
Problem 4

Table of Contents

Load Parameters	1
Batch Least Squares	1
Recursive Least Squares	2
Comparison	3

Load Parameters

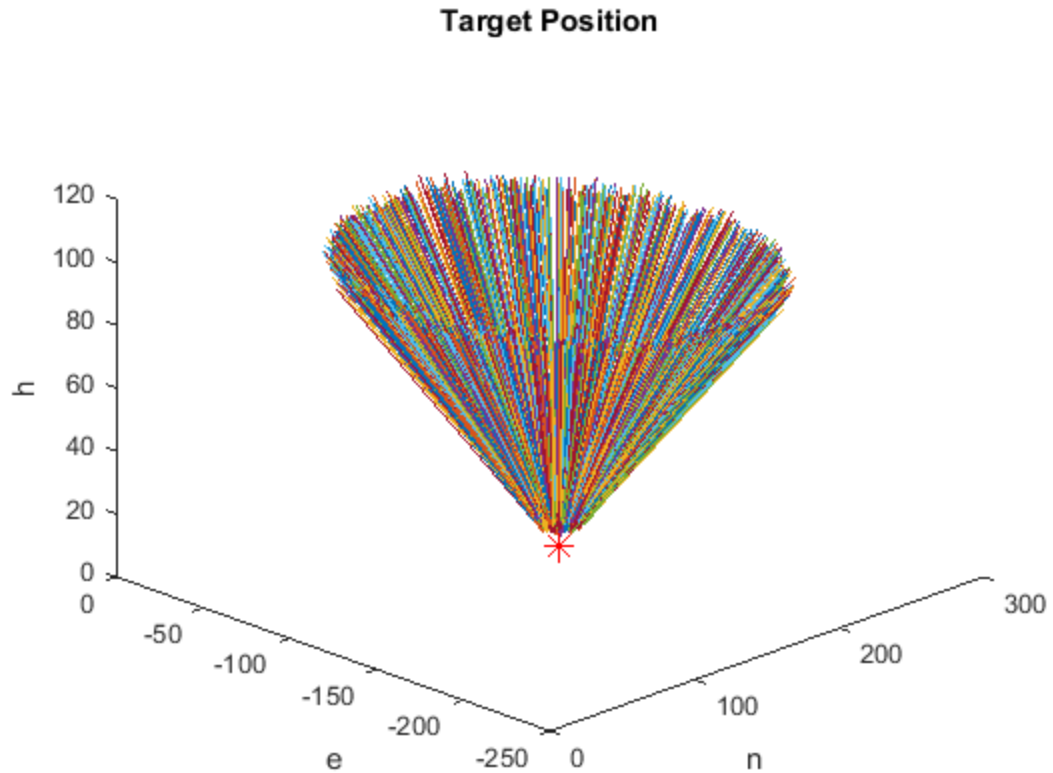
```
clear all
close all
load('mid2_prob4.mat')
p_uav = [pn.', pe.', h.']; %[pn pe h];
clear pn pe h
```

Batch Least Squares

```
z = 1;
for i = 1:length(t),
    A(z:z+2,:) = [eye(3) -ell(i,:).'];
    b(z:z+2,1) = p_uav(i,:).';
    z = z+3;
end
x_star = inv((A.'*A))*A.'*b;
p_t_batch = x_star(1:3);
L_batch = x_star(4);

clear x_star A b z i

%Plot p_uav + L*ell to verify answers
figure()
title('Target Position')
hold on
plot3(p_t_batch(1), p_t_batch(2), p_t_batch(3), '*r', 'MarkerSize',10)
for i = 1:length(t),
    quiver3(p_uav(i,1), p_uav(i,2), p_uav(i,3), L_batch*ell(i,1),...
            L_batch*ell(i,2), L_batch*ell(i,3))
end
xlabel('n')
ylabel('e')
zlabel('h')
view(-45,30)
```



Recursive Least Squares

```
%Use RLS Filter to step through all data and update the model
%Initialization
Pm = eye(4,4);
x_m = zeros(4,1);
x_store(1,:) = x_m;
iden = eye(3);
j = 1;
%For RLS I can only add 1 row at a time so looking at north, east,
  then h
for i = 1:length(t),
    for k = 1:3,
        aml = [iden(j,:), -ell(i,k)].';
        bml = p_uav(i,k);
        Km1 = (Pm*aml)/(1+aml.'*Pm*aml);
        xml = x_m + Km1*(bml - aml'*x_m);
        Pml = Pm - Km1*aml'*Pm;
        Pm = Pml;
        x_m = xml;
        x_store(i+1,:) = x_m';
        j = j+1;
        if j == 4,
            j = 1;
        end
    end
end
```

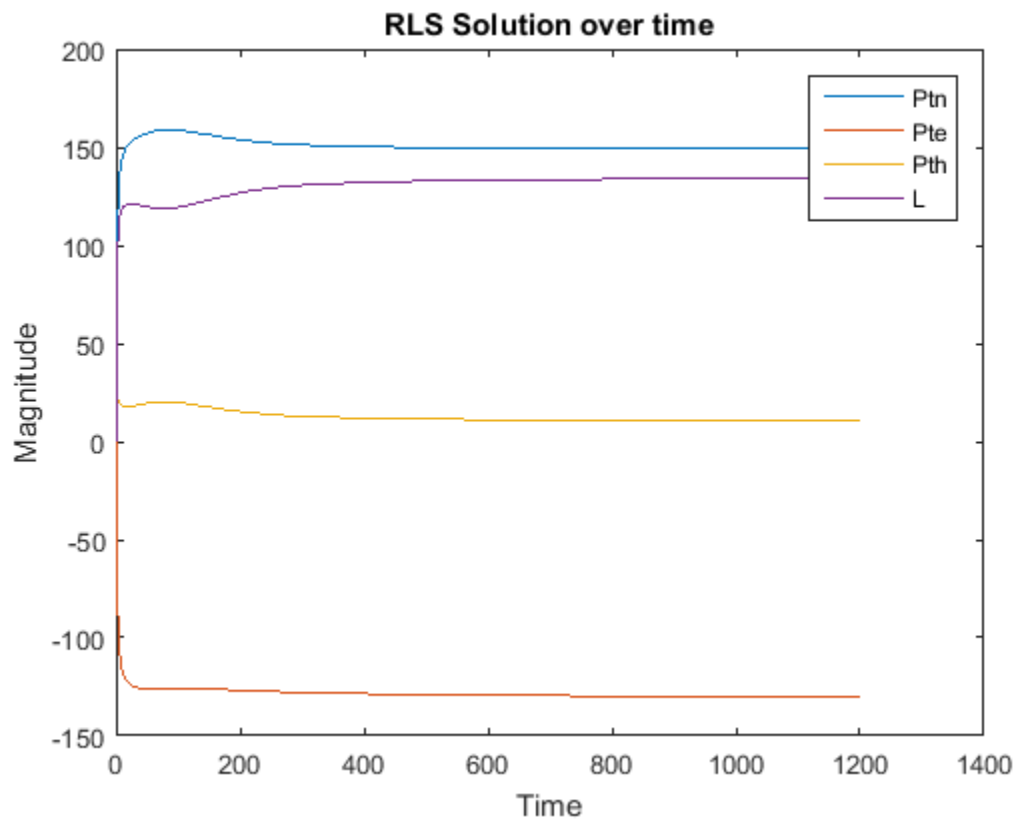
```

        end
    end

    p_t_RLS = x_m(1:3);
    L_RLS = x_m(4);

    figure()
    plot(x_store)
    title('RLS Solution over time')
    legend('Ptn', 'Pte', 'Pth', 'L')
    xlabel('Time')
    ylabel('Magnitude')

```



Comparison

```

p_t_batch
p_t_RLS
L_batch
L_RLS
%As shown, my RLS algorithm obtained nearly the same solution as batch
%least squares. Though it should be noted, that as shown on the plot,
%it
%took a little time to settle on the correct target and length
%parameters.
%This is because I never actually took an inverse. Rather than
%starting

```

```
%with a batch and then doing RLS, I started by seeding it with the
identity
%matrix and zeros, and it found the right parameters.
```

```
p_t_batch =
```

```
    149.9843
   -130.0600
    10.5321
```

```
p_t_RLS =
```

```
    149.8597
   -129.9518
    10.6500
```

```
L_batch =
```

```
    134.3088
```

```
L_RLS =
```

```
    134.1186
```

Published with MATLAB® R2015a