kalman滤波应用

Eason

目录

为加深对kalman滤波的理解。本文给出几个kalman滤波的例子。

1 无噪牛顿系统

假设一个牛顿系统，包含位置r，速度v和加速度a. 这个系统可以描述为：

$$\begin{bmatrix} \dot{r} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ v \\ a \end{bmatrix} \tag{1}$$

$$\dot{x} = Ax \tag{2}$$

这是连续系统的描述方式，离散化之后可以表示为：

$$x_{k+1} = Fx_k \tag{3}$$

其中$F = \exp(AT)$，展开有：

$$\begin{aligned} F &= \exp(AT) \\ &= I + AT + \frac{(AT)^2}{2!} + \ldots \\ &= \begin{bmatrix} 1 & T & \frac{(AT)^2}{2!} \\ 0 & 1 & T \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

其中T是离散化的采样时间。

这个系统的kalman滤波描述为：

$$
\begin{aligned}
\hat{x}_k^- &= F\hat{x}_{k-1}^+ \\
P_k^- &= FP_{k-1}^+F^T
\end{aligned}
$$

我们发现在$(k-1)^+$到k^-之间，估计误差的协方差是增加的。因为在$(k-1)^+$和k^-之间我们没有收到任何观察值，估计误差增大也是可以理解的。

现在假定我们对位置进行测量，测量方差为σ^2：

$$
y_k = H_k x_k + v_k \tag{4}
$$

其中

$$
\begin{aligned}
v_k &\sim N(0, R_k) \\
R_k &= \sigma^2
\end{aligned}
$$

我们可以计算出Kalman增益：

$$
K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5}
$$

把K_k，H_k，R_k带入，有：

$$
K_k = \begin{bmatrix} P_{k,11}^- \\ P_{k,12}^- \\ P_{k,13}^- \end{bmatrix} \frac{1}{P_{k,11}^- + \sigma^2} \tag{6}
$$

估计误差协方差：

$$
P_k^+ = P_k^- - K_k H_k P_k^- \tag{7}
$$

如果我们把P_k^-完整的写出来，并替换H_k, K_k，可得：

$$
\begin{aligned}
P_k^+ &= P_k^- - \frac{1}{P_{k,11}^- + \sigma^2} \begin{bmatrix} P_{k,11}^- & 0 & 0 \\ P_{k,12}^- & 0 & 0 \\ P_{k,13}^- & 0 & 0 \end{bmatrix} P_k^- \qquad (8) \\
&= P_k^- - \frac{1}{P_{k,11}^- + \sigma^2} \begin{bmatrix} (P_{k,11}^-)^2 & P_{k,11}^- P_{k,21}^- & P_{k,11}^- P_{k,31}^- \\ P_{k,12}^- P_{k,11}^- & (P_{k,12}^-)^2 & P_{k,12}^- P_{k,31}^- \\ P_{k,13}^- P_{k,11}^- & P_{k,13}^- P_{k,12}^- & (P_{k,13}^-)^2 \end{bmatrix} \qquad (9)
\end{aligned}
$$

通过式~(8)，我们将看到从k^-到k^+，协方差矩阵的迹下降。当我们得到一个新的观测时，我们希望对系统的状态估计会有所改进。也就是说，我们希望协方差降低。式~(8)告诉我们这个协方差的确降低了。

2 matlab实现

在使用matlab实现之前，让我们再次总结一下离散时间kalman滤波器的过程。首先，这个系统的描述为：

$$
\begin{aligned}
x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
E(w_k w_j^T) &= Q_k \delta_{k-j} \\
E(v_k v_j^T) &= R_k \delta_{k-j} \\
E(w_k v_j^T) &= 0
\end{aligned}
$$

Kalman滤波器初始化为：

$$
\hat{x}_0^+ = E(x_0) \tag{10}
$$

$$
P_0^+ = E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \tag{11}
$$

Kalman滤波器的整个过程为：

$$
\begin{aligned}
P_k^- &= F_{k-1}P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= P_k^+ H_k^T R_k^{-1} \\
\hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-) \\
P_k^+ &= (I - K_k H_k)P_k^-(I - K_k H_k)^T + K_k R_k K_k^T \\
&= [(P_k^-)^{-1} + H_k^T R_k^{-1} H_k]^{-1} \\
&= (I - K_k H_k)P_k^-
\end{aligned}
$$

这是个迭代的过程，在迭代过程中$k = 1, 2, \ldots$向前演进。

接下来是matlab代码实现，首先是初始化部分：

```
% sampling interval
T = 5;
% position measurement standard deviation
sigma = 30;
R = sigma^2;
% initial state estimate uncertainty
P0 = [100 0 0; 0 10 0; 0 0 1];
H = [1 0 0];
% state transition matrix
F = [1 T T*T/2; 0 1 T; 0 0 1];
x = [1; 1; 1]; % initial state
xhat = x; % initial state estimate
posArray = [];
xhatArray = [];
yArray = [];
Pplus = P0;
Varminus = [];
Varplus = [P0(1,1)];
KArray = [];
```

T 是采样间隔。=sigma= 是位置估计的标准差。=R= 是位置估计的方差。

然后是迭代计算部分：

```
for k = 1 : N
    % Simulate the system and measurement
    x = F * x;
    y = H * x + sigma * randn;
    % Estimate the state
    Pminus = F * Pplus * F';
    K = Pminus * H' * inv(H * Pminus * H' + R);
    xhat = F * xhat;
    xhat = xhat + K * (y - H * xhat);
    Pplus = (eye(3) - K * H) * Pminus * (eye(3) - K * H)' +
```

```
        K * R * K';
    % Save data for plotting
    posArray = [posArray x(1)];
    xhatArray = [xhatArray xhat];
    yArray = [yArray y];
    Varminus = [Varminus Pminus(1,1)];
    Varplus = [Varplus Pplus(1,1)];
    KArray = [KArray K];
end
```