5 - Graybox system Identification

Kalman filter is a method for software sensor but can be used also for graybox system identification.

Problem setting of a graybox system identification: we have a model built in a whitebox way

$$S: \begin{cases} x(t+1) = f(x(t), u(t); \vartheta) + v_1(t) \\ y(t) = h(x(t), u(t); \vartheta) + v_2(t) \end{cases}$$

$f()$ and $h()$ are known mathematical functions having some unknown parameters ϑ (with a physical meaning).
Problem: estimate $\hat{\vartheta}_N$ from a measured dataset
$\{u(1) \dots u(N)\}$ $\{y(1) \dots y(N)\}$

We can solve this problem using kalman filter (Trick: state extension)

$$S: \begin{cases} x(t+1) = f(x(t), u(t); \vartheta(t)) + v_1(t) \\ \vartheta(t+1) = \vartheta(t) + v_\vartheta(t) \\ y(t) = h(x(t), u(t); \vartheta(t)) + v_2(t) \end{cases} \qquad x_E(t) = \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix}$$

Focus on new state equation: $\vartheta(t+1) = \vartheta(t) + v_\vartheta(t)$

- it is a "fictitious" equation (we need it since each states need its state equation)
- $\vartheta(t+1) = \vartheta(t)$ is the core dynamical relationship. This is the equation of a constant quantity (ok since we want to estimate a constant value of parameter). $\vartheta(t+1) = \vartheta(t)$ is NOT asymptotically stable (No problem since kF can deal with stable/unstable systems)
- we need to add this noise otherwise kF would trust too much the initial condition and not search for the correct value

The problem is the noise definition. Usual assumption on $v_\vartheta(t)$

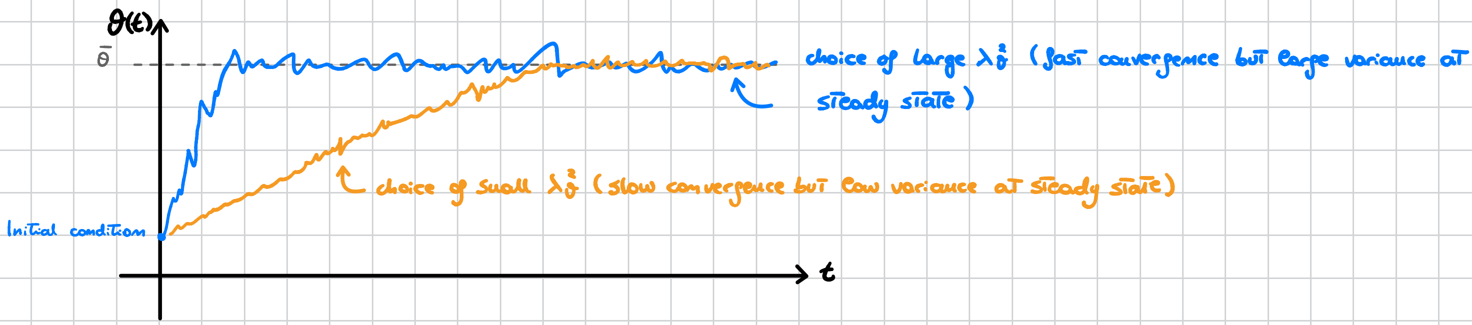
$$v_\vartheta(t) \sim WN(0, V_\vartheta) \qquad v_1, v_2 \perp v_\vartheta$$

Typical simplification of V_ϑ: all the features of V_ϑ are condensed into 1 single parameter λ_ϑ^2

This becomes a Tuning parameter

$$V_\vartheta = \begin{bmatrix} \lambda_\vartheta^2 & 0 & 0 \\ 0 & \lambda_\vartheta^2 & 0 \\ 0 & 0 & \lambda_\vartheta^2 \end{bmatrix}$$

this parameter λ_ϑ^2 is empirically tuned



Best choice? Depends on the length of the dataset and if the true parameter does change in time and we want to keep the algorithm always on.

Method very useful especially if used at runtime (ϑ is actually time-varying). KF in this case provides at the same time both state estimation $\hat{x}(t|t)$ and parameter

Example.

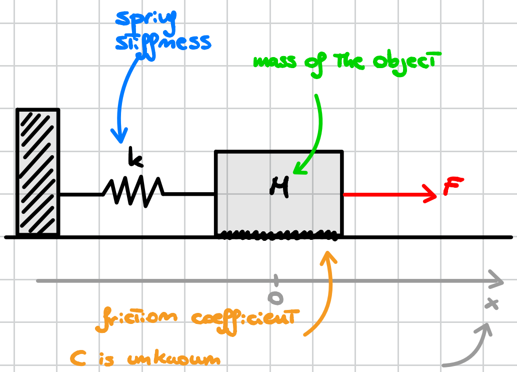
- 3 sensor
- 10 states
- 15 parameters

$3/25$ is too much

- 3 sensor
- 5 states
- 2 parameters

$3/7$ more reasonable

Example:



White Box equation of the dynamics of the system

$$\begin{cases} \ddot{x}(t)M = -k\,x(t) - C\,\dot{x}(t) + F(x) \\ y(t) = x(t) \end{cases} \qquad \vartheta = C$$

State space equation and discretization:

$$\begin{cases} x_1(t+1) = x_1(t) + \Delta x_2(t) \\ x_2(t+1) = -k\,\Delta/M\, x_1(t) + (1 - C\Delta/M)\, x_2(t) + \frac{\Delta}{M} F(x) \\ y(t) = x_1(t) \end{cases}$$

State extension: $x_3(t) = C(t)$. The overall system is

$$\begin{cases} x_1(t+1) = x_1(t) + \Delta x_2(t) + v_{11}(t) \\ x_2(t+1) = -k\,\Delta/M\, x_1(t) + (1 - x_3(t)\cdot\Delta/M)\, x_2(t) + \frac{\Delta}{M} F(x) + v_{12}(t) \\ x_3(t+1) = x_3(t) + v_{13}(t) \\ y(t) = x_1(t) + v_2(t) \end{cases}$$

Assume:

$$v_1(t) \sim WN(0, V_1)$$
$$v_2(t) \sim WN(0, V_2)$$
$$V_{12} = 0$$

$$V_1 = \begin{bmatrix} \lambda_{11}^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_\vartheta^2 \end{bmatrix} \qquad V_2 = \lambda_2^2$$

Ready for kalman filter application to get both state and friction

System typically becomes non-linear (due to true state multiplication $x_3(t)\cdot x_2(t)$)

gray box identification can be done in a more classical way with a parametric optimization

$$u(t) \longrightarrow \boxed{M(\vartheta)} \longrightarrow y(t) \quad \Rightarrow \quad \begin{matrix}\{u(1) \;\dots\; u(N)\} \\ \{y(1) \;\dots\; y(N)\}\end{matrix} \quad \Rightarrow \quad J_N(\vartheta) = \underbrace{\textstyle\sum_1^N (y(t) - M(\vartheta)u(t))^2}_{\text{simulation error}}$$

In many applications, we may know one possible model of a system $M(\vartheta)$ which depends on some physical parameters. In state space, the model becomes (as function of ϑ):

$$\begin{cases} \dot{x}(t) = A(\vartheta)\,x(t) + B(\vartheta)\,u(t) \\ y(t) = C(\vartheta)\,x(t) + D(\vartheta)\,u(t) \end{cases}$$

Our goal is to estimate ϑ from some input/output data $D = (u(t), y(t))_{t=0\dots N}$

Simulation Error Method: we can select one particular $\bar{\theta}$ and run a simulation of the model $M(\bar{\theta})$ and check the difference between the simulated output $y_{sim}(t;\bar{\theta})$ and the real output $y(t)$

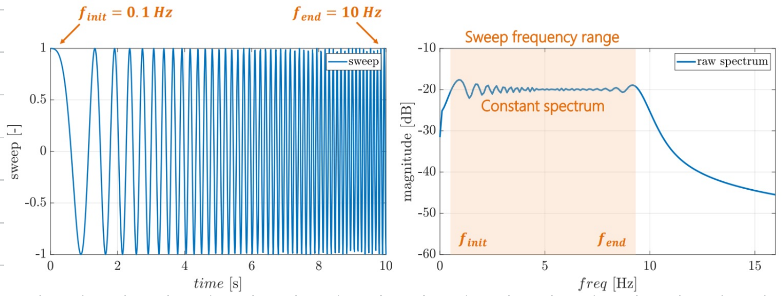
$$\theta_{opt} = \arg\min_{\theta \in \Theta} RMS\left(y(t) - y_{sim}(t;\theta)\right) \qquad y_{sim}(t;\theta) = M(\theta)\,u(t)$$

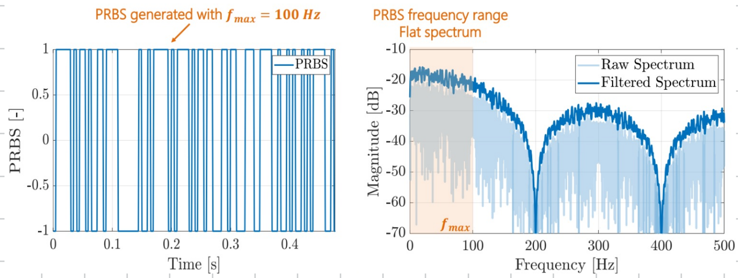
where RMS(·) stands for Root Mean Square and Θ is the set of all possible θ; $y_{sim}(t;\theta)$ is the simulation of the system for a given θ.

In general, the optimization is non-convex and cannot be written linearly in the parameter vector θ. The dataset D must be informative: we do not need any dataset but one which are able to excite all the system dynamics of interest. The identified parameters directly reflect the quality of data.

How to excite all the system dynamics can differ a lot from system to system, but in general the most used signals for identification are:

- **Sinesweep** (or multiple sine experiment): we can span all the frequency range which is important for identifying the system. The dataset contains useful info both in time/frequency domain.





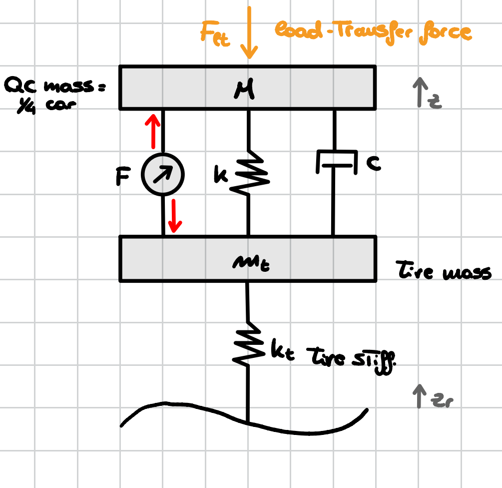
- **Pseudorandom Binary Signal**: it's a signal which only has 2 values ($\pm A$) and commutes between them at random. From a spectral perspective, the PRBS has a constant spectrum up to its cutting frequency. PRBS data contains a lot of information in frequency domain

Usually, the optimization is performed on at least one dataset and then validated on at least a qualitatively different dataset (the experiment is not just a repetition of the optimization dataset). The validation analysis must be performed both in time/frequency domain

---

Example: vehicle suspension identification

We can model a single vehicle suspension using the QC-model for an active suspension $\theta = [M\ k\ c]^T$



Body mass balance: $M\ddot{z} = F - F_{lt} - k(z - z_t) - c(\dot{z} - \dot{z}_t)$

Wheel mass balance: $m_t\ddot{z}_t = -F + k(z - z_t) + c(\dot{z} - \dot{z}_t) - k_t(z_t - z_r)$

State-Space model:

$$x(t) = \begin{bmatrix} \dot{z} \\ z \\ \dot{z}_t \\ z_t \end{bmatrix} \qquad y(t) = \begin{bmatrix} \dot{z} \\ z \\ \dot{z}_t \\ z_t \\ \ddot{z} \end{bmatrix} \qquad u(t) = F(t) \qquad d(t) = \begin{bmatrix} z_r(t) \\ F_{lt}(t) \end{bmatrix}$$

State space continuous-time model:

$$\begin{cases} \dot{x}(t) = Ax(t) + B_u u(t) + B_d d(t) \\ y(t) = Cx(t) + D_u u(t) + D_d u(t) \end{cases}$$

$$A = \begin{bmatrix} -c/M & -k/M & c/M & k/M \\ 1 & 0 & 0 & 0 \\ c/m_t & k/m_t & -c/m_t & -(k_t + k)/m_t \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B_u = \begin{bmatrix} 1/M \\ 0 \\ -1/m_t \\ 0 \end{bmatrix} \qquad B_d = \begin{bmatrix} 0 & -1/M \\ 0 & 0 \\ k_t/m_t & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -c/M & -k/M & c/M & k/M \end{bmatrix} \qquad D_u = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/M \end{bmatrix} \qquad D_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1/M \end{bmatrix}$$

Experiments:

1) Vehicle is still: $\dot{z}_r = 0$ ($d(t) = 0$)
2) No pitch and roll movement: $F_{lt} = 0$ ($d(t) = 0$)
3) $F(t)$ is a sinesweep