Embedded Controls Final Project

Tom Hall
EE 554
12/07/2011

Introduction:

The given task was to design a system that:

- -Uses at least one actuator and one sensor
- -Determine a controlled variable and suitable sampling interval for this system
- -Model if possible the system and compare the behavior of the model to the behavior of the actual system
- -Design and tune a controller to meet your design specifications, and compare the simulated performance to the actual performance of the system

The project developed to meet these criteria is to create a mobile unit that would avoid obstacles. This project was selected to create a platform that could be used and upgraded for other projects.

Hardware Design:

The hardware specifications designed for this project are as follows:

- -The unit needs to have at least one distance detecting sensor
- -The unit should use two motors
- -The controller should be an Arduino Duemilanove

The arduino duemilanove is a microcontroller based on the Atmega328 chip. It has the following specifications:

- -Bootable Memory: 1 Kb
- -Clock Speed: 20 MHz
- -A/D: 8-channel 10-bit resolution
- -GPIO Lines: 23
- -Temperature Range: -40°C to 85°C
- -Power Consumption: 0.2 mA

Original Design

The original system used two infrared emitter and detector pairs to determine the distance to an object. This combination detects the distance by emitting an infrared wave that would reflect off an object and return to the detector. As an object gets closer to the sensor the detector will receive the emitted wave faster. The original system also used a pair of low power motors that could be driven by the microcontroller through a pulse width modulated (PWM) signal. A PWM signal outputs a series of 5 volt pulses whose frequency is determined by a duty cycle. For example a duty cycle of 50% would have the motor running at half speed while a duty cycle of 100% would have the motor running at full speed. A diagram of the design can be seen in figure 1.

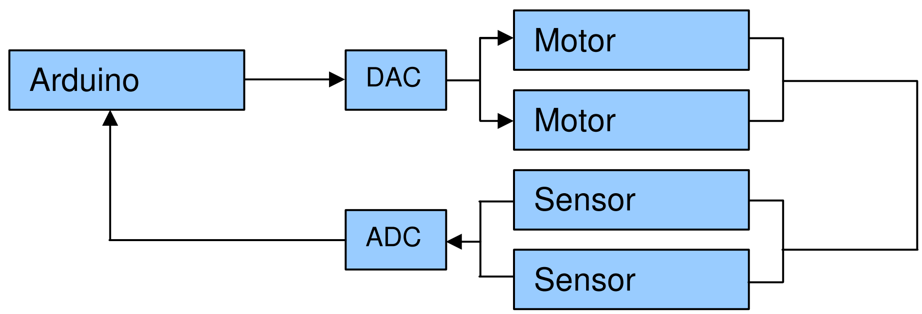

Figure 1: A block diagram of the original design

Issues with the original design

Due to funding and time constraints the parts to make the unit mobile were unable to be purchased. As a result an agreement was reached with Dr. Erives where the "mobile" unit could be laid out on a breadboard and tested to show a proof of concept. However a couple of hours before the project was to be presented the pair of infrared sensors were no longer responding to the program. The sensors output only zeros. After much troubleshooting it was necessary to start looking at another design.

Current Design

Because of the complications with the infrared sensors a new ultrasonic sensor was selected to replace the original sensors. This sensor measures distance by emitting a pulse that bounces off objects and reading the echo. The sensor requires a small 10us pulse to initiate the pulse sequence then requires 10ms to read the echo. To use this sensor a input capture device needed to be designed in order to determine the distance. An input capture device takes in a signal and determines the time between the current signal and the previous. Because of the limited amount of time, only one sensor could be implemented. This results in only one axis of movement because the sensor can only sense one dimension. Because of this there would only be a need for one motor which suited the time constraints. A diagram of this design can be seen in figure 2.

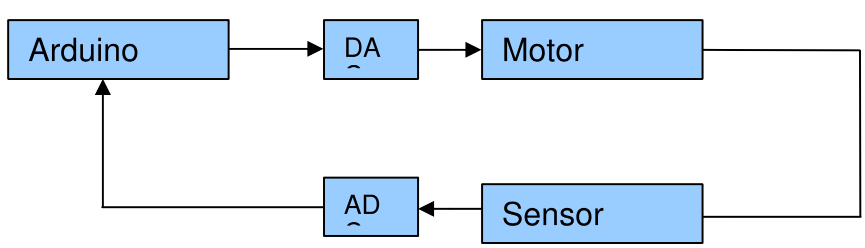

Figure 2: A block diagram of the current design

<u>Controller:</u>

The selected controller was a PID controller. This was because the PID is able to control an overshoot while keeping the system stable. The characterization of the infrared sensors can be seen in figure 3. This wave form is the readout from the sensor as the distance to an object is decreased. Figure 4 contains the characterization of the ultrasonic sensor. This characterization was used to calculate the tau value for the controller. The zeta value was then varied until a more stable signal was found. From there the other constants and coefficients were calculated. These values were then put into equation 1.

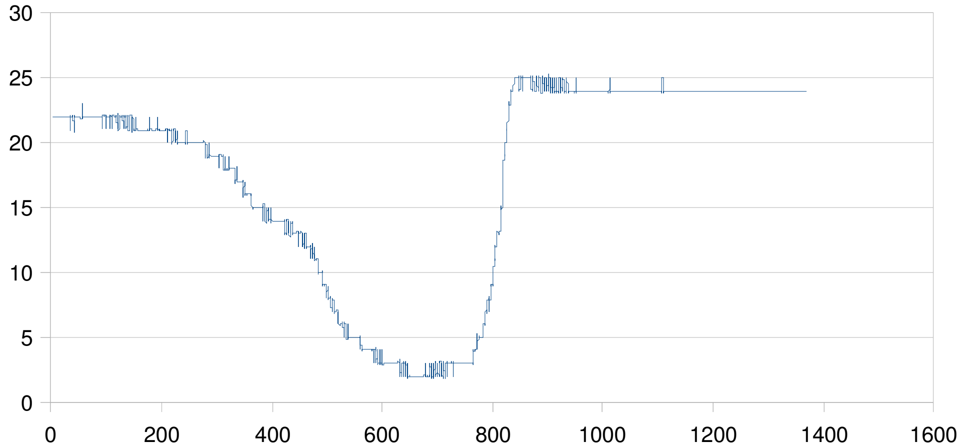

Figure 3: Characterization of the infrared sensor over time

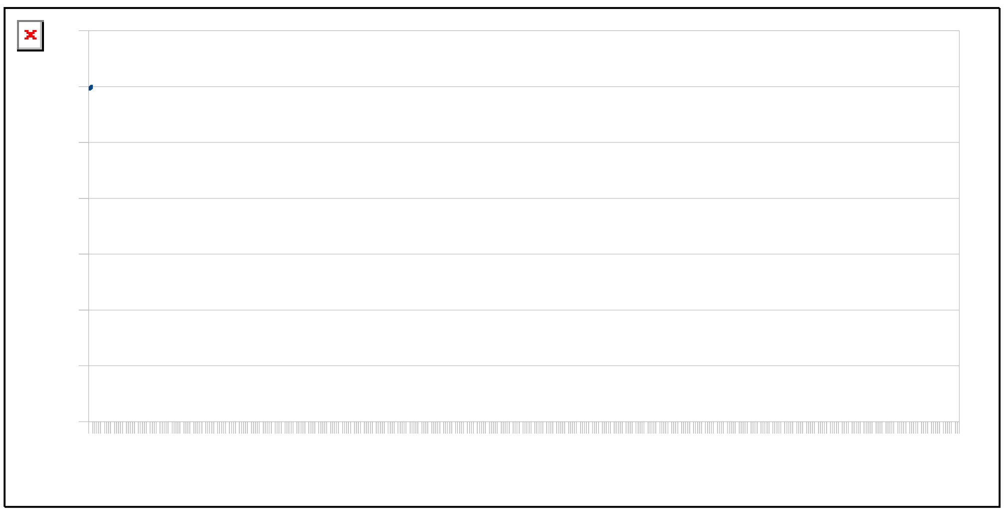

Figure 4: Characterization of the ultrasonic sensor over time

$$u(t_k) = u(t_{k-1}) + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right]$$

Equation 1: Formula used to implement the PID controller

Software Design:

The software design was fairly straightforward. First the 10us pulse is sent to initiate the ultrasonic sensor. After reading the distance, the program applies the new value to the controller which then corrects the motor speed. Finally PWM's duty cycle is calculated by normalizing the sensor value and in the process adding in the correction value obtained from the controller. The code can be seen in appendix 1.

Results:
The expected behavior of this system is to run the motor and adjust the motor speed as the object moves closer or further away. Since the system was immobile a distance was set that the motor had to "get" to. Next an object was moved closer and further away to the sensor to simulate the system moving. When the object was moved closer the motor would slow and when the object was moved further away the motor sped up. When the object would approach the desired distance the motor would stop. There was a slight margin of error that the controller could reduce with more time, however for this project the system was successful.

Conclusion:
The system that was designed met all applicable criteria that was needed for the project. The system used one motor as an actuator and one ultrasonic sensor as a sensor, the controlled variable was the motor speed, the characterization of the system was completed and then used to design a PID controller for the system. In the future two more of the ultrasonic sensors will be added to the system so that it can eventually navigate better. Also an amplifier circuit will be designed to amplify the voltage coming from the PWM so that higher power motors can be used. Finally a PCB will be designed to take the system off a breadboard.

Bibliography

Arduino - HomePage. Web. 07 Dec. 2010. <http://www.arduino.cc/>.

Fadali, M. Sami., and Antonio Visioli. *Digital Control Engineering: Analysis and Design*. Amsterdam [u.a.: Elsevier, Academic, 2009. Print.

"PID Controller." *Wikipedia, the Free Encyclopedia*. Web. 07 Dec. 2010. <http://en.wikipedia.org/wiki/PID_controller>.

Appendix 1: Code

```
int delta=0;
int prev=0;
#define del_t 10                    //Defines constants to be used in the controller
#define tau -5.5
#define zeta .3105
#define w_n (1.0/(tau*zeta))
#define k_d w_n*w_n
#define k_p (k_d*2.0*zeta*w_n)
#define k_i (w_n*w_n*k_d)
#define t_i (k_p/k_i)
#define t_d (k_d/k_p)

float e=0;
float e_1=0;
float e_2=0;
float u=0;
float u_1=0;


int ref=625;                        //Sets the distance

void setup(){
  Serial.begin(9600);               //Sets baud rate for serial communication
  pinMode(3, OUTPUT);               //Sets up pins to be used
  pinMode(9, INPUT);
  pinMode(6, OUTPUT);

}

void loop(){
  digitalWrite(6,HIGH);             //Sends a 10us pulse
  delay(.01);
  digitalWrite(6,LOW);
  //int val = digitalRead(9);

  //Serial.println(val);

  delta = pulseIn(9,HIGH);          //Input capture function
  Serial.println(delta);            //prints to serial communicator
  delay(del_t);

  int e=ref-delta;
                                    //Controller
  u=(int)(u_1+(e*(1.0 + del_t/t_i+t_d/del_t)+e_1*(-1.0-2.0*t_d/del_t)+e_2*(t_d/del_t))*k_p);
  u_1=u;
  e_2=e_1;
  e_1=e;
```

```
  //Serial.println(u);

  if (delta < 0 || delta > 10000){
    //analogWrite(3,255);
  }
  else{
    analogWrite(3,((25*(delta+u))/1000));     //PWM function
  }
//  Serial.print("Sensor 1 = ");
//  Serial.print(sensor1);
//  Serial.print("     Sensor 2 = ");
//  analogWrite(3, 8*sensor1);
//  analogWrite(5, 8*sensor2);

}
```