

13. Microcomputer-Based Control Systems

Chapter 13 objectives are to

- Introduce the general approach to digital control systems
- Design and implement some simple open loop control systems,
- Design and implement some simple closed loop control systems,
- Develop a methodology for designing PID control systems,

13.1. Introduction to digital control systems

$$e(t) = X^*(t) - X'(t)$$

then the control system will attempt to drive $e(t)$ to zero.

the effectiveness of a control system by determining three properties:

- steady state controller error,
- transient response, and
- stability.

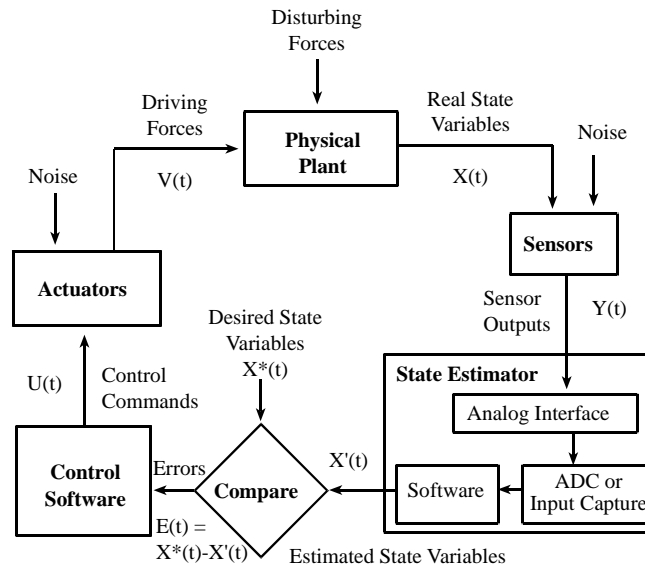


Figure 13.1. Block diagram of a microcomputer-based closed-loop control system.

FSM implementation of a line-tracker

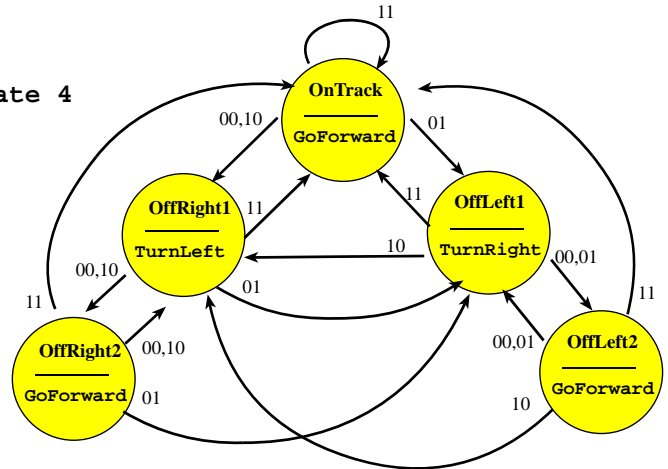
```

const unsigned char RTable[4]={0x05,0x06,0x0A,0x09};
const unsigned char LTable[4]={0x50,0x60,0xA0,0x90};
int Ri;          // right wheel index
int Li;          // left wheel index
void GoForward(void){
    Ri = 0x03&(Ri+1); // move right motor
    Li = 0x03&(Li+1); // move left motor
    PTT = RTable[Ri]+LTable[Li];}
void TurnLeft(void){
    Ri = 0x03&(Ri+1); // move right motor
    PTT = RTable[Ri]+LTable[Li];}
void TurnRight(void){
    Li = 0x03&(Li+1); // move left motor
    PTT = RTable[Ri]+LTable[Li];}
    
```

```

const struct State{
    void (*CmdPt)(void);    /* function to execute */
    const struct State *Next[4];}; /* Next states */
typedef const struct State StateType;
typedef StateType * StatePtr;
#define OnTrack &fsm[0]
#define OffLeft1 &fsm[1]
#define OffLeft2 &fsm[2]
#define OffRight1 &fsm[3]
#define OffRight2 &fsm[4]
StateType fsm[5]={
    {&GoForward, {OffRight1,OffLeft1,OffRight1,OnTrack}}, /* On the track */
    {&TurnRight, {OffLeft2,OffLeft2,OffRight1,OnTrack}}, /* Off little left */
    {&GoForward, {OffLeft1,OffLeft1,OffRight1,OnTrack}}, /* Off left */
    {&TurnLeft, {OffRight2,OffLeft1,OffRight2,OnTrack}}, /* Off little right */
    {&GoForward, {OffRight1,OffLeft1,OffRight1,OnTrack}}}; /* Off right */
StatePtr Pt;
unsigned char Input;
void main(void){
    TSCR1 = 0x80;    // Enable TCNT, default rate 4
MHz
    DDRT = 0xFF;    // PortT 7-0 are outputs
    DDRM = 0x00;    // PortM 1-0 are inputs
    PTT = 0x55;    // energize stepper
    Pt = OnTrack;    // Initial State
    while(1){
        (*Pt->CmdPt)(); // 1) execute function
        Input = (PTM)&0x03; // 2) input
        Pt = Pt->Next[Input]; // 3) next
    }
}

```



13.3.1. Bang-bang Temperature Control.

Other names for Bang-Bang include Two-position, On-off, or Binary Controller. The hysteresis extends the life of the relay by reducing the number of times the relay opens and closes.

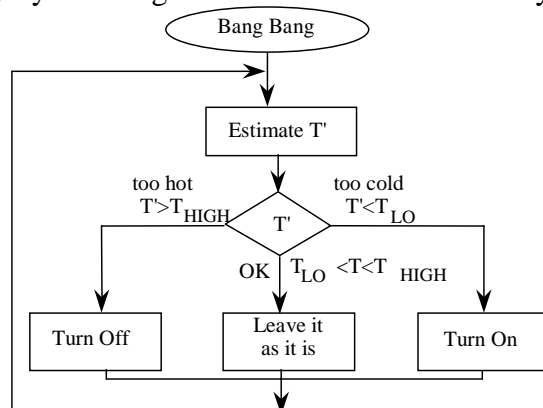


Figure 13.7. Flowchart of a Bang Bang Temperature Controller

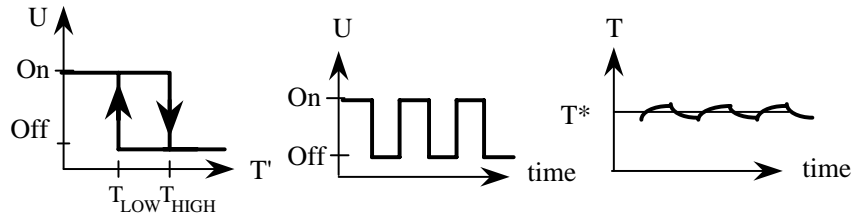


Figure 13.8. Algorithm and response of Bang Bang Temperature Controller

13.3.2. Closed-Loop Position using Incremental Control

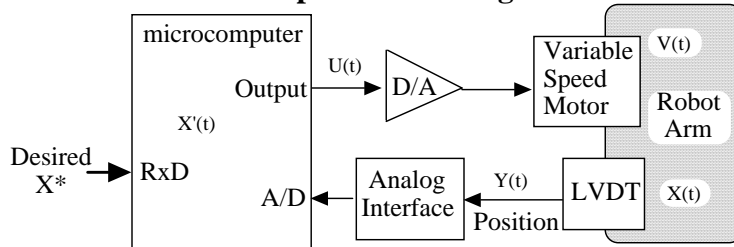


Figure 13.10. Interface of a position controller.

Common error: An error will occur if the software does not check for overflow and underflow after U is changed.

Observation: If the incremental control algorithm is executed too frequently, then the resulting system behaves like a simple bang-bang controller.

The rate is about 10 times faster than the time constant of the physical plant.

```
interrupt 13 void TC5handler(void){ int new;
  X=SE(A2D(channel)); // estimated (mm)
  E=Xstar-X; // error (mm)
  new=0x00FF&PTT; // promote to 16 bits
  if(E< -1) new--; // E<-1 so decrease
  else if (E>1) new++; // E>+1 so increase
  // leave as is if -1<E<1
  if(new<0) new=0; // underflow
  if(new>255) new=255; // overflow
  PTT=new; // output to actuator
  TC5=TC5+rate; // periodic rate
  TFLG1=0x20; } // ack C5F
```

Observation: Incremental control will work moderately well (accurate and stable) for an extremely wide range of applications. Its only short-coming is that the controller response time can be quite slow.