Department of Electrical Engineering

EEL 5666

Intelligent Machines Design Laboratory

Director: Dr. Arroyo

Range Rover

Autonomous Golf Ball Collector

Andrew Janecek
May 1, 2000

Table of Contents

ABSTRACT

The following paper describes the design on an autonomous robot that collects golf balls. The robot was built from an RC platform, more specifically a Losi truck. The robot was designed to avoid obstacles while picking up golf balls. This behavior was achieved by using a bumper. Thus the robot will first run into the object in order to go around the object. The platform used a 68HC11 microprocessor that relies on feedback from sensors that sense the surrounding environment. An electric fence was used to keep the robot in an area that is designated by the fence. The microprocessor was also used to keep track of the number of balls picked up. This number was displayed by using an LED array that showed the binary representation of the number of balls. The motivation of the robot was to build a mobile agent that could replace human intervention at golf ranges. The retrieval of golf balls should be a simple-minded task. The use of robots in swarms should allow for a large area of a driving range to be cleared by these robots.

Executive Summary:

Range Rover is an autonomous robot that is designed to clear a designated area of golf balls. This application is practical and could be used by any driving range in order to eliminate human intervention thus saving money. This idea could also be practical for home use assuming the customer has a large enough yard. The scope of the project is much small than a typical driving range would be. By using a swarm of these robots, the full area of a driving range could be covered.

The scope of the project is smaller due to the limitations of the platform that was used. The platform was a RC car, more specifically a Losi truck. Electric RC cars have limited amount of power due to the battery. Without any additional weight, these RC cars are expected to run for about eight minutes. Due to the extra weight Range Rover would typically last five minutes with each fresh battery. Since Range Rover is much smaller than a typical car used at ranges, the mechanical device to pick up the golf balls was much smaller as well. A normal drum (this term will be described later in the report) used at driving ranges will have over twenty slots for balls. The drum created for Range Rover only had eight slots for balls.

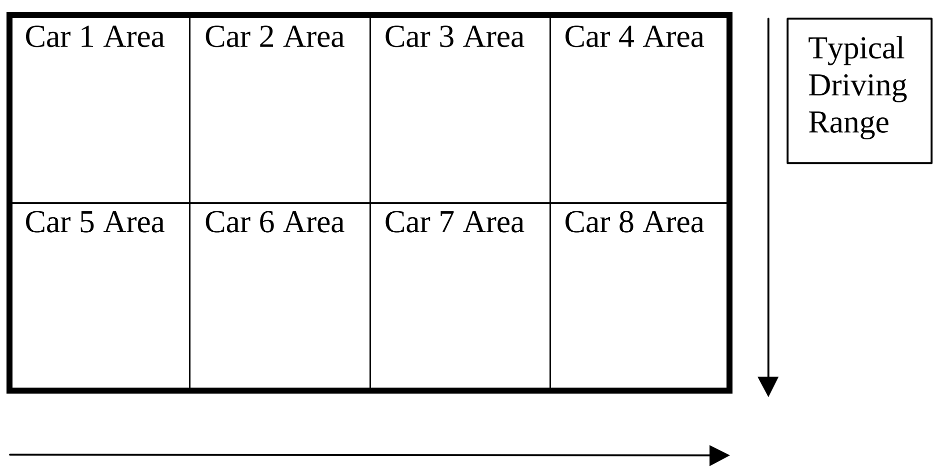
A 68HC11 TJ PRO Board, designed by Mektronix, is used to control the entire robot. This board is specifically designed for robots such as Range Rover. This board provides analog ports for sensors to be used. The readings from these sensors will determine the behavior of the robot. Most of the time spent by Range Rover is driving straight and picking up any balls in its path.

An electric fence is used to contain the robot in the area that is determined by the fence. The number of balls picked up is kept track by the microprocessor as well. This number is displayed on an LED array that can be seen by the user. This array will flash when the maximum numbers of balls have been picked up. The maximum is currently set to ten in the code, which could easily be changed. Finally there is a bumper on Range Rover that is used to avoid obstacles. The robot will run into the object and then back up to go around the object. This bumper will work anytime except for when Range Rover is taking appropriate action due to the electronic fence.

Introduction:

The motivation of Range Rover comes from the simple repetitive task of finding many of the same object, in this case golf balls. As a golfer I have had to shag my own golf balls many times. This task seems that it would be simple enough for an autonomous robot to complete.

The scope of this project is dependent on the customer. If the customer were just an individual, then one Range Rover would meet their needs. Range Rover is built to cover a typical yard. The yard would have to be somewhat clear of objects. Range Rover is built to avoid objects but would not work nearly as well with a cluttered yard. Second the case of a golf range is analyzed. In this case more than one Range Rover would be necessary. This could be done with a swarm of robots, each cleaning its own designated area. A figure is shown below to make this point for clear:



The objective of this project is to build an autonomous robot that will clear a designated area of golf balls. This area will be designated with an electronic fence. The robot will also avoid objects by first running into the object and then going around the object. These objectives are met by a series of behaviors that are controlled by various sensors that control the actuation of the robot.

The following paper will discuss first the integration of the whole system. The specific details of the components will not be discussed; rather how these components will work together. The platform of Range Rover will also be discussed. The design of this robot is clearly more mechanical making the platform important. The actuation of the robot will also be discussed. A DC motor controls the forward and backward motion of Range Rover. The steering is controlled by a servo. The next section will discuss how the sensors will control the actuation of the robot. The combination of the sensors and the actuation determine the behaviors of Range Rover. The behavior section will give more details on these sensor reactions.

Integrated System:

The following block diagram shows the complete system integration of Range Rover.

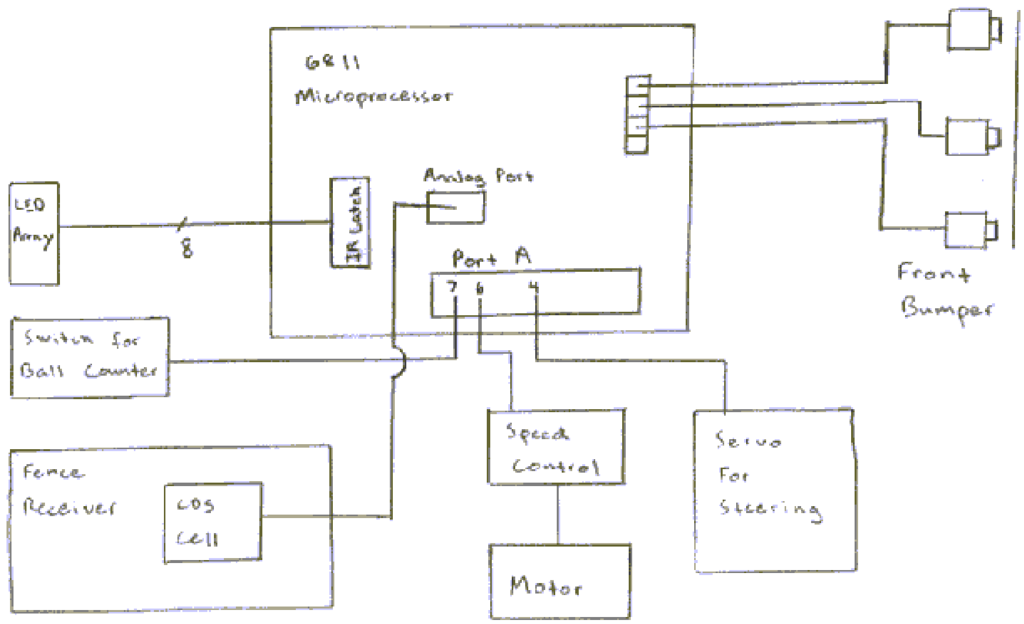


Figure 1: Integrated System for Range Rover

The complete system for Range Rover consists of one continuous front bumper, a servo and speed control (motor driver) for actuation, a switch for ball counting, and an electronic fence detector. The microprocessor board was powered with a 9 volt battery pack that was regulated on the board. The motor was driven with a separate 7.2 volt Ni-Cd battery.

The actuation of Range Rover is controlled by a pulse width modulated signal. The servo responds to the signal by rotating a certain amount depending on the pulse width of the signal. The motor is not directly controlled the same way. The electronic speed control of the RC car (same as the motor driver) is what controls the motor. Due to very high currents that are created by the motor, it would have been very difficult to make a motor driver myself. The electronic speed control makes the motor control exactly the same as the servo. The pulse width modulated signals were created through Port A of the microprocessor. More specifically, the steering used PA4 and the motor used PA6.

The readings from the sensors will determine what actuation will take place. Under normal operations, Range Rover will have the motor turning forward with the servo making the wheels straight. The program written for Range Rover is such that Range Rover sits in a continuous while loop checking the sensors. A function is written for each sensor so that the reading is taken and then appropriate action is taken. These functions are called in order.

For example, the function for the bumper does the following. First the value of the analog port of the bumper is checked. If the bumper has not been struck then Range Rover will not change the pulse width signal for the servo or motor. If the bumper has been struck then the pulse width signal will be changed. The same is true for the fence receiver.

The ball counter does not have any effect on the actuation. The function for the ball counter simply checks the value of the PACNT register. This register is automatically updated each time a rising clock edge has been detected. The value of the register is placed on the LED array for the user to see. Since no IR has been used for Range Rover, the IR latch is used to send the value of PACNT to the LED array and hold the value.

Mobile Platform:

The platform created for Range Rover consisted of two separate parts. The first part is the car itself. This part of the platform carried the bumper, housing for the microprocessor, electric fence receiver, and motor. The picture for the car is seen below.

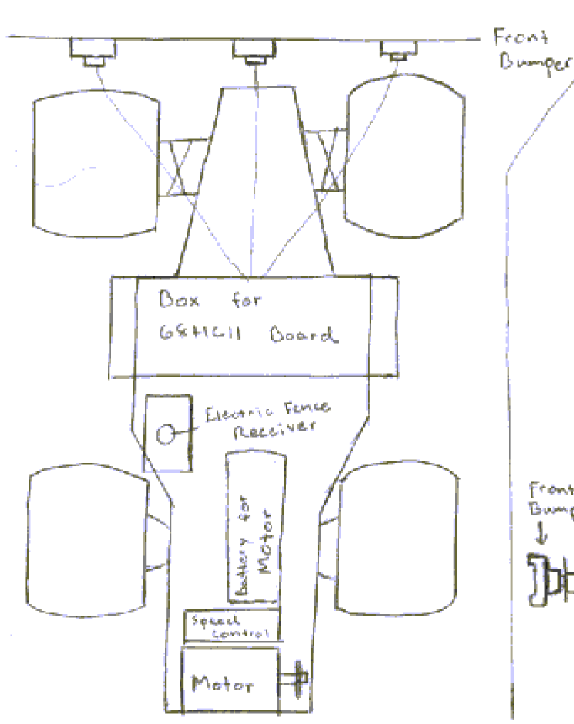


Figure 1: Top View of Platform

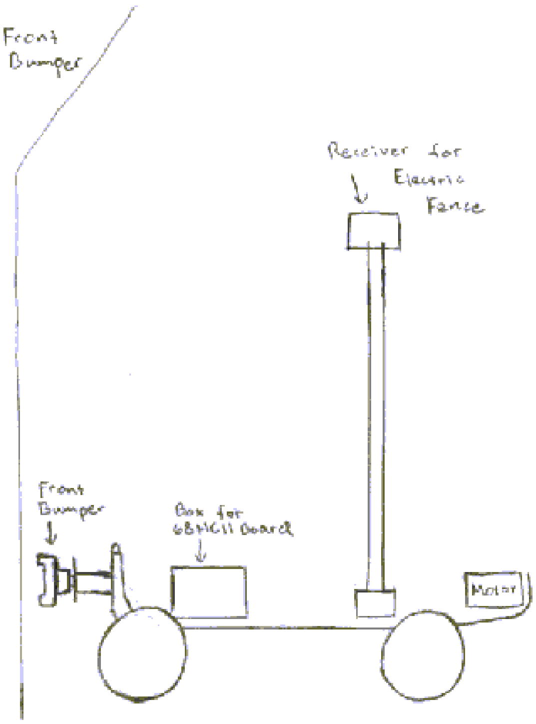


Figure 2: Side View of Platform

The reason for the receiver of the electric fence being so high is described in the sensor section. The housing for the 68 HC11 board is necessary because of the amount of dirt that will be thrown in the air. There was no body placed on the car because the body would not fit with the housing for the board on the car. The design to make the car separate from the trailer that will pick up the golf balls was used to make the design more portable.

The design behind the trailer to pick up the golf balls was based on currently commercially available designs. The design consisted of a drum and a basket in which the golf balls will funnel into. The drum consists of a central rotating axle with circular discs placed on this axle. The idea behind this design is that the discs are spaced at a distance slightly smaller than a golf ball. This will force the golf ball to wedge in between the discs. The ball will then rotate around the axle. The ball will then be funneled into the basket for collection.

This trailer is pulled behind the car. The connection is not shown on the figures but is not necessary. The drawing for the trailer can be seen below.

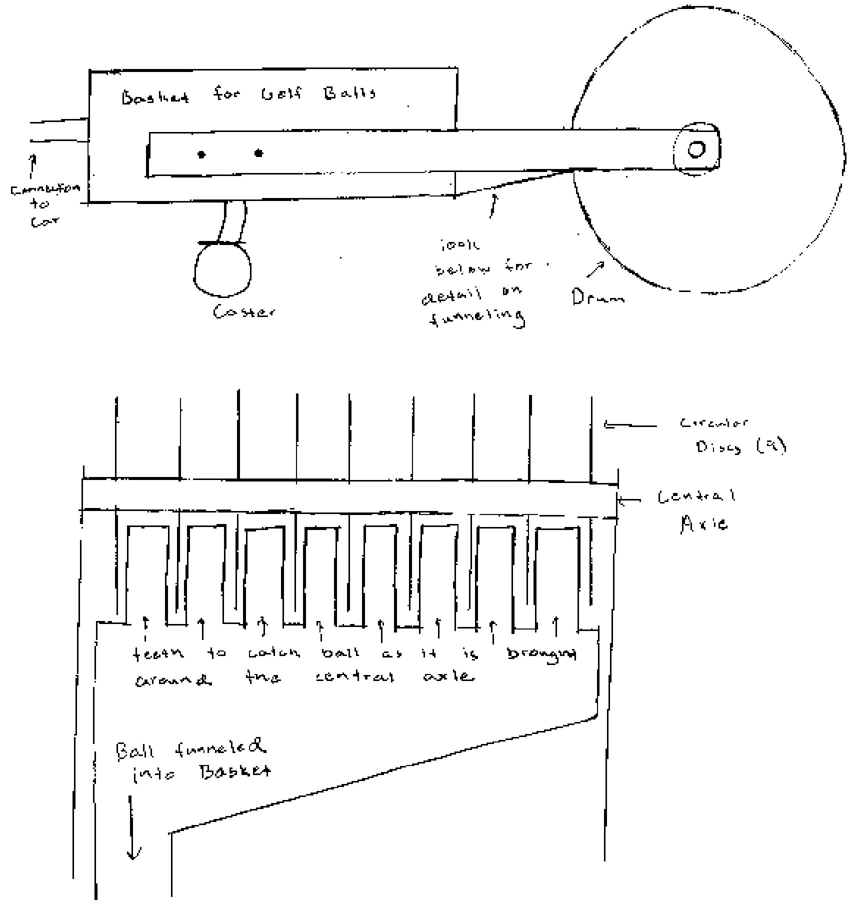


Figure 4: side and top view of trailer

Actuation:

Battery Power

Two separate power supplies power the car. The motor is driven by a 6 cell, 7.2 volt battery pack. The cells of this battery pack are larger than normal AA batteries. This is necessary because of the extremely high current that will be created from the DC motor. During stall the motor can produce over 100 Amps of current. The motor driver for this motor would have been very difficult to make. For this reason an electronic speed control was used to control the motor.

The 68HC11 board was powered with a battery pack consisting of six AA batteries. This battery pack was regulated with a chip on the 68HC11 board. The servo that was used to control the steering of the car was also powered off of the board. The power sources needed to be separate due to the difference in voltage. The board was used to send the signal to the electronic speed control. This signal would control the direction of the motor.

Servo and Motor Control

The control of both the servo and the motor was accomplished by pulse width modulation. The pulse widths were generated from the output compare pins on Port A of the 68HC11 board. The steering was controlled on PA 4 and the motor was controlled on PA 6. By observing on an oscilloscope it could be seen that the pulse width to go in reverse was larger than when it was going forward. The pulse width needed to control could not be seen on the oscilloscope.

Note:

When using separate battery packs they cannot be connected in any way. This will cause one power supply to try and charge the other power supply. A lot of current will be pushed through the traces on the board very quickly damaging the board. Unfortunately this was learned from experience.

Sensors:

Range Rover consists of the following sensors:

Sensor	Quantity	Purpose	Action
Front Bumper	3 bump sensors attached by one continuous bumper	Detect the robot running into an object in the front.	Stop the car. Back the car up with the wheels turned. Then continue forward.
Ball Counter	1	Keep track of the number of balls that have been picked up	Increment PACNT register when the pulse accumulator sees a rising edge.
Electric Fence	1	Keep the car in the pre-determined area.	Control the motor and steering so that the car does not leave the pre-determined area.
CDS cell	1	Used in conjunction with electric fence. Detects when neon bulb is lit by fence.	Determines when appropriate actions needs to be taken due to the fence. Range Rover goes in a circle once the fence is reached.

Table 1: Sensor Array

Front Bumper

The bumper was difficult to build due to the speed and force at which Range Rover would strike objects. The bumper was made to be sturdy so that it would not break upon striking an object. Despite thinking that it was sturdy enough, this was not the case. The switches that were used were from an old keyboard to make the bumper more rugged.

The bumper consisted of three switches that were connected with one continuous bumper. The bumper would cause the left or right switch to be pressed of either side of the bumper were pressed. All three switches would be pressed if the bumper were struck in the middle. The signal is read through PA0 of the microprocessor. The analog value will be between 40 and 140 depending on where the bumper is struck.

Ball Counter

The ball counter was created with the following circuit diagram below:

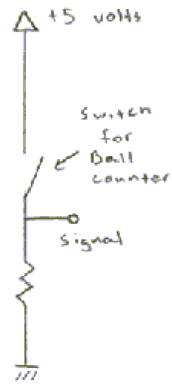


Figure 2: Circuit Schematic for Ball Counter

The circuit was simple to make. The function of this sensor was as follows. When the switch was open, the signal would be going to ground since there will be no current to produce any voltage. Once the switch has been closed, current is allowed to flow through the resistor. The voltage will drop across the resistor and five volts will now be seen on the signal. The rising edge from ground to power is used to count the balls. This is done with the pulse accumulator. The pulse accumulator needs to be initialized to recognize rising edges and needs to be placed in counting mode. This will increment the PACNT register every time a rising edge is detected. The PA7 pin is used for the pulse accumulator.

Electric Fence and Ball Counter

The following circuit diagram is used with the electric fence:

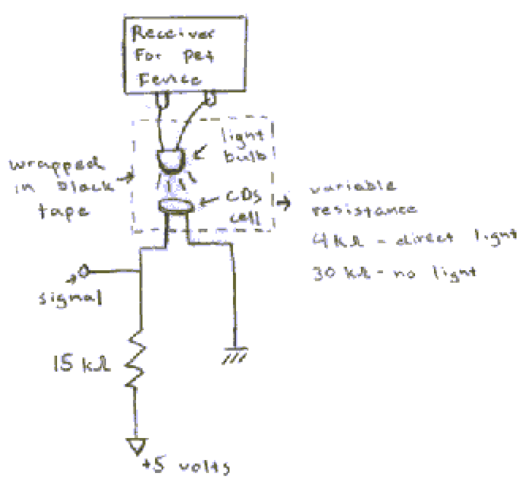


Figure 3: Circuit Schematic for Electric Fence detector

The transmitter for the electric fence is a typical pet fence. This transmitter emits a radio signal out of a wire loop. The radio signal sent seems to be a normal AM signal. The receiver will detect this radio signal and place over 2000 volts across the two contact probes. This high voltage is used to light a small neon light bulb. This method has to be used due to the extreme voltage the receiver uses.

The CDS cell is used as a voltage divider. When the CDS cell is exposed to direct light the resistance across the cell is 4 kohms. This will place almost 1 volt on the signal in theory. When the CDS cell is not exposed to light the resistance of the cell is 30 kohms. This will place 4 volts on the signal in theory. An analog port is used to measure the voltage of the signal. A voltage drop of one volt is plenty for detection. The way the receiver works is such that the bulb does not stay on continuously.

The CDS is placed directly on the light bulb. The light bulb and CDS cell are wrapped in black tape so that no light penetrated the tape. This would give a high analog value when the receiver is not active and a low analog value when the receiver is active. The PE2 pin is used to take measurements on the signal.

Problems with sensors

I had a few problems that set me back with the sensors. The first and most serious problem was the noise produced by the motor. The RF noise would completely eliminate the capability of the receiver to detect the fence. The receiver was originally placed on the front of the car. The distance was to close however.

It would be desirable to place the wire underground. This was not possible due to the noise. The first solution was to turn the motor off long enough to take a reading from the receiver. This solution did not work because the momentum of the car was not enough to keep the car going. The only solution was to place the receiver high. This would allow for readings to be taken while the motor is running.

The other problem was the bumper. The bumper that was built was not sturdy enough to absorb an object strike. The use of springs to absorb the force would be necessary. The function of the bumper was shown, but the car could not be allowed to run into a rigid object.

The final problem encountered was the switch used for the ball counting. A keyboard switch was used, similar to the bumper switches. However, the ball was not heavy enough to push down the switch. The proper operation of the counter was shown manually. With a different switch the counter would have worked.

Behaviors:

The basic function of Range Rover is to pick up golf balls in a designated area. This is done through a series of behaviors that are controlled by the sensors. The behaviors of Range Rover are as follows. In normal operation, Range Rover moves forward with the wheels straight. This will continue until one of the sensors shows that other recourse is needed. The first behavior to be looked at is the avoidance behavior.

Once the bumper detects that an obstacle has been struck the following happens. Range Rover will first back up for two seconds with the wheels straight. Then the wheels will be turned for one second while backing up. Range Rover will then continue normal operation after this behavior has finished. The reason for backing up straight for two seconds is for clearance of the trailer. If the wheels were turned immediately once Range Rover backs up, then the trailer will get up on the back wheels of the car. Any balls that may happen to be picked up during this behavior will not be displayed until after the behavior is done.

The most important behavior is for Range Rover to turn around when it has detected the electric fence. The idea of this behavior is to have Range Rover to leave the fence at the same spot it entered the fence except in the other direction. This behavior needs to be working properly in order for Range Rover to be truly autonomous. The circle that Range Rover does requires a significant area to turn around. One flaw in the design is that the bumper will not work during this behavior. Any object struck during turning around will cause a serious problem.

The last behavior is counting the golf balls. This behavior works all times that the robot runs. The result is only displayed during normal operation. When the other behaviors take place the result of any ball pick up will not be displayed until after the behavior is done.

Problems with behaviors

The biggest problem with the behaviors is the terrain. Because the fence behavior relies on timing, the terrain must be flat. The hilly land would cause Range Rover to not be able to complete a circle properly. This would force the user to place the electric fence on flat land where Range Rover will do the circle.

The ball counting behavior did work. The problem occurred with the mechanical bouncing of the switch. The bouncing would cause more than one rising edge to occur for each time the switch is pressed. This could be fixed by using a denouncing circuit or a switch that is already debounced. The lack of time did not allow for these solutions to be implemented.

Experimental Layout and Results:

It was necessary to test each behavior separately. It would have been impossible to try and realize what was wrong with all the behaviors at once. The first step was to make sure that the bumper was working properly. This behavior consisted of the car backing up once it has struck the object.

The behavior needed to be tweaked a little so that it would back up straight first then turn the wheels for a little while. This was necessary due to the trailer that it was pulling. If it were to back up right away with the wheels turned this would cause a problem. The trailer would get stuck on the back wheels. The right combination was to back up straight for two seconds then turn the wheels and continue backing up for one second. The car would continue to go forward after this behavior was done. Due to the timing no other behaviors worked during this time.

The next behavior checked was the ball counter. A program was first written to use the pulse accumulator without any other behaviors. It was convenient that the IR latch was still available. Since the value had to be displayed at all times a latch was needed anyways. There was trouble having the program working at first because of a jumper. Once the jumper was moved the LED array worked properly. The PA7 had to be used for the pulse accumulator. Once the counter seemed to work it was added to the Range Rover code.

The most important behavior was tested last. The first step was to figure out the right timing for the actuation. The behavior had to make the car do a circle and return to the same spot going the opposite direction. As said before there was difficulty in this behavior because flat land was necessary. Different terrains would also throw off the timing of turn. The assumption that had to be made was for the car to be on flat grass terrain. The actuation consisted of going forward while turning for two seconds, then going straight for one second, and then turning left for five seconds. This behavior would work for flat grass terrain. The turn behavior was then implemented with the electric fence. This behavior worked well once the timing was worked out.

Once all the behaviors proved to work individually they were put together in one file. The difficulties that have been discussed before mad e this process difficult. Once the difficulties were worked out the code could be put together. The functions were called one after the other in the code. This simulated a multitasking system.

Conclusions:

The original goals were met with the exception of implementing sonar for obstacle avoidance. The use of sonar would have been possible if other problems had not occurred. A working sonar receiving board was created. The lack of time did not allow for it to be implemented in the design. All other behaviors worked properly. The following limitations were obvious in my design:

- The battery power will only allow approximately five minutes of operation. This flaw was known before the design was started. The battery for the motor is used quickly. Operation of the robot could take place once the user is done hitting the golf balls.

- The speed of the car was too much for the bumper constructed. Range Rover would clearly take the appropriate behavior when the bumper was struck. The force would usually be too much for the bumper however. The bumper would need springs to absorb some of the impact.

- The design was more dependent on flat terrain that was originally intended. The turning behavior would not always work when the terrain was not flat. The slow decline of battery power would also cause problems with the turn behavior.

Future Work

The first thing that would need to be done is to complete the sonar system. The constant abuse of the front bumper would eventually cause problems. The sonar could be used to avoid collisions that could easily be avoided. Time of flight could be used with the sonar board. This system would still miss small objects like poles. This kind of shape is very common on golfing ranges. The sonar could be used to avoid more of the bigger obstacles.

The other addition that could be very useful would involve the ball counter. The current behavior of blinking on the LED array when the basket is full needs to be changed. A more desirable behavior would use a beacon. When the basket is full, Range Rover could go to the beacon. This would allow for Range Rover to be completely autonomous.

Appendix:

Final Code for Range Rover

```
/*************************************************************************
*
 *Range Rover program
 *created by: Andrew Janecek
 *
 *include files: tjpbase.h , stdio.h, hc11.h
 *functions: init_count() - set up pulse accumlator
 *                          test_bump() - checks front bumper
 *                          test_fence() - use analog port for electric fence reading
 *                 change_counter() - use IR latch to display balls picked up
 *                 circle_turn() - have car go in a circle
 *
 *This program is used to keep Range Rover in a pre-designated area

 *determined by an electric fence.

*************************************************************************
/


/*************************** Includes *******************************/

#include <tjpbase.h>
#include <stdio.h>
#include <hc11.h>

/************************ End of Includes ***************************/


/*************************** Constants ********************************/

#define IRLATCH *(unsigned char *)(0x7000)
#define AVOID_THRESHOLD 100
#define STEERING 1
#define FULLF servo(2,3800)
#define FULLB servo(2,2500)
#define RIGHT servo(0,2000)
#define LEFT servo(0,3500)
#define STRAIGHT servo(0,2800)
#define FENCE analog(2)

/************************ End of Constants ***************************/
```

```
void test_bump(void);
void test_fence(void);
void init_count(void);
void change_counter(void);
void circle_turn(void);

unsigned char count;
unsigned char done = 0;
unsigned char change;

void main(void)
/****************************** Main **********************************/
{

  init_analog();
  init_motortjp();
  init_clocktjp();
  init_servotjp();

  while(1)
  {

          init_count();
          FULLF;
          STRAIGHT;
          wait(1500);

          while(1)
          {

          test_bump();


          test_fence();
          change_counter();

    }
  }
}
/**************************** End of Main ******************************/


/***********************************************
* test_bump will test the value of the bumper   *
* port and will change the acuation accordingly *
***********************************************/
```

```
void test_bump()
{
	if (BUMPER < 20)
	{
		STRAIGHT;
		FULLF;
		wait(200);
	}

	else if (BUMPER > 45  && BUMPER < 65)
	{
		FULLB;
		wait(2000);
		RIGHT;
	   wait(1000);
		STRAIGHT;
	   FULLF;
	}

	else if (BUMPER > 75  && BUMPER < 95)
	{
		FULLB;
		wait(2000);
		LEFT;
		wait(1000);
		STRAIGHT;
	   FULLF;
	}

	else if (BUMPER > 160  && BUMPER < 180)
	{
		FULLB;
		wait(2000);
		RIGHT;
		wait(1000);
		STRAIGHT;
	   FULLF;
	}

	else
	{
		FULLB;
		wait(2000);
		RIGHT;
		wait(1000);
```

```
                STRAIGHT;
            FULLF;
        }
}


/********************************
* test_fence tests the value of  *
* analog port PE2, if the fence  *
* has been dtected then the car  *
* will do a circle               *
********************************/

void test_fence()
{
        if (FENCE < 230)
        {
                circle_turn();
        }
}


/**************************************
* Currently set to pick up 10 balls    *
* and then flash the LED array when    *
* the appropriate number of balls have *
* been picked up.                      *
**************************************/

void change_counter()
{
        count = PACNT;
        if (PACNT >= 10 && done == 0x00)
        {
                change = 0x00;
                IRLATCH = change;
                done = 0x01;
        }
        else if (PACNT >= 10 && done == 0x01 && change == 0x00)
        {
                IRLATCH = 0xFF;
                change = 0xFF;
        }
        else if (PACNT >= 10 && done == 0x01 && change == 0xFF)
        {
                IRLATCH = 0x00;
```

```
            change = 0x00;
        }
        else
        {
           IRLATCH = count;
        }
}



// Initialize the PACTL resister for pulse accumulater
// Set initial PACNT to zero

void init_count()
{
        PACTL = 0x50;
        PACNT = 0x00;
}


/*************************
* series of actions that *
* will make the car turn *
* around in a circle     *
*************************/

void circle_turn()
{
        FULLF;
        RIGHT;
        wait(2000);
        STRAIGHT;
        wait(1000);
        LEFT;
        wait(4500);
        STRAIGHT;
        wait(2000);

}
```