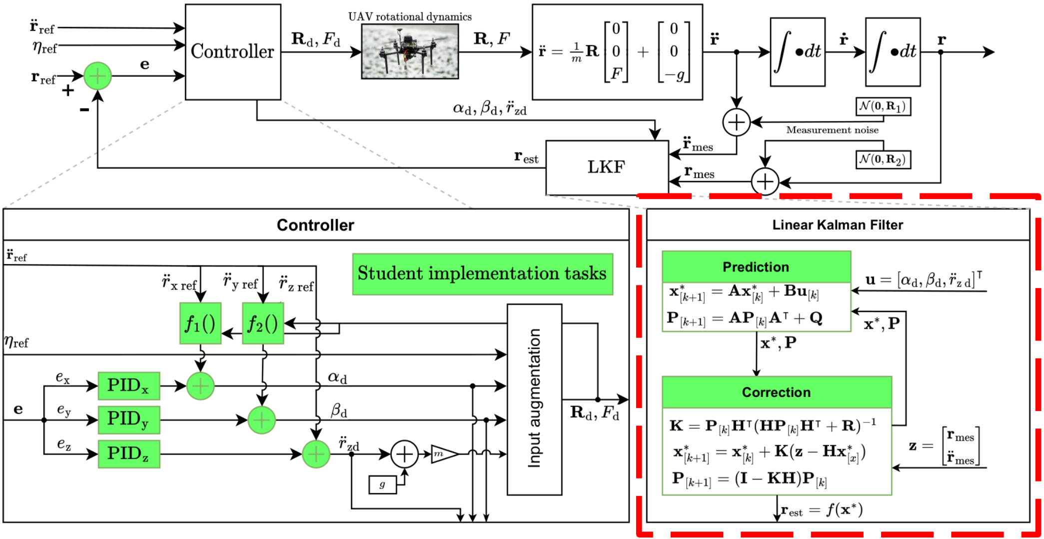
Linear Kalman Filter

B(E)3M33MRS — Aerial Multi-Robot Systems

Martin Jiroušek

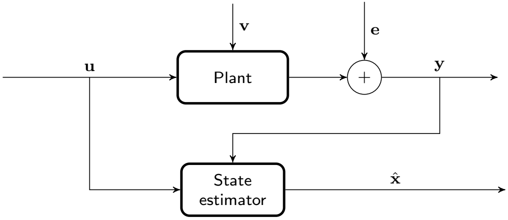
Multi-Robot Systems group, Faculty of Electrical Engineering
Czech Technical University in Prague

Task



Controller
UAV rotational dynamics
LKF
Measurement noise
Controller
Student implementation tasks
PIDx
PIDy
PIDz
Input augmentation
Linear Kalman Filter
Prediction
Correction

Estimation



Nonlinear System

$$\mathbf{x}[t+1] = \mathbf{f}(\mathbf{x}[t], \mathbf{u}[t]) + \mathbf{v}[t]$$
$$\mathbf{y}[t] = \mathbf{h}(\mathbf{x}[t], \mathbf{u}[t]) + \mathbf{e}[t]$$

Noise

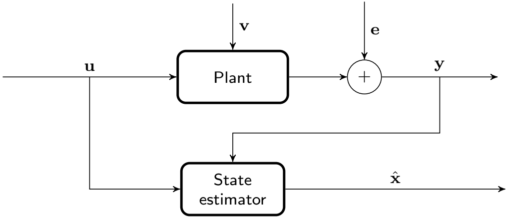
general $p(\mathbf{x}, \mathbf{y})$ from unknown distribution

The Task

Design a function:

$$\hat{\mathbf{x}}[t] = \mathbf{g}(\mathbf{y}[0:t-1], \mathbf{u}[0:t-1])$$

Estimation - LKF Assumptions



Linear System

$$\begin{aligned}\mathbf{x}[t+1] &= \mathbf{A}\mathbf{x}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{v}[t]\\ \mathbf{y}[t] &= \mathbf{C}\mathbf{x}[t] + \mathbf{D}\mathbf{u}[t] + \mathbf{e}[t]\end{aligned}$$

Gaussian White Noise

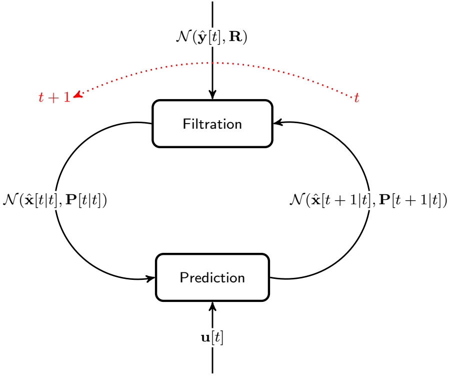
$$p\left(\begin{bmatrix}\mathbf{v}[t]\\ \mathbf{e}[t]\end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{Q} & \mathbf{S}\\ \mathbf{S}^T & \mathbf{R}\end{bmatrix}\right)$$

$$\varepsilon\left(\begin{bmatrix}\mathbf{v}[t_1]\\ \mathbf{e}[t_1]\end{bmatrix}\begin{bmatrix}\mathbf{v}[t_2]\\ \mathbf{e}[t_2]\end{bmatrix}^T\right) = \begin{bmatrix}\mathbf{Q} & \mathbf{S}\\ \mathbf{S}^T & \mathbf{R}\end{bmatrix}\delta(t_1 - t_2)$$

LKF Algorithm

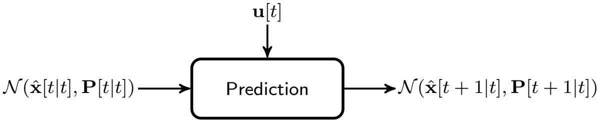
The algorithm can be divided into two separate steps, assuming the process noise $\mathbf{v}(t)$ and measurement noise $\mathbf{e}(t)$ are uncorrelated ($\mathbf{S} = 0$):

- Prediction (Time-Update) – predicts the state using our knowledge of the control action
- Correction (Data-Update) – updates the prediction with the new observation



Prediction

Simulates a single step based on the current control action $\mathbf{u}[t]$ using the system model $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and updates our estimated probability density function (described by the mean and covariance matrix):



where:

- $[t|t]$ denotes the value of a variable at time t, given the data up to the time t
- $[t+1|t]$ denotes the value of a variable at time $t+1$, given the data up to the time t

But how does the PDF of the predicted state evolve?

- Can we describe the evolution of the mean value of the state $(\hat{\mathbf{x}}[t|t] \rightarrow \hat{\mathbf{x}}[t+1|t])$?
- Can we describe the evolution of the covariance matrix of the state $(\mathbf{P}[t|t] \rightarrow \mathbf{P}[t+1|t])$?

We can use our model for that!

Prediction - Derivation

Stochastic dynamics of the system:

$$\mathbf{x}[t+1] = \mathbf{A}\mathbf{x}[t] + \mathbf{B}\mathbf{u}[t] + \mathbf{v}[t]$$

Since $\varepsilon(\mathbf{v}(t)) = 0$ (i.e., the process noise is unbiased), the mean value dynamics is simply:

$$\hat{\mathbf{x}}[t+1|t] = \mathbf{A}\hat{\mathbf{x}}[t|t] + \mathbf{B}\mathbf{u}[t]$$

Let us define the error dynamics as:

$$\tilde{\mathbf{x}}[t+1] = \mathbf{x}[t+1] - \hat{\mathbf{x}}[t+1] = \mathbf{A}\tilde{\mathbf{x}}[t] + \mathbf{v}[t]$$

Using the error dynamics, the development of the covariance is given by:

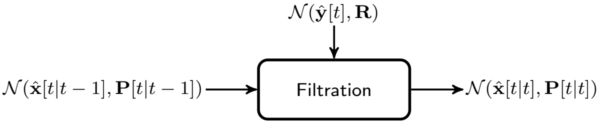
$$\mathbf{P}[t+1|t] = \text{cov}(\mathbf{x}[t+1]) = \varepsilon\left(\tilde{\mathbf{x}}[t+1]\tilde{\mathbf{x}}^T[t+1]\right) = \varepsilon\left(\left(\mathbf{A}\tilde{\mathbf{x}}[t] + \mathbf{v}[t]\right)\left(\mathbf{A}\tilde{\mathbf{x}}[t] + \mathbf{v}[t]\right)^T\right) =$$

$$\underbrace{\varepsilon\left(\mathbf{A}\tilde{\mathbf{x}}[t]\tilde{\mathbf{x}}^T[t]\mathbf{A}^T\right)}_{\mathbf{A}\mathbf{P}[t|t]\mathbf{A}^T} + \underbrace{\varepsilon\left(\mathbf{v}[t]\mathbf{v}^T[t]\right)}_{\mathbf{Q}} + \underbrace{\varepsilon\left(\mathbf{A}\tilde{\mathbf{x}}[t]\mathbf{v}^T[t]\right)}_{0} + \underbrace{\varepsilon\left(\mathbf{v}[t]\tilde{\mathbf{x}}^T[t]\mathbf{A}^T\right)}_{0} = \mathbf{A}\mathbf{P}[t|t]\mathbf{A}^T + \mathbf{Q}$$

These are the relations that we use to calculate the pdf of our prediction

Filtration

We are provided with a predicted p.d.f. and a measurement, and we are supposed to update the p.d.f. such that it reflects the measured data:



But what is the optimal way to estimate a random variable, in our case state $\mathbf{x}[t]$, by observing another random variable, in our case measurement $\mathbf{y}[t]$?

Filtration - Linear Mean Square Estimate

From a statistical point of view, it is reasonable to design the filtration method $\hat{\mathbf{x}}(\mathbf{y})$ such that it minimizes the **Mean Square Error**:

$$J_{MS} = \varepsilon((\mathbf{x} - \hat{\mathbf{x}}_{MS}(\mathbf{y}))^T (\mathbf{x} - \hat{\mathbf{x}}_{MS}(\mathbf{y})))$$

This is, however, generally difficult or even impossible.

We therefore make the following **simplification:**

- Let us assume that the estimate is a linear function of the observation:

$$\hat{\mathbf{x}}_{LMS}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{b}. \tag{1}$$

- and that they are drawn from a joint normal probability density function:

$$p\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xy} \\ \mathbf{P}_{yx} & \mathbf{P}_{yy} \end{bmatrix}\right).$$

Using these assumptions, we can find a closed-form solution for $\hat{\mathbf{x}}_{LMS}(\mathbf{y})$ that minimizes the Mean Square Error.

Filtration - Linear Mean Square Estimate - Derivation

The cost function can be rewritten as:

$$J_{LMS} = \mathcal{E}((\bar{\mathbf{x}} - \mathbf{A}\bar{\mathbf{y}} - \mathbf{b})^T(\bar{\mathbf{x}} - \mathbf{A}\bar{\mathbf{y}} - \mathbf{b})) = \mathrm{tr}\left(\mathcal{E}\left((\bar{\mathbf{x}} - \mathbf{A}\bar{\mathbf{y}} - \mathbf{b})(\bar{\mathbf{x}} - \mathbf{A}\bar{\mathbf{y}} - \mathbf{b})^T\right)\right) =$$
$$\mathrm{tr}\left(\mathbf{P}_{xx} + \mathbf{A}(\mathbf{P}_{yy} + \mu_y\mu_y^T)\mathbf{A}^T + (\mathbf{b} - \mu_x)(\mathbf{b} - \mu_x)^T + 2\mathbf{A}\mu_x(\mathbf{b} - \mu_x)^T - 2\mathbf{A}\mathbf{P}_{yx}\right),$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix.
Since the cost function is convex, the optimum can be obtained by finding the extrema points:

$$\begin{aligned} \frac{\partial J_{LMS}}{\partial \mathbf{A}} &= 2\mathbf{A}(\mathbf{P_{yy}} + \mu_y\mu_y^T) + 2(\mathbf{b} - \mu_x)\mu_y^T - 2\mathbf{P}_{xy} &= 0, \\ \frac{\partial J_{LMS}}{\partial \mathbf{b}} &= 2(\mathbf{b} - \mu_x) + 2\mathbf{A}\mu_y &= 0. \end{aligned} \tag{2}$$

The set of equations (2) is solved by:

$$\mathbf{A} = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}, \qquad \mathbf{b} = \mu_x - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mu_y \tag{3}$$

By substituting (3) into (1), we obtain the **closed-form solution for the LMS estimate:**

$$\boxed{\hat{\bar{\mathbf{x}}}_{LMS}(\bar{\mathbf{y}}) = \mu_x + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\bar{\mathbf{y}} - \mu_y)} \tag{4}$$

and the **covariance of the LMS estimate:**

$$\boxed{\mathbf{P}_{\hat{\bar{\mathbf{x}}}_{LMS}} = \mathcal{E}\left(\left(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}_{LMS}(\bar{\mathbf{y}})\right)\left(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}_{LMS}(\bar{\mathbf{y}})\right)^T\right) = \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx}} \tag{5}$$

Filtration - Joint p.d.f.

However, the LMSE equations (4), (5) contain the mean and covariance of the measurement, which are not explicitly given. Therefore, we take the predicted state

$$p\left(\mathbf{x}[t]|\mathcal{D}^{t-1}\right) = \mathcal{N}\left(\hat{\mathbf{x}}[t|t-1], \mathbf{P}[t|t-1]\right)$$

and use our measurement model

$$\mathbf{y}[t] = \mathbf{C}\mathbf{x}[t] + \mathbf{e}[t]$$

to calculate the predicted measurement p.d.f.:

$$\hat{\mathbf{y}}[t|t-1] = \mathbf{C}\hat{\mathbf{x}}[t|t-1]$$

$$\mathbf{P}_{\mathbf{y}}[t|t-1] = \varepsilon\left(\tilde{\mathbf{y}}[t|t-1]\tilde{\mathbf{y}}^T[t|t-1]\right) = \varepsilon\left(\left(\mathbf{C}\tilde{\mathbf{x}}[t|t-1] + \mathbf{e}[t]\right)\left(\mathbf{C}\tilde{\mathbf{x}}[t|t-1] + \mathbf{e}[t]\right)^T\right) = \mathbf{C}\mathbf{P}[t|t-1]\mathbf{C}^T + \mathbf{R}$$

Similarly, we can calculate the cross-covariance matrices $\mathbf{P}_{\mathbf{xy}}$ and $\mathbf{P}_{\mathbf{yx}}$, and obtain the joint p.d.f. as:

$$p\left(\begin{bmatrix} \mathbf{x}[t] \\ \mathbf{y}[t] \end{bmatrix} \middle| \mathcal{D}^{t-1}\right) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}[t|t-1] \\ \mathbf{C}\hat{\mathbf{x}}[t|t-1] \end{bmatrix}, \begin{bmatrix} \mathbf{P}[t|t-1] & \mathbf{P}[t|t-1]\mathbf{C}^T \\ \mathbf{C}\mathbf{P}[t|t-1] & \mathbf{C}\mathbf{P}[t|t-1]\mathbf{C}^T + \mathbf{R} \end{bmatrix}\right) \tag{6}$$

That's all we need to know to use the LMS estimate!

Filtration - Finalization

By plugging the provided p.d.f. (6) into the LMS estimate equations (4), (5), we obtain the **Filtration step**:

$$p\left(\begin{bmatrix} \mathbf{x}[t] \\ \mathbf{y}[t] \end{bmatrix} \middle| \mathcal{D}^{t-1}\right) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}[t|t-1] \\ \mathbf{C}\hat{\mathbf{x}}[t|t-1] \end{bmatrix}, \begin{bmatrix} \mathbf{P}[t|t-1] & \mathbf{P}[t|t-1]\mathbf{C}^T \\ \mathbf{C}\mathbf{P}[t|t-1] & \mathbf{C}\mathbf{P}[t|t-1]\mathbf{C}^T + \mathbf{R} \end{bmatrix}\right)$$

↓

$$\hat{\hat{\mathbf{x}}}_{LMS}(\bar{\mathbf{y}}) = \mu_x + \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}(\bar{\mathbf{y}} - \mu_y)$$
$$\mathbf{P}_{\hat{\hat{\mathbf{x}}}_{LMS}} = \mathbf{P}_{xx} - \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1}\mathbf{P}_{yx}$$

↓

$$\hat{\mathbf{x}}[t|t] = \hat{\mathbf{x}}[t|t-1] + \mathbf{L}[t](\mathbf{y}[t] - \mathbf{C}\hat{\mathbf{x}}[t|t-1] - \mathbf{D}\mathbf{u}[t])$$
$$\mathbf{P}[t|t] = \mathbf{P}[t|t-1] - \mathbf{L}[t]\mathbf{C}\mathbf{P}[t|t-1]$$
$$\mathbf{L}[t] = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} = \mathbf{P}[t|t-1]\mathbf{C}^T\left(\mathbf{C}\mathbf{P}[t|t-1]\mathbf{C}^T + \mathbf{R}\right)^{-1}$$

Overview

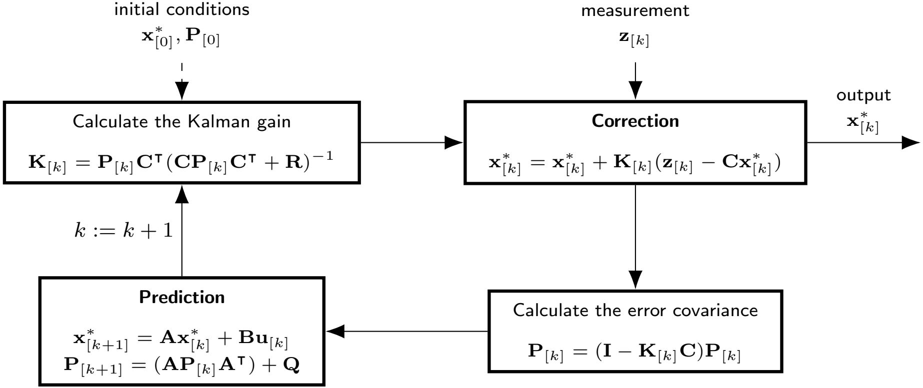
Prediction:

$$\hat{\mathbf{x}}[t+1|t] = \mathbf{A}\hat{\mathbf{x}}[t|t] + \mathbf{B}\mathbf{u}[t]$$
$$\mathbf{P}[t+1|t] = \mathbf{A}\mathbf{P}[t|t]\mathbf{A}^T + \mathbf{Q}$$

Filtration:

$$\hat{\mathbf{x}}[t|t] = \hat{\mathbf{x}}[t|t-1] + \mathbf{L}[t](\mathbf{y}[t] - \mathbf{C}\hat{\mathbf{x}}[t|t-1] - \mathbf{D}\mathbf{u}[t])$$
$$\mathbf{P}[t|t] = \mathbf{P}[t|t-1] - \mathbf{L}[t]\mathbf{C}\mathbf{P}[t|t-1]$$
$$\mathbf{L}[t] = \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1} = \mathbf{P}[t|t-1]\mathbf{C}^T\left(\mathbf{C}\mathbf{P}[t|t-1]\mathbf{C}^T + \mathbf{R}\right)^{-1}$$

Implementation



Takeaway

Optimality

- The LKF is a Linear Mean Square Estimator
- In the case of Gaussian white noise, the LKF is optimal
- In the case of general noise, the LKF is optimal among linear methods

Practical aspects

- Tuning by setting the covariance matrices of noise ($\mathbf{Q}$, $\mathbf{R}$)
- Large $\mathbf{Q} \rightarrow$ trust in measurement, resulting in a noisy, fast response
- Large $\mathbf{R} \rightarrow$ trust in model, resulting in a smooth, slow response