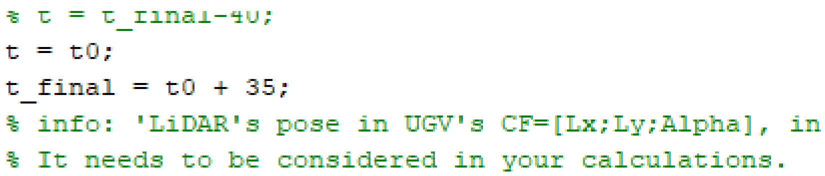
Part C explanation

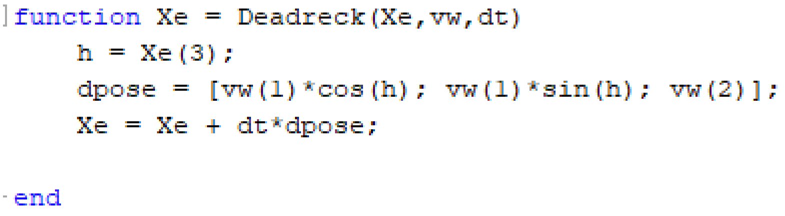
The program will ask for an initial guess to be inputted by the user.

The guess is used for our optimizer, which will use fminsearch to find the lowest cost function for a given bias. The cost function samples between 30seconds before the final sample is taken and the final time



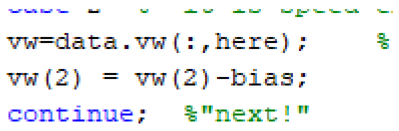
```
% t = t_final-40;
t = t0;
t_final = t0 + 35;
% info: 'LiDAR's pose in UGV's CF=[Lx;Ly;Alpha], in
% It needs to be considered in your calculations.
```

In a while loop, the cost function will get the position of the ugv estimated by dead reckoning using the measures the velocity and heading rate of the vehicle.



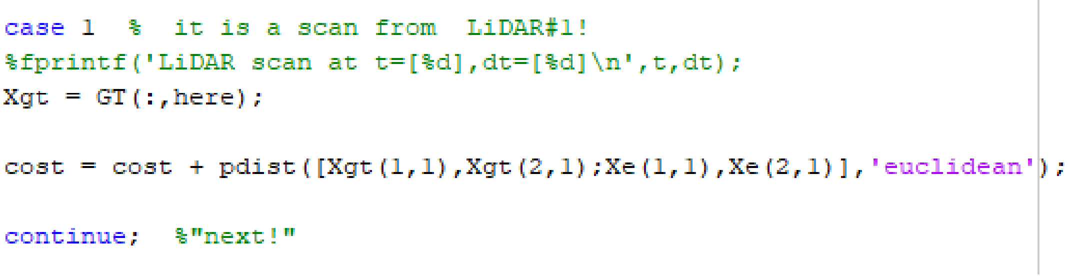
```
function Xe = Deadreck(Xe,vw,dt)
    h = Xe(3);
    dpose = [vw(1)*cos(h); vw(1)*sin(h); vw(2)];
    Xe = Xe + dt*dpose;

end
```

The heading rate of the vehicle will have the bias removed during each loop



```
case 2   %  it is speed &
vw=data.vw(:,here);    %
vw(2) = vw(2)-bias;
continue;  %"next!"
```

Whenever there is a scan from the lidar, the function will then use the known ground truth given the time sample and find the distance between the pose estimated and the ground truth. This distance will then be summed up and the total distance over the time sample will be the end cost.



```
case 1  %  it is a scan from  LiDAR#1!
%fprintf('LiDAR scan at t=[%d],dt=[%d]\n',t,dt);
Xgt = GT(:,here);

cost = cost + pdist([Xgt(1,1),Xgt(2,1);Xe(1,1),Xe(2,1)],'euclidean');

continue;  %"next!"
```