Robot design proposal

Scott Lawson 34359117

Liam Hodgson 29154119

Rowan Walsh 30817118

John Harvey

University of British Columbia

Engineering Physics 253

# Abstract

Presented here is the design of a novel 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 accurately and quickly. In accordance with the competition rules, the robot is designed to fit inside a 300mm3 cube and operate autonomously for at least 90 seconds. The robot will be able to follow tape at speeds up to 0.8m/s and shoot squash balls at 3-6m/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 has two ball casters mounted at the front. This wheel configuration allows the robot to navigate with differential steering while also being able strafe horizontally against a wall. Several reflectance sensors are utilized to track the location of black tape which is used for navigation.

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

# Introduction and Overview of Basic Strategy

This report is intended to serve as the plans for the construction of a robot for the 2013 Engineering Physics 253 robot competition, as well as a method of soliciting feedback from the instructors and TAs of the course.

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.

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.

# Electrical Design

The circuits used in the robot can be broken down into sensor and drive circuits. The circuits which interpret (compare, filter) the sensor signals will be placed near the sensors themselves. 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, to reduce the moment of inertia. 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.

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.

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.

|  |  |  |  |
| --- | --- | --- | --- |
| Name (quantity, size) (pins) | Function | Input/output values | Comments |
| H-bridge (1, 60\*120)  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, 15\*150) 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, 25\*60) 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, 75\*25) | 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, centred above the brush. Outputs are sent via. |

# Sensor system

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 VOLTAGE LiPo batteries.

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.

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 using a CIRCUIT, which switches between several amplification values) 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 DISTANCE apart, to ensure that THING.

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.

In all cases, cables leading from the sensors and to the TINAH board and batteries will be collected and routed as a single entity, as opposed to a group of 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.

# Software Code and Algorithm

## Overview

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.

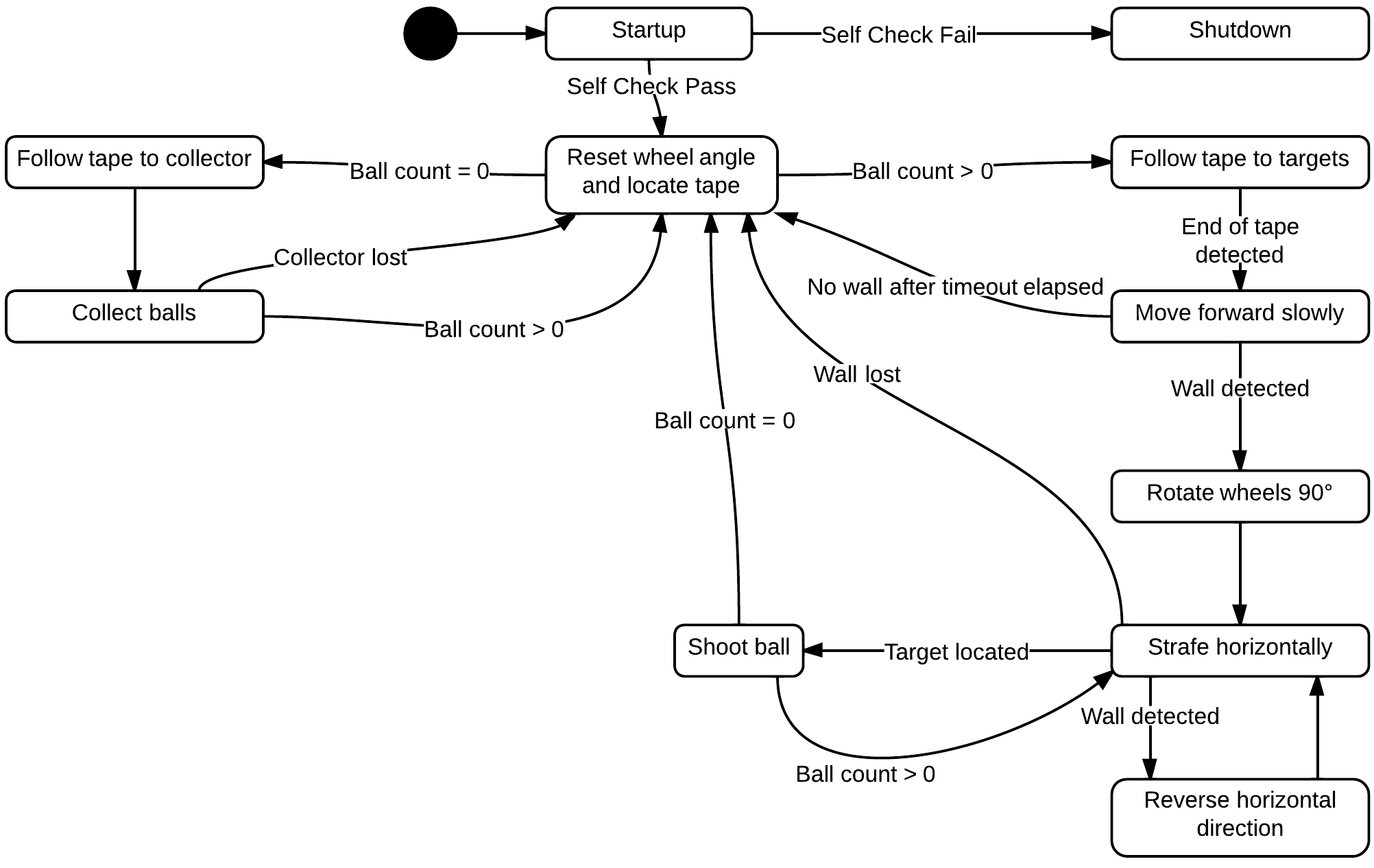
The robot can be modelled as a finite state machine. The highest level states are tape following, collecting, wall strafing, and shooting. The tape following state is central since it facilitates transitions between the collecting and wall strafing states. If the robot is disoriented at any point, this 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.

## 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();  } |

## 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();  } |

## 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;

}

## 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.

## 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.

## 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.

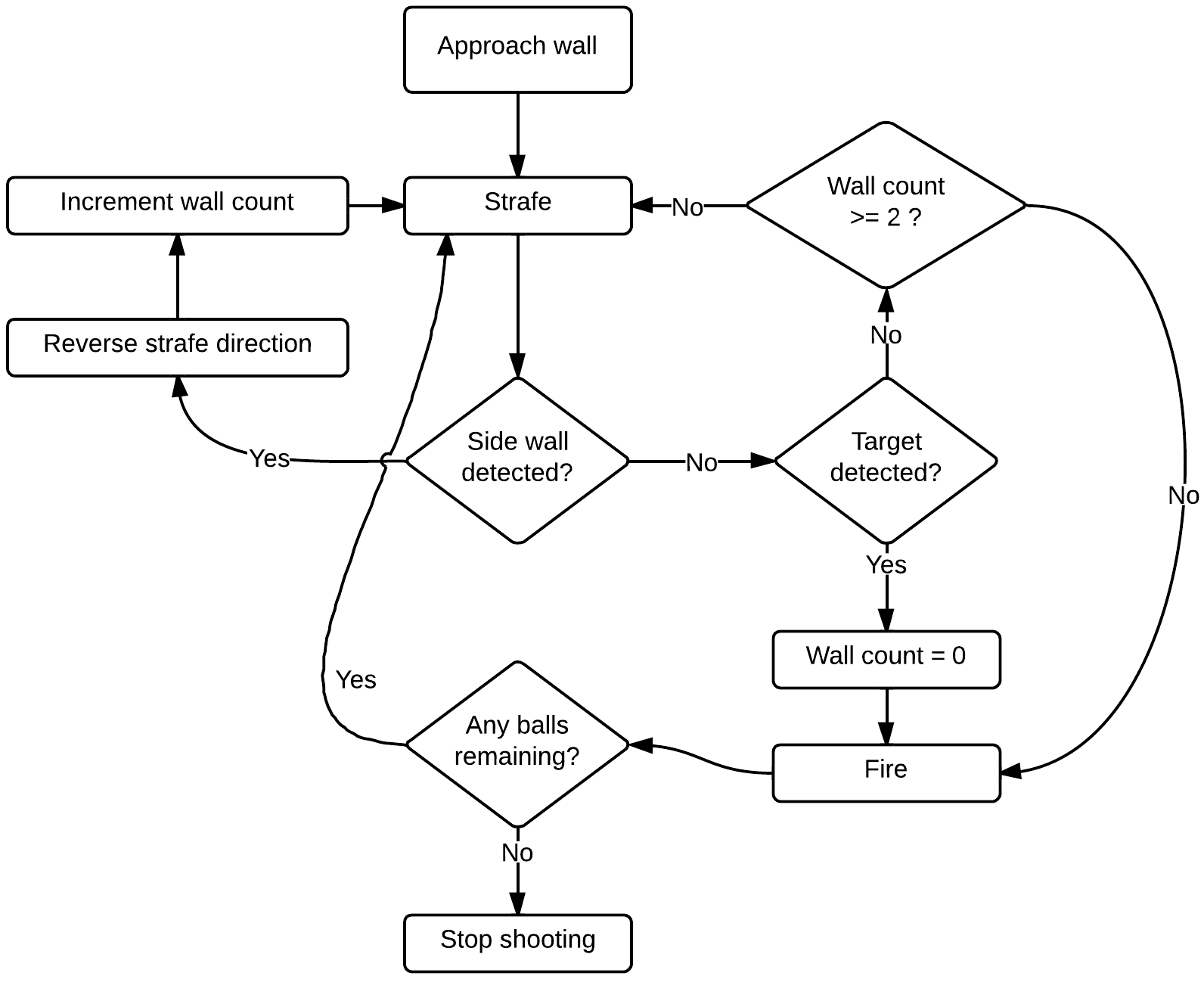


Figure 2. The shooting algorithm flowchart. When no balls remain, the robot stops shooting and enters a ball collection state.