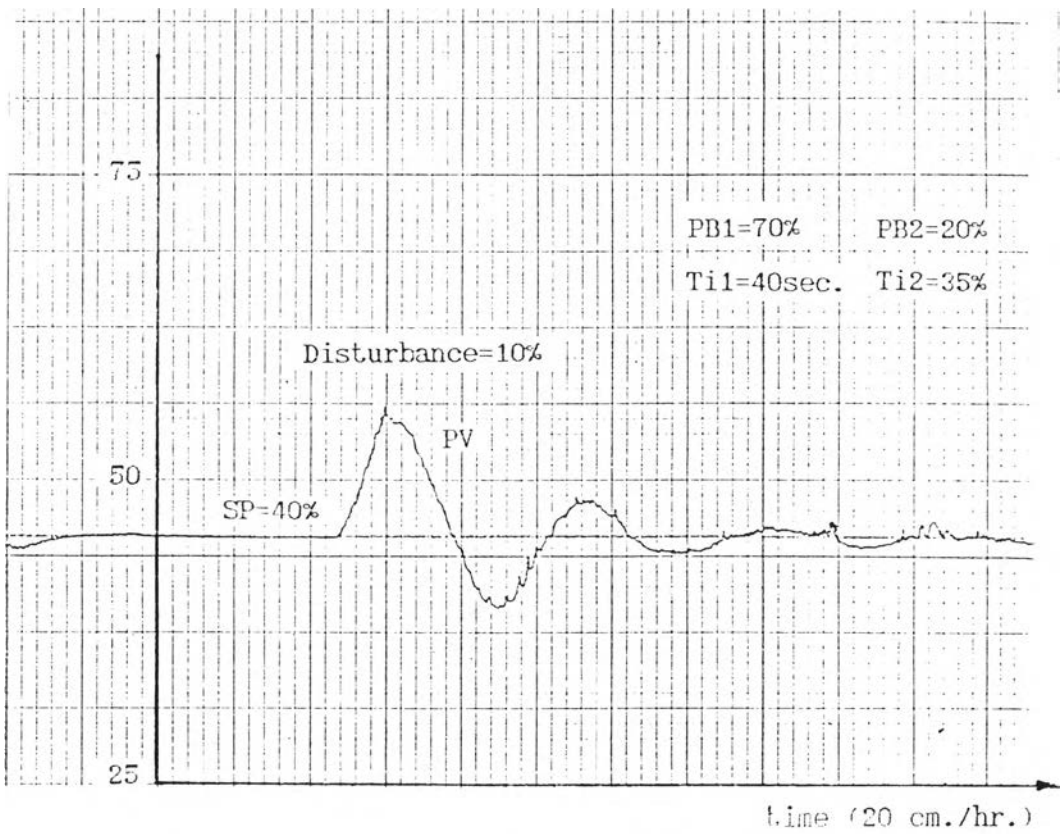
- Power On เครื่องควบคุมจะอยู่ในโหมด Manual
- ตั้งค่า PB2 = 20 %, Ti2 = 35 sec. ใช้แป้นพิมพ์ด้านข้างเครื่อง
- เปลี่ยนโหมดเป็น Auto โดยกดแป้นพิมพ์ A ด้านหน้าเครื่อง เครื่องควบคุมจะตัดลูปที่ 1 (ลูปใน) ออกจากระบบ เพื่อทำการปรับ (Tuning)ค่า PB,Ti และ Td ที่เหมาะสมของลูปที่ 2 (ลูปนอก)
- ตั้งค่า PB1 = 70 %, Ti1 = 40 sec.
- กดแป้นพิมพ์ C เพื่อเปลี่ยนโหมดเป็น Cascade control
- เปลี่ยนค่า Set point โดยกดแป้นพิมพ์ SV เพื่อตรวจสอบผลการตอบสนองของเครื่องควบคุมเมื่อมีการเปลี่ยนแปลงค่า Set point ดูรูปที่ 6.21
- ใส่สิ่งรบกวนระบบเข้าไป โดยเติมน้ำเข้าไปในถัง โดยที่ค่าเป้าหมายของระดับน้ำเท่าเดิม ได้ผลการตอบสนองดังรูปที่ 6.22

ผลการทดสอบ

เครื่องควบคุมเชิงเลขสามารถควบคุมระดับน้ำให้เท่ากับค่าเป้าหมายของระดับน้ำที่ต้องการได้ถูกต้อง เมื่อมีการเปลี่ยน Set point หรือเมื่อมี Process Disturbance



รูปที่ 6.21 กราฟแสดงผลของโปรเซสเมื่อมีการเปลี่ยนแปลงค่าเป้าหมาย ในการควบคุมแบบ Cascade



รูปที่ 6.22 กราฟแสดงผลของโปรเซสเมื่อมีสิ่งรบกวนระบบ
ในการควบคุมแบบ Cascade