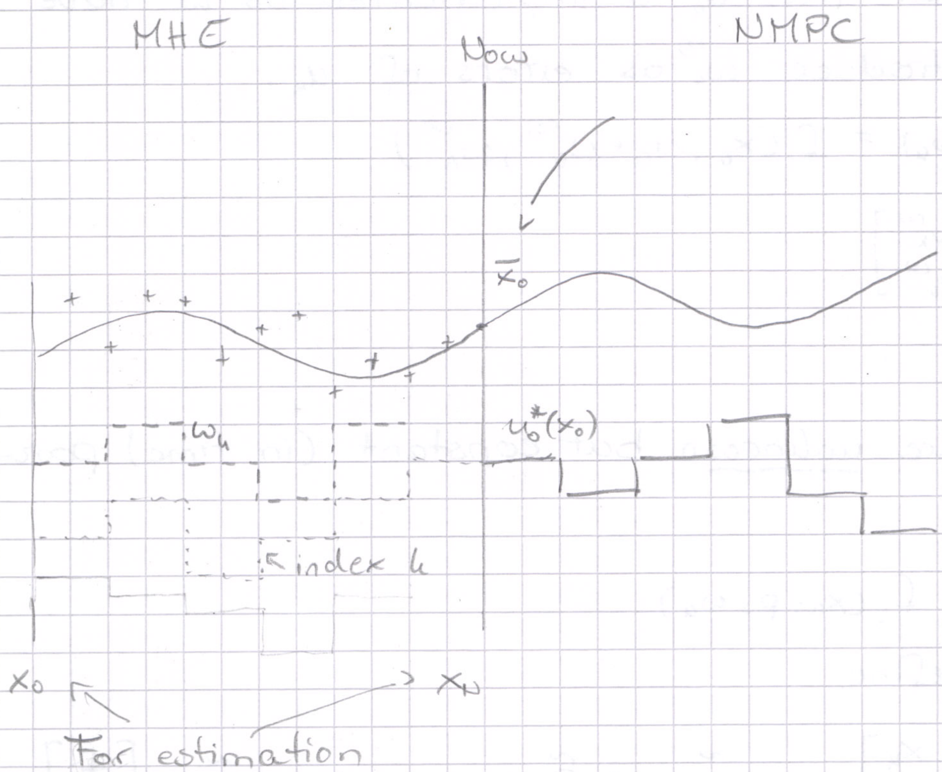
11. Moving horizon estimation

Aim: State and parameter estimation from measurements.

For online estimation: Use past measurements to get current state.



11.1. Offline estimation

Regard system

$$x_{k+1} = f_k(x_k, w_k), \quad k = 0, \ldots, N-1$$

$$y_k = g_k(x_k, w_k) + v_k, \quad k = 0, \ldots, N-1$$

with measurements of y_k, $k = 0, \ldots, N-1$ and all other variables unknown.

w_k: Disturbance

v_k: Measurement Errors

We additionally "know" the probability density func. (PDF)

for $w_0, \ldots, w_{N-1}$ for $v_0, \ldots, v_{N-1}$

and for x_0

We assume:

$$p_{w,k}(w_k) := c_{w,k} \cdot e^{-\beta_k(w_k)}, \quad p_{v,k}(v_k) = c_{v,k} \cdot e^{-\phi(v_k)}, \quad p_{x_0}(x_0) = c_{x_0} \cdot e^{-\alpha_0(x_0)}$$

Remark 1:

If we know past inputs u_k and the system

$x_{k+1} = f(x_k, u_k, w_k)$

then define

$\tilde{f}_k(x_k, w_k) := f(x_k, u_k, w_k)$

Remark 2:

If we doubt that u_k is implemented as we have sent it, we introduce $w_k^{(2)}$ as errors of u_k.

$$\tilde{f}_k(x_k, w_k) = f(x_k, u_k + w_k^{(2)}, w_k^{(1)})$$

$$w_k = \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix}$$

Remark 3:

If there are unknown but constant (in time) parameters p and

$$x_{k+1} = f_k(x_k, p, w_k)$$

We can define

$$\tilde{x}_k = \begin{bmatrix} x_k \\ p_k \end{bmatrix}, \quad \tilde{x}_{k+1} = \tilde{f}_k(\tilde{x}_k, w_k) \quad \tilde{f} = \begin{bmatrix} f \\ p_k \end{bmatrix}$$

$$x_{k+1} = f_k(x_k, p_k, w_k), \quad p_{k+1} = p_k$$

Remark 4:

Note: $(x_0, w_0, w_1, \dots, w_{N-1})$ determine $(x_1, \dots, x_N)$ uniquely.

11.2. Trajectory estimation problem

Given $y_0, \dots, y_{N-1}$

What is most likely trajectory

$x_0, \dots, x_N, w_0, \dots, w_{N-1}$?

Bayesian estimator:

$$p(x,\omega|y) = \frac{p(x,\omega,y)}{p(y)}$$

Measurement error

Model & Disturbance

$$= \frac{p(y|x,\omega)\cdot p(x,\omega)}{p(y)}$$

Const wrt. (x,ω)

Often, we look at

$$-\log p(x,\omega|y)$$

Often, maximise $p(x,\omega|y)$ wrt. x,ω to get "good estimater" (MAP).

Same as minimizing $-\log p(x,\omega|y)$

$$-\log p(x,\omega|y) = \underbrace{-\log p(x,\omega)}_{A} \underbrace{-\log p(y|x,\omega)}_{B} + \text{const}$$

$$A:\ p(x,\omega) = p(x_0,\dots,x_N,\omega_0,\dots,\omega_{N-1})$$

$$= \begin{cases} 0 & \text{if } \underline{\text{not}}\ x_{k+1} = f_k(x_{k+1},\omega_k) \text{ for } k=0,\dots,N-1 \\ p(x_0)\cdot p(\omega_0)\cdot p(\omega_1)\cdot \dots\ p(\omega_{N-1}) & \text{else} \end{cases}$$

$$-\log p(x,\omega) = \begin{cases} \infty \\ \alpha_0(x_0) + \sum_{k=0}^{N-1} \beta_k(\omega_k) \end{cases}$$

$$B:\ p(y|x,\omega) = p(y_0,\dots,y_{N-1}|x_0,\dots,x_N,\omega_0,\dots,\omega_{N-1})$$

$$= \prod_{k=0}^{N-1} p(y_k|x_k,\omega_k)$$

$$= \prod_{k=0}^{N-1} e^{-\phi_k(y_k - g_k(x_k,\omega_k))}\cdot \text{const}$$

$$y_k = g_k(x_k,\omega_k) + v_k$$

$$p(v_k) = e^{-\phi_k(v_k)}\cdot \text{const}$$

$$-\log p(y|x,\omega) = \sum_{k=0}^{N-1} \phi_k(y_k - g_k(x_k,\omega_k))$$

Map-estimate:

$$\boxed{\begin{aligned} &\arg\min_{x,\omega}\ \alpha_0(x_0) + \sum_{k=0}^{N-1}\left(\beta_k(\omega_k) + \phi_k(y_k - g_k(x_k,\omega_k))\right) \\ &\text{s.t.}\quad x_{k+1} = f(x_k,\omega_k), \quad k=0,\dots,N-1 \end{aligned}}$$

Examples for $\alpha_0, \beta_0, \ldots, \beta_{N-1}, \phi_0, \ldots, \phi_{N-1}$

1) $\alpha_0(x_0) = \frac{1}{2}(x_0 - \bar{x}_0)^T P_0 (x_0 - \bar{x}_0) =: \frac{1}{2}\|x_0 - \bar{x}_0\|_{P_0}^2$

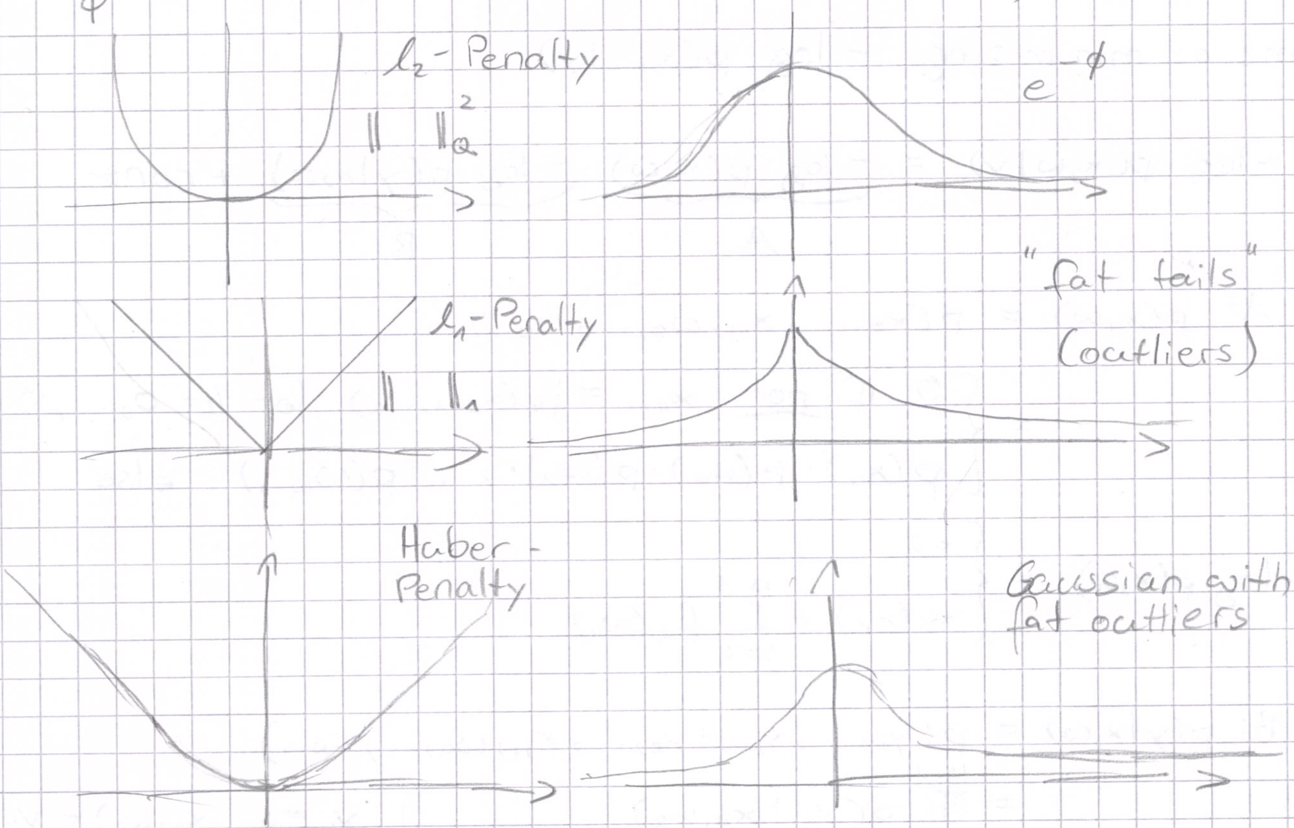
$P_0 \succ 0$, $\bar{x}_0$ fixed

(Gaussian with mean $\bar{x}_0$ and covariance $\Sigma_0 = P_0^{-1}$

2) $\alpha_0(x_0) = \begin{cases} \infty & \text{if } x_0 \neq \bar{x}_0 \\ 0 & \text{if } x_0 = \bar{x}_0 \end{cases}$

$\beta_k(\omega_k) = \frac{1}{2}\|\omega_k\|_Q^2$, $\phi_k(v_k) = \frac{1}{2}\|v_k\|_R^2$

(Gaussian errors)



How to detect parameter jumps?

$p_{k+1} = p_k + \omega_k$ $\qquad \beta_k(\omega_k) = \|\omega_k\|_1$

11.3. Recursive solution by dynamic programming

$$\min_{x_N} \min_{\omega_{N-1}} \min_{x_{N-1}} \min_{\omega_{N-2}} \cdots \min_{x_0} \quad \alpha_0(x_0) + \sum_{k=0}^{N-1} \beta \ldots + \phi \ldots$$

s.t. $x_{k+1} = f_k(x_k, \omega_k)$

$k = 0, \ldots, N-1$

<u>Define</u>: Arrival cost for $n = 0, \dots, N$

$$\alpha_n(x_n) = \min_{\substack{x_0, \dots, x_{n-1} \\ w_0, \dots, w_{n-1}}} \alpha_0(x_0) + \sum_{k=0}^{n-1} \beta_k(w_k) + \phi_k(y_k - g_k(x_k, w_k))$$

$$\text{s.t. } x_{k+1} = f_k(x_k, w_k), \quad k = 0, \dots, n-1$$

$$\alpha_n(x_n) = \min_{w_{n-1}, x_{n-1}} \alpha_{n-1}(x_{n-1}) + \beta_{n-1}(w_{n-1}) + \phi_{n-1}(y_{n-1} - g_{n-1}(x_{n-1}, w_{n-1}))$$

$$\text{s.t. } x_n = f_{n-1}(x_{n-1}, w_{n-1})$$

<u>Algorithm</u>:

1) Start with $\alpha_0(\dots)$
2) Compute $\alpha_1(\dots), \dots, \alpha_N(\dots)$ by DP-Recursion
3) Find x_N^* as $\arg\min_{x_N} \alpha_N(x_N)$
4) Find x_{n-1}^*, w_{n-1}^* from problem (n, x_n^*)

   for n from $N, \dots, 1$