SSY130 - Hand-in 3
Target Tracking Using the Kalman Filter

Lucas Rath

December 18, 2018

student id: 19930425
secret key: 'Vanillish'

Task 1)

Considering the following state equations:

$$\begin{aligned} \dot{x}(t) &= v_x(t) \\ \dot{v}_x(t) &= 0 \\ \dot{y}(t) &= v_y(t) \\ \dot{v}_y(t) &= 0 \end{aligned} \tag{1}$$

and the finite-difference approximation:

$$\dot{x}(t)|_{t=kT} = \frac{x(kT+T) - x(kT)}{T} \tag{2}$$

where T is the sampling time T=0.01 s.
The discrete state-space formulation can be derived by applying (2) in (1), resulting in:

$$\begin{aligned} s(k+1) &= A\ s(k) + w(k) \\ z(k) &= C\ s(k) + v(k) \end{aligned} \tag{3}$$

where:

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{aligned} w(k) &= WGN(0, Q) \\ v(k) &= WGN(0, R) \end{aligned} \tag{4}$$

The matrix C selects the position $x(k)$ and $y(k)$ from the state-space variables that are being measured and are subject to a Gaussian, zero mean, and uncorrelated noise $v(k)$ with covariance matrix R. The process is also subjected to noise $w(k)$, which affects only the states $v_x(k)$ and $v_y(k)$.

Task 2)

The main goal is then to apply a Kalman-Filter to filter the noise from the measured position. Figure 1 shows the measured and the ideal noise-free x and y positions.

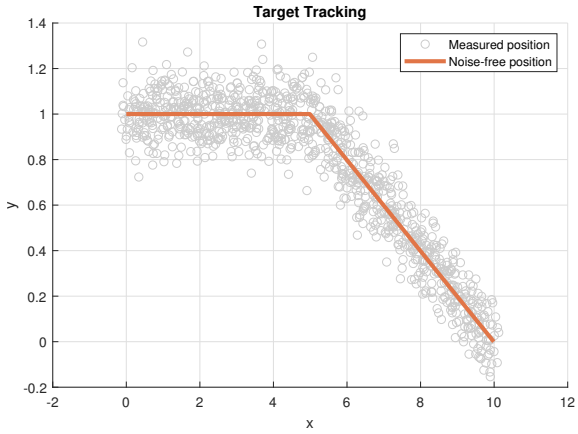


Figure 1: *Measured and noise-free positions x and y*

Task 3)

For sake of simplicity, the Kalman-filter equations, which are presented in the lecture notes, will not be showed here. The Matlab implementation can be seen below.

Listing 1: *Kalman-filter Matlab implementation*

```
% Filter initialization:
Xpred(:,1) = x0;          % Index 1 means time 0
P = P0;                   % Initial covariance matrix (uncertainty)

% Kalman filter iterations:
for t=1:N
    % Filter update based on measurement
    Xfilt(:,t) = Xpred(:,t) + P*C'* ((C*P*C' + R)\(Y(:,t) - C*Xpred(:,t)));
    % Uncertainty update, from (17)
    Pplus = P - P*C'* ((C*P*C' + R)\C*P);
    % Prediction, from (19)
    Xpred(:,t+1) = A * Xfilt(:,t);
    % From (20)
    P = A*Pplus*A' + Q;
end
```

Task 4)

In order to apply the Kalman-filter, we still need to specify the matrices P0, R and Q. Here, we consider zero vector as initial state vector, and P0 $= 10^6 \cdot I$ as initial state covariance matrix.

First, assuming the measurement noise is uncorrelated, then the R matrix is diagonal and each element corresponds to the variance of the noise concerning each measurement. Since we know the real position, we can estimate the covariance matrix R by analyzing statistically the measurement error, as can be seen in figure 2.

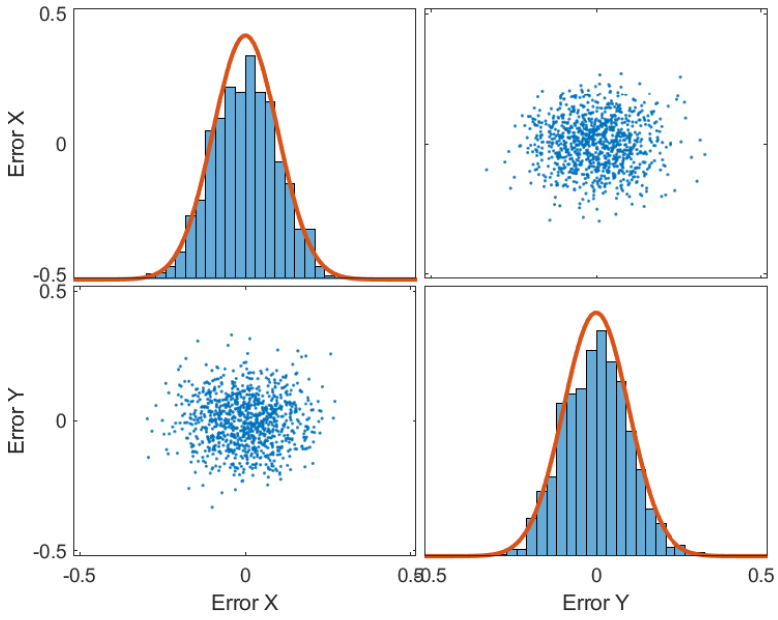


Figure 2: *Measurement error for position x and y*

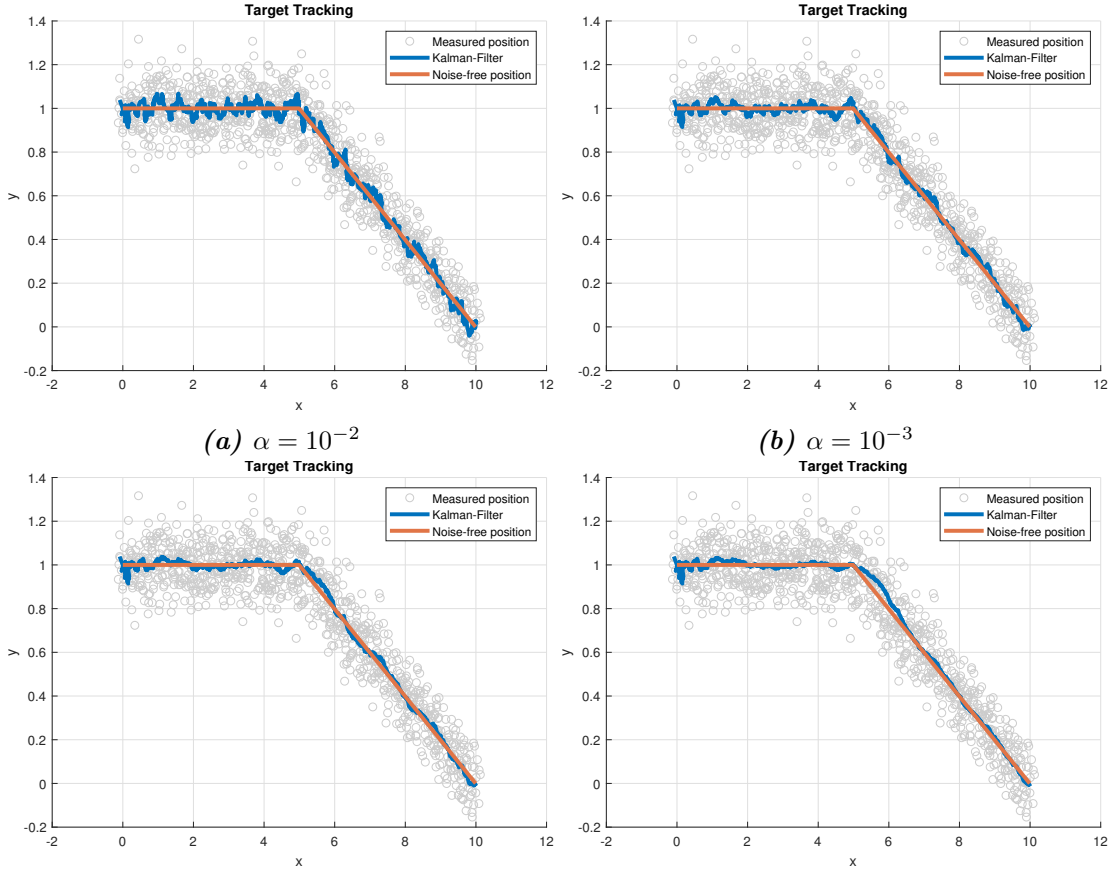
Fitting a zero-mean normal distribution, we get:

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} 9.6 \cdot 10^{-3} & 0 \\ 0 & 9.8 \cdot 10^{-3} \end{bmatrix} \tag{5}$$

In turn, matrix Q is related to the process-noise covariance. However, for our case there is no way to directly determine the matrix values. Assuming that the process noises only influence the velocities and we assume they are uncorrelated for each velocity, we can define Q as:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_{v_x}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_y}^2 \end{bmatrix} = \alpha \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and leave α as a tenable parameter. Figure 3 shows then the position tracking for some α values.



(a) $\alpha = 10^{-2}$ ***(b)*** $\alpha = 10^{-3}$

(c) $\alpha = 10^{-4}$ ***(d)*** $\alpha = 10^{-5}$

Figure 3: *Kalman-Filter position tracking*

Clearly, as we decrease α, we allow only very small variations in the velocity. As a result, the filtered position has much less noise influence, but at the same time it has much slower response to velocity variations. A good α value must then be chosen so it achieves a good balance between noise filtering and response speed. In this case a good α value is around 10^{-4}.

We can also plot the estimated velocities in the x and y directions versus time for the chosen α=10^{-4}, as showed in figure 4. As explained before, this value filter well the noise

but also did not allowed fast variations in speed.

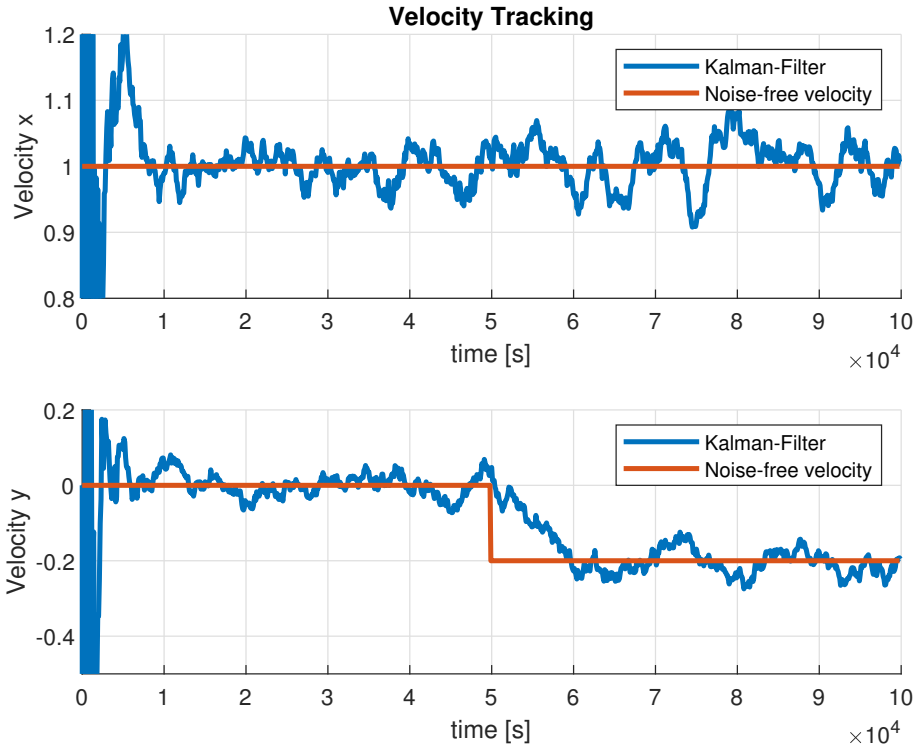


Figure 4: *Velocity tracking for* $\alpha = 10^{-4}$

This demonstrates the power of Kalman-filter. If we had decided to take the derivative of the original signal with noise, the result would have been catastrophic.