Electromagnetic propulsion / Coilgun
The University of Kent

Mustafa Omar, Partner: Jordan.T

25/02/2020

1 Abstract

In this experiment, an electromagnetic propulsion system was built, with the aim to make it as reliable as possible at launching a projectile.The system consists of two solenoids and two sensors connected to an Arduino [1], the objective is to successfully accelerate the projectile through both of these solenoids without having the projectile get stuck in the middle of at the equilibrium point, therefore the purpose of the sensors is to control the flow of current, this system was tested and a success rate of 40%, 60% rate and finally 100% were achieved when all the sources of failures were formally addressed. Furthermore the experiment also looked at using a calibrated timing method, where instead of the sensors controlling the flow of current to the solenoids it was all hard coded into the python code, the optimum activation period for this design (length of each part considered) was found using trial and error, these periods were 150 milliseconds (period of the activation of the magnetic fields) from the solenoid nearest to the projectile, and 350millisecond from the other solenoid, allowing the projectile to travel from the stationary point, into the solenoid and then turns off before the projectile gets stuck in the equilibrium point (both periods start when the first signal is triggered), aside from determining the optimum times, the success rate of this method is 100%.
This experiment was carried out with the goal of making a simple electromagnetic propulsion system that's reliable, and this was successfully achieved.

2 Introduction

Since the industrial revolution, electrical and electronic technologies have been seeing a constant evolution in terms of their quality and use-ability, from first generation electric generators, to processing machinery (computers), to the code used to program the computers to execute commands, there is always a consistent positive co-relation. At this point in time technology has become much better compared to it's earlier days and the aim of this investigation is to figure out new uses for already existing technology, by combining the cutting edge computer technology of ARDUINO, the advanced software language that's provide with the it, and a well made small sized solenoid, we will attempt to make an electromagnetic propulsion system in order to discover the imitations that come with the use of these technologies combined, an attempt will be had at making the technology reliable, and to see what potential applications it may have.

3 Procedure

3.1 Equipment:

Table 1: Show the main equipment used to build the device

Copper Wire	Optical Sensor
Power Supply	10k potentiometer
Arduino Nano	3x 10k, 2x220, 2x150 Resistor
Tactile Switch	PCB Terminal
MOSFET driver IC	PVC pipes
Diode	Cow magnets of varying length 1.5cm - 2.5cm (the projectile)

3.2 Theory

In this experiment we will attempt to make an Electromagnetic gun, that takes a projectile (Cow magnet) and accelerates it though the solenoid (Electromagnet), thus launching it at a velocity (V). We will aim to make the system reliable thus launching the projectile consistently without fail.
The Solenoid is capable of generating a Magnetic field of magnetic field strength (B), the magnitude of which is determined by equation (1).

$$B = \frac{\mu_0 I N}{L} \tag{1}$$

where (μ_0) is the permeability of free space, (I) is the current flowing through the loops of the solenoid, (N) is the number of turns of the solenoid and (L) is the length of the solenoid. Furthermore the potential energy (U) stored within the solenoid when a current is applied to it, is directly proportional to the magnetic field strength and follows equation (2)

$$U = \frac{B^2}{2\mu_o} AL \tag{2}$$

where (A) is the cross section of the solenoid assuming an ideal system (vacuum environment with a friction-less tube), a projectile that's placed directly in front of the solenoid will experience a force and thus a transformation in it's energy state from 0 to U; the potential energy stored by the solenoid translates into the maximum kinetic energy of the solenoid, therefore the maximum velocity of the projectile is calculated through equation (3):

$$V = \sqrt{\frac{\mu_0 I^2 N^2 A}{m_0 L}} \tag{3}$$

Where (m_0) is the mass of the projectile.This confirms that a magnet that's placed in front of a solenoid that's generating a magnetic field (B), will be accelerated towards the center of the solenoid, however it's apparent from source [4], that the projectile doesn't get released from the potential field of the solenoid and that the projectile experiences a negative force after it's center passes the center of the solenoid, this could only mean that there is a negative force being experienced by the projectile, after that point, which results in it being attracted back towards the center, the reason that this negative force is experienced is because the projectile is attracted to the point of equilibrium of the potential field (at the center), this is because the positive (north) and negative (south) regions of the projectile are attracted to the solenoid's opposite pole; the initial positive acceleration occurs because the attraction of the north pole of the projectile to the south pole of the solenoid in the positive direction (i.e from the entry to the exit) and vice versa for the south pole of the projectile, as the projectile passes the center, the net force is zero because both poles of the projectile are equally and oppositely attracted to opposite poles from the solenoid, furthermore as the projectile moves past the center point, it experiences a force in the negative direction now, this is because the attraction of the North pole of the projectile to the south pole of the solenoid is now in the negative direction i.e from the exit towards the entry. This means that in order to get the projectile firing, the solenoid will have to be stopped before the negative acceleration phase starts; before or exactly when the projectile start moving past the point of equilibrium, the solution for this is inspired from source [5], it involves the use of optical sensors in order to disable or cut the current to the solenoid at the right moment, 'moment' referring to the position of the center of the projectile relative to the solenoid, the condition is as follows: The current flow to the solenoid is stopped before or when the center of the projectile meets the center of the coil, this is because at this point the projectile is theoretically at it's maximum possible velocity. Therefore in order to correctly apply this condition, it's determined that the length of the solenoid relative to the length of the projectile should satisfy the condition (4):

$$\frac{L}{2} < x \tag{4}$$

Where L is the length of the solenoid and x is the length of the projectile, this means that at the time when the center of the solenoid meets the center of the projectile, the front end of the projectile is at the exit point of the solenoid or has already gone past it, this allows the projectile to fly past the line of sight of the optical sensors once it exits the solenoid. This condition has to occur as such because the alternative is placing the optical sensor inside of the solenoid, which is possible and more reliable than our current method, however it's difficult and beyond the scope of this experiment.

On the other hand, in order to produce a faster moving projectile, equation (3) can be manipulated. As (V) is directly proportional to (I) and (N), this means increasing either would increase the final velocity of the projectile,however,it's optimal for the potential applications of the device and for safety reasons (danger of high currents), to add a separate solenoid in order to increase the speed of the projectile. The new expected velocity given an ideal system is double the maximum velocity of the projectile after being accelerated to the center of the first solenoid.On the other hand, having one long solenoid to accelerate the projectile isn't ideal for the following reasons:
A) There would be no way to detect when the center of the projectile has reached the center of the solenoid.
B) getting a projectile of such a length that satisfies equation (4) is difficult and expensive.
C) The magnetic field generated will have to be flawless, which is difficult and expensive to achieve.
Reasons A and B are self explanatory, however, reason C would mean that any small irregularity in the alignment of the wire turns will produce an in homogeneous magnetic field, thus (at larger scales) causing the projectile to experience a force in the y or z axis thus increasing friction with the walls, reducing the final velocity.Having factored this and the issue of maintenance (it's cheaper to replace one small electromagnet than a larger one), it's optimal to use many (2x), smaller length solenoids in order to propel the projectile.
This completes the theory behind the mechanism of operation of the solenoid, and how some of the limiting factors resulted in the final design of the Electromagnetic propulsion system. A detailed outline of the construction of the circuit, use of code, testing of the different components is in section 3.3.

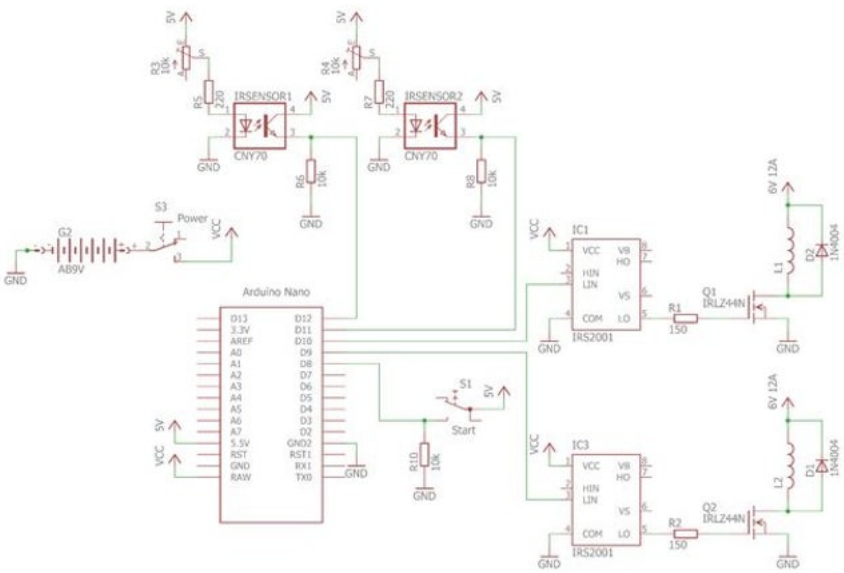
3.3 Practical construction of the device

3.3.1 Schematics

In this section a closer look will be had in regards to the schematics of this experiment, A) circuit schematics, B) Electromagnetic gun physical schematics and how each would produce a testing process in order to ensure a smooth construction of the final designs.

Starting with the circuit shown is figure 1

Figure 1: Showing the circuit schematics for the final design. [5]



This circuit shows the relevant components need to be linked to the Arduino in order for the final device design to function as expected, where (cny70)s[7] are the optical infrared sensors, (in4004)s[3] are the solenoid's diode preventing current back flow, (irs2001)[6] are the drivers, finally the device marked 'Arduino nano' is the Arduino device[1].

The optical sensors consists of two components connected to different other intermediate parts, including resistors, ground and the power supply as displayed in figure 1. The first component is the sensor if the infrared LED, this is a light source, upon the passing of an object that may reflect this wavelength e.g a finger or a projectile, the scattered photons are collected by the second component, which is the light sensitive transistor, which only responds to a narrow wavelength of light, most of which is produced by the on board LED, this allows the optical sensor to be sensitive to passing objects. This sensitivity is directly correlated to the number of photons produced by the LED, the LED's ability to produce (N) photons is dependant on the power passing thought the LED from the 5v power supply, this power can be controlled by using the variable 10k(ohm) variable resistor, this is used to calibrate the sensitivity of the sensor thus allowing to be reliable and protecting it from high voltages when lower sensitivity is preferred.

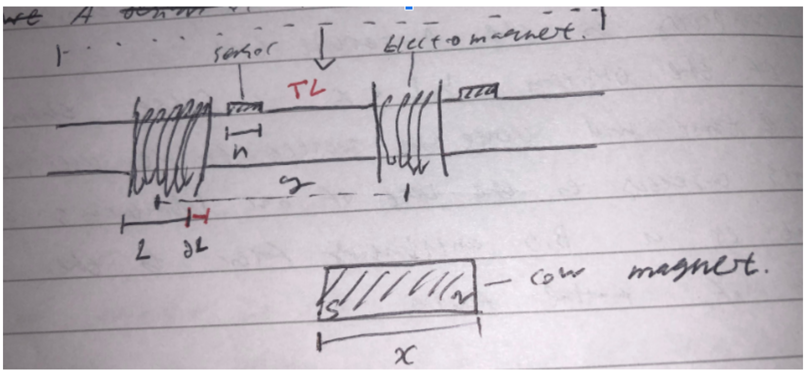
The solenoid electromagnet is connected to the Arduino through a couple of intermediate components, it's also connected to an external high voltage high current power supply. The connection to the driver is ignored due to the lack of necessity. The current and voltage of the power supply needn't be high, as a simple proof of reliability is the main objective, launching the projectile with a high velocity isn't an objective, thus high currents were avoided in order to improve the safety of the procedure.

The driver acts as a high speed switch, thus when the signal is received to trigger the electromagnets on or off, it executed that command quickly by increasing or decreasing the passing voltage very quickly, given that we are using a much lower voltage than what's demonstrated in source [5], our projectile wouldn't be moving as quickly as shown there, thus the time deficit in the activation of the coils is negligible, therefore the driver was considered unnecessary and it was removed.

Arduino. The Arduino nano is an electronic controller, it allows us to define input and output pins, to which components are connected. It then allows us to use a main looping function to check the status of the components through time, allowing us to code responses to changes that occur to the values of the pins, an example of the logic used:
If the button is pressed, turn on the light connected to pin (x).
The final code used in each of the testing phases and the final phase are displayed and explained (through comments) in section 8.

Further more the schematics for the final design of the system is displayed in figure 2

Figure 2: Showing the circuit schematics for the final design.



where the distances/areas (y) is the distance between the two solenoids,(L) is the length of the each solenoid, (dl) is the length uncertainty of (L), (n) is the cross section of the optical sensor, (x) is the length of the cow magnet and (TL) is the total length of pipe system, each having values: (y = 5cm),(L $=3 \pm 0.2$ cm),(dl = 0.2cm),(n= $0.5cm^2$) and (TL = 12.5cm). This schematic is inspired by source [5].
Before the construction and reliability testing of the main circuit and the final design of the system, a testing series was carried out and is detailed in section 3.3.2.

3.3.2 Testing builds

In this section, the testing circuits for individual components along with the testing Arduino code and testing of two methods of deactivating the electromagnets are displayed.

Starting off, the Arduino is connected to the PCB testing board and a simple blinker code (built into the Arduino source [2]) is uploaded to verify it's functionality, the test was a success.Furthermore, only one of the optical sensors was connected as detailed in figure 1, using a slightly modified version of the blinker code (shown in section 8.1), where the optical sensor is defined as an output and the condition coded so that the test LED would light up for one second if the optical sensor is triggered by a passing object, this was a success and the optical sensor is confirmed to be working.

Furthermore, a test solenoid was successfully created and it's basic functionality tested, by simply connecting it to a power supply when a projectile was placed at the entry, we were able to confirm the plausibility of the design, figure 3 shows how the projectile moved thought the single solenoid system.

Figure 3: Showing evolution of the projection when the magnetic field is active

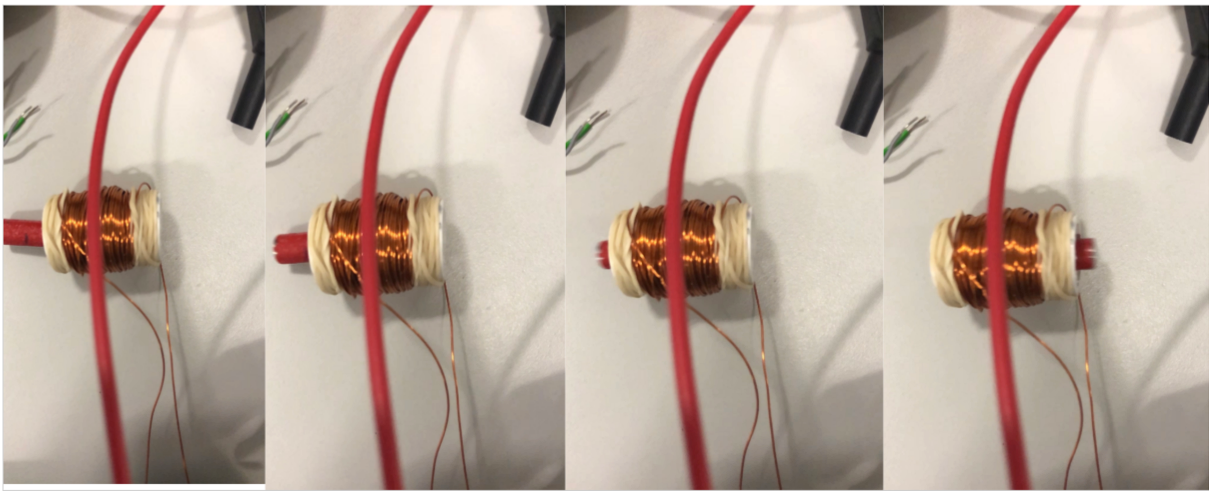
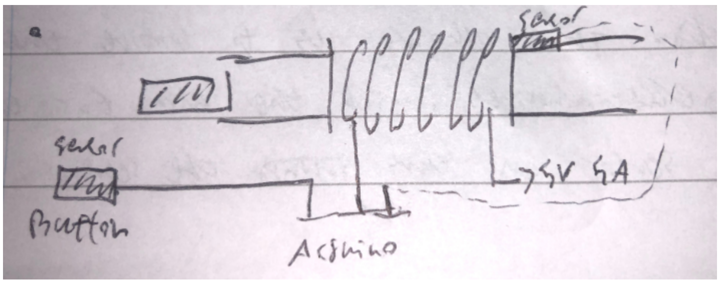


Figure 4: Showing testing system design of one solenoid connected to one optical sensor with one other optical sensor as a button, all of which are connected to the Arduino and the solenoid is connected to an external power supply.



This proves the theory (figure 3) and that's because on the far right image, right before the projectile is springing back into the center of the solenoid, we observe that a small portion of it has passed through the non-entry side, this is due to the condition in equation 4.Furthermore, after the installation of the solenoid electro magnet as detailed in the circuit schematic in figure 1, we are now able to coordinate the sensor and the solenoid and through the Creation of the testing system displayed in figure 4 and the use of the code listed in section 8.2, the aim is to test:
A) the functionality of the electromagnet under the control of the Arduino.
B) the ability of the sensor to detect a passing object that doesn't completely cover it and isn't directly above it and
C) the reliability of the system.

Furthermore, a comparative test was carried out, where the sensor was removed and only the button sensor remained to trigger the activation of the electro magnet, through the use of the timer method coded into the Arduino (in section 8.3), we were able to calibrate the period of activation of the coil directly though Arduino by trial and error to produce the best possible result, a reliability test was then carried out. A few adjustments were made to the one sensor and initial conditions for the one solenoid with the sensor test and the results re-measured.

Figure 5: Showing the final circuit created.

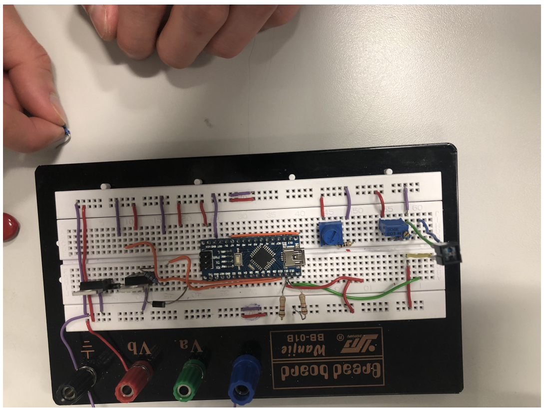
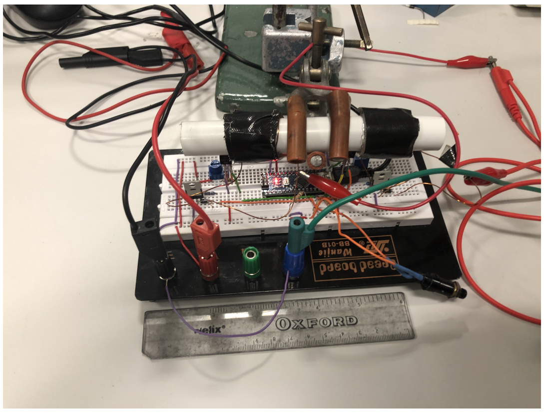


Figure 6: Showing the final system with two solenoids, and two sensors.



Nonetheless the final circuitry and system design were made, as detailed in figures 1 and 2 are shown in figures 5 and 6 respectively, allowing us to proceed with the final testing and the reliability of the system was again tested.
A) using two active sensors to deactivate the coils by using the code in section 8.5.
B) using two hard coded timers to activate the electromagnets, with the code listed in section 8.4.

The results and comparisons are displayed and discussed in sections 4 and 5.

4 Results

The reliability test we used in this experiment is simply a measure of how often the system successfully launches the projectile without having it decelerated and thus dragged back to the center of solenoid.Therefore a 'True' result is when the projectile successfully exits the solenoid regardless of how fast it's moving at the end and a 'False' is when the projectile gets decelerated at the end thus getting stuck in the potential field at the equilibrium point;(middle point).

We start by testing the reliability of the system made from one solenoid and one optical sensor, the results are in table 2. This is displaying a success rate of 40%, the discussion of which is in section 5

Table 2: Reliability testing for the single solenoid with a sensor.

Trial (N)	Result (T/F)
1	FALSE
2	TRUE
3	TRUE
4	FALSE
5	FALSE
6	FALSE
7	TRUE
8	TRUE
9	FALSE
10	FALSE

A reliability test was Re-preformed on the same one solenoid one sensor test with adjusted parameters, the results are displayed in table 3.this is displaying a sucess rate of 60% the reason for which is discussed in section 5.

Table 3: Reliability RETESTING testing for the single solenoid with a sensor.

Trial (N)	Result (T/F)
1	TRUE
2	TRUE
3	TRUE
4	FALSE
5	FALSE
6	TRUE
7	TRUE
8	FALSE
9	TRUE
10	FALSE

A reliability test was preformed on the system using only one solenoid and a timer function coded into Arduino, results are displayed in table 4.This is displaying a success rate of 90%, however the measurement in trial 6 was simply due to incorrect placement of projectile, this is discussed in section 6.

Table 4: Reliability testing for the single solenoid with a the timer method.

Trial (N)	Result (T/F)
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	FALSE
7	TRUE
8	TRUE
9	TRUE
10	TRUE

A reliability test was preformed on the final system using two solenoid and two sensors using the sensor activation method, the results are displayed in table 5.This is displaying a 100% success rate, because the first two measurements were considered invalid as a human error in circuitry was found, this is discussed in section 6.

Table 5: Reliability RETESTING testing for the final device, 2solenoids and 2 sensors

Trial (N)	Result (T/F)
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	TRUE
8	TRUE
9	TRUE
10	TRUE
11	TRUE
12	TRUE

A reliability test was preformed on the final system using two solenoids and two hard coded/calibrated values of the activation periods of each solenoid, the results are displayed in table 6. This is displaying a 100% success rate thus making the system reliable for more sophisticated uses.

Table 6: Reliability testing for the 2 solenoids with a the timer method.

Trial (N)	Result (T/F)
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	TRUE
8	TRUE
9	TRUE
10	TRUE

5 Discussion

For tables 2 and 3, we were testing the reliability of the single solenoid with the single sensor, this system effectively represents one out of the two units in the final device design displayed in figure 2, out results show, for table 2, that the success rate of the solenoid is 40%, which suggests that there is room for improvement and so the circuit was analysed to figure out what could be done to improve upon the result, this is discussed in section 6 and 3.2.Furthermore after the adjustments were made, the results were shown to improve by 20%. This brings the unit system reliability to 60% which we consider to be acceptable for the way this experiment was done, with clear improvements being possible a few of which include:
1) using a different and more reliable method of sensing the position of the projectile, this includes an optical sensor placed inside of the solenoid, thus giving us the ability disconnect the solenoid righ on the time that the projectile reaches the center of the solenoid, therefore allowing for the most optimum efficiency possible. 2) Using more current and/or more turns on the solenoid in order to increase the power output and thus an increased chance of sensor triggering.

Furthermore, for table 4, we used trial and error to calibrate the activation time of the solenoid once the button was pressed then carried out the same reliability test, which obtained a 100% success rate. This shows great promise in a method that requires the same masses to be fired at the same velocity all the times, therefore never requiring sensors to control the solenoid activation. This makes them cheaper to produce and more reliable for real world application purposes if they're ever needed but only for that specific purpose.

Nonetheless, for tables 5 and 6, a two solenoid system was used to test both methods,however the specific focus on optimising the sensor activation method, two main changes were made to the method, firstly, the length of the projectile was increased and the uncertainty in the length of the solenoid (dl) was decreased by using a glue gun as the barrier instead of plastic bands the combination of these two adjustments made it possible for the sensor to always be triggered much earlier before the the projectile got significantly decelerated, increasing the reliability of the system making it perfect (100% success rate).On the other hand, the timer method after calibration remained reliable(100% success rate), this was because the calibration made it so that negative acceleration hardly occurred and thus the projectile was always fired through successfully.

Furthermore comparing the two methods shows that the sensor triggering method is best for automatic control of variable systems e.g varying angles, masses, and etc, while the timer method is more reliable for constant systems or for systems where the projectile is too small to detect but it's velocity and position could be calculated thus allowing us to accelerate it even though we cannot observe it's exact location. However, it seems apparent that the combination of different methods (not only those two) is the best option but the most expensive.In practice we can control the exact velocity that a projectile exits a propulsion system with, for this to be done a light gate is needed to measure the speed of the projectile at each unit of the system (from solenoid end to solenoid end), this allows us to monitor the speed of the projectile at all times, then by the user of advanced code the method used at the next solenoid can be determined, to increase the velocity we can use the optical sensor inside of the solenoid in order to positively accelerate the projectile, this is done by simply activating the solenoid after a given period from when the light gate was broken, but we have to activate it before the projectile reaches the equilibrium point, otherwise we negatively accelerate it,how much we accelerate it by depends on the position of the projectile when the solenoid is switched on, this can be controlled using the timer method, this requires knowledge of the equation of motion of the projectile, thus, friction, air resistance, angle of firing (effects of gravity) will have to be accounted for in order to produce results good enough for real world applications an example of which is Electromagnetic cannons. On the other hand by using the timer method and the light gates, we can carefully calculate the exact time the electromagnet should turn on after the projectile has passed through the entry of the coil in order to decelerate the projectile to the wanted velocity.

6 Error analysis

In this experiment a few sources of error has been the most dominant, since the results obtained by our experimentation are simple T/F results there wasn't much uncertainty calculations needed. The only source of analytical uncertainty was the length of the solenoid, which was present due to the limitation of equipment, as rubber bands were used keep the solenoid from unraveling, an extra length is added to the end of the solenoid that effectively registers the sensor further away from the end of the solenoid (rubber doesn't conduct electricity as it's an insulator), thus this produces an uncertainty in the condition defined in equation 4, which effectively means as explained previously in section 3.2, that the projectile is more likely to be negatively accelerated back to the center of the solenoid as it reach the end of the length of the solenoid, thus failing to activate the sensor and release it self.Nonetheless this error is simply corrected by increasing the length of the solenoid, which in theory should effect the final velocity as a greater mass is inversely proportional to the maximum velocity (equation 3), however, this trade off increases the reliability of the system as increasing the length of the magnet means a favourable value emerges from the condition in equation 4.

On the other hand, it was apparent from the first set of data using the optical sensors that the system was unreliable, at only 40% success rate from table 2, the inputs to the sensor were modified,the 220(ohm) resistor connected to the LED was changed to only 110(ohm); we simply increased the current to the light source to allow it to produce more photons, thus increasing the sensitivity of the sensor. As well as bringing the sensor very close to the exit of the solenoid (vertical adjustment),this simple adjustment made the sensor respond more accurately to a projectile passing over it, increasing the success rate from 40% to 60% a 20% increase.

Furthermore, other errors are mainly involved with human miscalculations (brain failures more accurately).
A) In correct wiring of some components resulted in some anomalies in the results which were simply repeated.

B) projectile misplacement horizontally, this effects the final velocity of the projectile before it experiences a negative force, this is because when the center of the projectile spends more time/space being accelerated it will reach the point of equilibrium with a greater velocity, due to the limitation of the experiment we cannot avoid having the projectile negatively accelerated..ever, therefore the successful triggering of the sensors depends on the momentum of the projectile before it gets decelerated, which depends on the velocity and that depends on how long/far the projectile spends accelerating thus the initial horizontal position of the projectile. Once this was notice it was corrected for by making a mark on the projectile so the placement is always an optimal length, found through a short trial and error experimentation.

7 Conclusion

To conclude, the applications of this technology can be useful for real world heavy weaponry/ space weaponry and the Mass driver which is an Electromagnetic propulsion system scaled up massively, both in size and power, with the aim to be used as a catapult, in order to launch objects from the surface of the moon to moon orbit, thus saving fuel cost..given of course the fact that solar energy is free thus the device will pay for it self over time.

Nonetheless more experimentation and development is required in order to improve the quality and efficiency of the system advancements in material science,energy storage and such are required to make this technology worth while, however we can still continue experimenting, for example, how would the system be used in viscous medium, or humid environments, this will teach us the real limitations and could cause innovation in the design of system with aim of improving it.

8 Arduino Code

This section includes all the code used to test the equipment as well as the final code used in the final device.

8.1 Sensor test

```
#defining the sensors
const int sensor1 = 12;
#defining the built in led on the arduino
const int test = 13;

#settup of the connected devices
void setup() {
    #defining the inputs
    pinMode(sensor1, INPUT);
    #defining the test led
    pinMode(test,OUTPUT);
}

#starting the looping function
void loop() {

    #if the optical sensor is triggered, flash the led light for one second
  if (digitalRead(sensor1) == HIGH) {
      #turn on the on board led
      digitalWrite(test, HIGH);
      #pause for a second
      delay(1000);
      #turn off the onboard led
      digitalWrite(test, LOW);
  }
```

```
}
```

8.2 Electromagnet test

```
#defining the sensors
const int sensor1 = 12;
const int sensor2 = 11;
#defining the divers
const int driver1 = 10;
#defining the built in led on the arduino
const int test = 13;
#defining the timer and the condition peramiter
long timer;
int condition;
#settup of the connected devices
void setup() {
    #defining the inputs
    pinMode(sensor1, INPUT);
    pinMode(sensor2, INPUT);
    #defining the outputs
    pinMode(driver1, OUTPUT);
    #defining the test led
    pinMode(test,OUTPUT);
    #defining the condition peramiter
    condition = 0
    timer = 0;
}


#starting the looping function
void loop() {

    #if the first optical sensor is triggered, activate the electromagnet test sequence
  if (digitalRead(sensor1) == HIGH) {
      #set activation conditoin
      condition = 1;
  }
    #if the first sensor is triggered carry on with the following sequence
 if (condition == 1){
     #turn on the electromagnet
     digitalWrite(driver1, HIGH);
     #turn on the on board led
     digitalWrite(test, HIGH);
     #check the status of the second sensor throughout the next 250 milliseconds
     while(timer < 2500000){
         if(digitalRead(sensor2) == HIGH){
             #turn off the electromagnet
             digitalWrite(driver1, LOW);
             #turn OFF the on board led
             digitalWrite(test, LOW);
             # exit and reset
             timer = 2500000;
             condition = 0;
         }

        #incrument throught time
         delayMicroseconds(1);
```

```
        timer = timer + 1;
      }
      #if 250 millisecond pass and the sensor is'nt triggered, reset the system
      #turn off the electromagnet
      digitalWrite(driver1, LOW);
      #turn OFF the on board led
      digitalWrite(test, LOW);
      #reset
      timer = 0;
      condition = 0;
 }

}
```

8.3 Timer method code

```
#defining the sensors
const int sensor1 = 12;
#defining the divers
const int driver1 = 10;
#defining the timer and the condition peramiter
long timer;
int condition;
#settup of the connected devices
void setup() {
    #defining the inputs
    pinMode(sensor1, INPUT);
    #defining the outputs
    pinMode(driver1, OUTPUT);
    #defining the condition peramiter
    condition = 0
}


#starting the looping function
void loop() {

    #if the first optical sensor is triggered, activate the electromagnet test sequence
  if (digitalRead(sensor1) == HIGH) {
      #set activation conditoin
      condition = 1;
  }
    #if the first sensor is triggered carry on with the following sequence
 if (condition == 1){
     #turn on the electromagnet
     digitalWrite(driver1, HIGH);
     #delay for a time (t), THE CALIBRATED PERIOD
     delay(250);
     #turn off the coil and reset the condition
     digitalWrite(driver1, LOW);
     condition = 0;
     }


}
```

8.4 Timer method for two coils

```
#defining the sensors
const int sensor1 = 12;
#defining the divers
const int driver1 = 10;
const int driver2 = 9;
#defining the condition peramiter
int condition;
#settup of the connected devices
void setup() {
    #defining the inputs
    pinMode(sensor1, INPUT);
    #defining the outputs
    pinMode(driver1, OUTPUT);
    pinMode(driver2, OUTPUT);
    #defining the condition peramiter
    condition = 0
}


#starting the looping function
void loop() {

    #if the first optical sensor (temporarly used as a button) is triggered, activate the electromagn
  if (digitalRead(sensor1) == HIGH) {
      #set activation conditoin
      condition = 1;
  }
    #if the first sensor is triggered carry on with the following sequence
 if (condition == 1){
     #turn on the first electromagnet
     digitalWrite(driver1, HIGH);
     #delay for a time (t), THE FIRST CALIBRATED PERIOD
     delay(150);
     #turn off the coil and advance the condition
     digitalWrite(driver1, LOW);
     condition = 2;
     }

    #if the first squence is carried out, continue with the second sequence
    if (condition == 2){
         #turn on the second electromagnet electromagnet immeadiately
         digitalWrite(driver2, HIGH);
         #delay for a time (t), THE FIRST CALIBRATED PERIOD
         delay(350);
         #turn off the coil and reset the condition
         digitalWrite(driver2, LOW);
         condition = 2;

     }


}
```

8.5 Final complete test, two Electromagnets and two coils

```
#defining the sensors
const int sensor1 = 12;
```

```
const int sensor2 = 11;
#defining the divers
const int driver1 = 10;
const int driver2 = 9;
#defining the start button
const int startbutton = 8;
#defining a timer variable and a condition variable
long timer;
int condition;
#settup of the connected devices
void setup() {
    #defining the inputs
    pinMode(sensor1, INPUT);
    pinMode(sensor2, INPUT);
    #defining the outputs
    pinMode(driver1, OUTPUT);
    pinMode(driver2, OUTPUT);
    #defining the human controlled input
    pinMode(startbutton, INPUT);
    #setting the timer and the condition variables to the appropriate values
    timer = 0;
    condition = 0;
}
#starting the looping function
void loop() {
    #initial condition/action, in the case that the button is pressed before an initial condtion was
    #..->preventing unintional evaluation of the code
  if (condition == 0 && digitalRead(startbutton) == HIGH) {
      #if the button is pressed the first electro magnet is activated
      digitalWrite(driver1, HIGH);
      #the condition variable is changed to trigger a second event
      condition = 1;
  }
#the second action, runs directly after the first even ends
  if (condition == 1) {
        #the timer variable is reset
        timer = 0;
    #looing for a period of 50 milliseconds
    while (timer < 50000) {
    #if during this period, the first sensor is triggered
      if (digitalRead(sensor1) == HIGH) {
            #disable the fist electromagnet
            digitalWrite(driver1, LOW);
            #activte the second electromagnet
            digitalWrite(driver2, HIGH);
            #exit out of the loop by setting the timer function to a value that preforms a soft breal
            timer = 50000;
            #prepair for a second condition
            condition = 2;
      }
        #if the sensor isn't triggered keep looping by adding a delay and incumenting the value of t
        delayMicroseconds(1);
        timer = timer + 1;
    }
  }
#a double check procedure to make sure the second condition is moved past
    # this is a seafty net and a rest funciton
  if (condition == 1) {
```

```
        #set the first electro magnet off
        digitalWrite(driver1, LOW);
        #redefine the initial conditions, effectively resetting the syetem
        condition = 0;
        timer = 0;
  }
#the third action
  if (condition == 2) {
        #reset the timer variable
        timer = 0;
        #loop through 50 milliseconds
    while (timer < 50000) {
        #if, during this time the second sensor is activated
      if (digitalRead(sensor2) == HIGH) {
            #deactivate the second electromagnet
            digitalWrite(driver2, LOW);
            #exit out of the loop and reset the system
            timer = 50000;
            #reset the system to the iniaital condition
            condition = 0;
      }
      #if the sensor isn't triggered keep looping by adding a delay and incumenting the value of time
      delayMicroseconds(1);
      timer = timer + 1;
    }
  }
#a double check procedure to make sure the third condition is moved past
    #this is a seaftey net and a reset function
  if (condition == 2) {
        #set the second electromagnet off
        digitalWrite(driver2, LOW);
        #reset the initial codition and reset time
        condition = 0;
        timer = 0;
  }
}
```

References

[1] Arduino. "arduino$_n$*ano*". In: (x). URL: http://www.phablabs.eu/sites/default/files/datasheet%5C%20Arduino%5C%20Nano.pdf.

[2] Arduino. "Blinker". In: (x). URL: https://www.arduino.cc/en/tutorial/blink.

[3] diodes. "ir4004". In: (x). URL: https://cdn.sparkfun.com/assets/learn_tutorials/4/2/3/1N4004.pdf.

[4] ELECTROBOOM. "Just Try and Make Your Own Coil-Gun (Accelerator)". In: (2013). URL: http://www.youtube.com/watch?v=mdZo_keUoEs&t=2m0s.

[5] Instructables.com. "Make your own coil gun". In: (2016). URL: https://www.instructables.com/id/Make-Your-Own-Coilgun/.

[6] ior. "Rectifier". In: (x). URL: https://www.infineon.com/dgdl/Infineon-IRS2001-DataSheet-v01_00-EN.pdf?fileId=5546d462533600a401535675a760277e.

[7] vishay. "cny70". In: (x). URL: https://www.vishay.com/docs/83751/cny70.pdf.