Weekly Summary

Shiqi Duan

University of Science and Technology of China

sqduan@mail.ustc.edu.cn

December 2, 2019

Overview

Kalman filter Review

What is a kalman filter

The kalman filter is an estimator for what is called the linear quadratic estimator.
Used for estimating dynamic process perturbed by white noise.

- It's a mathematical tool
- It's a program
- It's a consistent statistical characterization of an estimation problem

Mathematical form

Estimation ("Predict with prior knowledge")

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \tag{1}$$

Update based on measurement ("Correct")

First we calculate the gains

$$\begin{aligned} &\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \\ &\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ &\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \end{aligned} \tag{2}$$

$\tilde{\mathbf{y}}_k$ is the measurement residual; $\mathbf{S}_k$ is the covariance matrix of measurement residual; $\mathbf{K}_k$ is the optimal Kalman gain

Mathematical form

Then we use them to update variable $\mathbf{x}$ and $\mathbf{P}$

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (I - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_{k|k-1}\end{aligned} \tag{3}$$

Main components

- Prediction
- Receding horizon
- Modelling
- Performance index
- Degree of freedom
- Constraint handling
- Multivariable

Main components

Prediction

Prediction horizen $>$ settling time

Prediction

Continually update our predictions and decision making using the most recent target and measurement data.

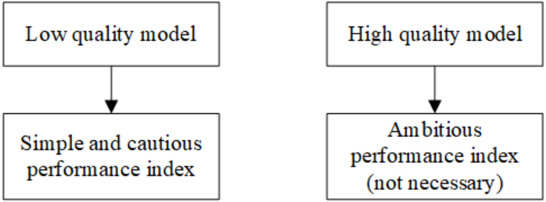
Modelling

The simplest model gives accurate prediction is usually the best.

Main components

Performance index

- What is the performance index for?
  The performance index is a numerical definition of what is the best.
- How to design the performance index?
  One should only increase the complexity where the benefits is clear.

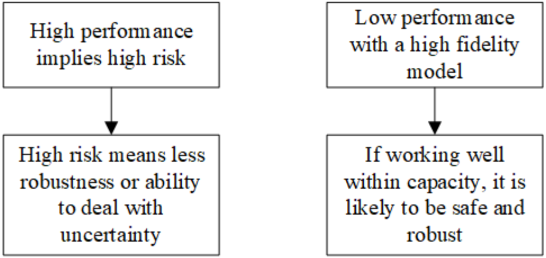


Main components

Typically quadratic performance is used, because:

1. It give us well conditioned optimization
2. Unique minimum
3. Smooth behaviours (unlike 1-norm or inf-norm)

- How to balance optimal and robust performance?



Main components

Degree of freedom

The useful num of DOF is related to the prediction accuracy.

In MPC, an ill-posed performance index means **a low prediction horizon compared to the system dynamic and use numerous DOF to optimal tracking with that horizon.**

The useful num of DOF is related to the prediction accuracy

Main components

Constraint handling

One major advantage of MPC is it **embeds constraints to strategy**, which means that it will not propose input flows that allow overshooting, the response time may become slower, but much more safer.

Why we use MPC

- Intuitive concept, easy to understand and implement
- Systematic handling of constrains
- Can handle MIMO and dead-time without modification
- Feed forward to make good use of future target information
- Handling challenging dynamics (unlike PID)

Model requirements

Simple model?

Simple manipulation and algebra requires linear models. If these are good enough, **use linear models in MPC**.

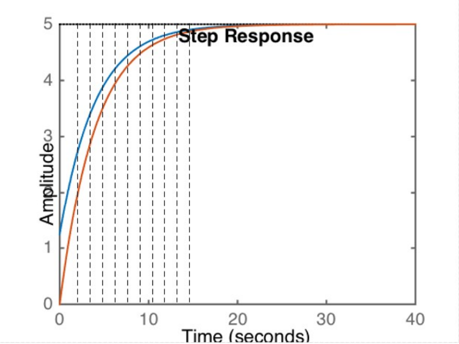
Discrete or continuous?

Decision making requires processing time, there for, **MPC laws are implemented in discrete time**.

Model requirements

What sample rate?

A typical argument is that one wants around **10 sample points** within a typical response (settling time or rise time)



Modelling

Model form

- State space model
- Transfer function model
- Step response model
- Independent model

Basic concepts of prediction

Given data at time k, we can determine the data at $k+1$

$$\left.\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + d_k \end{aligned}\right\} \Rightarrow \begin{aligned} y_{k+1} &= Cx_{k+1} + d_{k+1} \\ y_{k+1} &= CAx_k + CBu_k + d_{k+1} \end{aligned} \tag{4}$$

Nomally we can assume that $d_k = d_{k+1}$.

Basic concepts of prediction

Splitting predictions

We can saperate the predictions into known and unknown part

$$
\begin{aligned}
y_{k+n|k} = & CA^n x_k + d_k + \\
& C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \cdots + Bu_{k+n-1|k}\right)
\end{aligned} \tag{5}
$$

- known: $CA^n x_k + d_k$, based on current and past measurement
- unknown: $C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \cdots + Bu_{k+n-1|k}\right)$, based on future input choices.

Matrix form of ss prediction

Rewrite the prediction in matrix form:

$$\mathbf{x}_{k+1} = \begin{bmatrix} A\mathbf{x}_k \\ A^2\mathbf{x}_k \\ \vdots \\ A^n\mathbf{x}_k \end{bmatrix} + \begin{bmatrix} Bu_{k|k} \\ ABu_{k|k} + Bu_{k+1|k} \\ \vdots \\ A^{n-1}Bu_{k|k} + \cdots + ABu_{k+n-2|k} + Bu_{k+n-1|k} \end{bmatrix} \tag{6}$$

make seperation, we get

$$\mathbf{x}_{k+1} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B & A^{n-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ \vdots \\ u_{k+n-1|k} \end{bmatrix} \tag{7}$$

Matrix form of ss prediction

- State prediction

$$\mathbf{x}_{k+1} = P_x \mathbf{x}_k + H_x \mathbf{u}_k \tag{8}$$

- Output predictions

$$\mathbf{y}_{k+1} = P \mathbf{x}_k + L d_k + H \mathbf{u}_k \tag{9}$$

Unfinished Work

- Try to design a kalman filter for the LQG problem
- Continue on MPC and design a simple MPC demo

References

Mohinder S. Grewal (2014)
Kalman Filtering: Theory and Practice Using MATLAB
Wiley-IEEE Press

Welch G (1995)
An introduction to the Kalman filter
Journal