Date: 4/25/06
Student Name: Jon Azevedo
TA: Adam Barnett
Sara Keen

HIWAA

EEL 5666: Intelligent Machines Design Laboratory

University of Florida,

Drs. A. Antonio Arroyo and E. M. Schwartz, ECE

Table of Contents

Abstract

My project is a test platform for an autonomous highway vehicle. It has line tracking and adaptive cruise control capabilities. My robot is set to an initial velocity until an object interferes with its path. It then adjusts to the respective speed of the object and sets a safe distance. Once the object is out of the robots path it resumes to its original speed.

Executive Summary

HIWAA's chassis is completely made out of aluminum. I cut the body out of a 1/16 plate of aluminum. For the motor mounts, I milled them out of a solid piece of aluminum, along with the casing for the steering bearing, sonar bracket, and the servo mount. I used the lathe to turn down the aluminum for the rear motor shafts, steering column, and the front axel. The front axel, steering column, and the casing for the steering bearing were high precision parts. They were all press fitted with the steering assembly having a double press fit.

I knew that I was going to have HIWAA perform line tracking as one of his duties so I put extra time in designing the steering. I did not want to have the motors steer, as this produces choppiness. I wanted to design a simple and effective steering column that would allow both the rear wheels to be moving in the turns. I came up with having a 3/8" front axel with a threaded hole in the center. The steering column that screws into the axel consists of two pieces. One pieces attaches to the servo and the other is attached to the axel. The steering column is press fitted into a roller bearing, which is press fitted into the bearing casing. The casing is directly bolted to the chassis.

With this steering design HIWAA performs line tracking very smooth. The downfall to this design compared to steering with the motors is that my steering cannot turn ninety degrees. On the other hand, HIWAA is built as an autonomous highway vehicle so he only needs to be able to turn around bends on a highway.

I decided to have a sensor suite of sonar, encoder, photo reflectors, and a bump switch. I have the sonar to determine the distance, velocity, and acceleration of the vehicle that is directly in front of HIWAA. The encoder is to determine HIWAA's speed. The photo reflectors are to detect a contrast and follow a designated path. HIWAA has bump sensors for obstacle avoidance.

Between the design of HIWAA and his sensor suite he is able to track a line of any orientation within his turning radius and perform adaptive cruise control. The adaptive cruise control measures the acceleration of the car in front of HIWAA and the matches HIWAA's acceleration to match it. The same is true for deceleration.

Introduction

This project is inspired to make driving safer and more convenient. Diving is the most dangerous form of traveling, especially on the highway. With the technology that is gained from my platform, highway driving will be next to flawless. By eliminating human error of swerving, following too close and slow reaction, there will be a significant decrease in accidents.

Integrated System

The core system of my robot has all the components of a "smart" robot. The brain of my robot is an Atmega 128 microprocessor. The microprocessor controls several important functions. These functions include: line tracking, speed, distance traveled, and the speed of other objects. Photo reflectors are going to be used to determine the lines that form a lane. Speed and direction of my robot are determined by shaft encoders that are present on both of the motors. The speeds of the other objects that are in the way of my robot are measured by way of sonar. Steering will be done with a solid front axel and a single servo attached at the center point. There is also a LCD screen that will output important information that my robot will be computing.

Mobile Platform

My platform is made out of aluminum to support the weight and the torque of the DC motors used.

Actuation

My robot uses two Maxon DC motors to drive itself. There is one motor located in the left rear and one in the right rear. Each motor has a 6:1 gear head on it to reduce the output shaft speed to around 650 rpm at 12 volts. The output speed of the motors is linear to the amount of input voltage. The max voltage I am going to be powering the motor is 6 volts. With four inch wheels and a 6 volt input my robot will do around 2 MPH. The actuation for the steering is a high torque servo connected to the midpoint of the solid front axel.

Sensors

There are several sensors implemented on my robot. The sensors used are as follows: pressure sensors, photo reflectors, encoders, and sonar. There is one pressure sensor in the front of the vehicle and one in the back of the vehicle. These sensors are implemented into the platform as a safety switch to protect the motors. When the pressure sensor is pressed it grounds the circuit and kills the power going to the motors. Photo reflectors are going to be used to line track. The way they work is by transmitting a light, via a light emitting diode and a phototransistor for detection. The more light that is reflected to the phototransistor, the higher the output voltage is. The amount of light that is reflected is greatly dependent of the color of the surface. The shaft encoders are used mainly to determine the speed of my robot. The way shaft encoders work, is by counting the revolutions. By knowing the wheel diameter, the distance is known. Now, if you integrate the distance with respect to time you have the speed. Then if you integrate speed with respect to time you have the acceleration. Sonar is used as obstacle avoidance and to measure to respective speed of objects that are in front of my robot. The sonar works the same way as the encoders. It will measure the distance and then integrate it with respect to time and then take the difference between that reading and the reading of the encoder to get the speed of the object. The sonar will also use the distance reading to make sure that my robot is following the object at a safe distance.

Behaviors

HIWAA has a multitude of behaviors. The behaviors are determined by feedback from the sensor suite. Some of the behaviors are determined from a combination of sensors and some from just one.

The simplest of HIWAA's behaviors is his obstacle avoidance from his bump sensors. If HIWAA bumps his from bump switches, he stops, reverses, and then takes a reading with his sonar to see if he can move forward. If his rear bump sensors are pressed he stops, moves forward, and then measures with his sonar to see if he can continue moving forward.

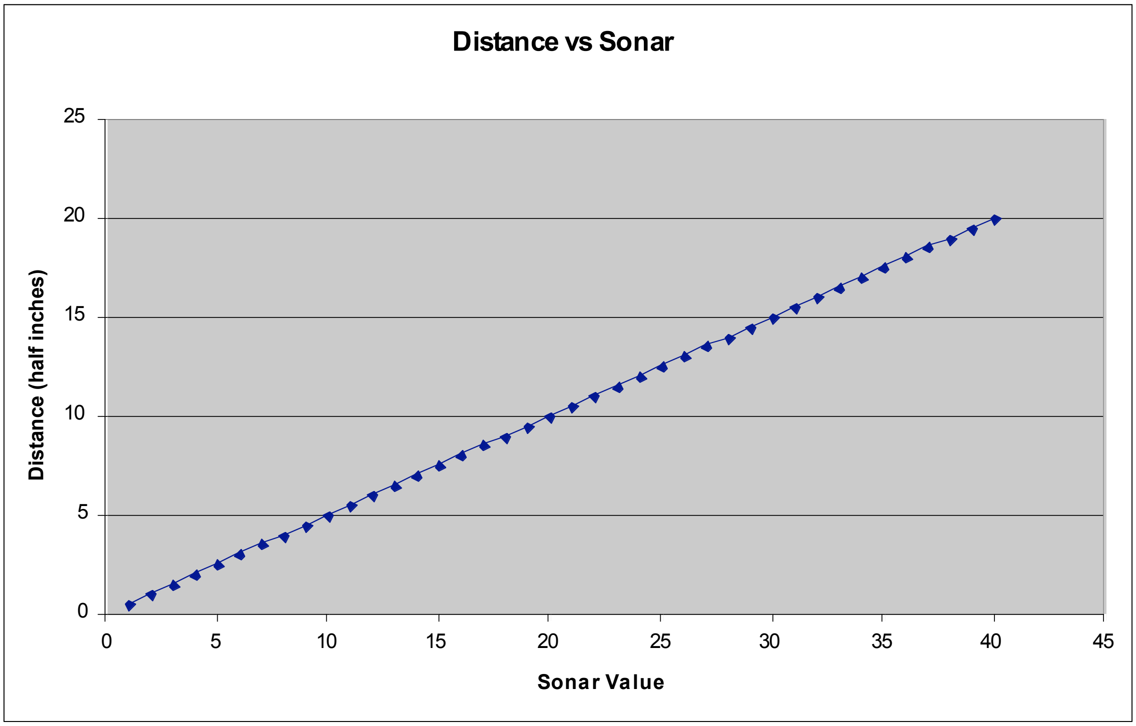
HIWAA's sonar creates lots of his behaviors. One behavior is if the sonar measures anything within six inches in front of him he stops until the object moves out of that range. If there is a vehicle within thirty-six inches of highway he engages adaptive cruise control. Adaptive cruise control works by having a velocity and acceleration algorithm.

Basically HIWAA matches the acceleration and velocity of the leading vehicle, all while keeping a safe distance.

HIWAA has two line tracking modes. One mode is enabled if there are no vehicles within thirty-six inches of his front bumper. This mode of line tracking has the line tracking determine the speed of HIWAA. The stringer presence of the line that HIWAA has the faster he can go. The weaker the presence of the line the slower he goes. This boils down to HIWAA going fast on straight a ways and slowing down for turns. The other mode of line tracking is if HIWAA is following a vehicle. If HIWAA is following a vehicle he still line tracks, but at the speed of the leading vehicle with the help of the adaptive cruise control.

Experimental Layout and Results

Luckily for me most of the sensors that I used for HIWAA are cut and dry. By that I mean that they either work or they don't. The only sensor that I had to experiment with numerically is the sonar. I initially set up the sonar on an interrupt that fired every micro second in order to have very good accuracy. Unfortunately this was tying up too much of the processor, which was quickly found when adding on other behaviors. Since a lot of HIWAA's features come from the sonar it was important to calibrate it while maintaining maximum resolution. With setting it up on a forty-eight second interrupt, the best resolution was about a half an inch. This is portrayed in the graph below.



Conclusion

All said and done I am really impressed with how HIWAA turned out. He drives fast and smooth on a straight or curved line. I am equally happy with how well HIWAA's adaptive cruise control system works. They are both extremely smooth systems.

I have learned many things about controlling a complete system in IMDL. In return there are a couple aspects that I would like to change about HIWAA. Ideally I would like to use I2C sensors or multiple microprocessors in order to run at a faster frequency. By running at a faster frequency HIWAA would be able to have faster sample rates with his sensors and faster reaction times with his servo and motors. This would improve he reliability and smoothness.

The other big improvement that I would make on HIWAA is to have my sonar attached to a servo. With the sonar attached to a servo, I could do better adaptive cruise control around tight turns. I would tie the servos movement in with the line tracking so that the sonar would be looking into the turn before the front of HIWAA is pointing into the turn. This improvement would make HIWAA adaptive to any environment you put in him.

I would like to thank William Dubel and his website for help and the design for my motor controller, Groung Ghondi for the design of my photo reflector board, Drs. A. Antonio Arroyo, E. M. Schwartz, and Dr. Crane for there help and support throughout HIWAA's creation.

Documentation

Atmel ATMega128 Documentation

http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf

AVRFreaks

http://www.avrfreaks.net/

Devantech SRF04 Ultrasonic Range Finder Reference Sheet

http://www.robot-electronics.co.uk/htm/srf04tech.htm

BDMICR

http://www.bdmicro.com/

Appendices

Sonar sample code:

```
uint16_t SRF04()
{
        PORTG = 0x00; // set port low
        DDRG = 0x02; // set output pin
        us_sleep(50000); // wait 4.8msec between pulses for echo to settle
        PORTG = 0x02; // trigger pin high
        us_sleep(10); // wait a few usec
        PORTG = 0x00; // end trigger
        us_count=0; //set counter to zero

        while(!(PING & 0X01));
            {
                //DO NOTHING AS LONG AS PING0 IS LOW
            }
                while(PING & 0X01);
                    {
                        //DO NOTHING AS LONG AS PING0 IS HIGH
```

```
                    }



        return_value2=(us_count);



        sonar_measure=return_value2;

        //velocity_count2=velocity_count;

        //velocity_count=0;



float return_value;

return_value=(((sonar_measure)*.7829)-11.146); // return average (calibrated for 1/2 inches)

sonar_measure=return_value;



        final=sonar_measure;

        velocity=((final-initial));///(velocity_count2); //half inch per 48 us

        initial=final;



return (sonar_measure);





//return (sonar_measure);



}
```

```
void sonar_motor(void)

{



        if (velocity < 2 & velocity > -2)

        {

                //do nothing

        }

        //==========================start
up======================================================


        else if (velocity == 2 & OCR1A <= 275 & OCR1C <= 275)

        {

                speed_up1=0;

                speed_up2=0;


                speed_up1=OCR1A;

                speed_up2=OCR1C;


                speed_up1 +=75;

                speed_up2 +=75;


                OCR1A=speed_up1;

                OCR1C=speed_up2;
```

```
}


else if (velocity == 3 & OCR1A <= 275 & OCR1C <= 275)
{
        speed_up1=0;
        speed_up2=0;


        speed_up1=OCR1A;
        speed_up2=OCR1C;


        speed_up1 +=100;
        speed_up2 +=100;


        OCR1A=speed_up1;
        OCR1C=speed_up2;
}


else if (velocity == 4 & OCR1A <= 275 & OCR1C <= 275)
{
        speed_up1=0;
        speed_up2=0;


        speed_up1=OCR1A;
```

```
        speed_up2=OCR1C;


        speed_up1 +=125;

        speed_up2 +=125;


        OCR1A=speed_up1;

        OCR1C=speed_up2;

}


else if (velocity == 5 & OCR1A <= 275 & OCR1C <= 275)

{

        speed_up1=0;

        speed_up2=0;


        speed_up1=OCR1A;

        speed_up2=OCR1C;


        speed_up1 +=150;

        speed_up2 +=150;


        OCR1A=speed_up1;

        OCR1C=speed_up2;

}
```

```
	else if (velocity >= 6 & OCR1A <= 275 & OCR1C <= 275)

	{

		speed_up1=0;

		speed_up2=0;



		speed_up1=OCR1A;

		speed_up2=OCR1C;



		speed_up1 +=200;

		speed_up2 +=200;



		OCR1A=speed_up1;

		OCR1C=speed_up2;

	}



	//===============================already moving=================================

	else if (velocity == 2 & OCR1A <= 950 & OCR1C <= 950 & OCR1A > 275 & OCR1C > 275)

	{

		speed_up1=0;

		speed_up2=0;
```

```
        speed_up1=OCR1A;

        speed_up2=OCR1C;


        speed_up1 +=25;

        speed_up2 +=25;


        OCR1A=speed_up1;

        OCR1C=speed_up2;

    }


    else if (velocity == 3 & OCR1A <= 950 & OCR1C <= 950 & OCR1A > 275 &
OCR1C > 275)

    {

        speed_up1=0;

        speed_up2=0;


        speed_up1=OCR1A;

        speed_up2=OCR1C;


        speed_up1 +=45;

        speed_up2 +=45;


        OCR1A=speed_up1;

        OCR1C=speed_up2;
```

```
	}

	else if (velocity == 4 & OCR1A <= 950 & OCR1C <= 950 & OCR1A > 275 & OCR1C > 275)
	{
		speed_up1=0;
		speed_up2=0;

		speed_up1=OCR1A;
		speed_up2=OCR1C;

		speed_up1 +=60;
		speed_up2 +=60;

		OCR1A=speed_up1;
		OCR1C=speed_up2;
	}

	else if (velocity == 5 & OCR1A <= 950 & OCR1C <= 950 & OCR1A > 275 & OCR1C > 275)
	{
		speed_up1=0;
		speed_up2=0;
```

```
                speed_up1=OCR1A;
                speed_up2=OCR1C;

                speed_up1 +=100;
                speed_up2 +=100;

                OCR1A=speed_up1;
                OCR1C=speed_up2;
        }

        else if (velocity >= 6 & OCR1A <= 950 & OCR1C <= 950 & OCR1A > 275 &
OCR1C > 275)
        {
                speed_up1=0;
                speed_up2=0;

                speed_up1=OCR1A;
                speed_up2=OCR1C;

                speed_up1 +=125;
                speed_up2 +=125;

                OCR1A=speed_up1;
                OCR1C=speed_up2;
```

```
}


else if (velocity == -1 & OCR1A <= 0 & OCR1C <= 0)
{
        speed_up1=0;
        speed_up2=0;


        speed_up1=OCR1A;
        speed_up2=OCR1C;


        speed_down1 -=25;
        speed_down2 -=25;


        OCR1A=speed_up1;
        OCR1C=speed_up2;
}


else if (velocity == -2 & OCR1A >= 0 & OCR1C >= 0)
        {
                speed_down1=0;
                speed_down2=0;


                speed_down1=OCR1A;
```

```
                    speed_down2=OCR1C;


                    speed_down1 -=45;

                    speed_down2 -=45;


                    OCR1A=speed_down1;

                    OCR1C=speed_down2;

          }


else if (velocity == -3 & OCR1A >= 0 & OCR1C >= 0)

          {

                    speed_down1=0;

                    speed_down2=0;


                    speed_down1=OCR1A;

                    speed_down2=OCR1C;


                    speed_down1 -=60;

                    speed_down2 -=60;


                    OCR1A=speed_down1;

                    OCR1C=speed_down2;

          }
```

```
else if (velocity == -4 & OCR1A >= 0 & OCR1C >= 0)

{

        speed_down1=0;

        speed_down2=0;


        speed_down1=OCR1A;

        speed_down2=OCR1C;


        speed_down1 -=100;

        speed_down2 -=100;


        OCR1A=speed_down1;

        OCR1C=speed_down2;

}


else if (velocity <= -5 & OCR1A >= 0 & OCR1C >= 0)

{

        speed_down1=0;

        speed_down2=0;


        speed_down1=OCR1A;

        speed_down2=OCR1C;
```

```
				speed_down1 -=150;

				speed_down2 -=150;


				OCR1A=speed_down1;

				OCR1C=speed_down2;

			}


}


void sonar_safty(void)

{

	while (return_value2 <= 31)

	{

		SRF04();


		photo();


		OCR1A = 0;

		OCR1C = 0;

	}

}
```