John Harvey

Marine Drive Residence

2205 Lower Mall

Vancouver BC V6T 1Z4

June 15th 2013

Mr. Randall Kerr

Faculty of Applied Science

University of British Columbia

2360 East Mall

Vancouver BC V6T 1Z3

Dear Mr. Kerr,

Subject: Formal Design Proposal for APSC 203

We have prepared the enclosed report, titled Brute Force and Ignorance: a Design Proposal for the Construction of an Autonomous Tape-Following Ball-Throwing Robot, in response to your request for a formal report, as an assignment for APSC 203.

The report details our plans for the construction of a robot intended to compete in the 2013 ENPH 253 robot competition, including complete circuits, mechanical design, proposed software design, and an analysis of risks, individual responsibilities, and a proposed timetable.

We hope that this report will meet with your approval. If you have questions or require additional information, please contact John Harvey at [johnharveybc@gmail.ca](mailto:johnharveybc@gmail.ca).

Respectfully submitted,

John Harvey

Liam Hodgson

Rowan Walsh

Scott Lawson

Enclosure

Brute Force and Ignorance: a Design Proposal for the Construction of an Autonomous Tape-Following Ball-Throwing Robot

Scott Lawson 34359117

Liam Hodgson 29154119

Rowan Walsh 30817118

John Harvey 29240116

June 16th, 2013

University of British Columbia

Engineering Physics 253

# Abstract

This report summarizes the design of a two wheeled robot capable of competing in the 2013 ENPH 253 robot competition. The robot is designed to collect squash balls and shoot them at targets as accurately and quickly as possible. In accordance with the competition rules, the robot is designed to fit inside a 0.027 m3 cube and operate autonomously for at least 90 seconds. The robot will be able to follow tape at up to 0.8m/s and shoot squash balls at 5-8m/s.

A lightweight spinning brush sweeps balls into a holding ramp for collection. The robot has two rear wheels that can be independently steered by two servo motors and two ball casters mounted at the front. This wheel configuration allows the robot to navigate with differential steering as well as granting the ability to strafe horizontally against a wall. Several chassis mounted reflectance sensors are utilized to track the location of black tape which is used for navigation.

The robot is controlled by an ATMega128 based Wiring board using code written in the Wiring language and Wiring IDE.

Contents  
List of Figures  
List of Tables  
List of Definitions

# 1.0—Introduction

The goal of the 2013 Engineering Physics robot competition, and the purpose of the robot detailed in this report, is to acquire and fire squash balls at eight targets over the course of 90 seconds within 3 m2 playing field, without any outside assistance. This report is intended to serve as a method of soliciting feedback from the instructors and TAs of the course, and plans for the construction of said robot.

In each section detailing a mechanical system (firing mechanism, chassis, etc.) the materials, dimensions, fabrication process, and method of assembly have been included, culminating in a step-by-step description of the function of the part. Rough calculations (force required, weight, speed) have been provided where appropriate. The chassis section includes the method by which each component will be fastened to the robot, as well as a description of how the balls will be moved from the collection mechanism and prepared for firing.

The electrical design and sensor system sections include detailed descriptions of the circuits involved in each: schematics and proposed methods of cable management, in particular. The sensor system section additionally includes a complete diagram of all sensors in relation to the TINAH board, with expected input/output values, and the electrical design section includes a list of each protoboard/PCB to be used, with approximate size, number of connections, and physical location.

Potential issues as well as associated solutions and alternative methods are included in the risk management and contingency planning section. Probabilities have been estimated and assigned to each problem, as well as impact and changes to the project each would cause.

The task list, major milestone, and team responsibilities is relatively self-explanatory. A proposed calendar, list of each team member’s main areas of responsibility, and rough Gantt chart are included.

## 1.1—Overview of Basic Strategy

The basic strategy or the robot is fairly simplistic, working from the idea that the simplest ideas are the easiest to implement successfully. The robot will initially acquire tape, move to the back of the arena, and collect balls by forcing the collecting wheel into the wall. After collecting, the robot will reverse, spin 180 degrees, and ‘wobble’ back and forth while moving forward until it acquires tape. At this point, it will advance until it reaches the end of the tape, and continue forward (no longer following tape) until it comes into contact with the front wall. It will then maneuver so that the front of the robot is perpendicular to the targets.

At this stage we rotate our wheels, so that, without having moved the chassis of the robot, we can move side-to-side in front of the targets. Each time a target is detected with both of our 1000 Hz detectors, the robot will stop and fire a ball. This ball, ideally, will be collected immediately after firing. After three seconds of waiting, to allow for ball collection, the robot will continue to move sideways, repeating the process at each detected target. When the robot reaches the opposite wall, detected by one of the side-mounted touch sensors, it will reverse direction. If and when it runs out of balls, it will continue to move in the direction it was moving, until the rear-mounted QRD sensors indicate that the robot is directly in front of tape. At this stage, we will reverse, leaving the wall far enough to rotate, acquire tape, and proceed to collect more balls in the same manner as previously described.

# 2.0—Mechanical Components

Contained within this section is a description of each primarily mechanical system contained within the robot: the chassis, firing mechanism, collection mechanism, drive system, and internal ball handling.

## 2.1—Chassis

The main characteristics we wanted in the chassis were simplicity, rigidity and modularity, as it will be the structure onto which the rest of the robot components will be mounted. The base of the chassis will consist of a flat rectangle of 3mm aluminium sheet metal, with arms extending forward leaving a gap for the brush. 3mm aluminium provides enough stiffness for our requirements, and allows us to add threaded holes to the chassis. Clearance holes and tapped holes will be cut in the base to allow for rapid mounting of components, and two large holes are located near the rear corners of the chassis for the wheels, where the drive system will be located. This part will be fabricated using the waterjet, and the majority of mounting will be done using M3 screws, either directly screwing into the chassis, or using clearance holes and nuts. Making the base out of one single flat piece means that we are able to add additional holes after the initial fabrication if we should need to modify or add component mounting.

### 2.1.1—Brush Mounting Structure

The second part of the chassis is structure for mounting the brush and wall that is used for ball collection, which is also made out of 3mm aluminum. The division of the chassis into two major parts was done to simplify fabrication (fewer sheet metal bends required) and to allow for independent development and testing. As it is, the brush and ball collection mechanism can be assembled and tested separately from the main chassis, which will have the TINAH, the sensors and motors mounted to it. This will allow us to test the robot’s movement and sensing without having mounted the collection mechanism. The brush mounting structure will be waterjet cut, and the flanges and two holding arms will be bent. While the two arms are bent to an angle just under 90o, there is sufficient flexibility in the design that the error which is likely to be present in our fabrication will not affect its effectiveness.

### 2.1.2—Brush Holders

There are two brush holders which act as an interface between the brush shaft and the chassis. They were designed to allow for easy removal of the brush assembly from the chassis, as we will likely have to modify it frequently in testing. These components will be 3D printed, as they contain angles which would be difficult to machine. The idea it that both holders will slide onto either end of the brush shaft, and the holder will then slide into the slot cut in the vertical sheet metal brush mounting structure. This allows for flexibility in the horizontal position of the brush shaft. Once the correct location has been determined, the screws at the top and bottom the part will be tightened, closing the upper and lower arms against the chassis structure and preventing sliding. To remove the brush, one needs only to loosen the screws and slide the entire assembly out of the slots.

### 2.1.3—Ball Guard

A ball guard will be mounted above the brush, connected at each end to the chassis using screws. Its purpose is to prevent any balls that are being conveyed along the top side of the brush from flicked back out of the robot. It will be made of hand or waterjet cut sheet metal aluminum, which will be bent at the ends. In testing we observed that it is possible (though unlikely) for a ball to get caught between this guard and the brush, and forced into the inside of the brush, so testing will be performed to ensure that the geometry of the ball guard prevents this.

### 2.1.4—Omni-Bearings

The robot will be driven by two powered wheels at the rear of the chassis, however it will also have unpowered bearing wheels near the front of the robot. These will consist of 3D printed holders for small (about 1cm) diameter ball bearings. The printed part will be screwed into the bottom of the chassis, allowing the bearing to roll freely between the printed holder and the chassis sheet metal.

### 2.1.5—Internal Ball Routing

A ramp will be fabricated out of sheet metal and riveted together; it will guide the ball from the brush to the firing mechanism. The design of this component has yet to be finalized, as it will be a simple design, and it is dependent on the finalization of the firing loading mechanism and location.

### 2.1.6—Mass Budget

Base Chassis Structure (aluminum) – 460g

Brush Mounting Structure (aluminum) – 370g

Brush (wood, fishing line, aluminum) – 30g

2x Brush holder (ABS) – 13g

3x geared Barber Coleman motors –

1x ungeared Barber Coleman motors –

2x servo motors –

## 2.2—Brush

The brush is the critical component of the robot’s ball collection mechanism. It spins at low speed, pulling in any balls that it encounters, lifting them up a vertical wall and funnelling them towards the storage and firing mechanisms.

The brush consists of two disks of different diameters (80mm and 60mm) mounted axially on a shaft 260mm apart. The two disks have inward-facing T-slots cut at equally spaced intervals around the outer diameter, which are strung with fishing line. The fishing line passes back and forth between the slots on both disks, forming a long cylindrical surface with which to grab and pull in balls. Fishing wire was chosen as it is very lightweight, yet has a high tensile strength, which is important as each loop is tensioned between the end disks. Fishing wire is also flexible, allowing it to deform inwards when it is rolling over a ball – this allows us to leave a gap smaller than the ball diameter (approximately 20-30mm for a 40mm ball) between the ground and the outer diameter of the brush. Testing has shown that the brush will easily pull in a ball and push it up the vertical wall behind the brush, however further testing will need to be done to determine the optimal spacing between the brush and the floor/wall, as well as to verify what surface finish will provide sufficient friction on the vertical wall to allow it to pushed up. The spacing of the strung wires is such that balls should not pass through the string into the interior of the brush, however an inner tube may be added if testing showing that this intrusion is a possibility. In the prototypes fabricated so far, the brush has been strung with a single length of fishing line. This presents a risk, as if one section of the line breaks, the entire brush will be rendered useless. To overcome this, we are considering stringing the brush with multiple shorter lengths of fishing line, or else fully threading the entire brush with a few long overlapping lengths of line. Another property of fishing wire is that it is fairly slippery, so that balls will not jam against it. This brings up the reason for the unequal wheel diameters: the lower edge of the brush will be parallel to the ground, meaning that the upper edge will be at an angle to the horizontal. Extending the vertical wall (up which the ball is pushed) above the brush allows the ball to roll between the top of the brush and the wall instead of simply being pushed over, and due to the slope of the brush, it will be pushed along the top of the brush towards one end. Because the brush is constantly rolling, if multiple balls are on top of the brush at once, they will not jam together, which was our concern with using a static ramp or tube; it also means that we do not need another mechanism for funnelling the balls to one point.

The size of the brush has been determined based on various constraints; however the optimal size has yet to be finalized. The maximum brush diameter is 130mm, as if the brush is any larger than this the brush can’t pull in a ball from the rear wall. Minimum diameter is harder to calculate, however smaller is better, as it leaves more space for other components on the chassis. The difference between the two disk diameters, which creates the slope for the ball to roll down, can be fairly small; about 20cm difference in diameter. This sizing is an initial estimate, however the size will likely change as we perform tests.

The brush will be driven by a belt running around the end of its shaft, outside the vertical mounting arms. The drive motor will be mounted rear of the vertical wall. A flexible belt allows the axes of the brush shaft and motor shaft to be slightly off-parallel, as will be the case.

The brush will be mounted at the front of the robot between the two arms of the brush mounting structure. Preliminary prototypes have used disks made using laser cut wooden disks constrained with nuts on a threaded rod. We are considering using metal disks for reasons of strength; however this will increase mass and increase the chance that the fishing wire will be cut by sharp edges.

## 2.3—Firing Mechanism

The firing system uses two rollers, spinning in opposite directions. The rollers use a similar design to the ball-collector brush: strings strung between two plates with holes along their circumferences. The rollers driven by a single motor and are geared together, ensuring both spin at the same rate. The ball is lifted from a ramp below the firing mechanism into place between the rollers by a servo-actuated arm.

### 2.3.1--Rollers

The firing system consists of two rollers spun opposite to each other by an Un-geared Barber Coleman. The rollers consist of two separated plates with string strung between them along the circumferences. It was observed in tests of the collector mechanism that this string design deforms well to the ball and provides excellent grip. The string used for the firing rollers will be Kevlar Size 5 thread, which provides the necessary strength with an approximate breaking strength of 600 N. To keep friction minimal between the rollers and the stationary axle, the roller plates will be cut from brass. Despite a longer spin-up time, the added weight of the brass will provide most of the energy when firing a ball (the motor's energy is stored in the angular momentum of the discs during spin-up).

Include render of roller

The un-geared Barber Coleman motor is used for its higher maximum speed. Assuming a 5 m/s exit velocity of the ball and the 50 mm diameter rollers used in the design, the discs will need to be spun to about 1900 rpm. A 1:1 gear ratio between the two rotors will ensure that their speeds are exactly synchronized, which is important for consistently firing the balls straight. 3D-printed guides will constrain the balls vertically between the rollers, away from the gearing.

### 2.3.2—Lifting Mechanism

Once balls have been collected, they are corralled into a line below the firing mechanism. This line ends with the next ball to be fired sitting in the small hollow of a servo-actuated arm. A reflectance sensor within this hollow confirms the presence of a ball before firing. The arm lifts the ball into the gap between the rollers; once the ball makes contact with the roller's strings it is quickly accelerated between the rollers and towards the targets.

Include diagram of lifting mechanism

### 2.3.3—Tuning of the Firing System

Calculations can be done to obtain rough estimates and checks for the roller diameter and speed, but as any friction and play in the physical mechanism cannot be fully taken into account, tuning the system will have an important role in determining its final configuration. This is taken into account with the design of the firing system's structure; with slots to adjust the angle of the entire firing assembly relative to the chassis and a ball-lifting system that can be adjusted to work with a range of firing heights.

In particular, the firing speed will need to be carefully tuned: the possibility of targets bouncing back from an "away-flipped" state limits the exit velocity, while the need to consistently knock-over the targets sets a lower bound.

# 3.0—Electrical Systems

Contained within this section is a description of the electrical systems contained within the robot: the sensor array and basic electrical design—inputs and outputs into the main board, PCBs, and voltages/power provided.

## 3.1—Sensor Systems

The sensor system consists of three varieties of sensors: touch, IR light, and reflectance. The touch sensors serve to align the robot with the front wall when firing at targets, and indicate when the robot is at the end of a wall. The IR light detectors are primarily used in acquiring targets, but a 10 kHz sensor has been included as a contingency measure, if other methods of orientation should fail. Five reflectance sensors are used to follow tape, and one is used to detect whether the robot has balls ready to fire. While the signals will be routed to the TINAH board, the sensors will draw their power from two 9V LiPo batteries.

### 3.1.1—Reflectance Sensors

Each reflectance sensor will be attached to an LM311 comparator and a potentiometer. This will allow the reflectance sensors to output a digital signal (the input voltage compared to the voltage across the potentiometer) reducing the number of analog pins required by six. Four of the sensors will be mounted on the front of the robot—two near the middle, and one at each side. The two near the middle will be used for following tape, and the two on the sides will be used to detect the end of tape near the targets. One sensor will be towed behind the robot, used to sense when the robot is directly in front of tape, allowing us to reverse directly onto tape from firing, as opposed to having to re-acquire tape blindly. The final reflectance sensor is internal, and used to sense when a ball is in the loading mechanism. Our intention is to replace the potentiometers with fixed-value resistors after calibrating them to sense tape/ball, as appropriate.

### 3.1.2—IR Sensors

The IR light detectors are the most complicated sensor. Each requires ten circuits: a detector, DC filter, amplifier, two active filters, a rectifier, and four unity-gain amplifiers. Our intention is to leave each of these as discrete circuits, as opposed to combining several in one. This is to ensure that debugging and tuning are relatively easy. Two of the sensors, mounted facing the front, will detect 1 kHz IR light at a distance of one foot, allowing us to aim at targets. The third will detect 10 kHz light at a variety of distances (achieved by splitting the detected signal between two amplification circuits designed for ranges of 1-3 and 4-6 feet) and will be used as a last-resort method of re-orienting, as well as a method of finding the back wall to collect balls. The two 1 kHz sensors will be mounted six inches apart, to minimize overlap between individual targets.

### 3.1.3—Touch Sensors

The four touch sensors are, at this stage in planning, somewhat in flux. The initial plan is to use digital touch sensors—buttons. One sensor will be mounted on each side of the robot, used to detect when a wall has been reached. The front sensors, while initially planned to be digital switches, might evolve into more complex analog sensors, putting bearings on metal ‘whiskers’ which rotate as the bearings come into contact with the wall, held in the default position using springs. Their rotation will be transformed into a measure of how hard we are pressing into the wall, allowing us to more accurately sense whether or not we are perpendicular to the wall. If it is possibly to use a PID algorithm to stay perpendicular to the wall using digital touch sensors, this will not be necessary, but we see no easy way to test this until our robot is moving: our first goal.

### 3.1.3—Sensor Cable Routing

Cables leading from the sensors and to the TINAH board and batteries will be collected and routed as a single entity wherever possible, as opposed to routing individual wires. We intend to use ribbon cable as frequently as is possible, and to have the battery inputs for all our circuits as close to the TINAH board as prudent, to ensure that the cables remain grouped as long as is possible. Our intention is to route the four QRD circuits at the front in conjunction with the touch sensors, simplifying the wiring of the majority of the components located away from the robot’s core. Each tape sensor has three inputs (VCC, ground, and signal) and each touch sensor, two, bringing us to a total of 20 wires required. 24-conductor ribbon cable will be routed from these sensors to a permanently mounted shrouded box header, which will, in turn, be routed to the TINAH board.

The three IR sensors will be routed similarly, using three three-wire insulated cables, tied together and routed to a permanently mounted connector, which in turn leads to the TINAH board. The trailing tape-follower will consist of three cables, also routed as a unit.

## 3.2—Electrical Design

The circuits used in the robot can be broken into two categories: sensor, and drive/power. The circuits which interpret (compare, filter) the sensor signals will be placed near the sensors themselves, and other circuits, such as the H-bridges used, will be placed near the TINAH board. The batteries will be placed as close to the TINAH board as possible. No cable will interface directly with the TINAH board: all cables will go through a permanently mounted circuit board which has leads to battery outputs and TINAH inputs.

### 3.2.1—Sensor Circuits

The comparators for the reflectance sensors used to follow tape at the front of the robot will be mounted behind the larger-radius wheel, on the right of the robot. This board will contain 4 LM311 comparator chips, and several resistors. It will be approximately 60 by 25 mm, and mounted sideways, parallel to the side of the chassis. Removing and replacing the circuit will be as easy as pulling it out—it will be held in place loosely and constrained by two bolts during the competition itself.

The touch sensors require no circuit to interpret their signal. They will be connected via header pins on the reflectance sensor interpreter board to the wire-to-board header mounted on the same board that goes to the ribbon cable.

The internal reflectance sensor (used to determine whether or not the robot has a ball to fire) and trailing tape follower will be individually routed to the TINAH input board.

The IR sensors will be individually routed to the TINAH board/battery via shielded cables. The two 1 kHz wires will be twisted together before joining the sensor ribbon cable and moving back towards the TINAH board. The 10 kHz cable will be routed alongside the trailing tape follower cable.

### 3.2.2—Drive/Power Circuits

The two H-bridge circuits will be enclosed in a bent sheet-metal box (approximately 60 by 120 by 40 mm) kept close to the TINAH board. The inputs to the H-bridges will come from the TINAH board/battery board, and the outputs will move to snap-fit connectors, which go to the motors. All of the servos (two wheel-rotation, one loading mechanism) will go directly to the TINAH/battery interface board. The two other geared Barber Coleman motors (collector and firing mechanism) will go directly to the TINAH/battery interface board, as neither requires H-Bridge circuits.

A long, thin protoboard will be placed directly above the TINAH board, serving as a TINAH board/battery interface. All wires from sensors and motors will plug into this board, as opposed to the TINAH board itself. It will provide various voltages (5V, 9V, and 12V) and be permanently, rigidly mounted. It will have a variety of inputs, from single wire to ribbon cable, and route the appropriate inputs/outputs from the TINAH board to the cables plugged into the battery/board interface, making unplugging and replacing/debugging individual sensors/motors easier.

TABLE SOMETHING: ELECTRONICS AND STUFF

|  |  |  |  |
| --- | --- | --- | --- |
| Name (quantity, size) (pins) | Function | Input/output values | Comments |
| H-bridge (1, 60x120)  Input pins: 12V, ground, 2 TINAH PWN  Output pins: 2 battery PWM | Locomotion motor inputs/outputs | Inputs: 12V, ~1.3 A (max) Outputs: 12V ~1.3 A (max) | Enclosed within a metal box, 40 mm tall. Cables to motors are three-strand shielded wire |
| TINAH board/battery interface (1, 15x150) Input pins: Battery ground, 5V, 9V, 12V, various sensors Output pins: VCC and ground, sensor data (to various TINAH inputs) | Provides a single interface for connecting inputs/outputs requiring both TINAH and battery. | Three rails of outputs/inputs (VCC, ground, signal) and three rails of constants (5V, 9V, 12V) | Mounted close to/over TINAH and battery, near back of robot. Signal rail is not continuous, and has individual wires leading to TINAH inputs |
| Reflectance sensor interpreter (1, 25x60) Inputs: 4 QRD outputs, 4 touch outputs Outputs: 4 9V/ground (QRD), 4 touch inputs (5V) | Four LM311 comparators, and touch sensor to ribbon cable inputs. | Inputs: Front tape-following sensors (5V, 40mA) touch sensors (5V) Outputs: compared (digital) tape-following signal (5V), touch sensors (5V) | A single ribbon cable comes from this board and is routed to the board/battery interface. It includes touch sensor and compared tape sensor outputs. |
| IR sensor filter (3, 75x25) | Amplifies and filters 1 kHz and 10kHz IR inputs. | Inputs: IR light, 9V, ground Outputs: 0-5 volts | Stored in bent sheet-metal boxes, mounted 6 inches apart, centered above the brush. Outputs are sent via. |

# 4.0—Software Code and Algorithm

The robot controller is the ATMega128 based Wiring board with a TINAH shield. The Wiring board runs programs created using the Wiring language and Wiring IDE.

## 4.1—Algorithm Overview

The robot can be modelled as a finite state machine. The highest level states are tape following, collecting, wall strafing, and shooting. The tape following state is central since it facilitates transitions between the collecting and wall strafing states. If the robot is disoriented at any point, this state provides moderate failure recovery since the robot can realign with the nearest tape. Shown below is the robot’s high level state diagram.

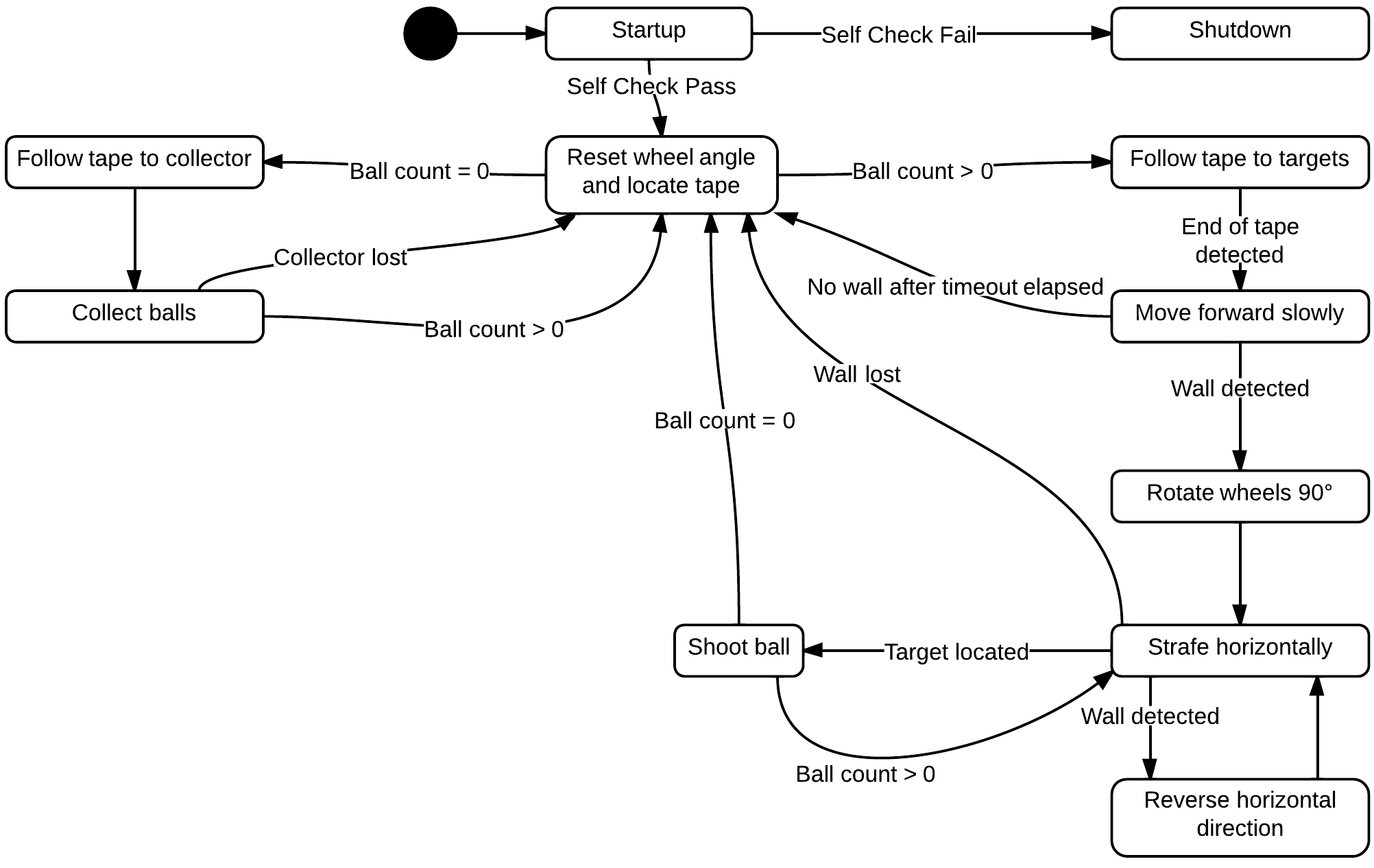


Figure 1. Robot control is implemented as finite state machine.

### 4.1.2—Locating Tape

The QRD sensor mounted on the tail permits tape detection while strafing. By counting the number of tape detections, it is possible to locate the centre tape. The below pseudocode provides one possible implementation of the centre tape algorithm. Since QRD readings are sampled dozens of times per second, it is important to note that this algorithm only counts *changes* in tape state. That is, it only detects tape when the previous reading did not detect tape. This ensures that each strip of tape is only counted a single time.

|  |
| --- |
| // PRE: Start facing target wall  // POST: Stopped with rear QRD sensor on center tape  void LocateCenterTape()  {  Strafe(LEFT);  while(!LeftTouchSensor()) // Wait for left wall alignment  Update();  Strafe(RIGHT); // Start moving right  int tapeCount = 0;  while(tapeCount < 2) // Look for center tape  {  // Prevents same tape from being detected twice  if (TapeRiseDetected()) tapeCount++;  else Update();  }  StopMotors(); // Stop at center tape  } |

Once the tape has been located, the next step is to move the robot onto the tape. This will be accomplished by rotating the robot until the front mounted QRD sensor has detected the tape. The following pseudocode demonstrates this manoeuvre.

|  |
| --- |
| // PRE: Situated on tape  // POST: Situated on tape, but rotated 180 degrees  void RotateOnTape(int turnRate = 100, unsigned int timeout = 10000)  {  SetWheelAngle(0);  LeftMotor(turnRate);  RightMotor(-turnRate);  while(!FrontTapeDetected())  {  Update();  if(timeout <= 0) TapeRecovery();  else timeout--;  }  StopMotors();  } |

### 4.1.2—Tape Recovery

If the tape is unexpectedly lost and can’t immediately be recovered, a tape recovery algorithm will be called. This algorithm will cause the robot to move in an expanding spiral. This ensures that the tape closest to the robot will be detected first. It is a recursive algorithm that will keep looking for tape *ad infinitum*. It is difficult to implement a reliable tape recovery algorithm due to lack of sensor input and the limited computing power of the ATMega128. Because of this, heuristic algorithms such as the below algorithm will be considered for implementation.

|  |
| --- |
| // PRE: Sensors cannot detect tape  // POST: Moves in expanding spirals until tape is detected  void TapeRecovery()  {  int leftSpeed = -200;  int rightSpeed = 200;  int timeout = 0;  LeftMotor(leftSpeed);  RightMotor(rightSpeed);  while(!FrontTapeDetected())  {  Update();  timeout++;  // Increase spiral radius  if (timeout % 100 == 0) LeftMotor(++leftSpeed);  if (timeout < 10000) continue;  // Call again to reset spiral  TapeRecovery();  return;  }  StopMotors();  } |

### 4.1.3—Tape Following

The robot has two tape following states, one that navigates to the collector (10 kHz emitter) and one that navigates towards the targets. A standard PID algorithm, as shown below, will be implemented to follow the tape. The end of the tape on the collector side does not have a tape intersection, while the target side does. The tape’s end will be detected by polling the two front-side mounted QRD sensors that act as intersection detectors.

// Computes new PID values and calculates a new motor speed for

// the left and right motors.

void Move()

{

if (leftDetected && rightDetected) error = 0;

else if (!leftDetected && rightDetected) error = TOO\_LEFT;

else if (leftDetected && !rightDetected) error = TOO\_RIGHT;

else if (!leftDetected && !rightDetected)

error = (previousError <= TOO\_LEFT) ? -OFF\_TAPE : OFF\_TAPE;

float proportional = (float)error \* proportionalGain;

float derivative = (float)(error - previousError) \* derivativeGain;

motor.speed(LEFT\_MOTOR, speed + (proportional + derivative));

motor.speed(RIGHT\_MOTOR, speed - (proportional + derivative));

previousError = error;

}

### 4.1.4—Ball Collection

The ball collection manoeuvre is executed by driving the robot forward in the direction of the collector (10 kHz emitter). QRD reflectance sensors paired with comparators will be mounted in the ball hopper to detect successful ball collections. When a ball rolls past a QRD sensor, it causes a temporary state change from LOW to HIGH. This will trigger a standard hardware CPU interrupt that will increment the ball count.

### 4.1.5—Wall Following

Two front mounted contact switches indicate whether the robot is currently contacting a wall. After the robot reaches the end of the tape on the target side, it will slowly approach the wall until contact is made. When this occurs, it will rotate the rear wheels by 90°. This wheel rotation lets the robot strafe horizontally along the wall. A PID algorithm will be implemented to keep the robot oriented aligned along the wall at all times.

### 4.1.6—Target Acquisition and Shooting

Once a ball has been collected and the robot is aligned along the target wall, it will be considered capable of firing at the targets. Infrared sensors provide analogue input to the Wiring board to indicate whether the target has already been hit. If the input voltage is above a certain threshold then a valid target has been acquired. A PWM command will tell a servo to load the ball into the spinning disks and fire.

Decision theory influences the shooting behaviour. If no targets can be detected, then either the robot’s sensors are not functioning properly or all of the targets have been hit. It is assumed that the robot is more likely to fail than it is to succeed. With this assumption, it is in the robot’s best interest to blindly fire at targets. If all targets had been hit, it implies that the robot is doing very well, thus making the deleterious effects of unnecessary firing negligible. The below flow chart provides an overview of the shooting algorithm. The *wall count* variable is used to determine whether no targets were detected.

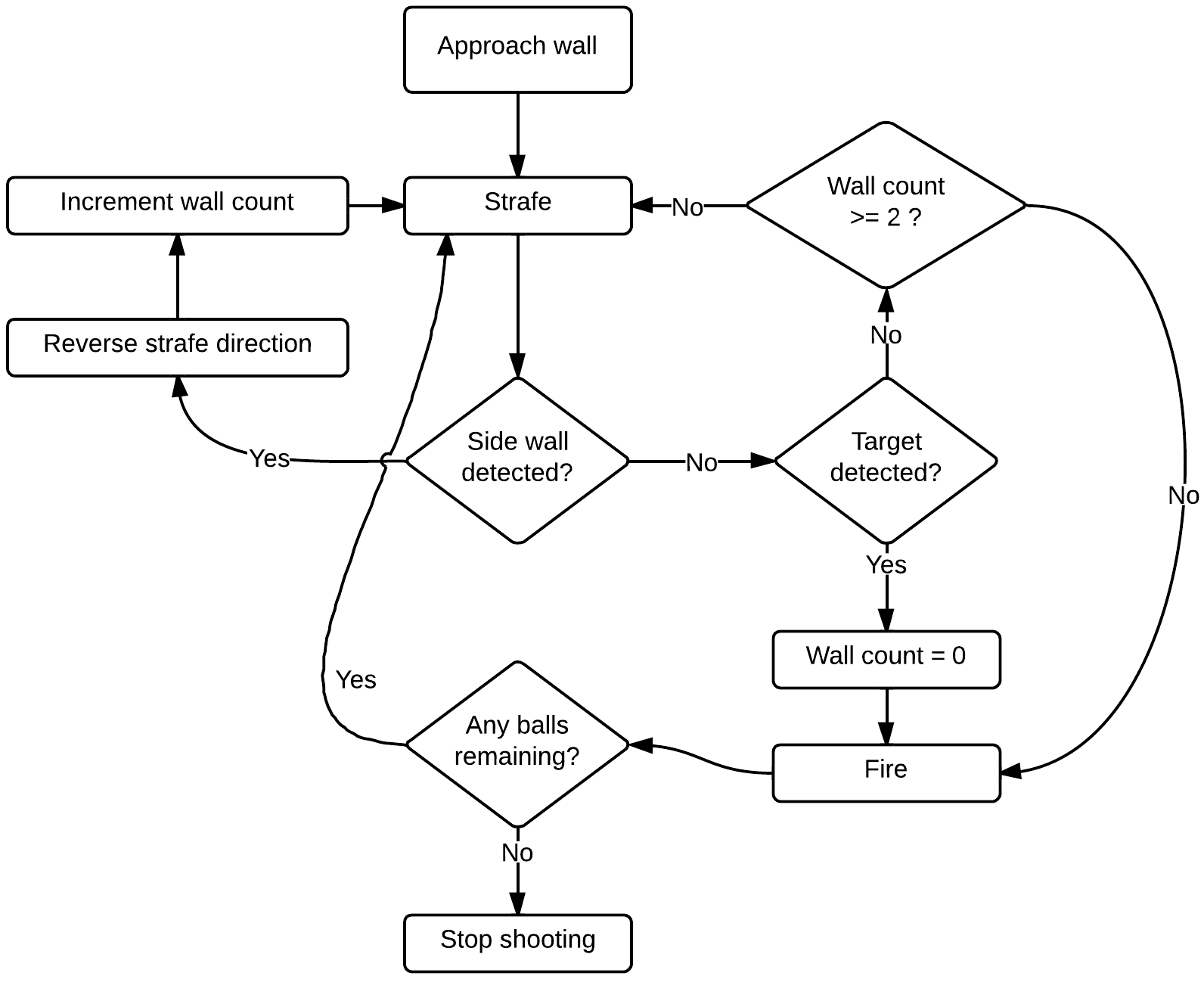


Figure 2. The shooting algorithm flowchart. When no balls remain, the robot stops shooting and enters a ball collection state.

# 5.0—Meta-Analysis

This section pertains to the more human elements of constructing a complex system: failures, risks, timelines, and individual responsibilities are detailed in this section.

## 5.1—Team Responsibilities, Major Milestones, and Timeline

### 5.1.1—Team Responsibilities

Each team member has taken a lead role in a certain part of the robot. While all team members take part in the design, fabrication, and implementation of each sub-assembly in the robot, the lead is responsible for keeping the team informed as to the state of their section, and keeping development of their section on-schedule.

John Harvey—Electronics lead, overall organization

Liam Hodgson—Collection System and Chassis lead

Rowan Walsh—Firing Mechanism and Drive System lead

Scott Lawson—Software lead

### 5.1.2—Major Milestones

By major milestone day (July 8th) the robot needs to be able to accomplish the following tasks

* Move
  + Chassis will be fabricated, and most electronics as well as drive system will be mounted.
  + H-bridge and other motor control circuits will be fabricated and tested.
  + Software will be written and implemented to drive motors.
* Follow tape
  + The tape following sensors will be mounted along with the comparator circuits.
  + The tape following algorithm will be implemented.
* Collect balls
  + Brush size and design will be finalized.
  + Brush will have been mounted on the chassis and connected to the drive mechanism.
  + Brush will be able to pull in balls and transport them to firing mechanism.

### 5.1.3—Timeline

The flow chart shows the major tasks to be completed, and the interdependencies of the tasks. The tasks furthest to the right of each branch do not have any dependencies.