Mobile Localization Using Radio Frequency Identification

Team 4 - te-AM Radio
Jake Junsay, George Cambel, Haniel Youlesivanson, Katlyn Chiu

Goals and Objectives

- To create a device that has the ability to locate stationary objects with passive RFID tags that require no power in an environment that has no signal or lack of

- Design an algorithm within Matlab that can localize the tag accurately and efficiently through the use of varying tags and antennas

- Ensure that the algorithm can locate through objects such as walls and other obstacles

Speaker: George Cambel

Problem Definition

- To establish a new way of assisting first responders in search and rescue situations
- Many times in these situations there is a need localize certain targets such as first responders themselves
- In those situations there often is no way to visually locate the the target
- Through the use of passive RFID tags we can locate first responders who are stuck or unable to move and in order to rescue them

Speaker: George Cambel

Design Attributes and Requirements

Attribute	O	C	F	M	Importance (1-5)
Should be able to locate the tagged object	X				5
Robot not in range of sensing tagged object initially		X			1
Uses passive RFID tags	X		X		5
Goes through a sweeping motion to pinpoint tagged location	X			X	4
Stops once target is reached	X		X		4
Realistic simulation parameters		X			3

Speaker: George Cambel

RFID Robot - Design Requirements

In our design of the RFID Robot, we considered our design in three parts:

1. The Robot
   - Robot design currently not a goal for project, but will most likely follow previous concepts.
2. The RFID System
   - Our main physical focus for this project, consisting of antennas, tags, and readers.
3. The Algorithm
   - Where our control aspect of the project is focused, particularly on the localization of tags and movement towards them.

In each part, our points for design were made to most effectively complete our goals for tasks of search and rescue

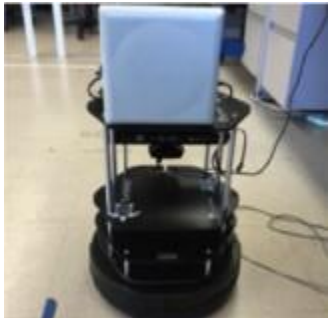


Image 1. Last year's RFID Robot

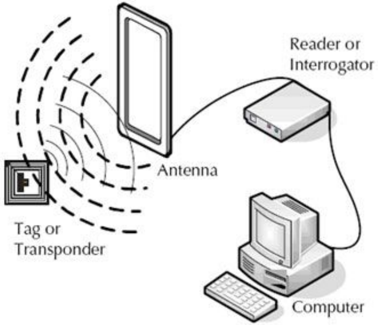


Image 2. Diagram of RFID System

Speaker: George Cambel

Our Key Design Points

So far into the project, our design points have mainly been centered around our RFID system and how it would be used in our algorithm. These design points are labeled a), b), c), d) and e) in the diagram to the right.

Given the complex nature of these components, along with the ease at which they can be purchased, many are of our design choices are set around components that could be bought and added to our system.

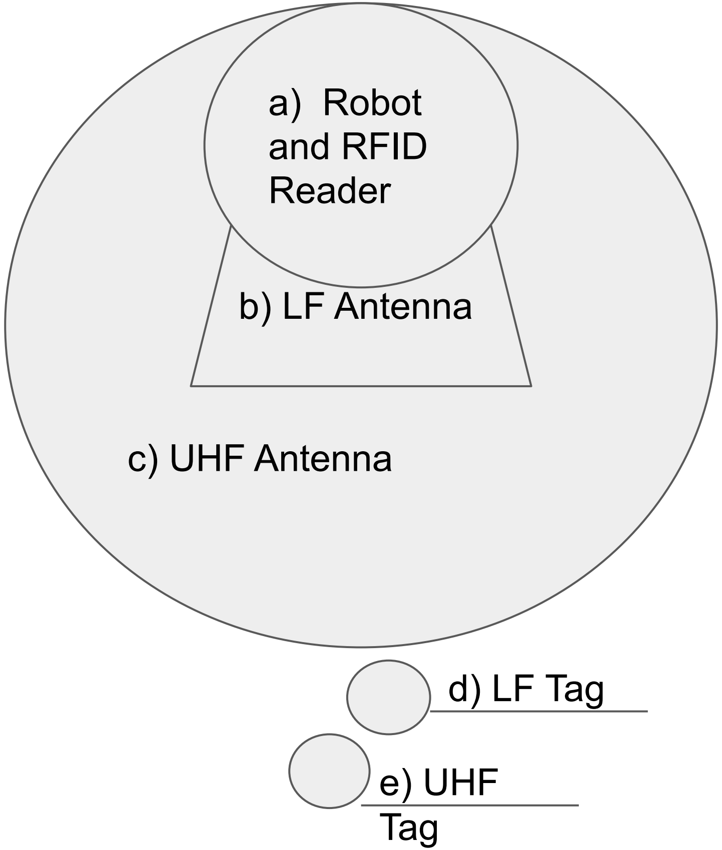


Image 3. Diagram of our robot design. Large circles represent antenna read radii

Speaker: Jake Junsay

Our Key Design Points (cont.)

Point a) Robot and RFID Reader

- Robot covered in previous slide
- RFID Reader
    - Most expensive component (~$200-$500)
    - Would allow for easy data acquisition between the robot and both antennas.
    - “Plug-and-play” aspects that allows ease of compatibility with whatever components we decided to use the RFID system with.

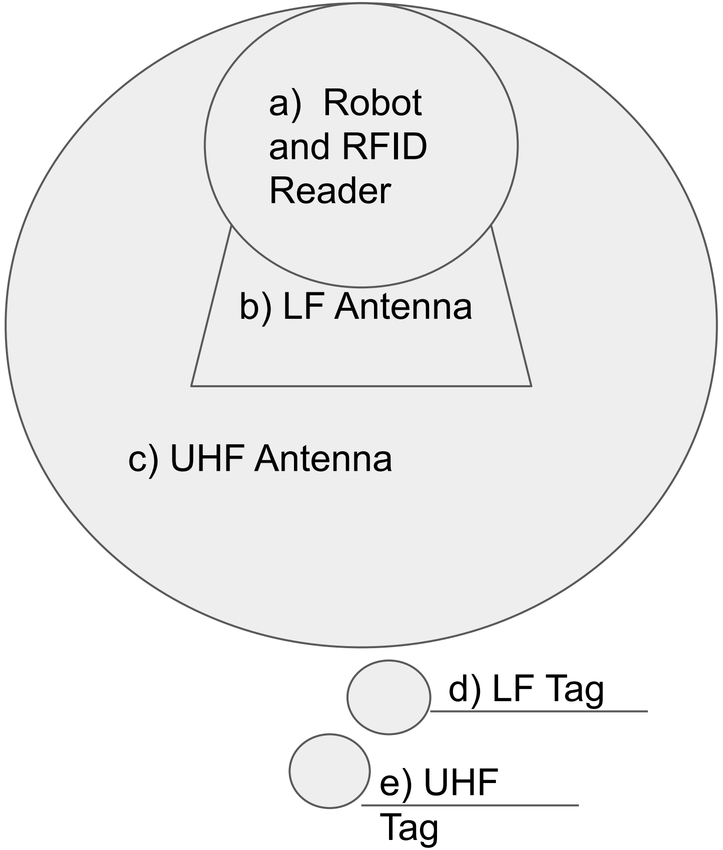


Image 3. Diagram of our robot design. Large circles represent antenna read radii

Speaker: Jake Junsay

Our Key Design Points (cont.)

Point b) LF Antenna

- Cheaper component ($20).
- High gain and circularly polarized to be able to read signals in any plane in range, with high front-to-back ratio to ensure read strength in front of antenna.
- Read range of 5-10 cm.

- In researching about previous years' work on the Mobile RFID project, we read that teams were having issues with stopping the robot once the target had been reached.
- We incorporated a short range antenna that could essentially fix this issue, as the short read range would ensure approach to target and stop once signal has been read.

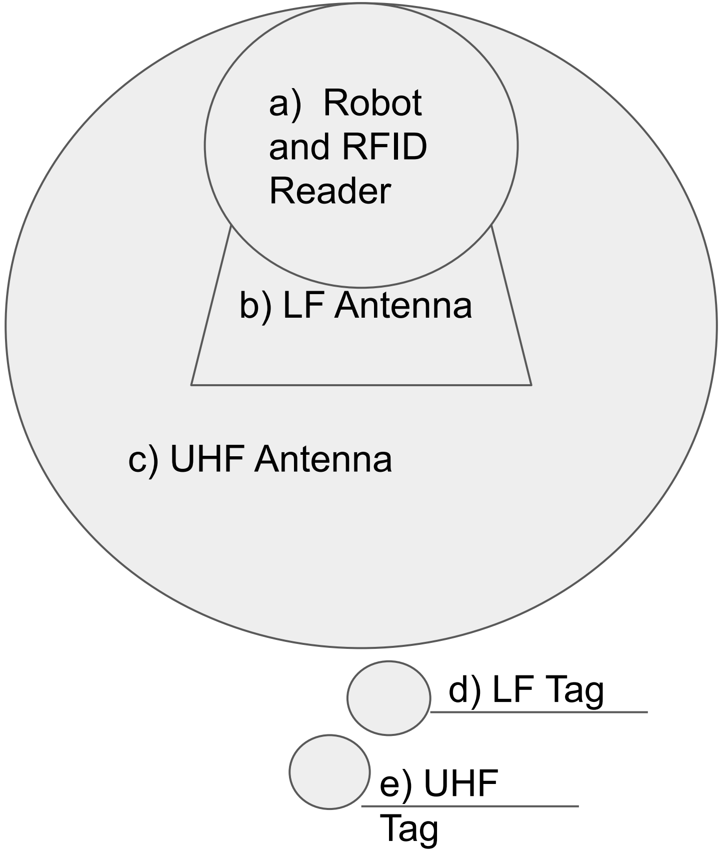


Image 3. Diagram of our robot design. Large circles represent antenna read radii

Speaker: Jake Junsay

Our Key Design Points (cont.)

Point c) UHF Antenna

- More expensive component ($150+)
- Circularly polarized, but with low gain and longer read range. This would allow the antenna to read on any plane, but within a broader range around the robot.
- Read range of 3-9 m.

- These capabilities would allow for our robot to have a wider range to simplify sweeping methods and to allow us to speed up our localization algorithm.

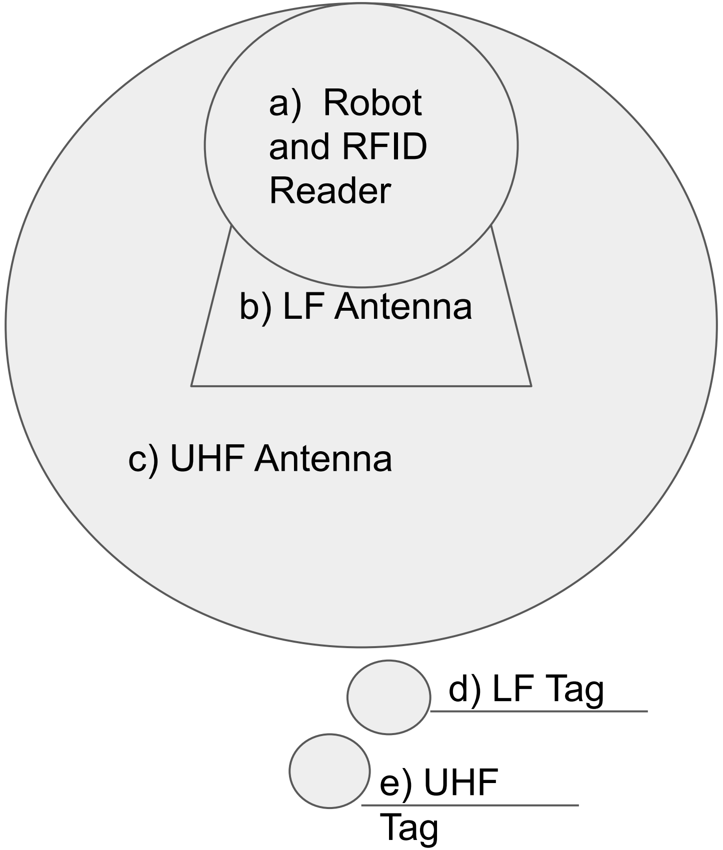


Image 3. Diagram of our robot design. Large circles represent antenna read radii

Speaker: Jake Junsay

Our Key Design Points (cont.)

Point d) LF Tag

- Cheap option (~$2-$3)
- Passive tags, such that they can be read at all times without risk of power loss
- Hard tag that would allow it retain integrity in extreme conditions

Point e) UHF Tag

- Cheap option (~$5-$7)
- Passive tags, such that they can be read at all times without risk of power loss
- Hard tag that would allow it retain integrity in extreme conditions

Both tags necessary to allow for both antennas to be able to find them respectively without issues of interference from either one.

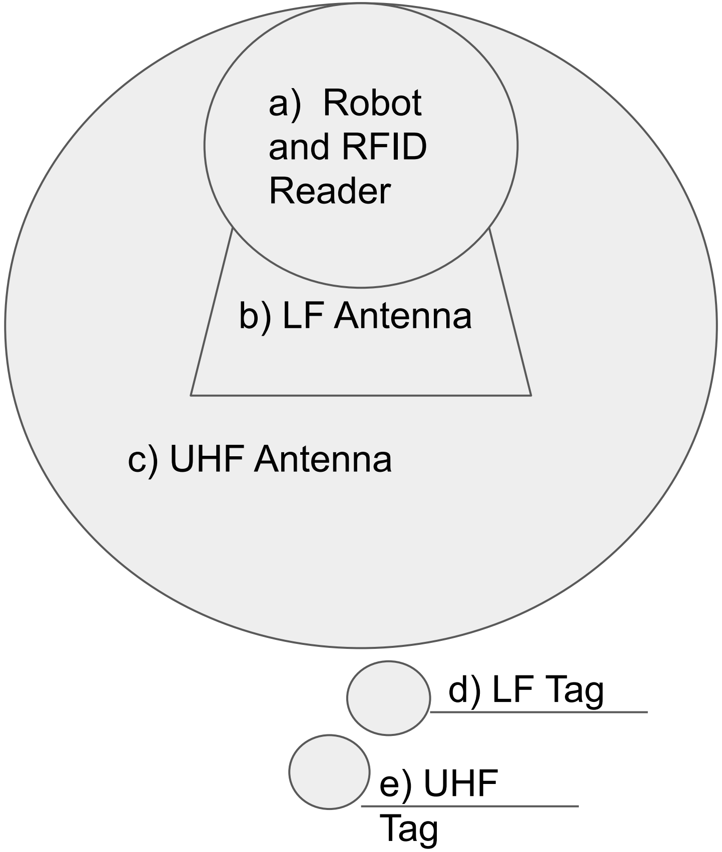


Image 3. Diagram of our robot design. Large circles represent antenna read radii

Speaker: Jake Junsay

Proposed Design

Create 3 variables that represent the input

- Signal 1 (UHF antenna input sensing UF tag)
- Signal 2 (LF input sensing UF tag)
- Signal 3 (LF antenna input sensing LF tag)

While (Signal 1 not = True or Signal 2 not= True)

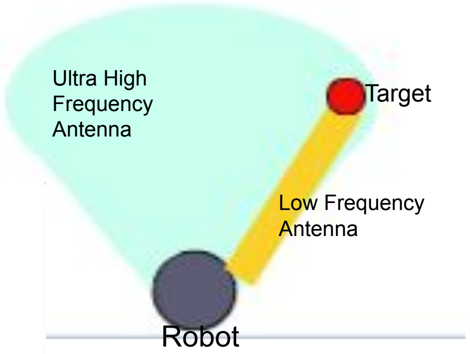
Sweeping movement code

While (Signal 2 not = True)

Rotate robot

While (signal 3 not = True)

Move forward



Speaker: Katlyn Chiu

Proposed Design

1. Robot will begin sweeping until the target (Ultra High Frequency tag) is detected by the Ultra High Frequency antenna
2. Low Frequency antenna will rotate until it detects the target (Ultra High Frequency tag)
3. Robot will move forward in the direction it was detected until the target (Low Frequency tag) is detected.
4. The robot has arrived within 10 cm of the target

Speaker: Katlyn Chiu

Current Code Breakdown

```
clear, clc;
%% Set up robot and surroundings
R = 0.1; % Wheel radius [m]
L = 0.5; % Wheelbase [m]
vehicle = DifferentialDrive(R,L);
x=round((15*rand)) ;
y=round((15*rand)) ;
map = occupancyMap(x,y);
```

- The robot parameters are defined. The room dimensions are randomized, and a map is created

- Two sensors are defined. One has a wider scan and the other has is narrow.

```
detector = ObjectDetector;
detector.fieldOfView = 1/4*pi;
detector.maxRange=2;
objects = [round(15*rand),round(15*rand), 1];
RFIDantenna= LidarSensor;
RFIDantenna.sensorOffset = [0,0];
RFIDantenna.scanAngles = linspace(0,2*pi,51);
RFIDantenna.maxRange = 2;
```

```
viz = Visualizer2D;
viz.hasWaypoints = true;
viz.mapName = 'map';
viz.objectColors = [1 0 0];
viz.objectMarkers = 'so^';
attachObjectDetector(viz,detector);
attachLidarSensor(viz,RFIDantenna);
```

- Define what should be visible in the simulation

Speaker: Haniel Youlesivanson

Current Code Breakdown

```
% simulation parameters
sampleTime = 0.1;
tVec = 0:sampleTime:45;
r = rateControl(1/sampleTime);
% initial conditions
startingPos = [round(15*rand);round(15*rand);round(360*rand)];
pos = zeros(3,numel(tVec));
pos(:,1) = startingPos;
```

- Sets waypoints for sweeping motion based on dimensions of the room

- The rate of the simulation is set at .1s, position matrix is initialized and will be updated throughout the simulation.

```
%% Path Planning
waypoints=zeros((x*(y-1)), 2);
rowCounter=1;
for j=1:x
    waypoints(rowCounter:(rowCounter+(y-2)),1)=j;
   if mod(j,2) ~= 0
    waypoints(rowCounter:(rowCounter+(y-2)),2)= 1:(y-1);
   else
    waypoints(rowCounter:(rowCounter+(y-2)),2)=(y-1):-1:1;
   end
     rowCounter=(rowCounter+(y-1));
end
```

Speaker: Haniel Youlesivanson

Current Code Breakdown

```
% Pure Pursuit Controller
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.5;
controller.DesiredLinearVelocity = 0.75;
controller.MaxAngularVelocity = 1.5;
```

- Defines a pure pursuit controller to follow the waypoints based on the look ahead distance. It is a path tracking algorithm that computes the angular velocity command needed to get from waypoint to waypoint.

```
vfh = controllerVFH;
vfh.DistanceLimits = [0.05 3];
vfh.NumAngularSectors = 36;
vfh.HistogramThresholds = [5 10];
vfh.RobotRadius = L;
vfh.SafetyDistance = L;
vfh.MinTurningRadius = 0.25;
```

- Defines a Vector Field Histogram (VFH) for obstacle avoidance. Not needed since open space is assumed.

```
%% Simulation parameters for narrow antenna
initPose = startingPos ;
tVec = 0:sampleTime:10;
vxRef = zeros(size(tVec));
vyRef = zeros(size(tVec));
wRef = zeros(size(tVec));
ref = [vxRef;vyRef;wRef];
pose = zeros(3,numel(tVec));
pose(:,1) = initPose;
```

- Sets the location of the narrow antenna, and sets its velocity and angular velocity to zero as it won’t move, the robot will.

Speaker: Haniel Youlesivanson

Current Code Breakdown

```
%% Simulation
for i = 2:numel(tVec) % loops 450 times
     % Get the sensor readings
    currentPos = pos(:,i-1);
    ranges = RFIDantenna(currentPos);
     % Run the path following and obstacle avoidance algorithms
    [vRef,wRef,lookAheadPt] = controller(currentPos);
    targetDir = atan2(lookAheadPt(2)-currentPos(2),lookAheadPt(1)-currentPos(1)) - currentPos (3);
    steerDir = vfh(ranges,RFIDantenna.scanAngles,targetDir);
    if ~isnan(steerDir) && abs(steerDir-targetDir) > 0.1
        wRef = 0.5*steerDir;
    end
```

- Start of the simulation. The wider antenna sensor readings are used and fed to the controller. The controller moves based on the position of the robot and the waypoints within its look ahead distance.

Speaker: Haniel Youlesivanson

Current Code Breakdown

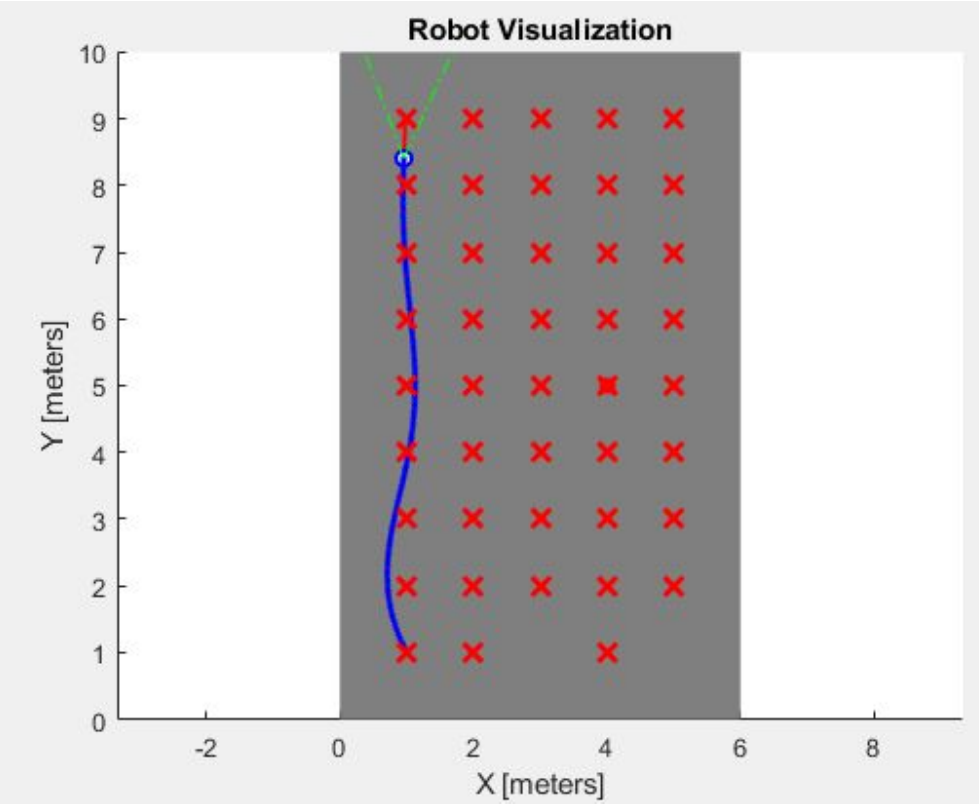
```
velB = [vRef;0;wRef];
vel = bodyToWorld(velB,currentPos);
pos(:,i) = currentPos + vel*sampleTime;
viz(pos(:,i),waypoints,ranges,objects)
vel = bodyToWorld(ref(:,i-1),pose(:,i-1));
pose(:,i) = pose(:,i-1) + vel*sampleTime;
detections = detector(pose(:,i),objects);
viz(pos(:,i),waypoints,ranges,objects)
```

- Uses current positions and velocity to get and display the new position.

- Checks to see if the narrow antenna has sensed the tag, and sends a message to the command window

```
% Display object detections every 10th iteration
if mod(i,10) == 0
    if ~isempty(detections)
        nearestLabel = detections(1,3);
        disp(['Nearest object is of label ' num2str(nearestLabel)]);
    else
        disp('No objects detected');
    end
end
```

Current Simulation



- Robot goes through the waypoints until it scans the object.
- Currently, object isn’t being displayed on the simulation and the robot stops before it turns when both sensors are used.

SWOT Analysis

Strengths (S)

- Very simple sensor algorithm, will make it easy to implement to robot
- With only 2 passive tags it will be no burden for individuals to carry

Weaknesses (W)

- Antenna Radii are based on exact shapes that may not be realistic
- Matlab simulation accuracy

Opportunities (O)

- Due to the nature of the passive RFID tags, it gives us the opportunity to apply to many cases such as soldiers, firefighters, police officers, and etc

Threats (T)

- Obstacles, currently only works in open areas
- Autonomous movement may interrupt sensor input

Speaker: Katlyn Chiu

Future Design and Testing Schedule

During the progress of this project, we have mainly been working off the assumptions of putting our system together from scratch. Moving forward, we are planning to test the plausibility of our design choices and adjusting our design accordingly.

- Real-world testing - Week 5
  - Begin testing of real robot/sensors and try tests using only one antenna.
- Detailed RFID system - Week 6-Week 7
  - We will start to flesh out the RFID system on a more concrete, numerical analysis
- Simulation and Algorithm testing - Week 8-9
  - We will finish our algorithm and begin testing it via MATLAB simulations
- Robot design - Week 10+ (extending into Winter Quarter)
  - If given the chance to continue to work on this project, we will begin design plans for the physical robot and integrating our RFID design

Speaker: George Cambel

Questions and Concerns

- At the beginning of our project, we based much of our research and design plans off using two antennas. However, given available testing materials we will be creating another algorithm that utilizes only one antenna.
- Oversimplification in Matlab simulation, unrealistic for actual robot
- Concerned about how to apply our algorithm to the actual robot and what issues may come up with all the varying sensors.

Speaker: Katlyn Chiu