Homework 4 Report

AERO626, Spring 2023

Name: David van Wijk   UIN: 932001896

Problem 1 Results

The solution and derivation for Problem 1 can be found below the analysis of Problem 2 (past the appendix).

Problem 2 Results

To solve this problem, a Kalman filter was implemented in MATLAB, the code for which can be found in Appendix A. The plots showing $\pm 1\sigma$ of the position and the velocity covariance at each time are shown in Figure 1 prior to and after each measurement. For the position, we can see that at each second (i.e. at each measurement), the covariance decreases, and then increases until the next measurement. This makes sense because our uncertainty in our position will increase during the propagation step, and then will be reduced once we use the information that the measurement provides. For the velocity plot, the uncertainty decreases between measurements, and then drops even more after the update step. At $t = 0$ we receive a measurement, which immediately allows us to clamp down on the position uncertainty, but has no effect on the initial standard deviation of velocity. This makes sense because we only have position measurements and have not propagated the states at $t = 0$. The final standard deviation of the position is 0.030582 $[m]$ and the final standard deviation of the velocity is 0.031183 $[\frac{m}{s}]$. The result therefore, is that we are able to achieve a standard deviation for our position that is tighter (lower) than the standard deviation of the measurement noise (which is 0.1 $[m]$). This makes sense because the filter is conditioned on all the previous measurements, rather than one individual measurement.

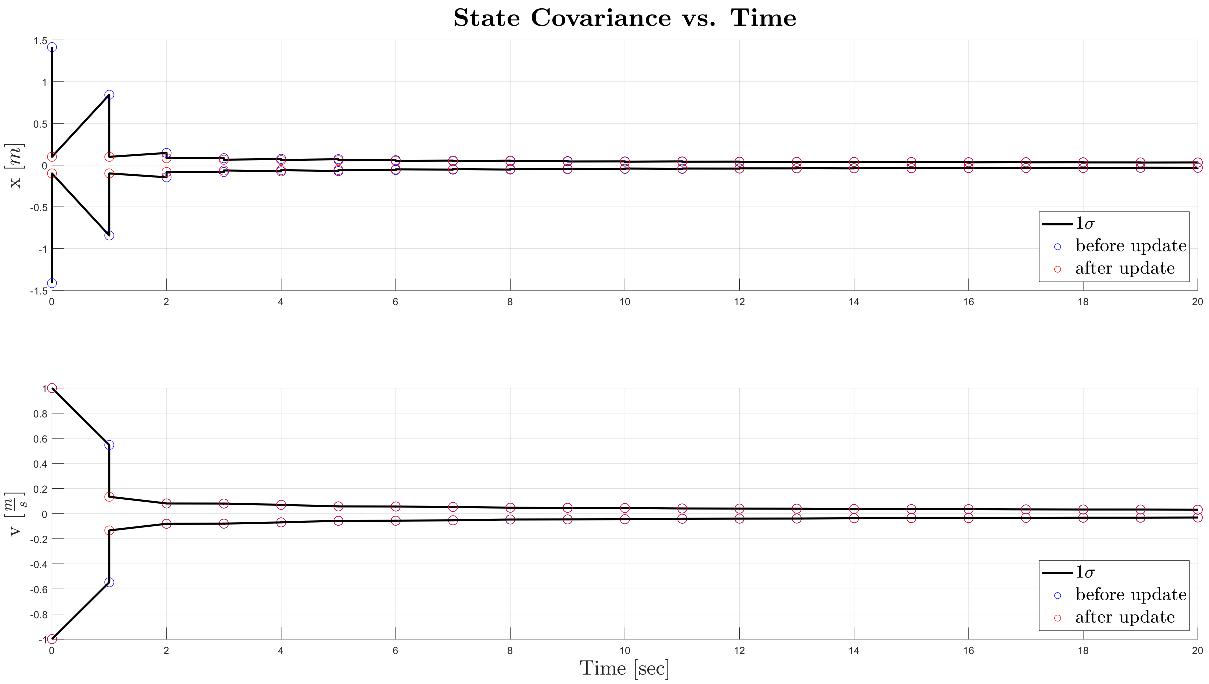

Figure 1: Position and Velocity Covariance versus Time.

The $\pm 1\sigma$ of the measurement noise covariance (constant for this system) and the innovation covariance is plotted in Figure 2. We can observe that the $\pm 1\sigma$ of innovation covariance approaches the $\pm 1\sigma$ of the measurement noise covariance ($\sqrt{P_{vv}}$) as time increases. At the final time of $t = 20$, the $\pm 1\sigma$ of the innovation covariance is 0.10503 $[m]$, as compared with the constant $\pm 1\sigma$ of measurement noise of .1 $[m]$. This makes sense because the $\pm 1\sigma$ of the innovation shouldn't be lower than $\pm 1\sigma$ of the measurement noise covariance, but will approach this value as more and more measurements are received.

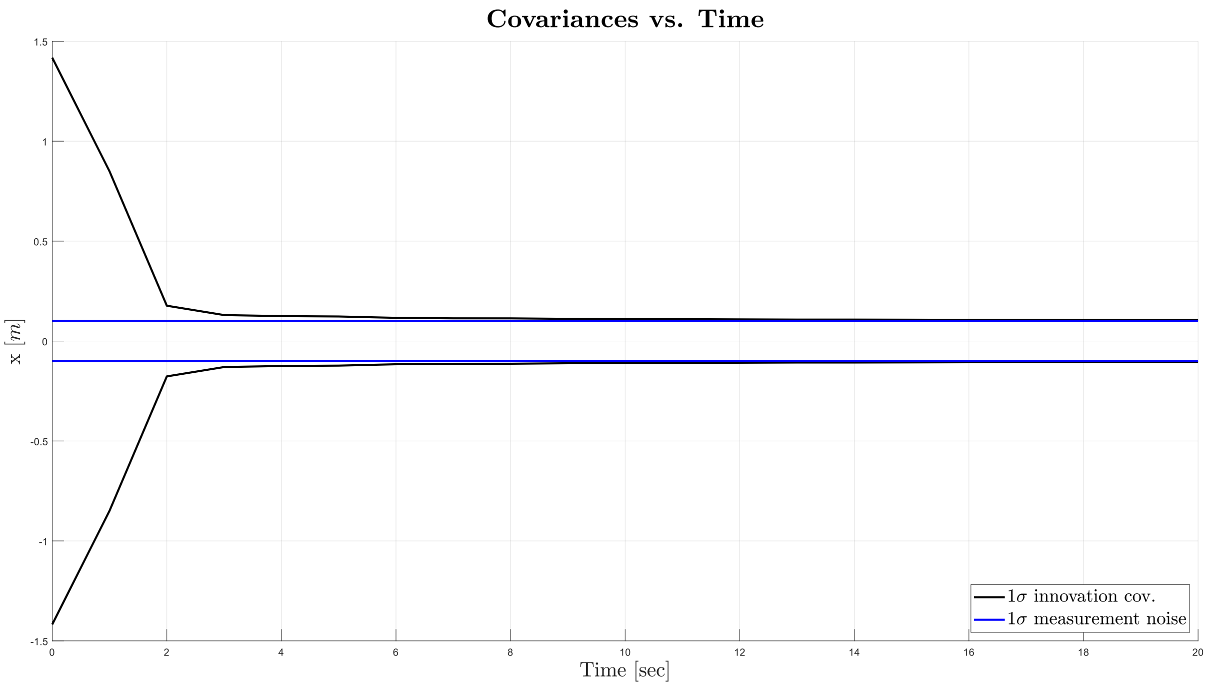

Figure 2: Measurement noise and Innovation Covariance versus Time.

A Matlab Code

```
%% AERO 626 Homework 4
%
%   Texas A&M University
%   Aerospace Engineering
%   van Wijk, David

%% Problem 2

plot_flag = true;

rng(100) % Seed to reproduce results
opts = odeset('AbsTol',1e-6,'RelTol',1e-6);

mx0 = [1; 0];          %[m], [m/s]
Pxx0 = diag([2,1]);   %[m]^2, [m/s]^2
Pvv = .1^2;           %[m]^2 constant covariance

N = 21;
tspan = linspace(0,20,N);
Fx = [0 1; -1 0];
Hx = [1 0];
Hv = 1;
x0 = mx0 + chol(Pxx0)'*randn(2,1);

% Store data
std_x_full = zeros(N*2,2);
std_innov_full = zeros(N,2);

count = 0;
for i = 1:N
    % Propagate the truth state
    if i == 1
        xk = x0;
    else
        [~,X] = ode45(@(t,x) prop(t,x,Fx),[tspan(i-1),tspan(i)],xkm1,opts
           );
        xk = X(end,1:2)';
    end

    % Generate measurement
    zk = Hx*xk + chol(Pvv)'*randn;

    % Propagate step of Kalman filter
    if i == 1
        mxkm = mx0;
        Pxxkm = Pxx0;
    else
        [~,X] = ode45(@(t,x) prop(t,x,Fx),[tspan(i-1),tspan(i)],[mxkm1;
           Pxxkm1(:)],opts);
        mxkm = X(end,1:2)';
        Pxxkm = reshape(X(end,3:end)',2,2);
```

```
    end

    % Store stuff
    count = count + 1;
    std_x_full(count,:) = sqrt(diag(Pxxkm))';

    mzkm = Hx*mxkm;
    Pxzkm = Pxxkm*Hx';
    Pzzkm = Hx*Pxxkm*Hx' + Hv*Pvv*Hv';

    % Kalman Gain
    Kk = Pxzkm/Pzzkm;

    % Update mean and covariance
    mxkp = mxkm + Kk*(zk - mzkm);
    Pxxkp = Pxxkm - Pxzkm*Kk' - Kk*(Pxzkm)' + Kk*(Pzzkm)*Kk';

    % Re-initialize for next loop
    xkm1 = xk;
    mxkm1 = mxkp;
    Pxxkm1 = Pxxkp;

    % Store stuff
    count = count + 1;
    std_x_full(count,:) = sqrt(diag(Pxxkp))';
    std_innov_full(i,:) = sqrt(diag(Pzzkm))';
end

disp(['The final standard deviation of the position is: ', num2str(
   std_x_full(end,1)),' [m]'])
disp(['The final standard deviation of the velocity is: ', num2str(
   std_x_full(end,2)),' [m/s]'])
disp(['The final standard deviation of the innovation covariance is: ',
   num2str(std_innov_full(end,1)),' [m]'])

if plot_flag
    xaxis_sz = 20; yaxis_sz = 20; legend_sz = 18;

    % Figure 1
    txt = '1$\sigma$';
    opts_pts1 = {80,'b',"o"};
    opts_pts2 = {80,'r',"o"};

    figure; grid on; set(gcf, 'WindowState', 'maximized');
    subplot(2,1,1); hold on; grid on;
    title('\textbf{State Covariance vs. Time}','Fontsize',25,'interpreter
       ','latex')
    t_mod = sort([tspan tspan]);
    a1 = plot(t_mod,std_x_full(:,1),'-','Color','k','LineWidth',2,'
       MarkerSize',20);
    b1 = scatter(tspan,std_x_full(1:2:end,1),opts_pts1{:});
    b2 = scatter(tspan,std_x_full(2:2:end,1),opts_pts2{:});
    plot(t_mod,-std_x_full(:,1),'-','Color','k','LineWidth',2,'MarkerSize
```

```
        ',20)
    scatter(tspan,-std_x_full(1:2:end,1),opts_pts1{:})
    scatter(tspan,-std_x_full(2:2:end,1),opts_pts2{:})
    ylabel('x [$m$]','Fontsize',yaxis_sz,'interpreter','latex')
    legendtxt = {txt, 'before update', 'after update'};
    legend([a1 b1 b2],legendtxt,'FontSize',legend_sz,'interpreter','latex
        ','location','southeast')

    subplot(2,1,2); hold on; grid on;
    a2 = plot(t_mod,std_x_full(:,2),'-','Color','k','LineWidth',2,'
        MarkerSize',20);
    b3 = scatter(tspan,std_x_full(1:2:end,2),opts_pts1{:});
    b4 = scatter(tspan,std_x_full(2:2:end,2),opts_pts2{:});
    scatter(tspan,-std_x_full(1:2:end,2),opts_pts1{:})
    scatter(tspan,-std_x_full(2:2:end,2),opts_pts2{:})
    plot(t_mod,-std_x_full(:,2),'-','Color','k','LineWidth',2,'MarkerSize
        ',20)
    xlabel('Time [sec]','Fontsize',xaxis_sz,'interpreter','latex')
    ylabel('v [$\frac{m}{s}$]','Fontsize',yaxis_sz,'interpreter','latex')
    legend([a2 b3 b4],legendtxt,'FontSize',legend_sz,'interpreter','latex
        ','location','southeast')

    % Figure 2
    figure; grid on; set(gcf, 'WindowState', 'maximized'); hold on;
    title('\textbf{Covariances vs. Time}','Fontsize',25,'interpreter','
        latex')
    a1 = plot(tspan,std_innov_full(:,1),'-','Color','k','LineWidth',2,'
        MarkerSize',20);
    plot(tspan,-std_innov_full(:,1),'-','Color','k','LineWidth',2,'
        MarkerSize',20)
    a2 = plot(tspan,-ones(N,1)*sqrt(Pvv),'-','Color','b','LineWidth',2,'
        MarkerSize',20);
    plot(tspan,ones(N,1)*sqrt(Pvv),'-','Color','b','LineWidth',2,'
        MarkerSize',20)
    ylabel('x [$m$]','Fontsize',yaxis_sz,'interpreter','latex')
    xlabel('Time [sec]','Fontsize',xaxis_sz,'interpreter','latex')
    legend([a1, a2],{'1$\sigma$ innovation cov.','1$\sigma$ measurement
        noise'},'FontSize',legend_sz,'interpreter','latex','location','
        southeast')

end

%% Functions

function dx = prop(~,x,Fx)
% Propagate state or mean
dx = Fx*x(1:2);

if length(x) > 2
% Reshape cov. matrix
P = reshape(x(3:end)',2,2);
% Prop cov.
dP = Fx*P + P*Fx';
```

```
dx = [dx; dP(:)];
end
end
```

1. Consider the noiseless, scalar, continuous-time dynamical system

$$\dot{x}(t) = 0$$

that is accompanied by scalar, discrete-time measurements of the form

$$z_k = x_k + v_k\,,$$

where v_k is a zero-mean measurement noise with constant variance P_{vv}. If the initial mean and variance of the state are known to be $m_{x,0}$ and $P_{xx,0}$, respectively, derive closed-form expressions for the Kalman gain and posterior variance at the ℓ^{th} update time, where $k = 1, 2, \ldots, \ell$. Your results should be expressed entirely in terms of $P_{xx,0}$, P_{vv}, and ℓ.

Given :

$\rightarrow \mathbb{E}\{\underline{v}_k\} = 0$

$\rightarrow P_{vv,k} = P_{vv} \quad \forall k$

$\rightarrow \mathbb{E}\{\underline{x}(t_0)\} = \underline{m}_{x,0}$

$\dot{\underline{x}}(t) = 0$, thus $\underline{x}(t) = a$ where a is a constant, $a \in \mathbb{R}$

Thus, $x_k = a \quad \forall k$

Starting with the definition of the Kalman gain, given on page 205 of the notes :

$$K_k = (P_{xz,k}^-)(P_{zz,k}^-)^{-1}$$

where $\underline{z}_k = \underline{H}_{x,k}\underline{x}_k + H_{v,k}\underline{v}_k$

$$P_{xz,k}^- = P_{xx,k}^- \cancel{H_{x,k}^T}^{\,1}$$

$$P_{zz,k}^- = \cancel{H_{x,k}}^{\,1} P_{xx,k}^- \cancel{H_{x,k}^T}^{\,1} + \cancel{H_{v,k}}^{\,1} \cancel{P_{vv,k}}^{\,P_{vv}} \cancel{H_{v,k}^T}^{\,1}$$

Need to find $P_{xx,k}^-$ ....

$P_{xx,k}^-$ is found by propagating the covariance forward, under the propagation model described by:

$$\dot{P}_{xx}(t) = F_x(t)P_{xx}(t) + P_{xx}(t)F_x^T(x) + \cancel{F_w(t)Q_{ww}(t)F_w^T(t)}^{\,0 \text{ (no modelled process noise)}}$$

Since $\dot{x}(t) = 0$, $F_x(t) = 0$, $\therefore \dot{P}_{xx}(t) = 0$

To find the results at an arbitrary update time, ℓ lets find the pattern by starting with $k = 1$ ...

For $k=1$, $P_{xx,1}^{-} = P_{xx,0}$ because $\dot{P}_{xx}(t) = 0$

Then,
$$\left.\begin{aligned} P_{xz,1}^{-} &= P_{xx,0} \\ P_{zz,1} &= P_{xx,0} + P_{vv} \end{aligned}\right\} \quad k_1 = (P_{xx,0})\left(P_{xx,0} + P_{vv}\right)^{-1}$$

The state covariance is then updated according to the equation given on page 228 of the notes, which is valid for this case since we have linear measurements:

$$P_{xx,1}^{+} = \left[I_2 - K_1 H\right] P_{xx,1}^{-}$$

$$= \left[I_2 - (P_{xx,0})\left(P_{xx,0} + P_{vv}\right)^{-1}\right] P_{xx,0}$$

$$= \left[I - \frac{P_{xx,0}}{P_{xx,0} + P_{vv}}\right] P_{xx,0}$$

$$= \left[\frac{\cancel{P_{xx,0}} + P_{vv}}{P_{xx,0} + P_{vv}} - \frac{\cancel{P_{xx,0}}}{P_{xx,0} + P_{vv}}\right] P_{xx,0}$$

$$P_{xx,1}^{+} = \frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}$$

The same procedure can be done with $k=2$, using $P_{xx,1}^{+}$ as $P_{xx,2}^{-}$

$$P_{xz,2}^{-} = P_{xx,2}^{-}$$

$$P_{zz,2}^{-} = P_{xx,2}^{-} + P_{vv}$$

Thus, $$k_2 = (P_{xx,2}^{-})\left(P_{xx,2}^{-} + P_{vv}\right)^{-1} = \left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right)\left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}} + P_{vv}\right)^{-1}$$

$$= \left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right)\left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}} + \frac{P_{vv}(P_{xx,0} + P_{vv})}{P_{xx,0} + P_{vv}}\right)^{-1}$$

$$= \left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right)\left(\frac{P_{vv} P_{xx,0} + P_{vv}(P_{xx,0} + P_{vv})}{P_{xx,0} + P_{vv}}\right)^{-1}$$

$$= \left(\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right)\left(\frac{P_{xx,0} + P_{vv}}{P_{vv} P_{xx,0} + P_{vv}(P_{xx,0} + P_{vv})}\right)$$

$$= \frac{P_{vv} P_{xx,0}}{2 P_{xx,0} P_{vv} + P_{vv}^2}$$

$$K_2 = \frac{P_{xx,0}}{2 P_{xx,0} + P_{vv}}$$

And again, we can solve for $P_{xx,2}^{+}$ which will be $P_{xx,3}^{-}$ in the next iteration using

$$P_{xx,2}^{+} = [I_2 - K_2 H] P_{xx,2}^{-}$$

$$= \left[\frac{2 P_{xx,0} + P_{vv} - P_{xx,0}}{2 P_{xx,0} + P_{vv}}\right]\left[\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right]$$

$$= \left[\frac{P_{xx,0} + P_{vv}}{2 P_{xx,0} + P_{vv}}\right]\left[\frac{P_{vv} P_{xx,0}}{P_{xx,0} + P_{vv}}\right]$$

$$P_{xx,2}^{+} = \left[\frac{P_{vv} P_{xx,0}}{2 P_{xx,0} + P_{vv}}\right]$$

Doing the same for $k=3$ will result in $K_3 = \dfrac{P_{xx,0}}{3 P_{xx,0} + P_{vv}}$, & $P_{xx,3}^{+} = \dfrac{P_{vv} P_{xx,0}}{3 P_{xx,0} + P_{vv}}$

This pattern holds for all $k = 1, 2, 3, \ldots$ and so for an arbitrary $k = \ell$ the Kalman gain, and the posterior variance will be:

$$\boxed{K_\ell = \frac{P_{xx,0}}{\ell P_{xx,0} + P_{vv}}} \quad \text{and} \quad \boxed{P_{xx,\ell}^{+} = \frac{P_{vv} P_{xx,0}}{\ell P_{xx,0} + P_{vv}}}$$