TTK4190 Guidance and Control of Vehicles

Assignment 2 Part 2

M. Brandt & I. Kingman

Problem 3 - State Estimation using a Kalman filter

The continuous time linear lateral aircraft model with $\hat{\mathbf{x}} = [\beta, \phi, p, r]^\top$ is:

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \mathbf{A}_k\hat{\mathbf{x}} + \mathbf{B}_k\delta_a^c + \mathbf{E}_k\mathbf{w},\\ \mathbf{y} &= \mathbf{C}_k\hat{\mathbf{x}} + \mathbf{v},\end{aligned} \tag{1}$$

with system matrices:

$$\begin{aligned}\mathbf{A}_k &= \begin{bmatrix} -0.322 & 0.052 & 0.028 & -1.12 \\ 0 & 0 & 1 & -0.001 \\ -10.6 & 0 & -2.87 & 0.46 \\ 6.87 & 0 & -0.04 & -0.32 \end{bmatrix}, \quad \mathbf{B}_k = \begin{bmatrix} 0.002 \\ 0 \\ -0.65 \\ -0.02 \end{bmatrix},\\ \mathbf{C}_k &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{E}_k = \mathbf{I}_{4x4}.\end{aligned} \tag{2}$$

It is assumed that the bias d is zero.

(a) $\mathbf{Q}$ is the 4x4 process noise covariance matrix, $\mathbf{R}$ is the 2x2 measurement noise covariance matrix and $\mathbf{P}$ is the 4x4 error covariance matrix. $\mathbf{Q}$ then describes the noise covariance of the aircraft dynamics. This covariance will represent the model uncertainties and inaccuracies, and is therefore the most important tunable parameter of the Kalman filter. $\mathbf{R}$ is the covariance of the noise measurement, and can be found in a datasheet or estimated from measured data from the sensor. Finally $\mathbf{P}$ is the estimated covariance between our estimate and the true state, and will change over time as we run the Kalman filter.

(b) Rate gyros will measure roll rate and yaw rate. They are typically modelled with a bias and white noise:

$$\omega^b_{\text{gyro}} = \omega^b_{m/n} + b^b_{\text{gyro}} + w^b_{\text{gyro}} \tag{3}$$

The assumption that the noise is additive white Gaussian noise is of course an approximation. The noise amplitude will probably decline for higher frequencies. Furthermore, electromagnetic interference from the microcontroller and other electronic systems onboard the aircraft will impact the rate gyro, which will definitely not act like white noise at all.

(c) The Kalman filter is the optimal linear state estimator if we have a perfect model of the system that is linear, with white process noise and measurement noise and exactly known noise covariances. In this simulation the aircraft dynamics are linear and exactly known, and the noise is (approximately) white Gaussian noise with exactly known covariance. The optimality assumptions therefore hold in simulation.

In reality the dynamics are nonlinear and the parameters of the linearized model are estimated. Furthermore, the noise is definitely not white. For instance wind disturbances are usually approximated as first and second order white-noise driven systems (Dryden gusts) with a steady-state bias added. Also the noise covariances in the system will in reality be unknown and dependent on time, temperature and other factors.

(d) The Kalman filter was implemented as a MATLAB function. See appendix A for the code.

(e) The course loop with the Kalman filter was simulated to provide estimates for β, ϕ, p and r. Process noise was added to the state-space model using the provided code. The process and

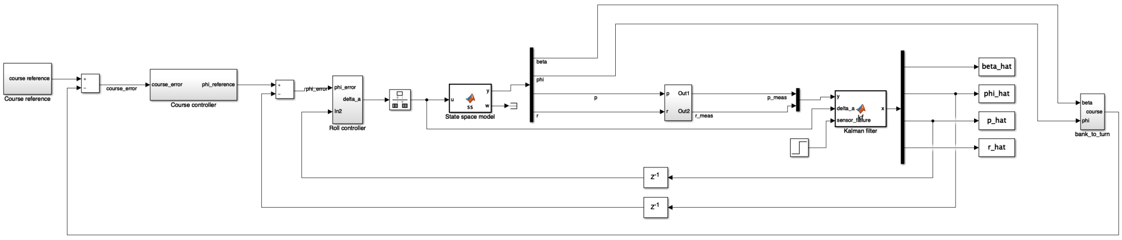

Figure 1: Simulink block diagram of autopilot system.

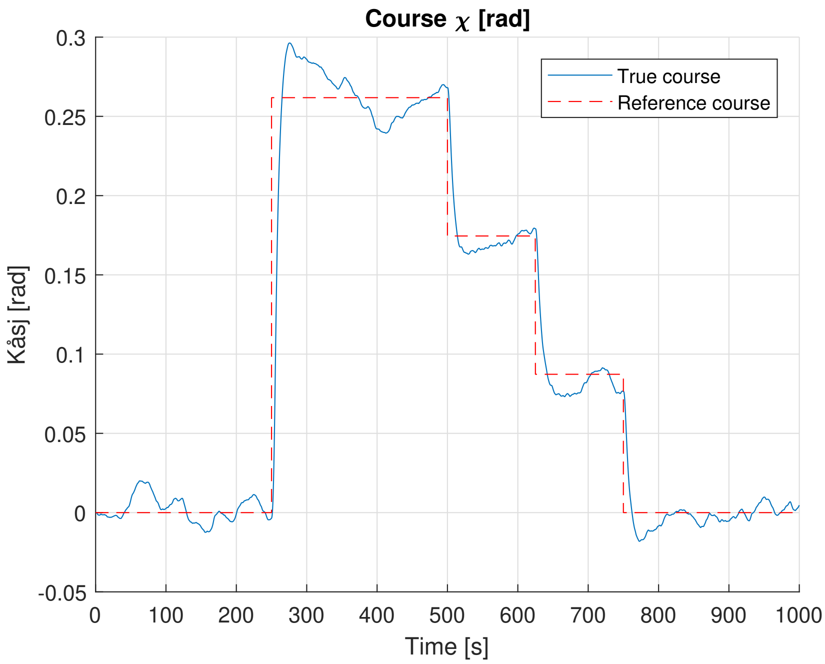

Figure 2: Reference and true course using noisy measurement for roll rate p in the feedback-loop.

measurement noise covariance is:

$$\mathbf{Q} = h * 10^{-6} * \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix} \tag{4}$$

The initial values were chosen as:

$$\mathbf{P}_0 = 10^{-5} * \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad \hat{\mathbf{x}}_0 = 0. \tag{5}$$

The Simulink block diagram can be found in fig. 1 The course reference was given as a series of steps, two 15 degree steps, followed by three more frequent 5 degree steps. The roll rate used in the feedback-loop was given by the noise-contaminated measurements, whereas the states that could not be measured were obtained from the Kalman filter. The resulting course, aileron input, and states are shown in fig. 2, fig. 3, and fig. 4, respectively.

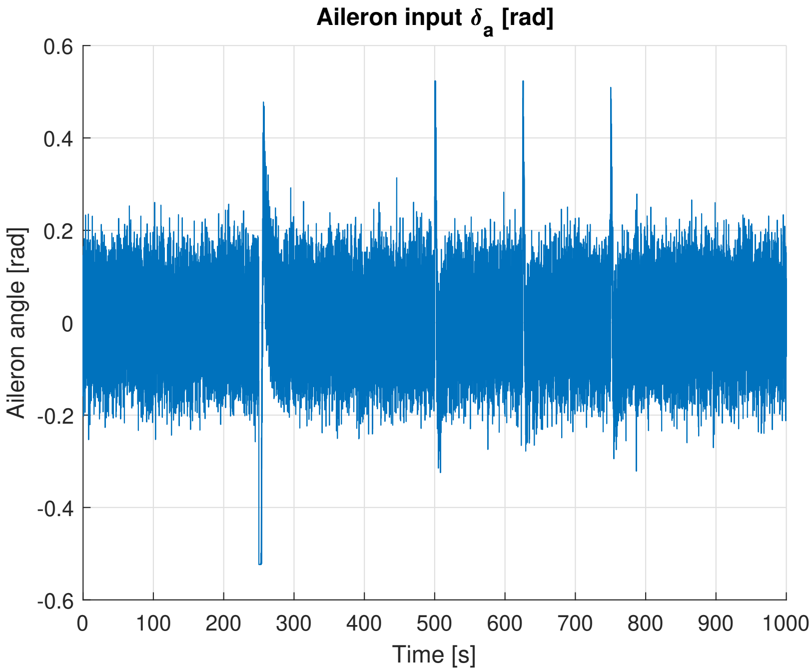

Figure 3: Resulting aileron input angle using noisy measurement for roll rate p in the feedback-loop.

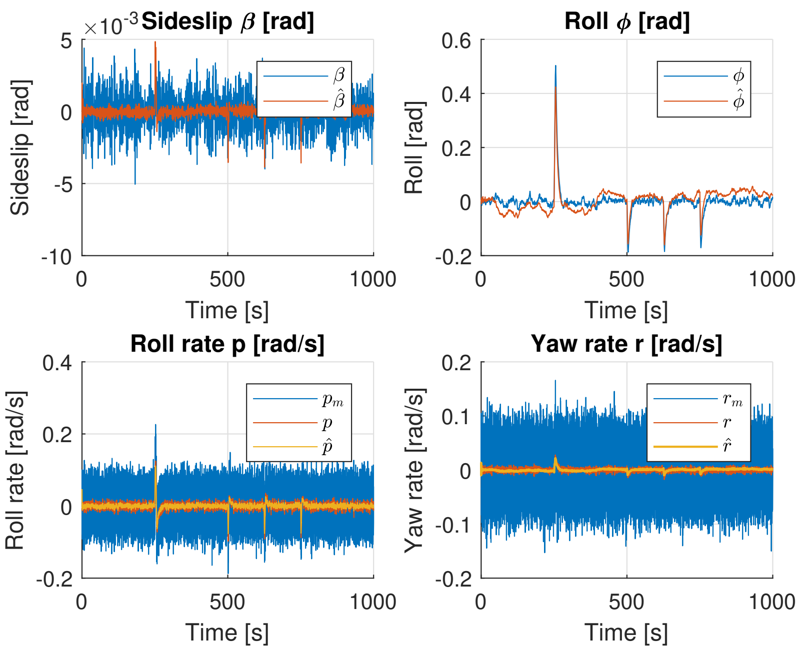

Figure 4: System states using noisy measurement for roll rate p in the feedback-loop. The subscript m indicates state measurement, the caret indicates state estimate.

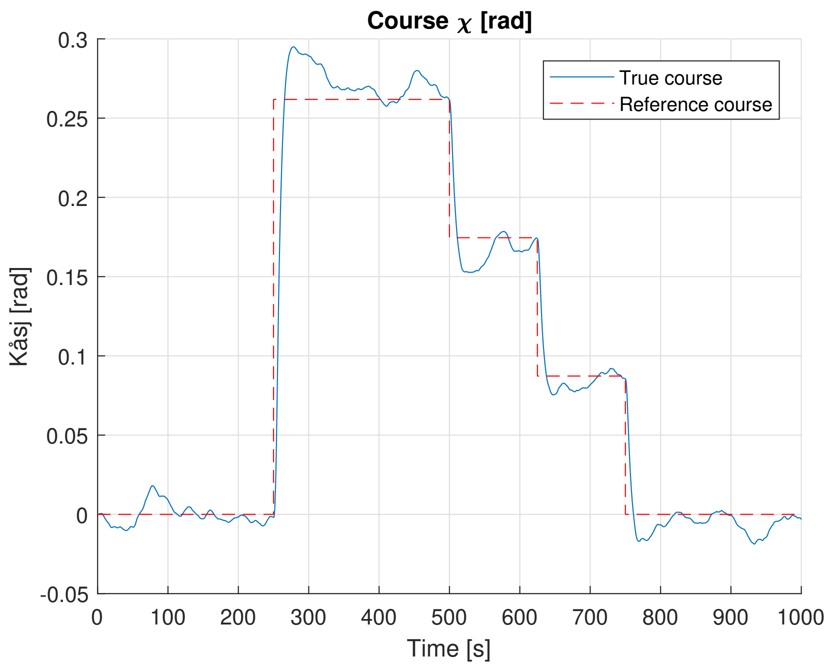

Figure 5: Reference and true course using estimated roll rate p from the Kalman filter in the feedback-loop.

From fig. 2, we note that the true course stays relatively close to the referenced course, although with somewhat oscillatory behaviour. From fig. 4, we note that the rate states p and r are particularly prone to noise. However, the estimates stay remarkably close to the true value, although with high frequency components. A significant consequence of this is seen in fig. 3, where the aileron deflection angle is severely influenced by the high frequency components of the roll rate, resulting in unrealistic and unnecessary aileron behaviour.

(f) The course loop was once again simulated. For this simulation, the roll and roll rate used in the feedback-loop was obtained from the Kalman filter, i.e. state estimates instead of measured states were used. The course reference was the same as in problem (e). The resulting course, aileron input, and states are shown in fig. 5, fig. 6, and fig. 7, respectively.

Comparing fig. 2 and fig. 5 reveals no obvious improvement in the course control. The aileron input, however is significantly different, as evident by comparing fig. 3 to fig. 6. The latter is to a much lesser extent subject to high frequency components, as a result of using estimated roll rate in the feedback-loop. This is a clear advantage as less effort is used in changing the aileron angle. Furthermore, the ailerons are subject to less wear and tear, making the course control algorithm using estimates for control feedback advantageous. When implementing a Kalman filter, the bandwidth of the estimator should be higher compared to that of the control loop, typically 10-20 times faster.

(g) The system was simulated with a complete sensor failure occurring in the middle of the simulation. This was implemented by increasing the entries of $\mathbf{R}$ towards infinity at a certain time, producing useless measurements. The resulting course, aileron input, and states are shown in fig. 8, fig. 9, and fig. 10, respectively.

Comparing the resulting course when a sensory failure occurs as seen in fig. 8 to fig. 5 shows suprisingly little difference. This is due to the fact that the model is very accurate, and that no disturbances such as wind is added to the simulation. In the case of complete system failure, the Kalman filter will rely only on the mathematical model and discard the information provided by the sensors. However, if the system were to be subject to strong winds, for

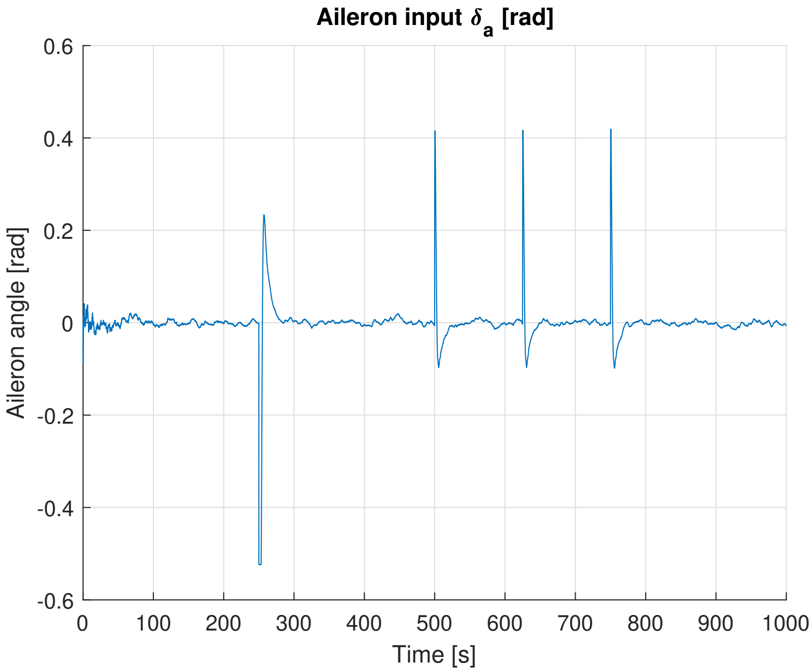

Figure 6: Resulting aileron input angle estimated roll rate p from the Kalman filter in the feedback-loop.

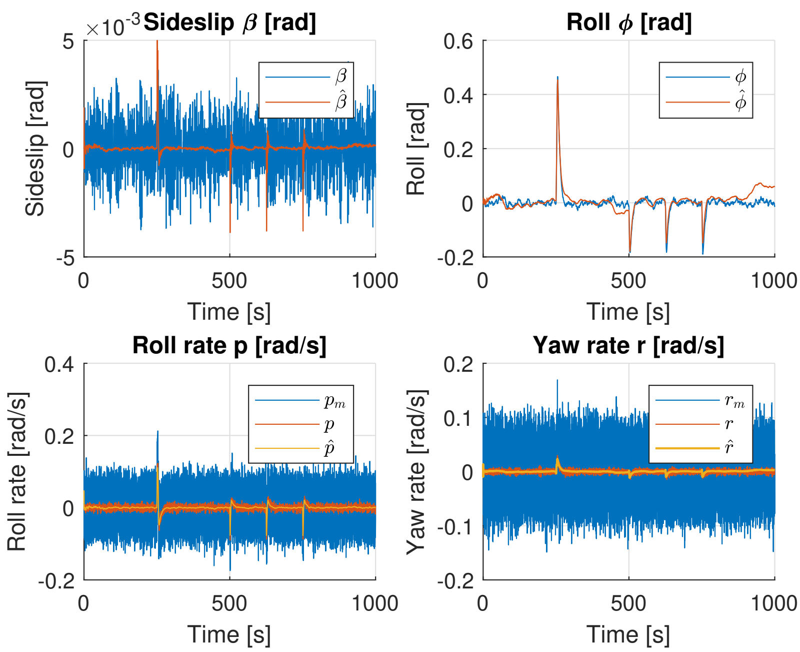

Figure 7: System states using estimated roll rate p from the Kalman filter in the feedback-loop.

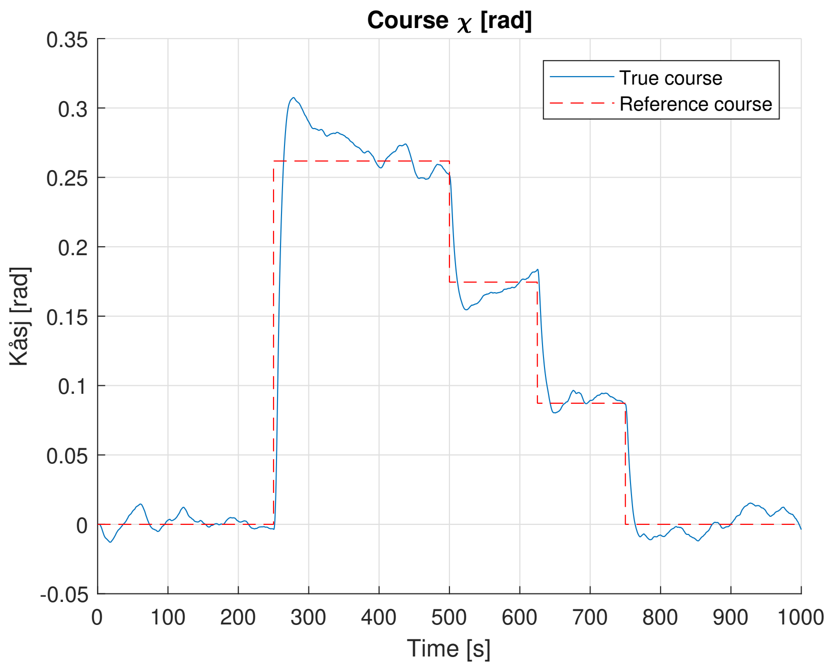

Figure 8: Reference and true course when a sensory failure complete occurs during the flight.

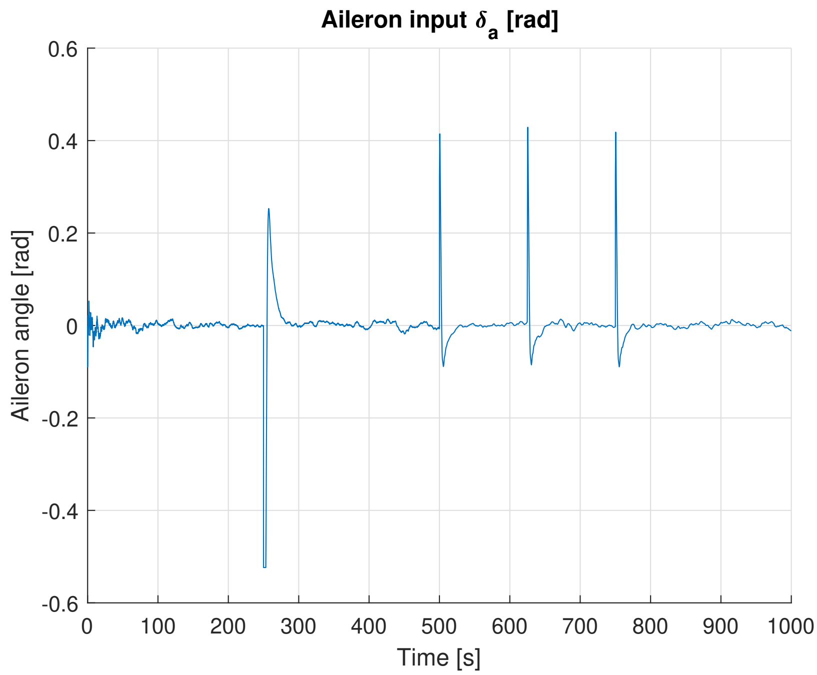

Figure 9: Resulting aileron input angle when a complete sensory failure occurs during flight.

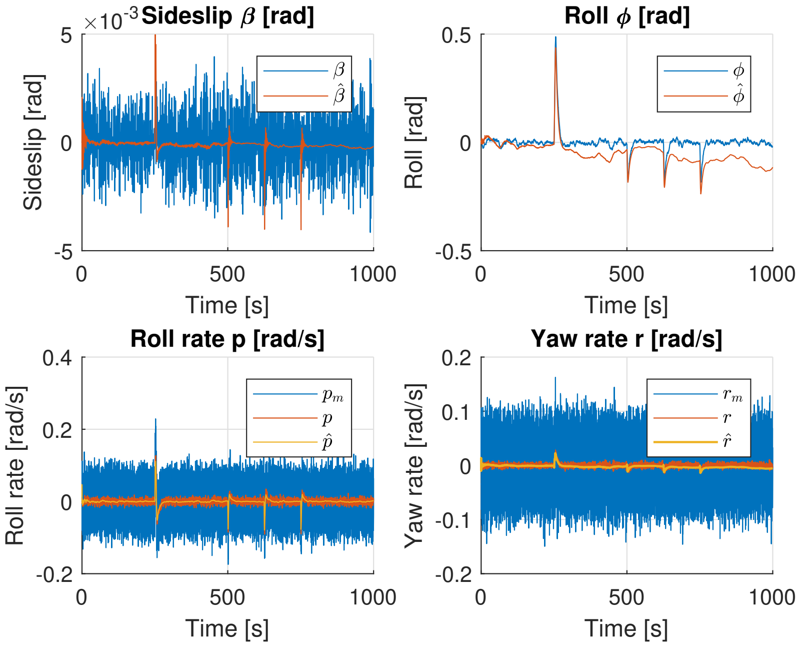

Figure 10: System states when a complete sensory failure occurs during flight.

example, in addition to sensor failure, efficient and safe control would be impossible.

A kalman_filter.m

```
function [x] = kalman_filter(y, delta_a, P_0, x_0, R, Q)
    persistent P;
    persistent x_bar;
    h = 1/100;

    A = [-0.322 0.052 0.028 -1.12;
        0 0 1 -0.001;
        -10.6 0 -2.87 0.46;
        6.87 0 -0.04 -0.32];

    B = [0.002; 0; -0.65; 0.02];
    H =[0 0 1 0; 0 0 0 1];
    E = eye(4);

    if isempty(P)
        P = P_0;
        x_bar = x_0;
    end

    %Kalman gain
    K = P*H'/(H*P*H' + R);

    %Corrector
    P = (eye(4) - K*H)*P*(eye(4) - K*H)' + K*R*K';
    x = x_bar + K*(y - H*x_bar);

    %Predictor
    phi = expm(A*h);
    delta = A\(phi-eye(4))*B;
    gamma = A\(phi-eye(4))*E;

    x_bar = phi*x + delta*delta_a;
    P = phi*P*phi' + gamma*Q*gamma';
end
```