

Kalman approach for position prediction in an undersampled tracking problem

Salah-Eddine Missri

December 16, 2014

Abstract

Idea : Use a Kalman filter as a predictor, with the prediction loop running at a rate faster than the correction loop (aka update loop) so waypoints can be extrapolated from measurements even though we don't receive data.

1 Kalman Filter summary

1.1 Prediction steps

The first step is to predict the a priori state estimate

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} \quad (1)$$

Where $\hat{\mathbf{x}}_{k|k-1}$ is the estimate of the state vector at iteration k , with iteration k not included. \mathbf{F}_k is the state propagation matrix for iteration k and $\hat{\mathbf{x}}_{k-1|k-1}$ is the estimate of the state vector at iteration $k-1$, with iteration $k-1$ included. Notice that in our case, the command u is modeled as noise that's why it's absent from the equation.

The second step is to predict the a priori estimate covariance

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \quad (2)$$

$\mathbf{P}_{k|k-1}$ is the parameter covariance matrix at iteration k , with iteration k not included. \mathbf{Q}_k is the process covariance matrix (aka state propagation estimate error).

1.2 Update steps

In order to compute the optimal Kalman gain \mathbf{K}_k , we first need to compute the measurement residual $\tilde{\mathbf{y}}_k$ and the residual covariance \mathbf{S}_k

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (3)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (4)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (5)$$

Where \mathbf{z}_k is the measurement. \mathbf{H}_k is the design matrix that maps the state \mathbf{x} into the measurement space and is defined such that $\mathbf{z} = \mathbf{H}\mathbf{x}$. \mathbf{R}_k is the measurement covariance matrix. The last two equations can be compressed into a more compact equation

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (6)$$

Once we have the optimal Kalman gain, we correct (update) the estimates with measurement data. The updated a posteriori state estimate is therefore given by

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (7)$$

And the updated a posteriori estimate covariance is given by

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (8)$$

\mathbf{I} being of course the identity matrix

1.3 Checklist

- Choice of state vector components \mathbf{x}
- Choice of measurements \mathbf{z}
- Description of measurement noise \mathbf{R}
- Choice of design matrix \mathbf{H}
- Choice of state propagation matrix \mathbf{F} and its associated covariance matrix \mathbf{Q}
- Initial setup conditions

1.4 Decoupling the Kalman predictor

Our quadrotor tracking problem can be simplified into a $3D$ position prediction problem. The quadrotor can be considered as a point moving in a $3D$ plane, we consider that it has three degrees of freedom. We observed that the axis x , y and z can be treated separately, therefore we can use our Kalman predictor on the axis separately.

2 Constant velocity motion model

2.1 State vector choice

By definition, the state of a deterministic dynamic system is the smallest vector that fully sums up the past of the system. Therefore, knowledge of the state should allow theoretically to predict the future outputs of the deterministic system.

So we can take as state

$$\mathbf{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (9)$$

Where x is the position that defines an object in the x axis and \dot{x} the associated linear velocity. Referring to classical mechanics equations, position and velocity should allow us to predict the system's behaviour.

2.2 Measurement considerations

The measurement z is considered to be the data received from the other quadrotor. This mainly contains positions and velocities

$$\mathbf{z} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (10)$$

Then, state space and measurement space can be linked through a design matrix \mathbf{H} such that $\mathbf{z} = \mathbf{H}\mathbf{x}$ with

$$\mathbf{H}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 11 \end{pmatrix} \iff \mathbf{H} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (11)$$

Otherwise, the error of the measurement could be modeled with a gaussian profile considering the GPS noise, but we can first suppose that the measurement is perfect and try to implement our Kalman predictor with no measurement noise. So we fix

$$\mathbf{R} = \mathbf{0} \quad (12)$$

As $\mathbf{R} = \mathbf{0}$, this back propagates into the following equations. Equations 4 and 5 become

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T \quad (13)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T)^{-1} = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{H}_k^{T-1} \mathbf{P}_{k|k-1}^{-1} \mathbf{H}_k^{-1} = \mathbf{H}_k^{-1} \quad (14)$$

Equations 7 and 8 become

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{H}_k^{-1} \tilde{\mathbf{y}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{H}_k^{-1} (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) = \mathbf{H}_k^{-1} \mathbf{z}_k \quad (15)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} = (\mathbf{I} - \mathbf{H}_k^{-1} \mathbf{H}_k) \mathbf{P}_{k|k-1} = \mathbf{0} \quad (16)$$

These equations (15 and 16) show that as we consider the measurement to be perfect, each time we correct the prediction using the measurement, we are in fact overwriting the state with the measured state. And the state covariance matrix is reset to $\mathbf{0}$ as we consider the measured state to be perfect.

2.3 State propagation

As we want to predict the position of the targeted quadrotor between two measurements of its actual position, we will consider a constant velocity model. Between two measurements, we have to solve a simple differential equation to extrapolate positions from the last measurement data, this can be done using the Euler method which yields

$$\{ x_k = x_{k-1} + \dot{x}_{k-1} \Delta t \quad (17)$$

Therefore, state propagation \mathbf{F} would be

$$\mathbf{F} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \quad (18)$$

As we assumed a constant velocity model between measurements, a change in velocity causes errors. This can be taken into account through the state propagation covariance matrix \mathbf{Q} that would depend on the maximal accelerations that the quadrotor is able to perform. Once again, using the Euler method, we can get the errors through linear integration of acceleration values. To fit $\sim 99.9\%$ of the case, we consider that the maximal acceleration cases cover 4σ , so

$$\begin{cases} 4\sigma_x = \frac{1}{2}\ddot{x}_{max}\Delta t^2 \\ 4\sigma_{\dot{x}} = \ddot{x}_{max}\Delta t \end{cases} \Leftrightarrow \begin{cases} \sigma_x = \frac{1}{8}\ddot{x}_{max}\Delta t^2 \\ \sigma_{\dot{x}} = \frac{1}{4}\ddot{x}_{max}\Delta t \end{cases} \quad (19)$$

The resulting state propagation error matrix is given by

$$\mathbf{Q} = \begin{pmatrix} \sigma_x & \sigma_v \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_v \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & \sigma_x \sigma_v \\ \sigma_x \sigma_v & \sigma_v^2 \end{pmatrix} = \frac{1}{16} \ddot{x}_{max}^2 \begin{pmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 \end{pmatrix} \quad (20)$$

To set this, we only need one parameter: \ddot{x}_{max} .

2.4 Implementation

initialisation;

while true do

$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}$ // compute the a priori state estimate;

$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$ // compute the a priori estimate covariance;

if new measurement taken then

$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ // update measurement residual;

$\mathbf{K}_k = \mathbf{H}_k^{-1}$ // compute optimal Kalman gain;

$\hat{\mathbf{x}}_{k|k} = \mathbf{H}_k^{-1} \mathbf{z}_k$ // correct the a posteriori state estimate;

$\mathbf{P}_{k|k} = \mathbf{0}$ // correct the a posteriori estimate covariance;

end

 new waypoint = $\hat{\mathbf{x}}_k$ // set current state estimate as waypoint;

end

Algorithm 1: Pseudocode for Kalman predictor implementation with simplified equations

3 Constant acceleration motion model

A constant velocity motion model may be a too simple approximation of the physical model of the quadrotor's motion. Therefore, a more precise model that allow non linear motion can be considered, here we propose a constant acceleration motion model.

3.1 State vector choice

To account for acceleration, we simply add it to our state vector. The previous $4D$ problem becomes a $6D$ problem

$$\mathbf{x} = \begin{pmatrix} x \\ \dot{x} \\ \ddot{x} \end{pmatrix} \quad (21)$$

Where \ddot{x} is the linear accelerations.

3.2 Measurement considerations

As our measurement does not contain acceleration information (see equation 22), we need to guesstimate the acceleration from the measurement. Therefore, we propose to guesstimate acceleration from the velocity. For this, we need to add the previous velocity of the target to the measurement

$$\mathbf{z}_k = \begin{pmatrix} x_k \\ \dot{x}_k \\ \dot{x}_{k-1} \end{pmatrix} \quad (22)$$

That allows us to infer acceleration using the Euler method

$$\left\{ \ddot{x}_k = \frac{\dot{x}_k - \dot{x}_{k-1}}{\Delta t} \right. \quad (23)$$

So the design matrix \mathbf{H} becomes

$$\mathbf{H}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{\Delta t} & -\frac{1}{\Delta t} \end{pmatrix} \iff \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -\Delta t \end{pmatrix} \quad (24)$$

With Δt the time step at which the loop is ran.

As we guesstimate the acceleration, we cannot consider that $\mathbf{R} = \mathbf{0}$. To account for the error on the acceleration, we consider that to fit $\sim 99.9\%$ of the case, we consider that the maximal acceleration cases cover 4σ . So $\sigma = \frac{1}{4}\ddot{x}_{max}$, and \mathbf{R} is given by

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\ddot{x}_{max} \end{pmatrix} \quad (25)$$

3.3 State propagation

Similarly, between two measurements, we have to solve a simple differential equation to extrapolate positions from the last measurement data, this can be done using a 2nd degree Taylor expansion, and a 1st degree Taylor expansion to predict the velocity which yields

$$\begin{cases} x_k = x_{k-1} + \dot{x}_{k-1}\Delta t + \frac{1}{2}\ddot{x}_{k-1}\Delta t^2 \\ \dot{x}_k = \dot{x}_{k-1} + \ddot{x}_{k-1}\Delta t \end{cases} \quad (26)$$

Therefore, state propagation \mathbf{F} would be

$$\mathbf{F} = \begin{pmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \quad (27)$$

As we assumed a constant acceleration model between measurements, a change in velocity causes errors. This can be taken into account through the state propagation covariance matrix \mathbf{Q} that would depend on the maximal accelerations that the quadrotor is able to perform. Once again, using the Euler method, we can get the errors through linear integration of acceleration values. To fit $\sim 99.9\%$ of the case, we consider that the maximal acceleration cases cover 4σ , so

$$\begin{cases} 4\sigma_x = \frac{1}{2}\ddot{x}_{max}\Delta t^2 \\ 4\sigma_{\dot{x}} = \ddot{x}_{max}\Delta t \\ 4\sigma_{\ddot{x}} = 0 \end{cases} \Leftrightarrow \begin{cases} \sigma_x = \frac{1}{8}\ddot{x}_{max}\Delta t^2 \\ \sigma_{\dot{x}} = \frac{1}{4}\ddot{x}_{max}\Delta t \\ \sigma_{\ddot{x}} = 0 \end{cases} \quad (28)$$

We suppose that $\sigma_{\ddot{x}} = 0$, as we do not predict the acceleration, we only measure it and correct it, so the error is accounted in \mathbf{R} as treated previously.

The resulting state propagation error matrix is given by

$$\begin{aligned} \mathbf{Q} &= \begin{pmatrix} \sigma_x^2 & \sigma_x\sigma_{\dot{x}} & 0 \\ \sigma_x\sigma_{\dot{x}} & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{16}\ddot{x}_{max}^2 \begin{pmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & 0 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (29)$$

To set this, we only need one parameter: \ddot{x}_{max} .

3.4 Implementation

initialisation;

while true do

$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1}$ // compute the a priori state estimate;

$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$ // compute the a priori estimate covariance;

if new measurement taken then

$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ // update measurement residual;

$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$ // compute optimal Kalman gain;

$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$ // correct the a posteriori state estimate;

$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$ // correct the a posteriori estimate covariance;

end

 new waypoint = $\hat{\mathbf{x}}_k$ // set current state estimate as waypoint;

end

Algorithm 2: Pseudocode for Kalman predictor implementation with general quations

4 Results & Discussion

Appendices

A Notation index

$\hat{\mathbf{x}}$	State estimate vector
\mathbf{F}	State propagation matrix
\mathbf{P}	State estimate covariance matrix
\mathbf{Q}	Process noise covariance matrix (State propagation error)
\mathbf{z}	Measurement
$\tilde{\mathbf{y}}$	Measurement residual
\mathbf{R}	Measurement covariance matrix
\mathbf{K}	Kalman gain
\mathbf{H}	Design matrix
\mathbf{S}	Residual covariance matrix