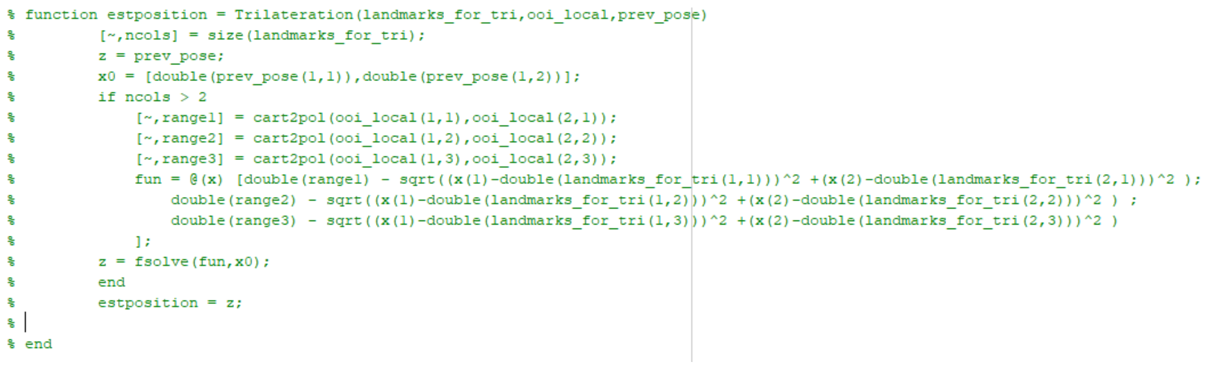
Explanation for approach of part D

Steven LY – Z5257127 -

The first approach taken is a basic Trilateration seen by the script:



```
% function estposition = Trilateration(landmarks_for_tri,ooi_local,prev_pose)
%         [~,ncols] = size(landmarks_for_tri);
%         z = prev_pose;
%         x0 = [double(prev_pose(1,1)),double(prev_pose(1,2))];
%         if ncols > 2
%             [~,range1] = cart2pol(ooi_local(1,1),ooi_local(2,1));
%             [~,range2] = cart2pol(ooi_local(1,2),ooi_local(2,2));
%             [~,range3] = cart2pol(ooi_local(1,3),ooi_local(2,3));
%             fun = @(x) [double(range1) - sqrt((x(1)-double(landmarks_for_tri(1,1)))^2 +(x(2)-double(landmarks_for_tri(2,1)))^2 );
%                 double(range2) - sqrt((x(1)-double(landmarks_for_tri(1,2)))^2 +(x(2)-double(landmarks_for_tri(2,2)))^2 ) ;
%                 double(range3) - sqrt((x(1)-double(landmarks_for_tri(1,3)))^2 +(x(2)-double(landmarks_for_tri(2,3)))^2 )
%             ];
%         z = fsolve(fun,x0);
%         end
%         estposition = z;
% 
% end
```

The script uses the known position of 3 landmarks observable to the lidar to locate the ugv's position in the global frame, using following set of equations:

$$\left\{\begin{array}{l} r_1 - \sqrt{(x-x_1)^2 + (y-y_1)^2} = 0 \\ r_2 - \sqrt{(x-x_2)^2 + (y-y_2)^2} = 0 \\ r_3 - \sqrt{(x-x_3)^2 + (y-y_3)^2} = 0 \end{array}\right\}$$

Figure 1. 3 Constraint Equations

The code will use fsolve to solve our 3 equations, using the previous pose calculated as its initial guess, with the first estimate being the initial pose of the ugv. This approach yielded the result for data file 15.

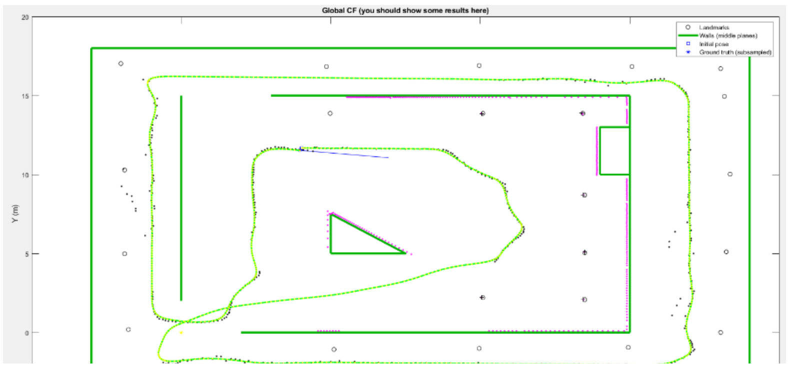


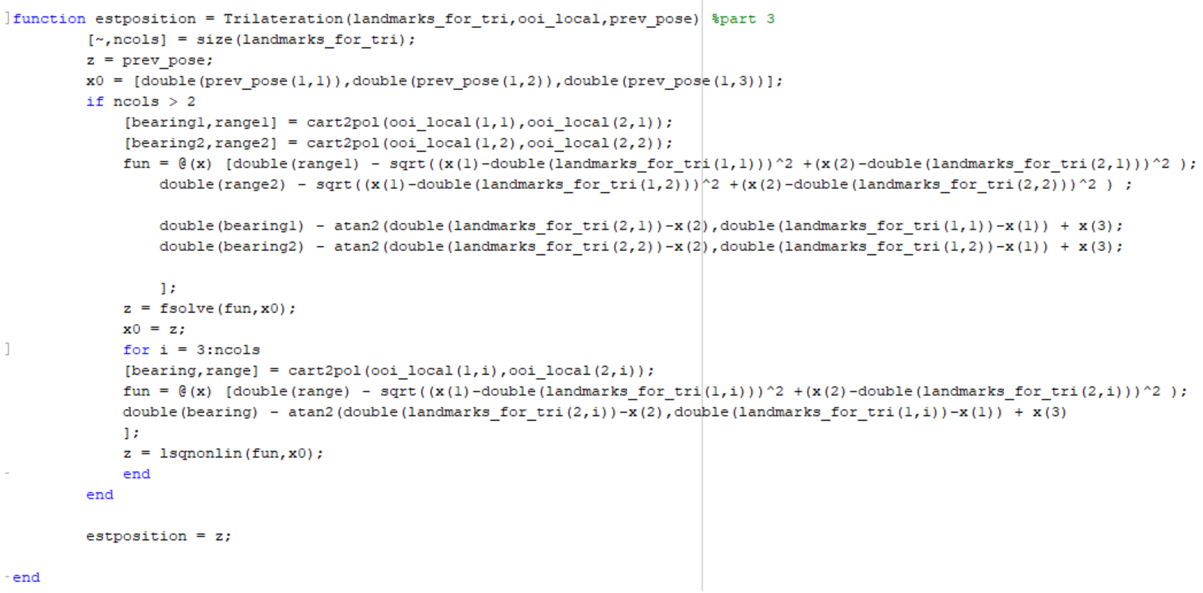
Figure 2. 1st approach-black dots are the estimated positions

In figure 2, it can be observed that there are 3 areas in which the estimated has greatly deviated. I took another approach to try to limit this deviation. Using the equations:

$$\left\{\begin{array}{l} r_i - \sqrt{(x-x_i)^2 + (y-y_i)^2} = 0 \\ \alpha_i = \operatorname{atan2}(y-y_i, x-x_i) - \phi \end{array}\right\}_{i=1}^{N}$$

Figure 3.2D Localization based on Range and Angular measurements

This equation will not only allow us to only need 2 observable landmarks, it will also give us the estimated heading. The code:



```
function estposition = Trilateration(landmarks_for_tri,ooi_local,prev_pose) %part 3
        [~,ncols] = size(landmarks_for_tri);
        z = prev_pose;
        x0 = [double(prev_pose(1,1)),double(prev_pose(1,2)),double(prev_pose(1,3))];
        if ncols > 2
            [bearing1,range1] = cart2pol(ooi_local(1,1),ooi_local(2,1));
            [bearing2,range2] = cart2pol(ooi_local(1,2),ooi_local(2,2));
            fun = @(x) [double(range1) - sqrt((x(1)-double(landmarks_for_tri(1,1)))^2 +(x(2)-double(landmarks_for_tri(2,1)))^2 );
                double(range2) - sqrt((x(1)-double(landmarks_for_tri(1,2)))^2 +(x(2)-double(landmarks_for_tri(2,2)))^2 ) ;

                double(bearing1) - atan2(double(landmarks_for_tri(2,1))-x(2),double(landmarks_for_tri(1,1))-x(1)) + x(3);
                double(bearing2) - atan2(double(landmarks_for_tri(2,2))-x(2),double(landmarks_for_tri(1,2))-x(1)) + x(3);

                ];
            z = fsolve(fun,x0);
            x0 = z;
            for i = 3:ncols
            [bearing,range] = cart2pol(ooi_local(1,i),ooi_local(2,i));
            fun = @(x) [double(range) - sqrt((x(1)-double(landmarks_for_tri(1,i)))^2 +(x(2)-double(landmarks_for_tri(2,i)))^2 );
            double(bearing) - atan2(double(landmarks_for_tri(2,i))-x(2),double(landmarks_for_tri(1,i))-x(1)) + x(3)
            ];
            z = lsqnonlin(fun,x0);
            end
        end

        estposition = z;

end
```

This code uses fsolve to solve four constraint equations that is obtained when there is at least 2 known landmark positions, using previous estimates as an initial guess. Any redundant equations gained from extra landmarks are then looped through a for loop that uses least square non linear solver using the estimated position we got from the first 2 landmark positions. The following code yielded these results.

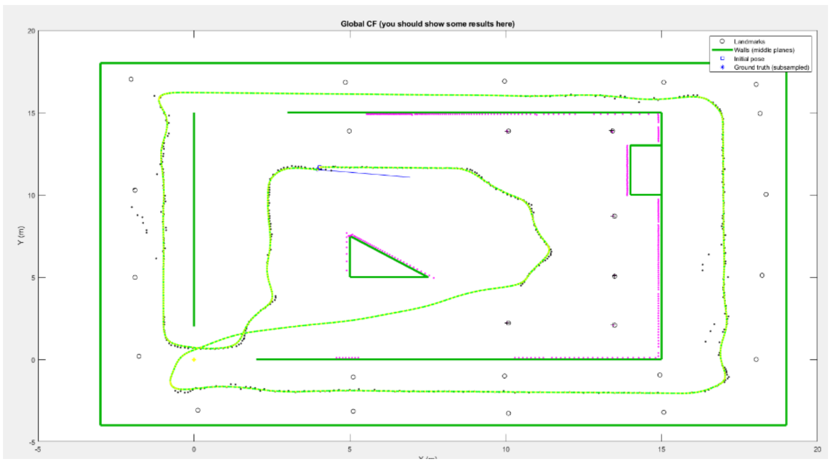


Figure 4. 2nd approach

Both approaches gave the same estimations, however approach 2 will allow us to the estimated heading and hence is the main approach I will use. There are many limitations to my approach, one being that if not enough landmarks are detected, there will no estimations occurring as seen by lack of points during the top of the ugv trip. There are also limitations in accuracy, as the lidar is polluted by noise, possible errors in the reflective surfaces, and the angular resolution of the lidar. Note that the estimated location for the landmarks in local coordinate frame is also an estimated value that takes the mean of a cluster of points as the centre of it, which will also impact the accuracy of our estimates. Another limitation is seen in the deviations observed, which occurs when there are 3 landmarks collinearly vertical to each other. This could be solved by applying EKF, which is not necessary to this project.