AEROSP 584 Final Project

Question 1:

We used R to represent the covariance of the measurement noise and Q to represent the covariance of the process noise. Then we make a slight modification to our Pk prediction, simply adding in this covariance:

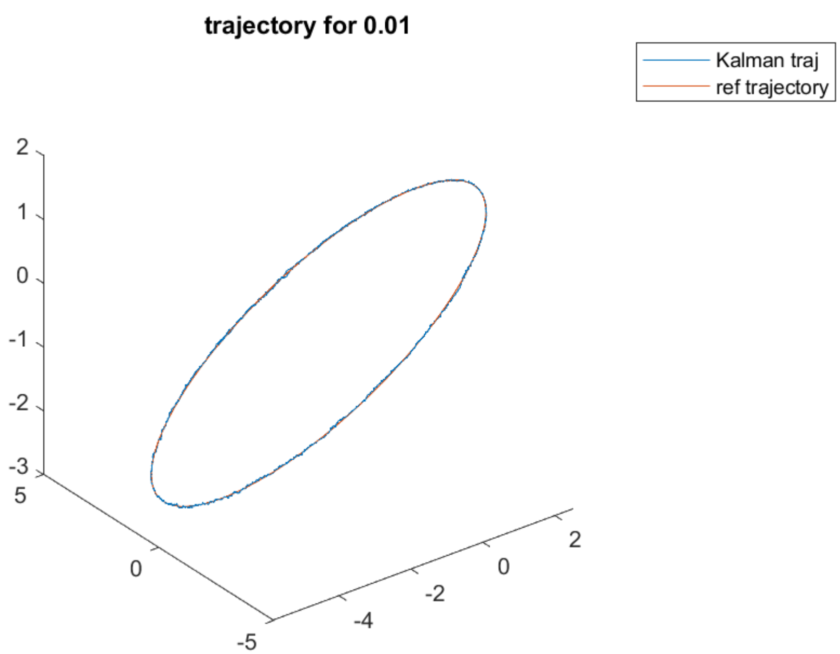
$$Pk=APk-1AT+Q$$

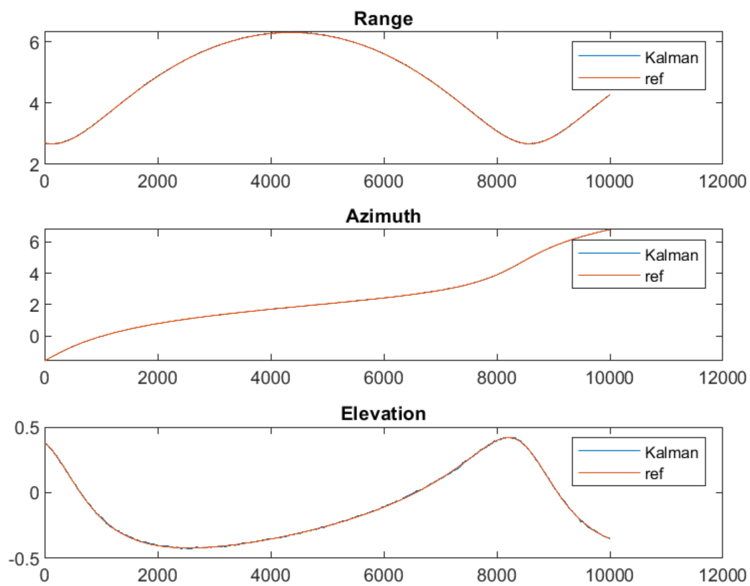
The very small values for the Q matrix turn out to be very helpful in keeping our estimated state values on track. These values help us determine the system accuracy and bias in the estimation. These values are set by looking at the the system noise given and taking into account that the steady state of P will lie between Q and R.

As the given error variance is 0.01, keeping that into account,

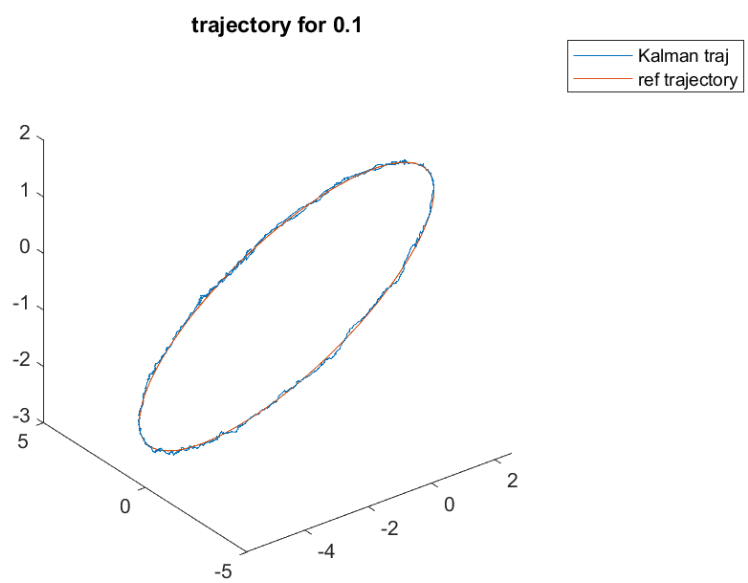
Q = 0.001*eye(6); R = 0.01*eye(3);

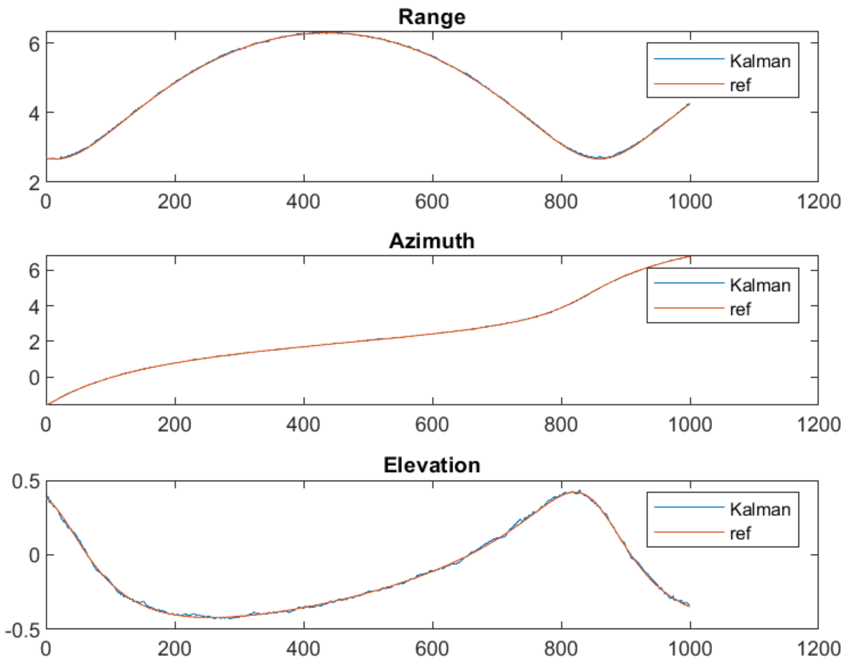
Case T_measure = 0.01:



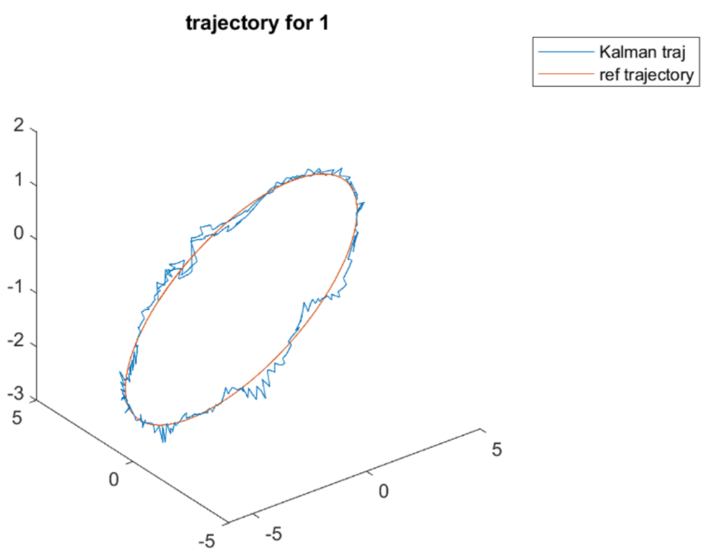


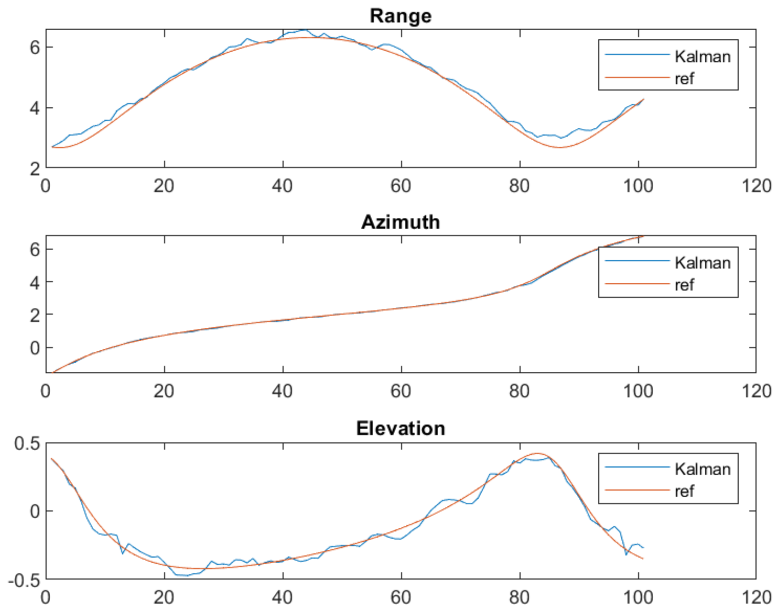
Case T_measure = 0.1:



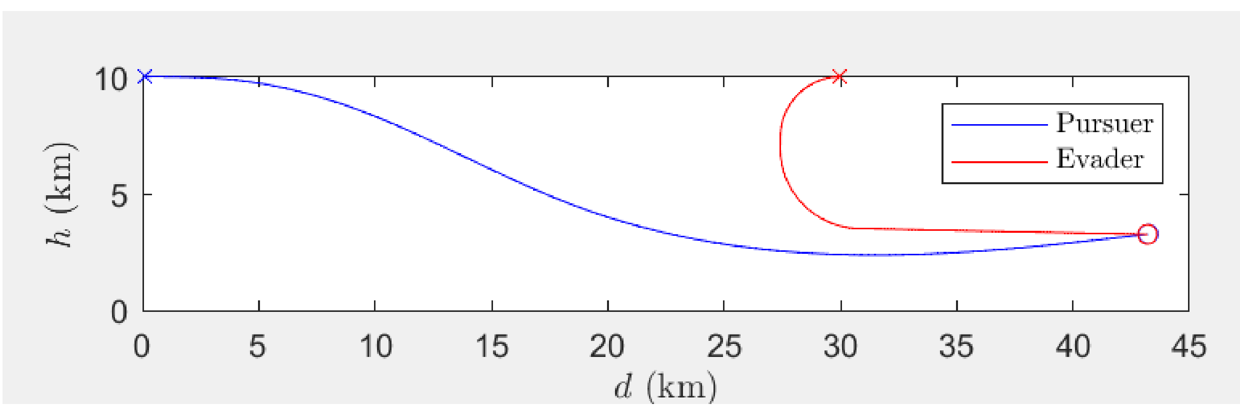


Case T_measure = 1:



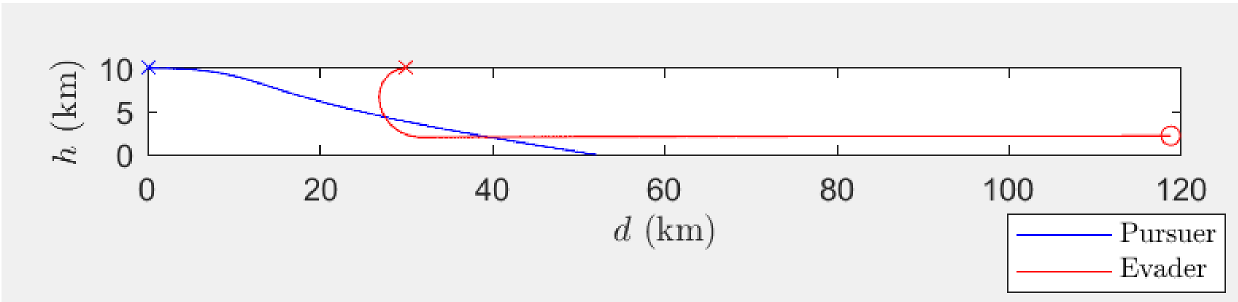


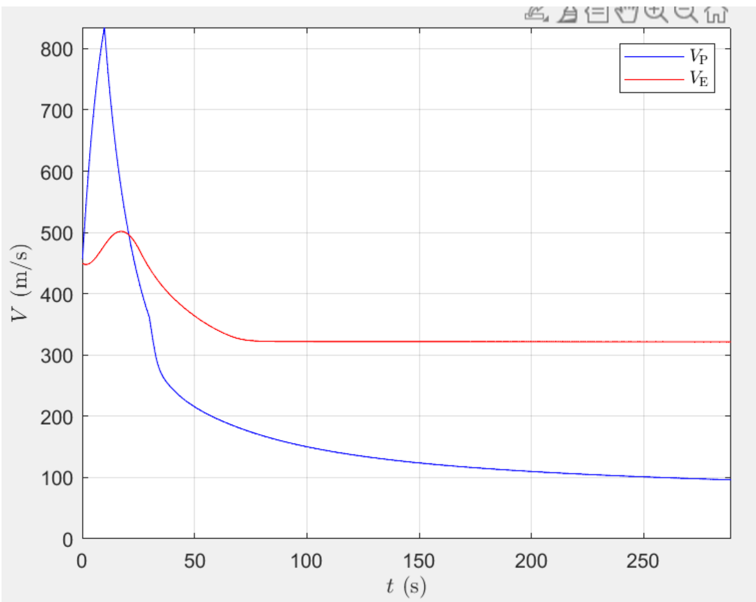
Question 2:
Case 1: Hit
Miss distance: 0.8077



Case 2: Miss

Miss Distance: 6.6665e+04

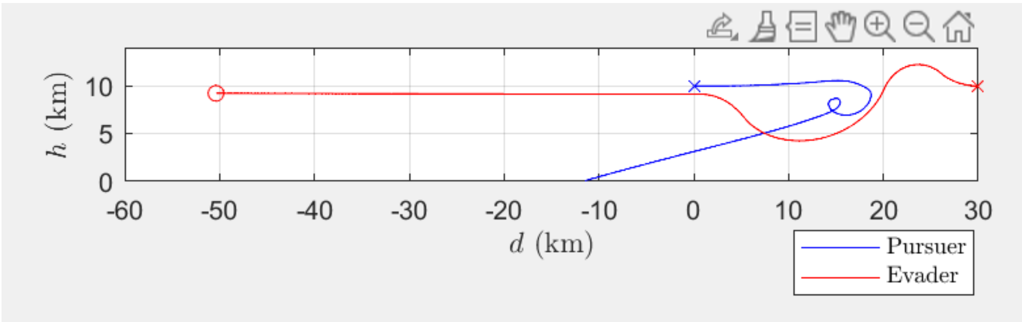


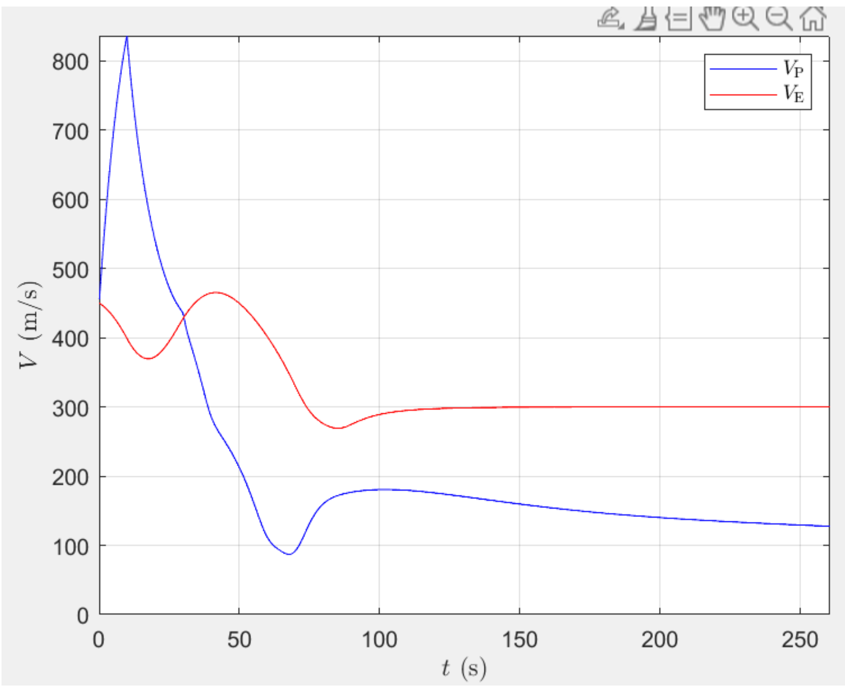


Question 3:

Case 1: Miss

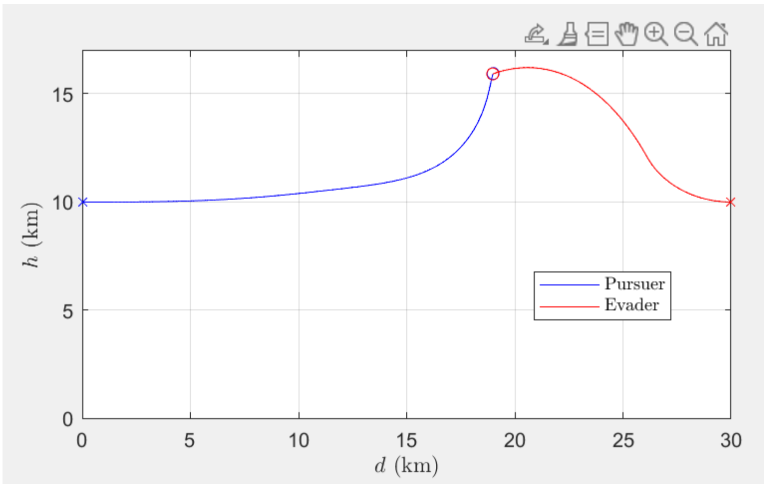
Miss Distance :   3.9866e+04

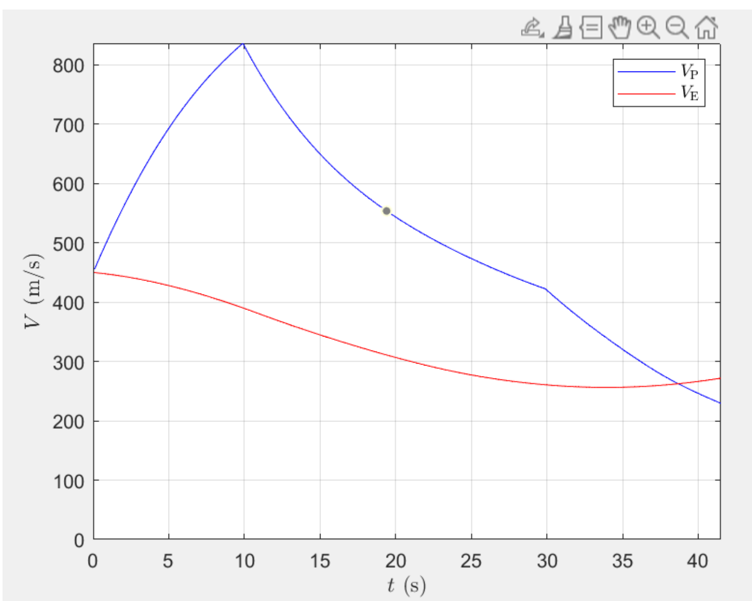




Case 2: Hit

Miss Distance: 0.3440





Question 4:

It is observed that in case of proportional guidance law, when both missile and target are in the same line. Due to the same gamma values the rate of R_dot and Beta_dot goes to zero. This leads to cases where only the gravity term is working and no additional acceleration is provided. So incase of the same speed it's a never ending chase.

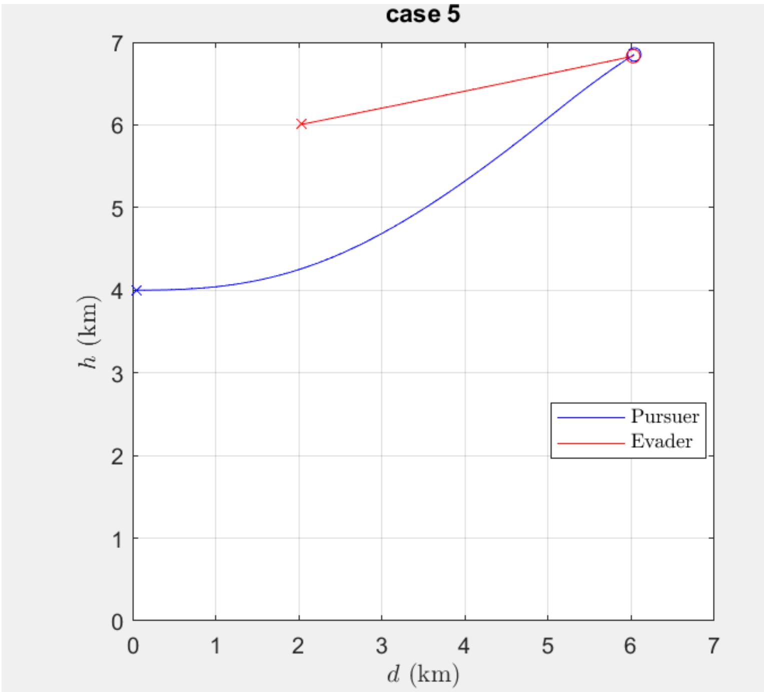
To tackle this *from the observation of Optimal guidance law data*, we can see that there is more acceleration to go up and the missile descends back. To reproduce this case, the used guidance law is as follows:

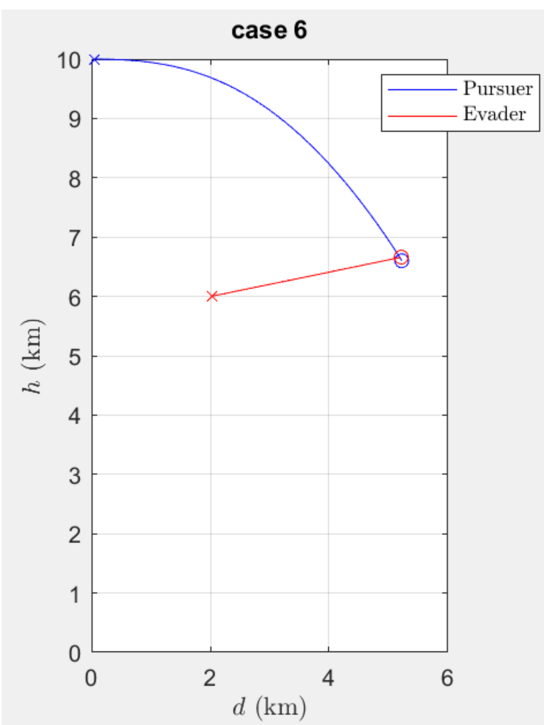
```
nzP = -7*(abs(r_dot)*beta_dot) - g*cos(theta)
If (R<lim):
        if(flag ==1)
                nzP = nzP + correction_term1
                if(height_pursuer  - height_envader  >= limit or R_dot>=lim2 or betadot>lim3)
                        flag =0;
                end
        end
        if(abs(R_dot)<lim_act1 or abs(beta_dot)<lim_act2)
                flag =1
        end
end
```

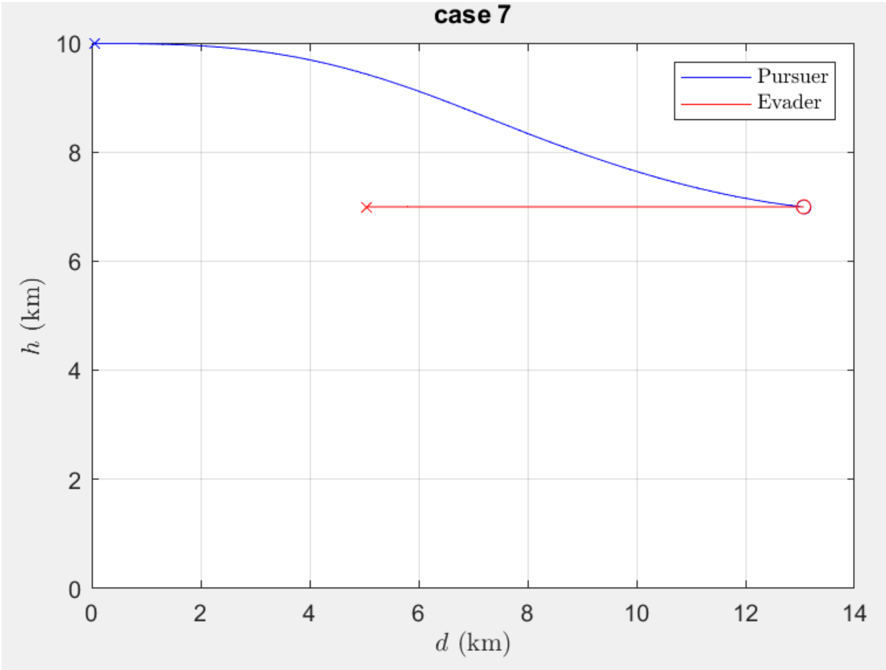
This system is **performing better than proportional law by hitting in four cases**:

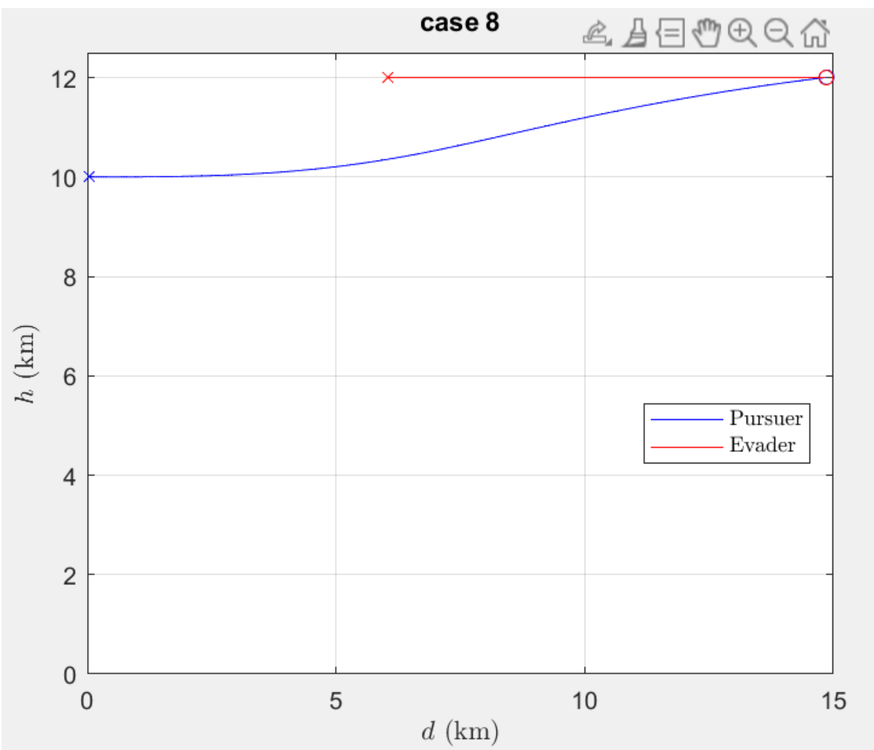
- So this law will produce a certain thrust when both are aligned together and that creates some offset in the guidance to generate some offset in pursuer to generate additional thrust to chase.
- The additional thrust generation will be activated if R_dot or Beta_dot equals zero and that will trigger the system to generate additional acceleration in upwards direction.
- That can create some room for missiles to hit the target.
- One more idea was to add a derivative term to the system, but it might not act when the system is completely aligned.

Hitting cases:

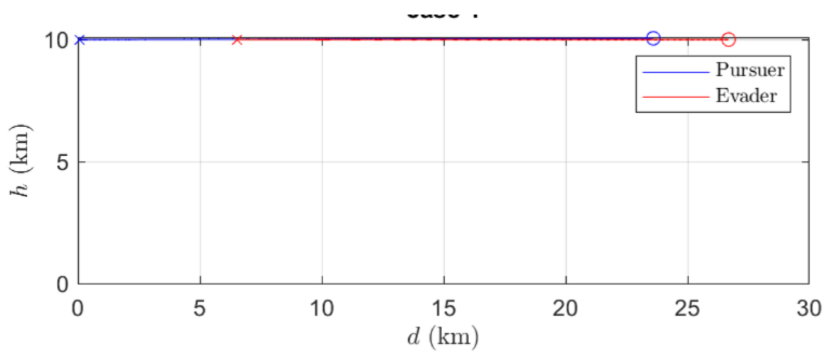


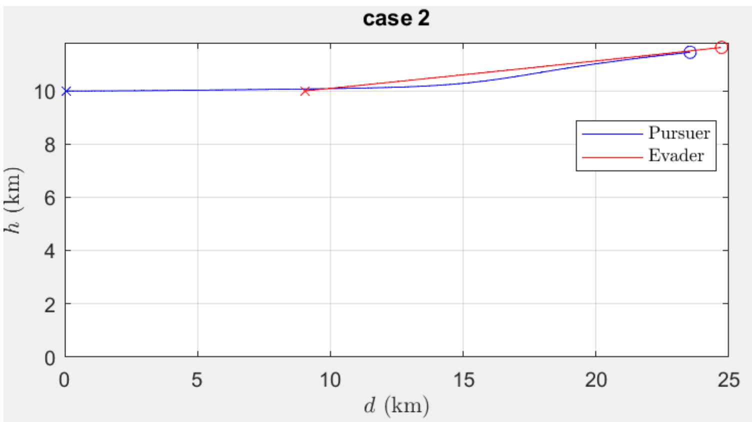


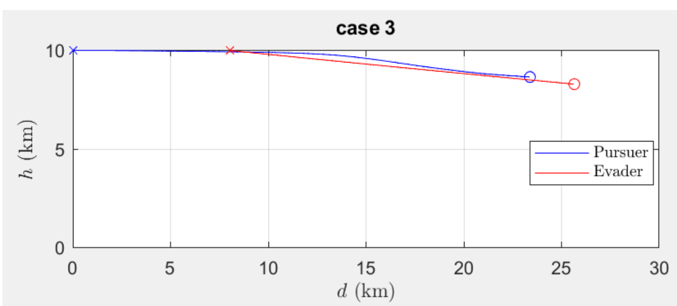

case 7
Pursuer
Evader
h (km)
d (km)
0
2
4
6
8
10
12
14

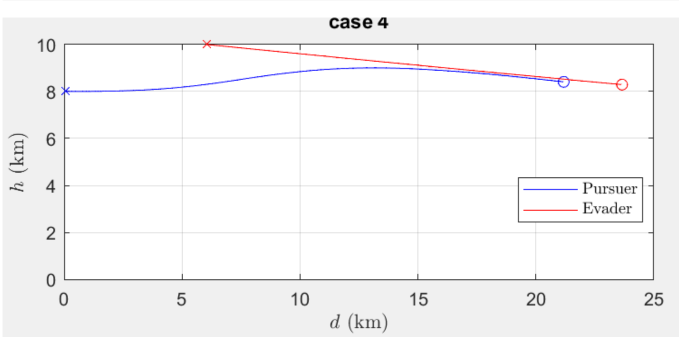

case 8
Pursuer
Evader
h (km)
d (km)
0
5
10
15

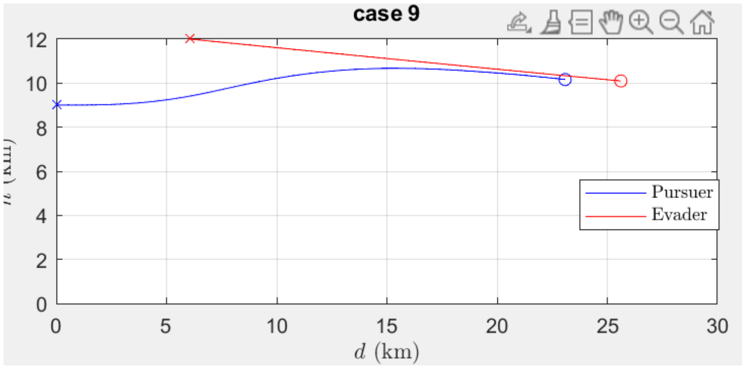
Missing cases:

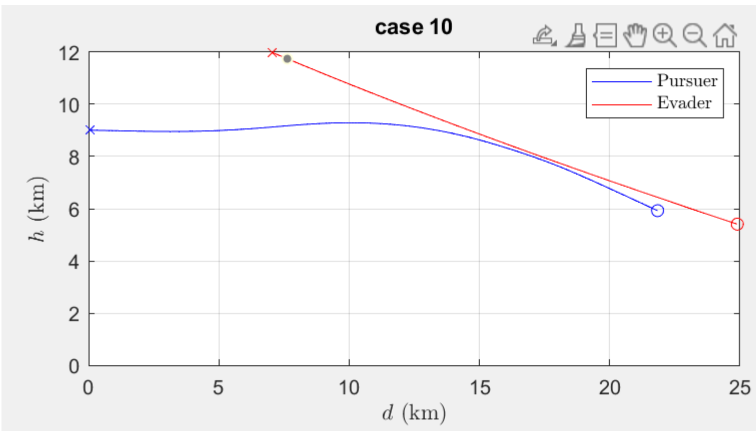









case 9
Pursuer
Evader
d (km)
0
5
10
15
20
25
30
0
2
4
6
8
10
12


case 10
Pursuer
Evader
h (km)
d (km)
0
5
10
15
20
25
0
2
4
6
8
10
12