

ASEN 5070-Statistical Orbit Determination-HW

6

Zach Dischner

10-18-2012

1 Problem 1

Given:

$$y = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad (1)$$

$$E[\epsilon] = 0 \quad (2)$$

$$R = W^{-1} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$H = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (4)$$

$$\bar{x} = 2 \quad (5)$$

$$\sigma^2(\bar{x}) = \frac{1}{2} \quad (6)$$

1.1 1a - Solve for \hat{x} using the batch processor with R^{-1} as the weighting matrix

I solved for \hat{x} using Equation 4.3.25 in the text, which results in the least squares solution for \hat{x} including *a priori* and weighting information.

$$\hat{x} = (H^T W H + \bar{W}_k)^{-1} (H^T W y + \bar{W} \bar{\mathbf{x}}_k) \quad (7)$$

Equation 7 was solved in MATLAB, since all components are readily available. The code accomplishing this is in Appendix A.

$$\boxed{\hat{x} = 1.5}$$

1.2 1b - Find standard deviation of the estimation error of \hat{x}

Finding the standard deviation of the estimation error first involved solving equation 4.3.7 in the book, which solves for the estimation error ϵ .

$$\hat{\epsilon} = \mathbf{y} - \mathbf{H}\hat{\mathbf{x}} \quad (8)$$

Once that was solved, the standard deviation could easily be found.

$$\boxed{\sigma(\hat{\epsilon}) = 0.57735}$$

1.3 1c - Find best estimate of estimation error $\hat{\epsilon}$

The estimate of the estimation error $\hat{\epsilon}$ was already included in part 1b, equation (8).

$$\boxed{\hat{\epsilon} = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}}$$

Again, all code is provided in Appendix A.

2 Problem 2

Given:

$$X = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 1 & t & \frac{1}{2}t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (9)$$

$$\bar{P}_0 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (10)$$

$$X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (11)$$

$$t_0 = 0 \quad (12)$$

At time t_1 :

$$Y(t_1) = G * X = x(t_1) = 2 \quad (13)$$

Therefore:

$$G = \begin{bmatrix} x & 0 & 0 \end{bmatrix} \quad (14)$$

$$\sigma(\epsilon_Y) = 1 = R \quad (15)$$

2.1 2a - Find \hat{X}_{t_0} Using the Batch Processor Algorithm

Following the Batch Processing Algorithm outlined in Chapter 4 of the book I first initialized my variables, and found the state transition matrix. I did this using a definition of the state transition matrix:

$$X(t_1) = \Phi(t_0, t_1) * X_{t_0} \quad (16)$$

Equation (9) gives us the relationship for X between times at this instant, so Φ is simple enough to find.

$$\Phi(t_0, t_1) = \begin{bmatrix} 1 & t & \frac{t^2}{2} \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Next, I used this to determine the H matrix at t_1 . We were given G (more or less), and knowing that $\hat{H} = \frac{dG}{dX}$, finding H was trivial.

$$H_1 = \tilde{H}_1 * \Phi_{t_0, t_1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & \frac{1^2}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Which yields:

$$G = \begin{bmatrix} 1 & 1 & \frac{1}{2} \end{bmatrix} \quad (19)$$

Due to the fact that our dynamics and observation relationships are both linear, there is no need to linearize and find state and observation deviation vectors (x,y). Instead we can solve directly for X and Y.

Continuing through the batch algorithm in the book, I formed my batch calculation terms:

$$\Lambda = \bar{P}_0^{-1} + H^T R^{-1} H \quad (20)$$

$$N = \bar{P}_0^{-1} \bar{X}_0 + H^T R^{-1} Y \quad (21)$$

And finally, I solved for \hat{X}_{t0}

$$\hat{X}_{t0} = \Lambda^{-1} N \quad (22)$$

Resulting in a final value of

$$\hat{X}_{t0,a} = \begin{bmatrix} 21/29 \\ 25/29 \\ 28/29 \end{bmatrix}$$

This result was found in MATLAB, and the accompanying code is found in Appendix A.

2.2 2b - Find \hat{X}_{t_0} Using the Sequential Processor Algorithm

Similar to 4.21a, I followed the procedure outlined in the book to perform the sequential processing algorithm. The catch is that the process in the book involves finding \hat{X}_{t_1} , using \tilde{H} matrices to map just between observations. To find \hat{X}_{t_0} , we have to map the sequential algorithm back to the t_0 epoch using the H matrix. Using the H matrix to map the observation back to the epoch, I first found the Kalman Gain.

$$K_0 = \bar{P}_0 H (H \bar{P}_0 H^T + R)^{-1} \quad (23)$$

All terms in equation (23) were found previously, and were simply re-used for this calculation, in MATLAB. Next, I found \hat{X}_{t_0} using the equations in the book.

$$\hat{X}_{t_0} = \bar{X}_0 + K_0 [Y_1 - H \bar{X}_0] \quad (24)$$

Which results in a final answer of:

$$\hat{X}_{t_0,b} = \begin{bmatrix} 21/29 \\ 25/29 \\ 28/29 \end{bmatrix}$$

This answer matches perfectly that found in part a, with the batch processor algorithm. Essentially, we were just adjusting our initial value, using the original initial value and an adjustment with the Kalman gain and a future observation error.

2.3 2c - Find \hat{X}_{t0} by mapping \hat{X}_{t1} back to t_0 with $\Phi(t_1, t_0)$

For this problem, I went through the sequential processing algorithm as the book outlines it, by using \tilde{H} matrices and covariance matrices found at the time of the observation, t_1 .

First, I mapped the original priori covariance matrix (eq (10)) to the current time with the pre-determined state transition matrix (eq (17)).

$$\bar{P}_1 = \Phi(t_0, t_1)\bar{P}_0\Phi^{-1}(t_0, t_1) \quad (25)$$

Next, I obtained a new \bar{X}

$$\bar{X}_1 = \Phi(t_0, t_1)\bar{X}_0 \quad (26)$$

Next, the Kalman gain was found for this observation time.

$$K_1 = \bar{P}_1\tilde{H}(\tilde{H}\bar{P}_1\tilde{H}^T + R)^{-1} \quad (27)$$

Which then allows me to calculate the new best estimate of \hat{X}_1

$$\hat{X}_{t1} = \bar{X}_1 + K_1[Y_1 - \tilde{H}\bar{X}_1] \quad (28)$$

Finally, I then mapped the result of eq (28) back to the initial state using the inverse of the state transition matrix, knowing $\Phi(t_0, t_1) = \Phi^{-1}(t_1, t_0)$.

$$\hat{X}_{t0} = \Phi^{-1}(t_0, t_1)\hat{X}_{t1} \quad (29)$$

When performed, the exact same result is found as in problem 4.21a and 4.21b.

$$\hat{X}_{t0,c} = \begin{bmatrix} 21/29 \\ 25/29 \\ 28/29 \end{bmatrix}$$

Appendix A - MATLAB Code

A Main Script

Contents

- Problem 4.20
- Problem 4.21

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
% Zach Dischner-10/14/2012
%
% ASEN 5070-Statistical Orbit Determination
%
% Homework 6
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
clc;clear all;close all; format compact;format long g;
```

Problem 4.20

```
[a b c] = Problem420();
fprintf('4.20-a          xhat = %3.5f \n\n',a)
fprintf('4.20-b      std(error) = %3.5f \n\n',b)
disp(['4.20-c          error = [',num2str(c),' ','']])
fprintf('\n\n')
```

```
4.20-a          xhat = 1.50000
```

```
4.20-b      std(error) = 0.57735
```

```
4.20-c          error = [-0.5          0.5          -0.5]'
```

Problem 4.21

```
format rat
[a b c] = Problem421();
format rat
```

```

disp(['4.21-a      Batch Xhat0    = [' ,num2str(a'),']',''])
disp(['4.21-b      Sequential Xhat0    = [' ,num2str(b'),']',''])
disp(['4.21-c      Phi(1,0)*Xhat1 -->Xhat0    = [' ,num2str(c'),']',''])

fprintf('\n\n Or in a more RATIONAL way...\n')
a
b
c

4.21-a      Batch Xhat0    = [0.72414      0.86207      0.96552]'
4.21-b      Sequential Xhat0    = [0.72414      0.86207      0.96552]'
4.21-c      Phi(1,0)*Xhat1 -->Xhat0    = [0.72414      0.86207      0.96552]'

Or in a more RATIONAL way...
a =
    21/29
    25/29
    28/29
b =
    21/29
    25/29
    28/29
c =
    21/29
    25/29
    28/29

```


B Problem 4.20

```
function [a b c] = Problem420()

%% Givens
y      = [1 2 1]';
Ee     = 0;
R      = [1/2 0 0; 0 1 0; 0 0 1]; % E(ee')
H      = [1 1 1]';
xbar   = 2;
var_xbar = 2; % sigma^2(xbar)

%% Part A: Find xhat with batch processing algorithm
W      = inv(R);
Wbar = var_xbar;
xhat = (H'*W*H + Wbar)\(H'*W*y+Wbar*xbar);
a      = xhat;

%% Part B: Find Standard Deviation of the Estimation Error with xhat
epsilon = y-H*xhat;
b      = std(epsilon);

%% Part C: Find Estimation Error
c      = epsilon;
end
```

C Problem 4.21

```
function [a b c] = Problem420()

%% Givens
y      = [1 2 1]';
Ee     = 0;
R      = [1/2 0 0; 0 1 0; 0 0 1]; % E(ee')
H      = [1 1 1]';
xbar   = 2;
var_xbar = 2; % sigma^2(xbar)
```

```

%% Part A: Find xhat with batch processing algorithm
W = inv(R);
Wbar=var_xbar;
xhat = (H'*W*H + Wbar)\(H'*W*y+Wbar*xbar);
a = xhat;

%% Part B: Find Standard Deviation of the Estimation Error with xhat
epsilon = y-H*xhat;
b = std(epsilon);

%% Part C: Find Estimation Error
c = epsilon;
end

```