Automated Metal detection robotics

Design documentation for ece 445

Mar 1st 2022

Partners:
Jack Li and Sumukh Kenkere Vasudeva Murthy
(shunjie2 and sumukhk2)
TA- Akshatkumar Sanatbhai Sanghvi

Group 57

1. Introduction

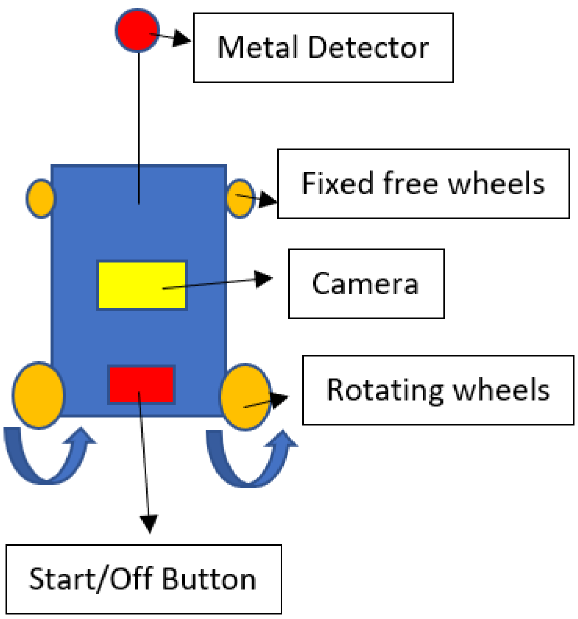
1.1. Problem and solution

Our team will build an automated metal detection robot for people in need with help of searching for lost objects efficiently and for those working in industries. Potential customers may have a problem spending too much time searching for lost objects, and safety issues may arise due to some special cases. Therefore, our product can solve this problem by creating a robot which does not need manual control, so that customers can be rest assured and let the robot do all the work for us.

To be more specific, this problem is closely associated with the time and safety involved in lost objects investigation. Currently, to solve this issue, people have been developing robotics that can move around by manual control, essentially through data transmission between the robot itself with other interfaces like smartphones[1]. Such design can alleviate the issue associated with safety since there is no actual human present physically at the investigation site. However, the accuracy is limited by the method used to transmit the data along with the camera's resolution constraint, and it still requires human control which may cause issues when the control cannot be ensured to be precise enough. There have also been attempts to create automated robots in the past [2], however, the algorithms mostly rely on the GPS system, which still needs human monitoring in order to ensure the progress. Therefore, our design endeavors to solve this issue by adding automation into the robotics system without human engagement until the suspected object is found, which aims to reduce human forces (and therefore, time) involved in the investigation process and enhance safety.

1.2. Visual aid

To help illustrate how the robot will be used, a simple figure can be used to demonstrate some of the properties of usage:



Where we aim for a simple control system by humans (start/off button). Once a suspected object is found and compared with the database, an alarm will be triggered which signifies the point human forces can engage in.

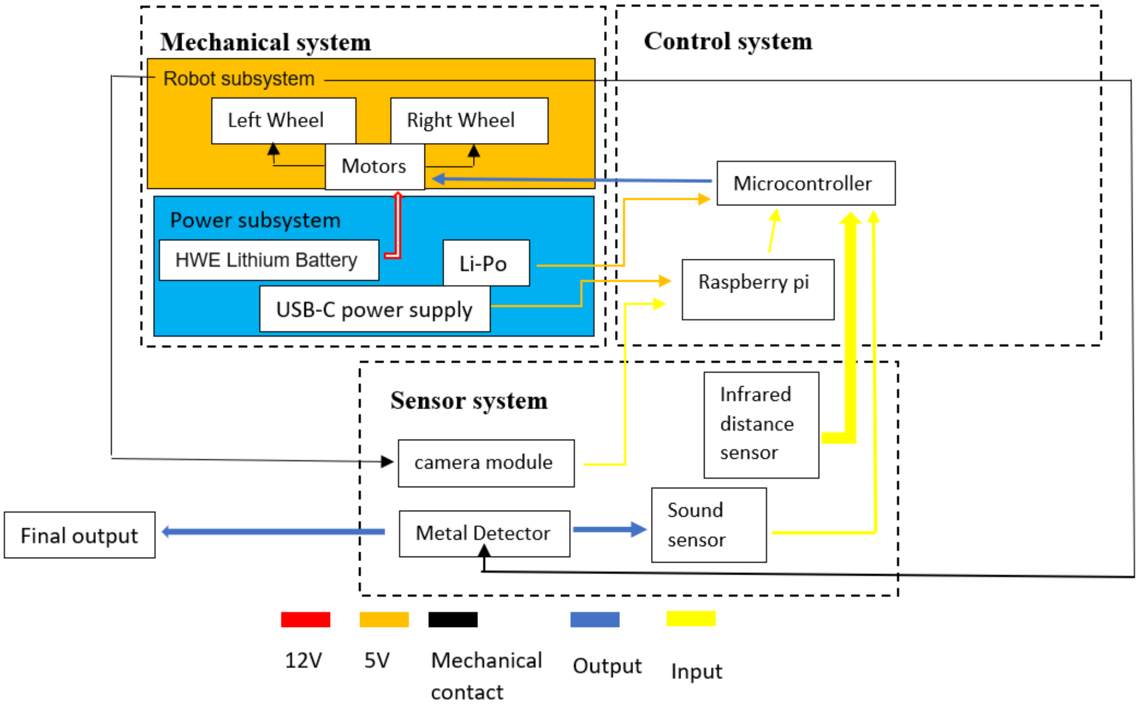
1.3. High level requirements

To solve the problem faced by the potential customers and enhance functionality, several high level requirements are expected to be met in order to make this project better:

- The robot is expected to search within a 5*5 meter room within 10 minutes to find potential targets (efficiency).
- The robot should be able to navigate through different obstacles without crashing into them (safety).
- The finally found object should at least resemble similarities to the target object (for example, a lost key is found to compensate for the target key) (accuracy).

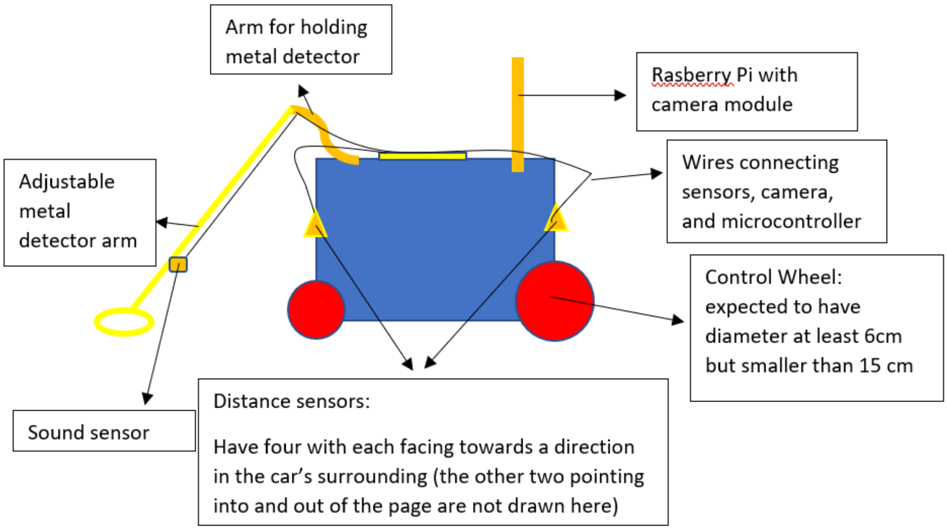
2. Design details

2.1. Block diagram



The design is divided into three systems: mechanical system which is responsible for power and robot car itself, sensor system which receives external information in the form of sound, images, and infrared light (distance signals), and lastly the control system which uses a microcontroller with the help of Raspberry pi to convert the signals detected from sensor system to the control signals sent to mechanical system.

2.2. Physical design diagram



This physical design diagram shows the layout of the potential sensors' position and some possible dimensions (for example the wheels). The metal detector arm has adjustable distance, therefore could be adjusted to be the right length once the robot car can be built from the machine shop. We have two pairs of distance sensors on the four sides of the car so that the distance information can be obtained without assuming the dimensions from the camera. Wires are connected to the PCB where mostly we have a microcontroller as the center of the signals coming from all the sensors (including the camera).

2.3. Subsystem description

Mechanical system—Power subsystem

The power subsystem is the energy source to all of our other systems and subsystems. Typically this includes power to the motor which can control the movement of the robot by precise amount, power to the circuit boards including Raspberry pi, microcontroller, as well as different sensors. We planned on using three different types of energy sources for the different systems without the use of a voltage regulator as this will be a more compatible design for the overall system.

We have a 12 V Lithium battery with a large energy reservation (6 Ah) to make sure that the robotics can be operated with a relatively longer duration of time. The Raspberry pi will be powered through an external USB C power source which can be bought economically from stores. We also need an extra lithium polymer battery (5V) to power the microcontroller, where one output pin of the microcontroller has 3.3 V power output, thus, providing power to the rest of the sensors integrated into the whole robotics system.

R&V requirements:

Requirements	**Verification**
Provide Ideal voltage generated by a 12 V 6A battery to power our device within a 5% accuracy.	Connect Battery to Voltmeter initially and verify that the voltage range that is supplied does not exceed 12V +/- 5% range. Supply 12 V to a basic short circuit and verify that the current supplied does not fluctuate.
Verify that the battery is able to power the entire system it's intended to power .	Connect all components of the subsystem the battery is connected to and run a basic image processing algorithm on RaspberryPi using fixed inputs from Pico and verify that the Pi is able to operate optimally

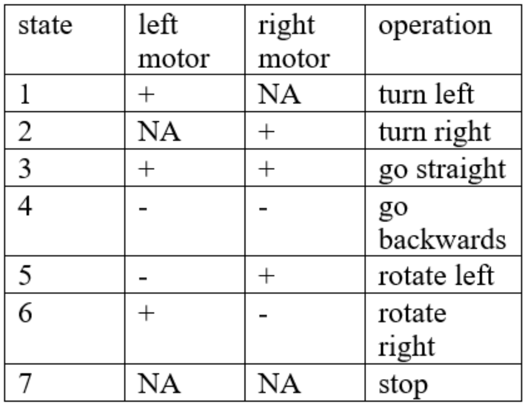
Mechanical system—Robot Subsystem:

The robotic subsystem utilizes the fact that the rotation of a metal detector can be achieved by using the motion of the car, thus, we can easily program the whole robotics subsystem by using two motors. Each of the motors have control signals coming from the control system described in the following sections, where the signals can indicate information about which motor should turn on at a given specific time in which direction. The motors are connected to the wheels and thus, we can control the movements of the robot in various directions. The metal detector also has a mechanical contact with the robot itself, as has been shown in the physical design diagram, where a camera is also mounted on top of the robot car to provide visual feedback.

Requirements	**Verification**
The (motor+wheel) must be able to support the weight of the entire robot which would weigh approximately 3 pounds with all the components+the body.	Verified by mounting weights on a board and adding wheels to see if they are able to support the entire weight without fracturing
The motors must be able to respond upon instruction from the Microcontroller with a minimal delay.	Verified by connecting the motor to the Raspberry Pi and sending arbitrary inputs to ensure the delay is within a minimal threshold of operation.
The wheels must be able to rotate to assist in maneuvering operations and turn as required	Verified after the model is completed by connecting the motors to the circuit and passing inputs to direct operations like turn left by a certain angle, reverse, rotate etc. and ensure that the motors are able to respond accurately in a timely manner.

Control system—Microcontroller (Raspberry Pico)

Our control system relies on two processing units, a Raspberry pi which is described below and a microcontroller. Typically the microcontroller is the heart to two of the high level requirements, which is key to the final efficiency (requirement 1) as well as safety of operation (requirement 2). The microcontroller receives inputs from all sensors, including the distance sensor, processed data from Raspberry pi, and sound sensor. Each of these signals received will contribute to the control signals output to the motors, and thus, realizing the control for automation. More specifically, the processed signal from Raspberry pi will contain information about the detected object's shape information, which is key to requirement 2, the accuracy; the distance sensor provides signals for the information in the robot's surroundings (four sensors covering the whole 360 degrees in x-y plane), thus, ensuring that the robot will not crash into places accidentally (safety requirement). We can program the microcontroller to provide signals divided into 7 states:



state	left motor	right motor	operation
1	+	NA	turn left
2	NA	+	turn right
3	+	+	go straight
4	-	-	go backwards
5	-	+	rotate left
6	+	-	rotate right
7	NA	NA	stop

Each of the operations will be accompanied with a specific time, typically with the help of the distance sensors which serve to change the operation based on specific conditions locally. The time limit for each operation is also designed to form a state machine, so that the robot can perform operation 3 (go straight) first (this is the marching state we call), and then stop (7), rotate left (operation 5, this is the detection state) (if there is no obstacle present, rotate right(6) is not necessary to be performed until the point the robot has rotated by a full circle) and repeat this process. By doing this, the robot will gradually search the entire area thoroughly to find the object. When obstacles present, rotate left and right will switch to each other in the detection state, to ensure safety high level requirements (that the robot will not crash into some of the obstacles); in marching state, turn left is prioritized unless the obstacle is also presented in the left direction, thus, will switch to turn right operation. There are also algorithms we will use to ensure that we will not constantly repeat the detection that has been done, so the operations related to turn left/turn right/go backwards will have more room for improvements.

Requirements	**Verification**
The microcontroller is required to direct the robot based on signals from the various sensors.	Verify the functioning of the microcontroller using an arbitrary program to test it's functioning by synthesizing our code and flashing the microcontroller while observing the test bench outputs.
The microcontroller must be able to function on input power of 3.3V 90 mA current.	Verified by providing the required power and running a sample program on the microcontroller to receive inputs from an IR sensor and check if Pico is able to provide the required response.

Sensor system—Image Processing Subsystem (Raspberry Pi+camera)

This subsystem processes data coming from the camera in the sensor system. Basic purpose is to satisfy the high level requirement 3 (accuracy). When the metal detector receives metallic signals, the sound sensor will be triggered, which will prompt the microcontroller to accept signals from the camera. The camera then provides signals which can be processed by Raspberry pi by pattern recognition. The image from the camera is stored in a suitable format- either a png or a HVEC on board the hard drive of the raspberry pi. This image is then accessed by our tensorflow program on board the PI which will then perform an identification on the contents of the image using classifiers and then determine whether the object is the same we were looking for. In case this requirement is met, the robot stops and indicates that the object has been detected. If not, a control signal is sent from the Pi to the Pico which directs the Pico to continue searching for the object until it's found.

Requirements	**Verification**
The camera must be able to capture an image of suitable size and format	Plug the camera into the raspberry Pi and access local storage on board the Pi to verify a clear image of appropriate size was generated
Raspberry Pi must be able to run a local python script using tensorFlow	Verify by installing a python IDE on board the raspberry Pi and run a basic TensorFlow script to identify images of Cats and Keys.
Raspberry Pi must be able to integrate with the microcontroller and respond to control and interrupt signals.	Verify by modifying the test program for the Pico by sending an interrupt signal from the microcontroller while using the IR sensor and verify that the interrupt signal from the Pico initiated a capture response from the Pi

Sensor subsystem:

For our sensor Subsystem, we're using a metal detector, a couple of IR distance sensors and a sonar sound sensor. The metal detector will emit a beep when a metal is detected and this is captured by the sonar sensor. The sonar sensor has 4 pins- an Analog pin which we will not use, a digital pin which will give us HI/Low outputs when a beep is detected, a ground pin which will be connected to the ground on the pcb and a3.3V input pin to power the device.

The distance sensor will serve to detect obstacles and help orient the robot by giving us accurate distance reading which will be used to specify interrupt signals to the motors using the microcontroller whenever we see an available path to explore, a wall and other obstacles.

Requirements	**Verification**
Metal detector should be able to detect a metallic object within a distance of 50 cm, a radius of 15 cm with a 5% accuracy at least 90% of the time.	Verified by using the metal detector on objects made from different metals like brass, copper, aluminum while simultaneously varying the distance and statistically recording the results to verify our requirements are met.
The Sonar sensor should be able to detect beeps from the metal detector 99% of the time and accurately send signals to our microcontroller(Pico)	Verified by connecting the sonar sensor to the microcontroller or an oscilloscope and observing the signal received from the sonar sensor when exposed to a noise.

2.4. Tolerance analysis

We have debated between ourselves and observed three challenges that we would like to highlight that could be critical to the functioning of our robot.

a) One important fact is that we plan to focus on ensuring a working product is checking if our voltage regulators to step down current from 5A(battery current) can successfully be stepped down to [500mA(Pi)+ 500mA(Pico)+15mA(IR)+15mA(Sonar)+ 16mA(Pi_camera)]= 1.046A. This would be accomplished using a transformer or a voltage controlled current source(VCCS) integrated into our PCB which would serve to regulate the current flowing into different sensors and ensure that there is no excess current flowing which could destroy the components.

b) Another factor that is critical to our application is to ensure that the robot is able to run for at least an hour on the battery supply. To achieve this, we've to ensure that the power supplied to all the components in an hour is able to function at maximum efficiency under high stress for the entire duration. Averaging 12 V and 1A every second for all the components, we have identified that the battery can clearly supply the power demand from the devices. The worst case scenario is that practically, we might end up losing power to the PCB due to heating and other common losses. In case the battery proves insufficient, we would have to use another lighter battery to power the sensors separately and power just the control system with the battery.

c) Lastly, as the camera is integral to the identification of objects and the Raspberry Pi is the central Processing Unit, we would have to apply non-linear Classification mathematics to evaluate how the image processing occurs on chip and for this, there’s a simple calculation involved. The images are stored as a numpy array on the Pi of size

[image Height]x[image Width]x7 where 7 is the total number of RGB bands available.

The camera captures images with a resolution of 2592 x 1944 which is 5038848 pixels.

In our operations, each pixel is essentially a value accounting for one byte. Due to the autonomous nature of our robot, we will have to store each image on site which will cost us about 5.04 MB of storage space per image. But, Tensorflow runs based on a classifier that looks at pre-existing images to identify current objects. Thus, we’re looking at a requirement of about 200 images for reference for each object. But, we have the option to use images of lower resolution which can be found on MIT’s image database. These images are usually in the order of 2-3 MB so we would require storage of at least 500 MB on board for the images. In addition, to handle Tensorflow and aid processing, we would require 2 GB of ram.

Calculating the runtimes of tensorflow is hard because it’s difficult to identify the processing times of different classifiers.

But, runtimes for a simple image identification algorithm on tensorflow using a pre-trained classifier depending on a 2 GB ram would require a minute approximately.

Raspberry Pi processing speed- 1.4 GHz

Size of image- 5MB, size of all images in database - 500 MB

Camera refresh rate- 60 Hz(60 frames per second) so, the delay in transmission is about 0.01 S.

Time taken to create EigenVectors from images per session

500 MB/1.4 Ghz=0.35 S

Time taken for classifier to match each image= 500 MB* 200 images/1.4 Ghz = 71.4 S

This calculation is an approximate calculation because there are times when the classifier finds similar data points within a few images and sometimes, it has to run through all the images.

So, the net delay per instance where the robot believes the desired object lies is one minute.

In addition, if there are multiple metallic objects, we believe that we may require more time than the expected 5 minutes to evaluate an area of 5x5 m room.

3. Cost and Schedule

3.1. Cost of the project (including the salary of workers needed to complete the projects together with the cost of parts)

· **Labor:**

Based on the average salary for undergraduate and graduate researchers, it is reasonable to assume an income of 15 dollars per hour. Thus, the cost of labor is:
15*2*15*10 = 4500 dollars in total.

· **Parts :**

Raspberry Pi	50$(approximate based on price fluctuation)
Raspberry Pico	20$ (4 pieces X 5 Dollars per piece)
Lithium Battery	38$
IR sensors	10$
Motors and wheels	40$
Camera	15$
Sonar Sensor	15$

· **Grand Total (**cost of labor + parts**):** ***<u>4688</u>*** *$ {projected cost}*

3.2. Timeline to complete the project
(still under development due to one partner leaving the course)

4. Ethics and Safety

To ensure safety to the customer, the engineering team and the robot, we are choosing to refer to section 1.1 of IEEE standards regarding battery usage which states "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment" [7]. As our product includes a lithium ion and or a lead acid battery, we would like to alert customers that there may be accidental malfunctions due to leakage of battery acid, short circuits in case the robot is subject to mechanical damage or in rare cases, combustion of the battery due to random errors. These issues must be handled with care by either calling the Fire Brigade or Contacting a specialist in chemical spills to avoid endangering lives as shocks can cause mortal danger and ingesting any chemical from the battery could lead to heavy metal poisoning, cancer and other serious illnesses.

In addition, we would also like to draw attention to consumer safety guidelines issued by the Government of United States(Lee, 1999) [8] which addresses the endurance testing, temperature and component requirements which we would like to follow as much as possible with the exception that we are able to find newer regulations that may be more suitable to our product.

Any redressal or claim will be subject to United States regulations prescribed in the consumer rights section of the website.

Citations

1. Bin Parsusah, Mujiarto & Sambas, Aceng & Haerudin, Irpan. (2021). Design of Arduino-Based Metal Detector Robot. Solid State Technology. 63. 12401-12411.

2. H. M. Ishak, H. A. Kadir, Lim Chain Fat and Mohd Helmy Abd Wahab, "Autonomous metal detector robot with monitoring system," *2008 International Symposium on Information Technology*, 2008, pp. 1-7, doi: 10.1109/ITSIM.2008.4631876.

3. T-Kuha, deep-image-processing [computer software], https://github.com/t-kuha/deep-image-processing

4.Arun Michael Dsouza, Tensorflow-image-detection [computer software], https://github.com/ArunMichaelDsouza/tensorflow-image-detection

5. R, S. K. (2021, August 19). *Principal component analysis: In-depth understanding through image visualization*. Medium. Retrieved February 27, 2022, from https://towardsdatascience.com/principal-component-analysis-in-depth-understanding-through-image-visualization-892922f77d9f

6. MIT. (n.d.). *Database of common household objects*. ADE20K dataset. Retrieved February 28, 2022, from https://groups.csail.mit.edu/vision/datasets/ADE20K/

7. IEEE. (n.d.). *IEEE code of Ethics*. IEEE. Retrieved February 25, 2022, from https://www.ieee.org/about/corporate/governance/p7-8.html

8. Lee, D. A. (1999, July 12). *CPSC staff comments on F963 toy safety regarding battery ...* Consumer Product Safety Commission. Retrieved February 27, 2022, from https://www.cpsc.gov/s3fs-public/pdfs/blk_media_batteryJuly99a.pdf