ME 6102: Design of Mechatronics Systems

Closed loop control of motor

P.S. Gandhi
Mechanical Engineering
IIT Bombay

PRASANNA S GANDHI
gandhi@me.iitb.ac.in

Fundamental Points

- In automatic control systems (example motor here) there are following tasks to be completed continuously:
  1. Read sensors (potentiometer in our case: ADC interface)
  2. Compute control law (Say P or PD control)
  3. Give control input computed in 2 to actuator (Motor in our case: via PWM interface duty cycle)
- Keep on doing these tasks at every "sampling time" set by using PIT module (1ms in this case)

Putting Things Together

Today's lab we will put all things together to feedback control motor

- Q: what will we use as feedback?

Position feedback from encoder.

- Q: what is proportional control law?

V = Kp * (desired position - actual position)

- Q: how will you apply this law to motor? What controls voltage seen by the motor?

PWM duty cycle. Assign the duty cycle a value proportional to V. Think what if the value obtained is – ve. Think what should happen and how..

- Q: what should we set as a desired position??

Putting Things Together: proportional control

V = Kp * (desired position - actual position)

- Q: what will happen if we keep the desired position constant?

motor will move from whatever current position to desired and we will not have enough opportunity to observe how control is working. We will need to re load the program by changing motor position.

- Q: How can we observe continuous operation of control action?

Generate a square wave desired position signal for motor to follow this position.

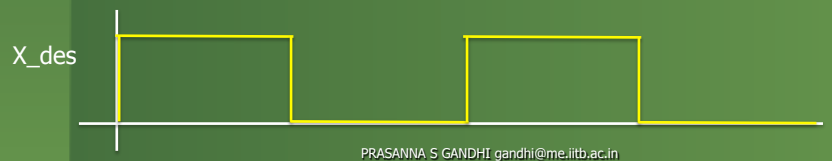
Putting Things Together

- Q: what should be amplitude of square wave?

Should roughly correspond to 2 rev of the wheel attached to gearbox shaft so that we can observe things nicely

- Q: what should be frequency of square wave?

Much larger than sampling time and such that we can observe the response of motor visually



Putting Things Together

- Q: what will be typical response of motor?.
- Our goal is to get the data recorded and processed in MATLAB to observe similar response, we may not have overshoot
- We set zero for encoder position (observe value of Enread =0) and implement

- What will you do if the response observed is exactly same as desired position?? Decrease sampling time

Typical Program

```
*/
#include <hidef.h>          /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */

void initPIT( void );     // function declaration for initiation of PIT module
void initPWM( void );
void initADC( void );

void main(void) {
  /* put your own code here */
  initPIT();
  initPWM();
  initADC();
    EnableInterrupts;
  for(;;) {
     _FEED_COP(); /* feeds the dog */
  } /* loop forever */
  /* please make sure that you never leave main */
}
```


Typical Program

```
//we can write VectorNumber_Vpit0 or 66- refer MC9S12XEP100.h file
Line no.173

void interrupt 66 MotorControl ( void )     //VectorNumber_Vpit0  and
MotorControl can be any name
{

     // Clear the interrupt flag
     // calculate updated desired position
     // Read ADC position use appropriate bias and gains
     // compute motor control law Proportional in this case
     // update duty cycle of PWM

}
void initPIT( void )
  { //You will initiate various registers pertaining to PIT in this program
}
Similar programs for ADC and PWM
```

Derivative control action

- We can implement it by using backward difference formula
- x_dot ={ x(i) – x(i-1)} / T f where T is sampling time.
- V = kp (x-x_ref) + kd x_dot
- Q: how can we know x(i – 1)?

Additional Task

- If you are done with PD control carry out the following task
- We would like to generate sinusoidal waveform as desired waveform instead of square waveform we did so far in XEP 100. Think of the way to do it and implement. Show the algorithm and output to TA.

Thank You