Project SafeSweep

Final Written Report

Joseph Magaña

EEL5666 IMDL

Instructors:

A. Antonio Arroyo, PhD

Eric M. Schwartz, PhD

TAs:

Mike Pridgen

Tim Martin

Ryan Stevens

Devin Hughes

Thomas Vermeer

Table of Contents

Abstract

The SafeSweep system is designed to move through an arena, detecting and collecting small metallic objects that it encounters. The purpose of the system is to evaluate the practicality of designing and implementing an automated robotic system which collects Foreign Object Debris (FOD) from runways and road surfaces. The SafeSweep system is based around a circular wooden platform, with two drive wheels. The system uses bump sensors and sonar to detect obstacles, and a metal detector which is used to detect FOD. Once FOD is detected, the system collects it into a storage hopper using a brush mechanism and continues to search for more FOD.

Executive Summary

The SafeSweep system is designed to locate and collect small metal debris. The system is based around a wooden double platform with the lower platform carrying the major components, and the upper platform carrying the sonar and power switch, as well as providing protection to the components below. Below the lower platform, the drive system as well as the debris collection system is attached.

SafeSweep is equipped with 4 bump sensors and a Maxbotix LV-MaxSonar-EZ3 for obstacle detection. The system is also equipped with a Future Kit metal detecting circuit for detecting metal debris. The systems behavior is controlled by the Atmel xmega128 micro-controller, and consists of obstacle avoidance and debris collection behaviors.

Debris collection is accomplished through a combined brush and hopper assembly. Once debris is detected by the metal detector, the brush is engaged and debris is swept up into the hopper's magnetic collection assembly. Actuation of the brush, as well as the drive system, is accomplished by using Hitech HS-485HB servos which are hacked for continuous rotation.

Introduction

Foreign Object Damage is the damage to vehicles which encounter Foreign Object Debris (FOD).† This damage can result in a degradation of a vehicle's safety and performance, and therefore increased maintenance costs and downtime. FOD is often small and difficult to detect from casual observation; for example, a small nail or other sharp object on a road surface ahead of a traveling car will most likely not be detected before it is passed over and causes damage.

One of the most common locations for FOD to cause damage are active runways, where aircraft are operating. Runways can harbor a variety of FOD, some common examples being: garbage, broken pavement pieces, parts or tools accidentally left by maintenance personal, debris from damaged aircraft or ground vehicles.[1] During all runway operations, but especially during takeoff and landing, aircraft may suffer damage when FOD strikes the landing gear or is sucked into a turbofan engine. While catastrophic failures are rare, FOD causes enough damage to cost the aerospace industry US$1.1-2 billion a year in direct costs alone, with a total cost (including delays, inefficiencies, etc.) of US$12 billion a year.[2]

† In literature, FOD commonly refers to both Foreign Object Damage and Foreign Object Debris. However, for clarity, In this report FOD refers only to Foreign Object Debris.

Because of the risks and costs involved, clearing FOD from runways is a necessary task for aircraft operators, especially the military which may need to operate high performance aircraft on a runway with minimal preparation and in less than ideal conditions. Unfortunately, many of the current techniques for clearing FOD are labor intensive, such as performing “walkdowns” of the runway or requiring personnel to meet a daily quota of FOD collection. If an automated robotic system could be developed that implemented FOD detection with collection and disposal, this labor could be redirected towards tasks which require it. This may mean cheaper costs for a commercial airport, or less non-essential base personnel for a remote military instillation.

The SafeSweep system is intended to be an experimental attempt at meeting some of the objectives that a proper automated FOD clearing system would require. These objectives are divided into primary and secondary, with the primary objectives being those that will be addressed first, and the secondary being those that will be addressed subject to the difficulty in implementation. The primary objective of the SafeSweep system is to be able to move within an area, avoiding obstacles, while detecting and collecting any small metallic FOD it comes across. The secondary objectives of the system include: being able to sweep the entirety of a closed arena, being able to identify larger FOD and signaling its location upon encountering it, being able to dispose of collected FOD in a special location once sweep is complete or FOD container is full.

This report consists of a full description of Project SafeSweep, including the design and implementation of the platform, power, sensors, actuation, and behavior. The descriptions include discussions on the specific details of each system as well as their overall integration. In addition, the report includes the results of the finished system and an evaluation of how the system performed in relation to objectives.

Integrated System

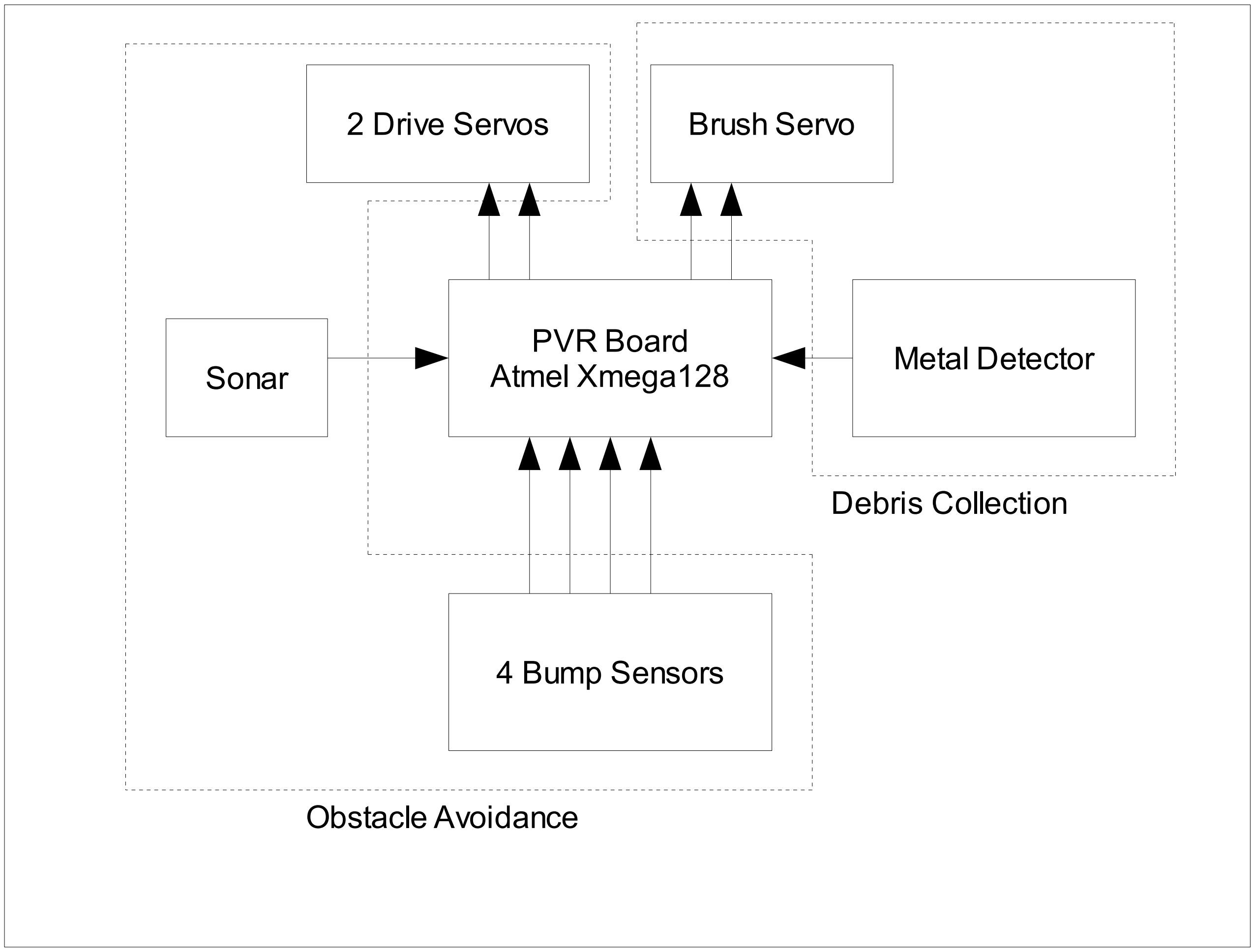

Figure 1: Block diagram illustrating the integrated system of SafeSweep. Devices are divided into two main categories, those that are used to avoid obstacles, and those that are used to collect debris. The xmega128 micro-controller controls the system's behavior by receiving sensor data, calculating behavior based on programming, then controlling for the necessary actuation.

Mobile Platform

The SafeSweep's mobile platform consists of two layers of wood, which are both circular in shape. The bottom layer is 9” diameter and carries the micro-controller, metal detector circuit, and power pack on top, as well as 4 plastic stands which hold the top layer above it. The bottom layer also carries the drive servos, brush assembly, metal detector coil, and hopper assembly below it. In addition, the bottom layer also has 4 bump sensors glued around its perimeter to detect collision with obstacles.

The top layer is 6” in diameter and carries the sonar sensor and the power switch. It also provides protection from the elements to the board and metal detector circuit below. The layer also has holes provided for the mounting of a diagnostic LCD screen.

The drive wheels are two 2” diameter plastic wheels, which are positioned at the sides of the bottom layer. A 1” diameter caster ball at the rear of the platform provides a third point of contact between the platform and the ground, and allows the system to rotate in place. This allows the system to navigate obstacles more easily by being able to maneuver in a tighter space. It also simplifies the code used to control maneuvering.

The brush assembly consists of a single 6” long Hoover Twist-n-Vac brush roll. The brush is fastened to the actuating servo on one side, and an L bracket and rotational bearing on the other. This allows the brush to rotate in place while remaining firmly fastened to the platform. The hopper assembly is a 6” wide by 3” long by 1.25” high cardboard box, with a thin masking tape lip at the front and 4 magnets attached inside. The hopper is fastened to the platform with 2 metal L-brackets. During debris collection, the brush rolls debris up the thin lip into the magnetic collection area, where the debris is trapped by magnetic attraction. Removal of debris after operation can be done either from the front opening of the hopper, or by removing 2 easily accessed screws fastening it to the platform.

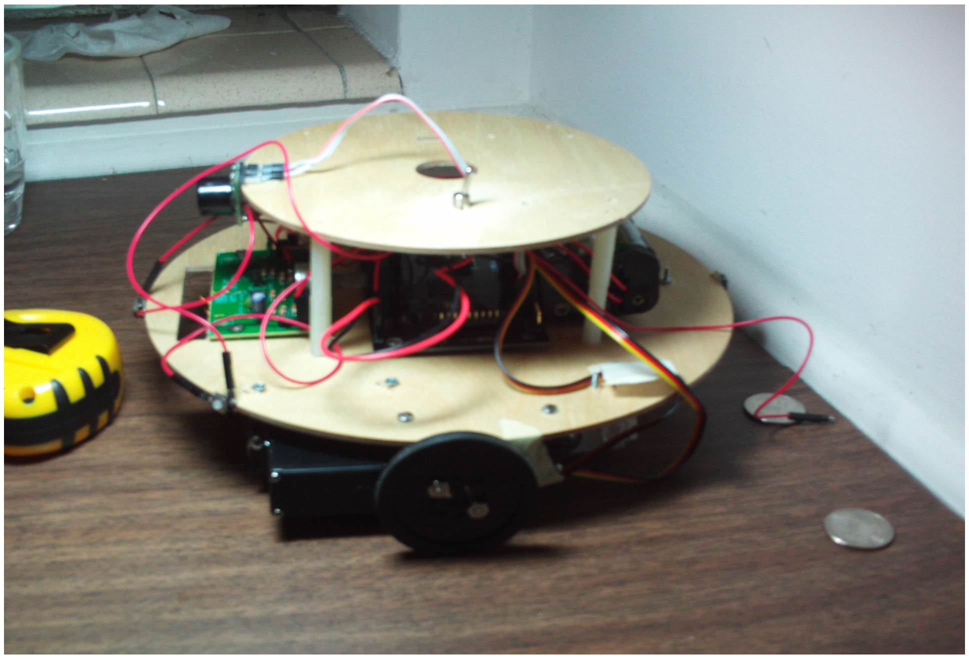
Figure 2: Side view picture of SafeSweep platform. The top layer carries the sonar and power switch. Other components are fastened to the bottom layer.

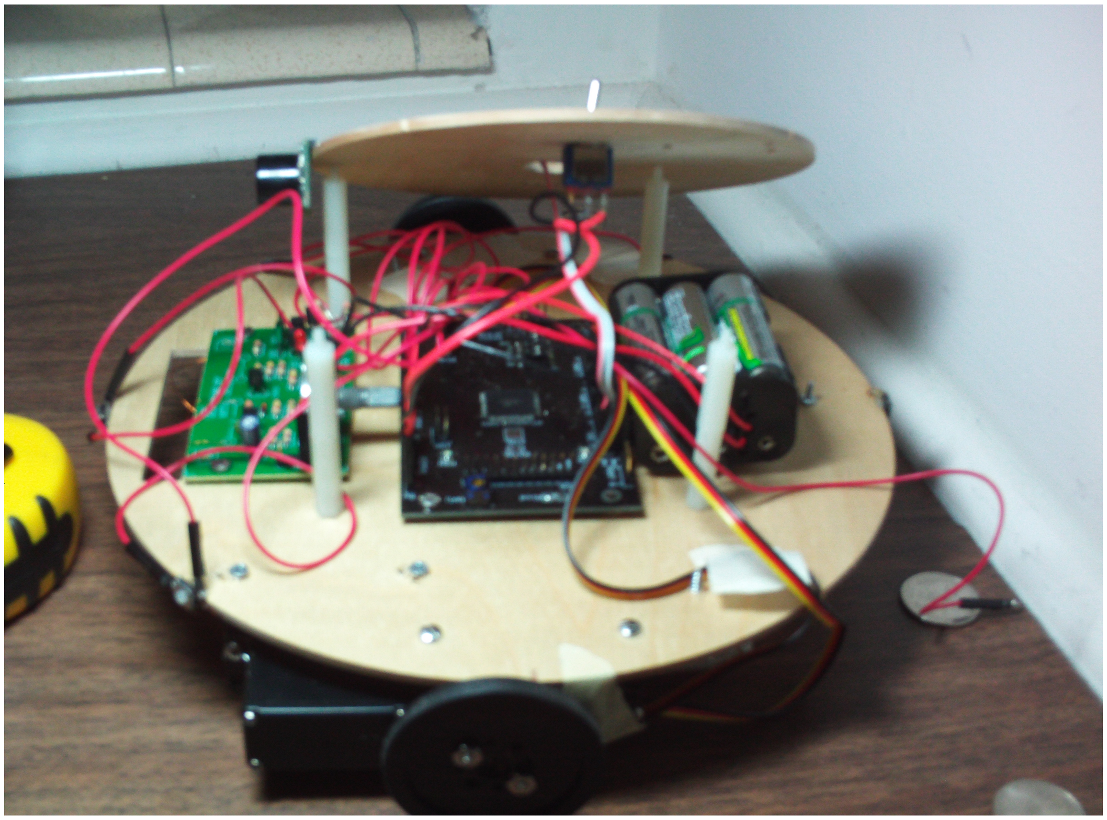
Figure 3: Picture of platform with top layer removed. The PVR board, power pack, and metal detector circuit are all now clearly visible on the bottom platform.

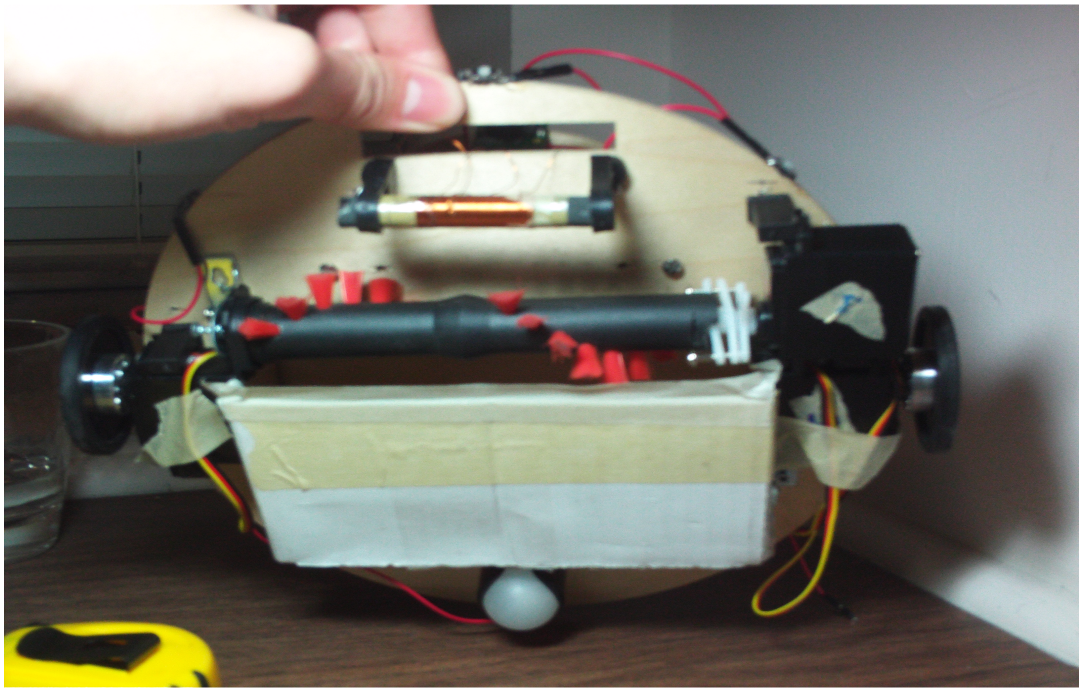
Figure 4: Picture of bottom of lower platform. The detector coils, brush assembly, and hopper, as well as drive wheels and caster ball, are all below the lower platform. Near my fingers is the forward bump sensor on the lower platform.

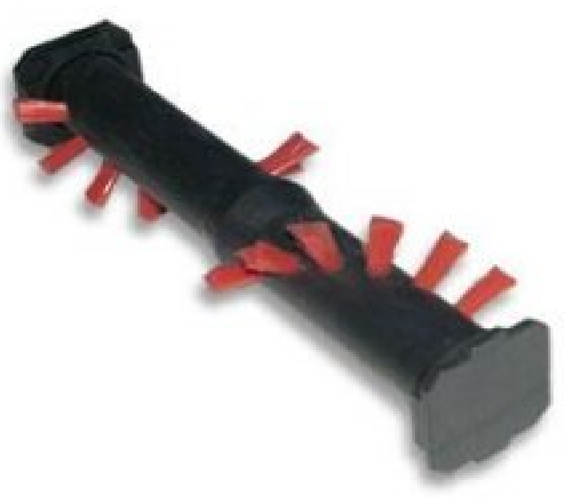

Figure 5: The 6" Hoover Twist-n-Vac brush roll used for the brush assembly. (www.hoover.com)

Actuation

The SafeSweep system's locomotion is provided by two Hitech HS-485 HB servos, which are hacked to allow for continuous rotation. Each locomotion servo drives a single drive wheel. Forward or reverse motion is accomplished by having the servos drive in the same direction. In order to turn in place, the servos are driven in opposite directions.

The brush assembly at the bottom of the system is also driven by a hacked HS-485 HB servo. This assembly rotates at a higher speed than the drive system in order to collect debris into the hopper.

The servos are controlled by code created by Thomas Vermeer and Mike Pridgen, which provides a pulse width modulated signal to the servos which is proportional to an integer input from -100 to 100. Servos were selected rather than DC motors in order to allow for power directly from the micro-controller without additional controller circuitry required. In addition, servos provided enough torque to actuate the system's components at an adequate speed.

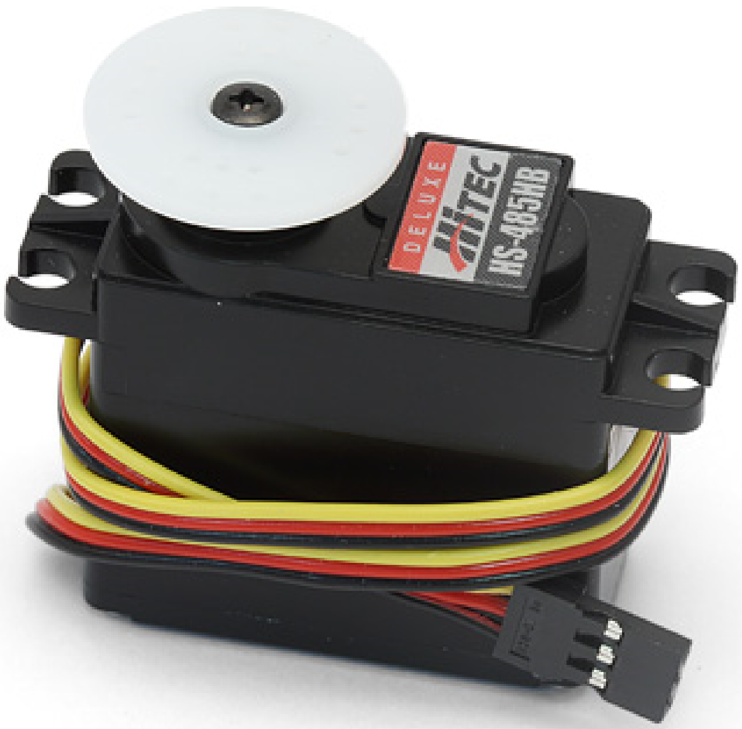

Figure 3: A Hitech HS-485 HB servo. The white plastic mounting hub was replaced with aluminum hubs specifically designed for the drive wheels used. (www.servocity.com)

Sensors

The SafeSweep system uses an array of bump sensors and a Maxbotix sonar to detect obstacles. The bump sensors are configured so that each connected an input pin on the micro-controller to ground when closed. These pins are configured to use the xmega128's built in pull-up resistors so that they are pulled high when the switch is open. When the bump sensor is depressed by obstacle collision, the switch is closed and the pin pulled low.

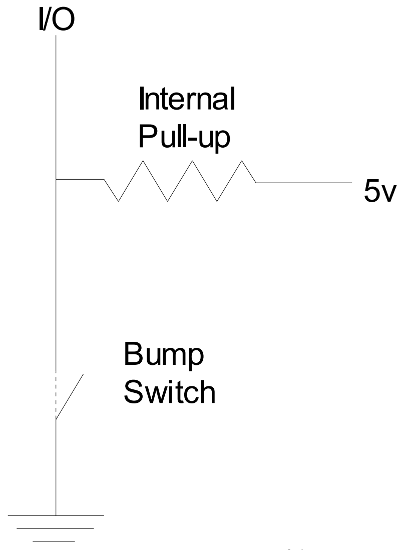

Figure 4: Diagram of bump sensor circuit. The depression of the bump switch closes the switch and pulls the I/O pin low.

The Maxbotix LV-MaxSonar-EZ3 is the primary obstacle detection sensor. It is positioned facing the forward direction of the platform in order to detect obstacles that the platform is driving towards. While powered, the sonar transmits sound pulses, then counts the time before receiving the echo from a detected object. The sensor's internal system then calculates the distance traveled by the pulse during that time, and outputs the result as an analog voltage proportional to the distance from the obstacle.

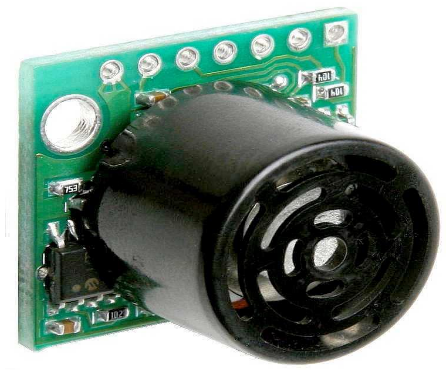

Figure 5: A Maxbotix LV-MaxSonar-EZ3. (www.maxbotix.com)

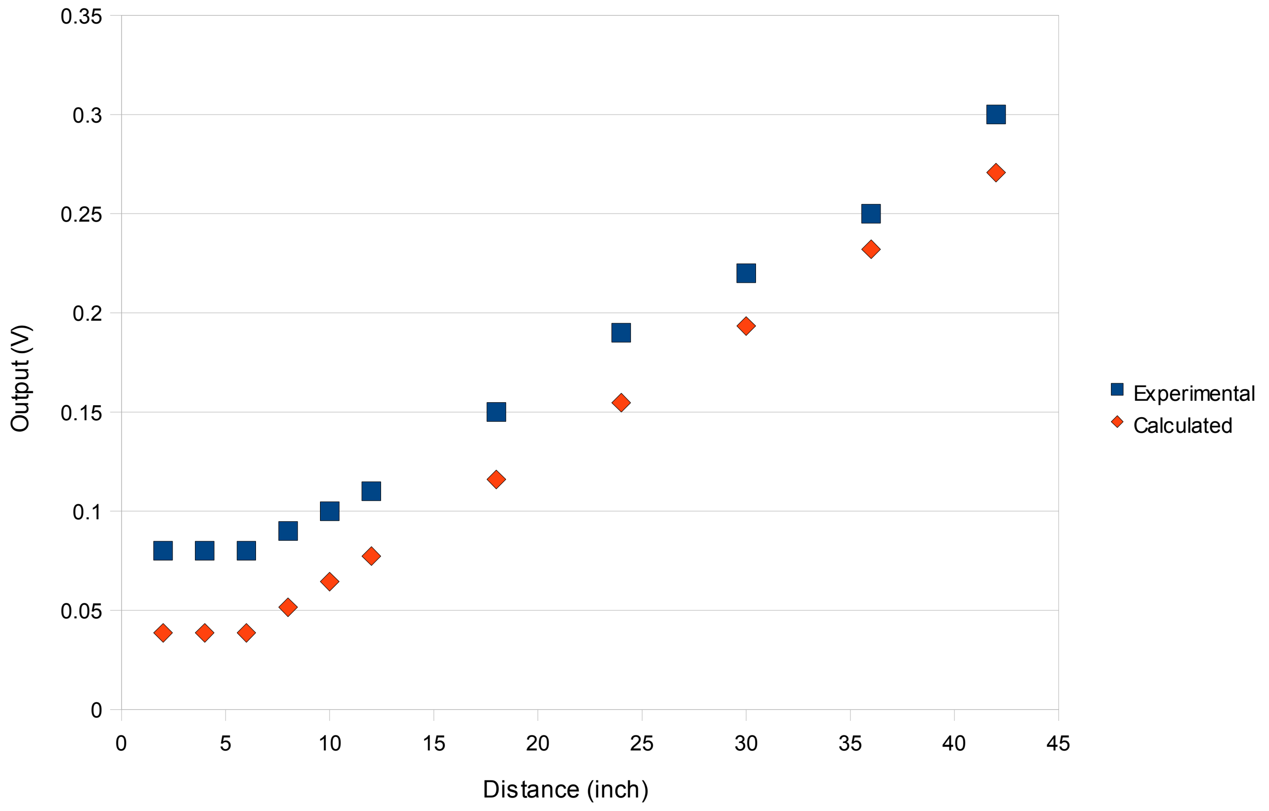

Figure 6: An output vs. distance characterization of the Maxbotix sonar. The experimental values were obtained by using a large rectangular target positioned perpendicularly to the sonar, in order to present a large, flat, surface, like a wall. The calculated values were developed from the relation provided by the sensor's data sheet.

SafeSweep uses the Future Kit FK919-1 metal detecting circuit to detect debris. The circuit uses two induction coils to detect the presence of metal, and has a range of ~50 mm from the coils. The circuit is also powered directly from the 9V battery pack. The detection coils are mounted to the bottom of the lower platform layer, and are near the ground. They are positioned forward of the brush and hopper assembly.

When the FK919-1 detects metal, a normally inactive portion of the circuit is powered. The micro-controller detects this change by measuring the analog voltage level at a component in the activated part of the circuit. In addition, the circuit includes a red LED which provides a visual indication of detection.

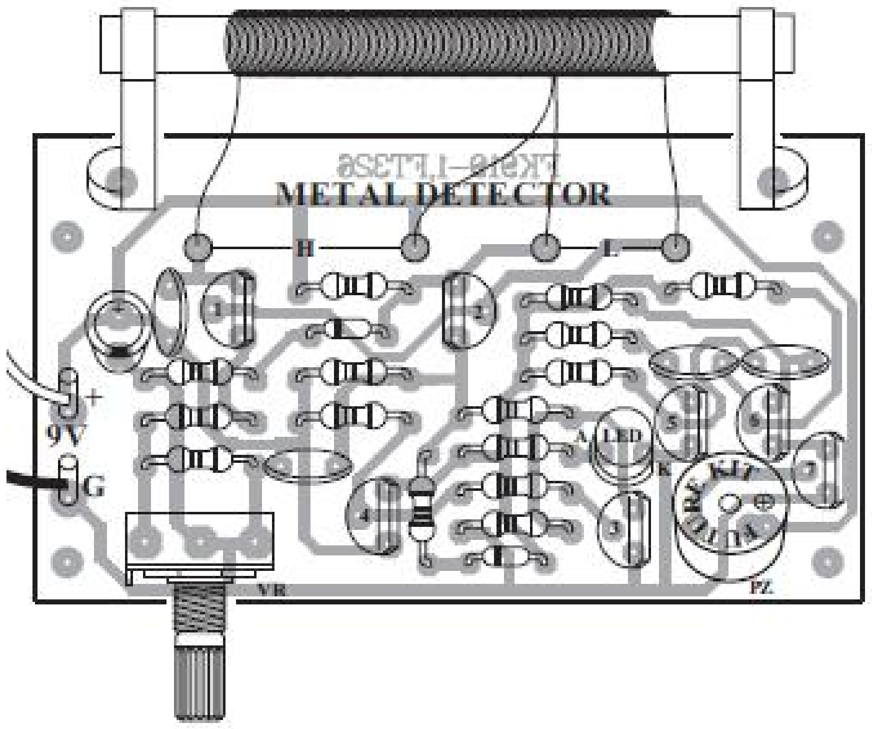

Figure 6: The Future Kit FK919-1 metal detection circuit. The coil assembly at the top of the picture is mounted at the bottom of the platform. The potentiometer at the lower right controls the sensitivity of the sensor. (From Data sheet)

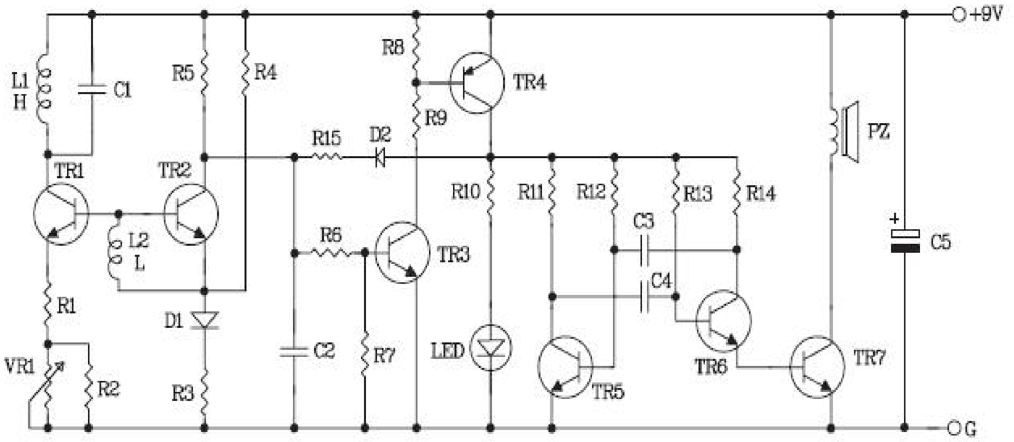

Figure 7: Circuit diagram of the FK919-1 metal detector. L1 and L2 are the detection coils. The node connecting R10 and TR4 is where the micro-controller measures voltage in order to determine if detection has occurred. The speaker, indicated by PZ, is removed in order to provide for quiet operation. (From Data sheet)

Behaviors

The SafeSweep system carries out what is essentially a random search of an enclosed area. When activated, the system drives forward in a main behavior loop until either debris or an obstacle is detected. If debris is detected, the system engages in debris collecting behavior. If an obstacle is detected, the system instead carries out obstacle avoidance.

The system avoids obstacles by immediately reversing direction and driving backwards for a few seconds, or until the rear bump sensor detects collision with a obstacle behind the platform. Then the system executes a right turn for .5 to 2 seconds, depending on the value of a counter which continuously increments during the main behavior loop. Once this turn is completed, the system resumes forward movement and returns to the main loop.

When the system detects debris, it dramatically slows forward movement while activating the brush assembly rotation. The system slowly drives over the debris to collect it, while continuing to check for obstacles. If an obstacle is detected, the system immediately ceases debris collection and engages obstacle avoidance, otherwise it continues collection for a few seconds before reengaging forward motion and returning to the main loop.

Experimental Layouts and Results

During a short demonstration on Dec. 6, 2010, SafeSweep was able to successfully collect a fair portion of small metal debris scattered in a small arena, consisting of a collection of #4 machine screws and wood nails. It managed to successfully detect debris that was driven over, and also avoided collision with obstacles.

Conclusion

SafeSweep was a successful first step is designing a debris collection system. While working on this project, I learning numerous things about robot design specifically, as well as the design process in general. I hope to use the knowledge and experience obtained working on this system on future tasks, both in school and with any future employer.

Some of the issues with the current system include very fine sensitivity of the metal detector, as well as difficulties in reliably collecting debris. The platform could also use additional mechanical components to increase the rigidity of the mechanical configuration, and make it better able to survive transport and operation without damage.

I hope to continue to work on SafeSweep, by building on its existing platform with new hardware and behaviors. I am interested in adding some of the more advanced features of my proposal that were not implemented in the final design, such as more advanced search algorithms. I may even carry over the SafeSweep concept to a new platform with more capabilities, such as better operation outside and on rough surfaces. This has been a genuinely fun and informative project for me and I would like to continue exploring applications of robotics.

Documentation

References

1. http://www.xsightsys.com/whatisfod.htm
2. http://insightsri.com/system/files/The+Ecomonic+Cost+of+FOD+-+Jul08.pdf

Appendix

Primary Code:

```
//Code for project SafeSweep
//By Joseph Magaña
//IMDL Fall 2010
//Requires functions found in PVR.c by
//Thomas Vermeer and Mike Pridgen
#include <avr/io.h>
#include "PVR.h"
//Globals//
int sonar_level;              //Tracks sonar output
int metal = 0;                //Tracks metal detection
int mode = 0;                 //Tracks mode for LCD display
int n = 0;                    //Obstacle avoidance turn randomizer
int z = 0;                    //Tracks obstacle detection during debris collection
const int left_go = -35;      //Forward motion constants
const int right_go = 40;      //some correction for servo drift
const int detect = 400;               //Sonar detection threshold
//Initialize functions//
void avoid(void);             //Obstacle avoidance
void lcdUpdate(void);         //LCD display update
int readSonar(void);          //Reads sonar output
int readMetal(void);          //Reads metal detector output
void sweep(void);             //Debris collection
//Main function//
void main(void)
{
        xmegaInit();          //Device/Peripheral initialization
        lcdInit();
        delayInit();
        ADCAInit();
        ServoCInit();
        PORTB_PIN0CTRL |= 0b00111000;//Activates pull-up resistor for I/O ports
        PORTB_PIN1CTRL |= 0b00111000;//used for bump sensors
        PORTB_PIN2CTRL |= 0b00111000;
        PORTB_PIN3CTRL |= 0b00111000;
        delay_ms(2000);                       //Initial delay before behavior starts
        ServoC2(right_go);            //Forward motion
        ServoC0(left_go);
        while(1)                      //Main loop
        {
                delay_ms(50);         //delay
                n++;                  //Turn randomizer counter
                if (n == 11){         //varies from 0-10
                        n = 0;
                }
                sonar_level = readSonar();     //Updates sonar level
                metal = readMetal();           //Updates metal detection status
                lcdUpdate();                   //Updates LCD display
                if (sonar_level < detect){     //Engages obstacle avoidance if sonar detects obstacle or
                        avoid();               //forward bump sensors detect collision
                }
                if ((PORTB_IN & 0b00000100) == 0b00000000){
                        avoid();
                }
                if ((PORTB_IN & 0b00000010) == 0b00000000){
                        avoid();
                }
                if ((PORTB_IN & 0b00001000) == 0b00000000){
                        avoid();
                }
                if (metal){            //Engages debris collection if metal detected
                        sweep();
                }
        }
}

void avoid(void)                       //Obstacle avoidance
{
int turn = 500 + 150*n;                        //Calculates random turn length (.5 to 2 seconds)
mode = 1;                              //Sets mode for LCD
lcdUpdate();                           //Updates LCD display
ServoC2(left_go);                      //Backwards motion
ServoC0(right_go);
if ((PORTB_IN & 0b00000001) == 0b00000000){  //Checks for rear bump switch collision
ServoC2(0);                            //stops motion if detected, otherwise
ServoC0(0);                            //backwards for 2 seconds
}
delay_ms(2000);
ServoC2(right_go);                     //Right turn for randomized time
ServoC0(right_go);                     //interval
delay_ms(turn);
ServoC2(right_go);                     //Forward motion
ServoC0(left_go);
mode = 0;                              //mode reset for LCD
}

void lcdUpdate(void)
{
//lcdGoto(0,0);                                \\This code displays the current
//lcdString("Sonar Level ");           \\sonar level as well as behavior
//lcdInt(sonar_level);                 \\mode to the LCD. It was commented
//lcdGoto(1,0);                                \\out in current configuration as the
//lcdString("Mode ");                  \\LCD display is not currently attached.
//if (mode == 0){
```

```
            //lcdString("Normal ");
            //}
//if (mode == 1){
            //lcdString("Avoid  ");
            //}
//if (mode == 2){
            //lcdString("Metal  ");
            //}
}

int readSonar(void)             //Reads sonar input. This function
{                               //reads 3 analog values from the
int i;                          //sonar and then averages them
int k;                          //to determine the sonar level
int signal[3];
for (i = 0; i < 3; i++){
signal[i] = ADCA0();
}
int sum = 0;
for (k = 0; k < 3; k++){
sum = sum + signal[k];
}
return sum/3;
}

int readMetal(void)
{                               //This function reads the analog
int signal;                     //voltage from a component of the
signal = ADCA7();               //metal detector and returns detected
if (signal > 2000){             //if the voltage is higher than
            return 1;           //a certain threshold.
}
else{
            return 0;
}
}

void sweep(void)                //Debris collection
{
mode = 2;                       //sets LCD mode
z = 0;                          //Obstacle flag
ServoC2(0);                     //Stops motion
ServoC0(0);
lcdUpdate();                    //Updates LCD
delay_ms(250);                              //Initial delay
ServoC5(95);                    //Engages brush rotation
ServoC2(right_go/3);            //Slow forward motion for 2 seconds
ServoC0(left_go/3);
delay_ms(2000);
```

```
if ((PORTB_IN & 0b00000100) == 0b00000000){  //At 2 seconds, function checks for
                    ServoC5(0);         //collision with bump sensors. If
                    avoid();            //collision is detected, shuts down
                    z=1;                //brush and goes into obstacle avoid,
}                                       //otherwise continues for 2 additional
if ((PORTB_IN & 0b00000010) == 0b00000000){  //seconds
                    ServoC5(0);
                    avoid();
                    z=1;
}
if ((PORTB_IN & 0b00001000) == 0b00000000){
                    ServoC5(0);
                    avoid();
                    z=1;
}
if (z==0){
delay_ms(2000);
ServoC2(right_go);                  //Resumes forward motion
ServoC0(left_go);
ServoC5(0);                         //Shuts down brush
mode = 0;                           //resets LCD mode
}
}
```