- **Lecture 1:**
  - Course organization
  - Motivations
  - Introductive Examples
  - Review of Probability
- **Lecture 2:**
  - Estimation theory
  - Exercises
- **Lecture 3:**
  - Linear estimator
  - Exercises
- **Lecture 4:**
  - Linear estimator: Kalman Filter
  - Exercises & Matlab class
- **Lecture 5:**
  - Nonlinear estimation: EKF and UKF
  - Exercises & Matlab class
- **Lecture 6:**
  - Nonlinear estimation: PF and RBPF
  - Exercises & Matlab class

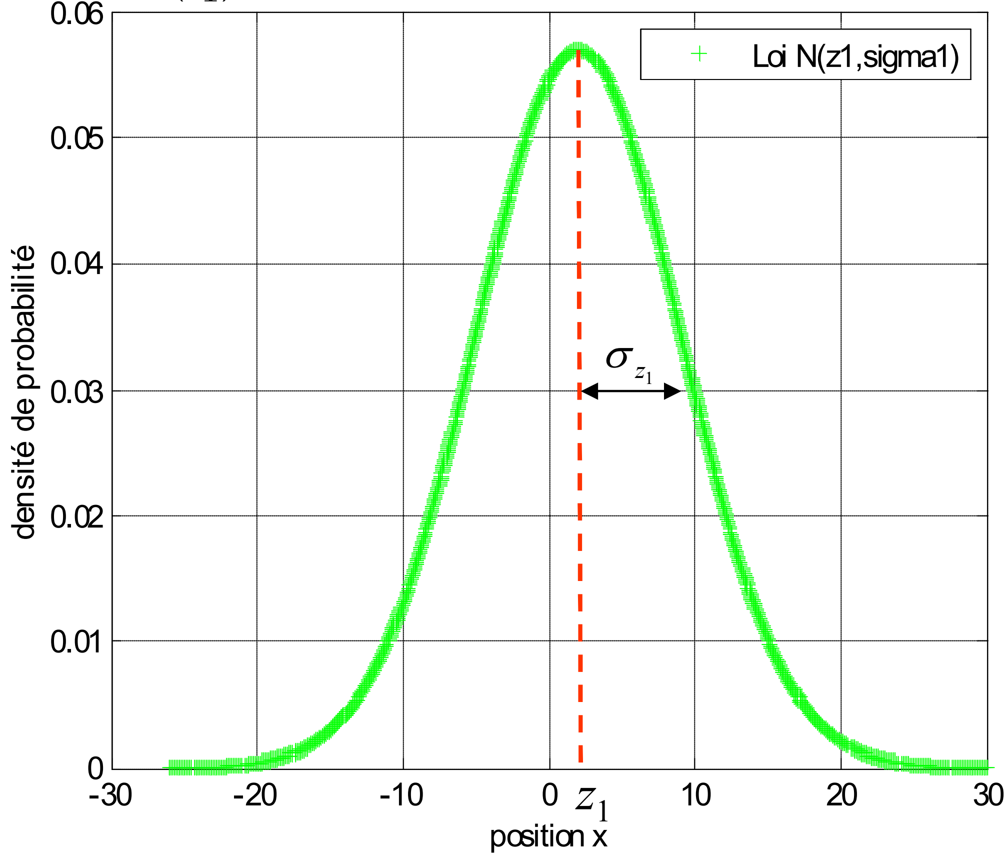
Mechbal N.

- **Introductive example:** *estimating the position of a ship*
  - *You are located on a boat, and you have no idea about your position (we assume one-dimensional localization). To determine your position, you have at your disposal only a "sextant" !*

Introductive example:

- At time t_1: a first attempt gives the position z_1
- z_1 is a realization of a random variable x with variance: $\sigma_{z_1}^2$
- The conditional probability of our position $x(t_1)$ has the form below:
- The distribution is Gaussian
- The best estimate is:

$$\hat{x}(t_1) = z_1 \ \ and \ \ \hat{\sigma}_x^2(t_1) = \sigma_{z_1}^2$$



✷ Introductive example:

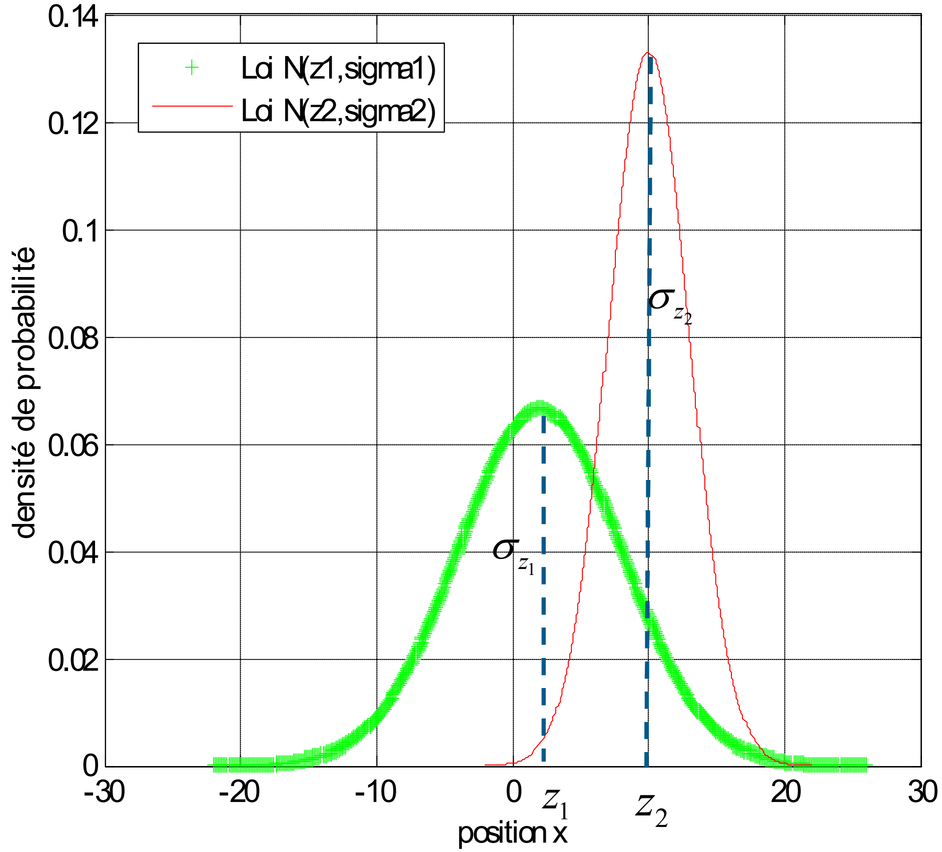
- At time t_2: It is assumed that another attempt is performed by a more experienced skipper at time t_2 near time t_1:
- This gives the position z_2
- z_2 is another realization of the random variable x with variance : $\sigma_{z_2}^2$
- As more experienced, the variance will be lower

$$\hat{x}(t_2) = ? \qquad and \quad \hat{\sigma}_x^2(t_2) = ?$$

We have two measures:

How do we proceed?

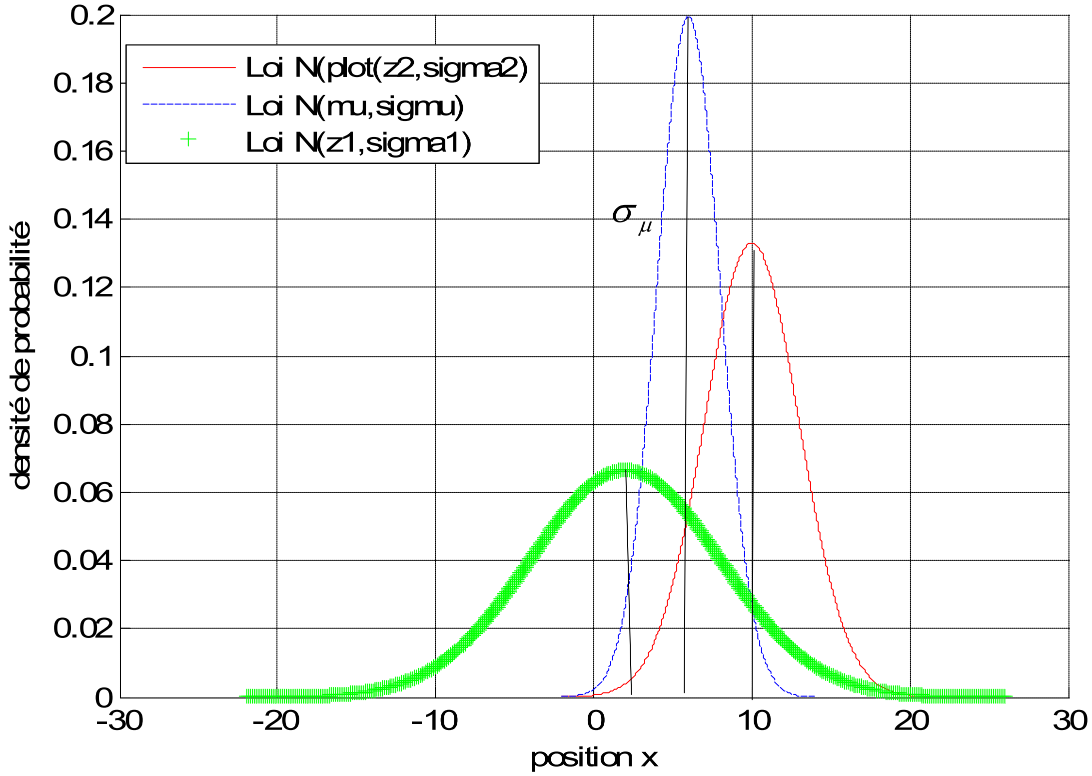
Should we eliminate the first measure?



✷ Introductive example:

✦ We propose the following estimate: *the two estimation are weighted inversely proportional to the relative uncertainty of each one, i.e.,*

$$\hat{x}(t_2) = \mu = [\sigma_{z_2}^2/(\sigma_{z_2}^2 + \sigma_{z_1}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_2}^2 + \sigma_{z_1}^2)]z_2$$



$$If \quad \sigma_{z_2}^2 << \sigma_{z_1}^2 \Rightarrow [\sigma_{z_2}^2/(\sigma_{z_2}^2 + \sigma_{z_1}^2)] \to 0$$

$$then \;\; \mu \to z_2$$

Introductive example:

- Recall the previous estimator

$$\hat{x}(t_2) = \mu = [\sigma_{z_2}^2/(\sigma_{z_2}^2 + \sigma_{z_1}^2)]z_1 + [\sigma_{z_1}^2/(\sigma_{z_2}^2 + \sigma_{z_1}^2)]z_2$$

- We show that this estimator could be rewritten as

$$\boxed{\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)\left(z_2 - \hat{x}(t_1)\right) \quad with \quad K(t_2) = \left[\sigma_{z_1}^2 \big/ \left(\sigma_{z_2}^2 + \sigma_{z_1}^2\right)\right]}$$

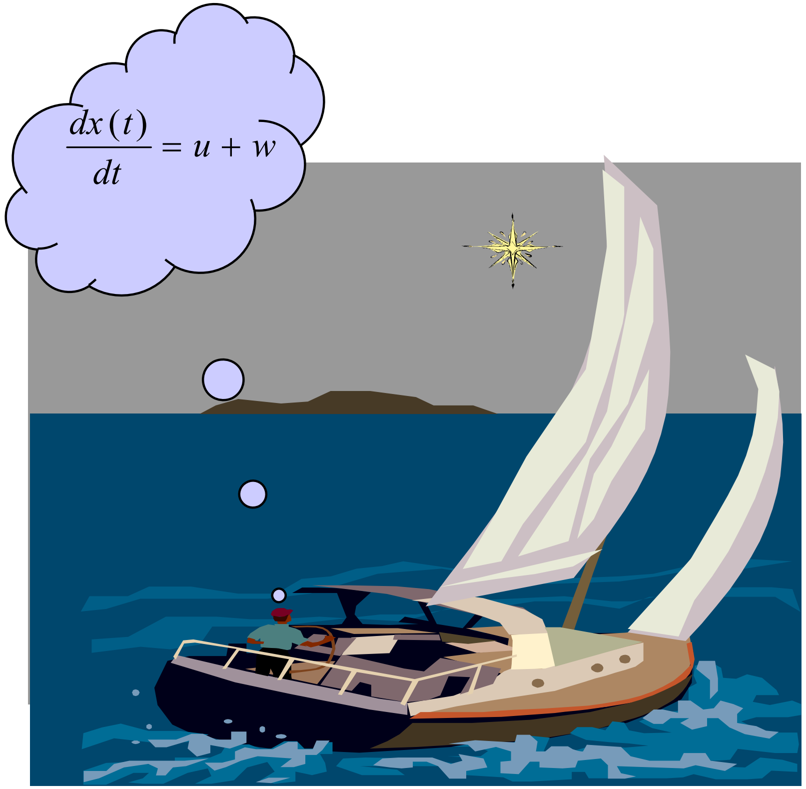
- With a variance (uncertainties)

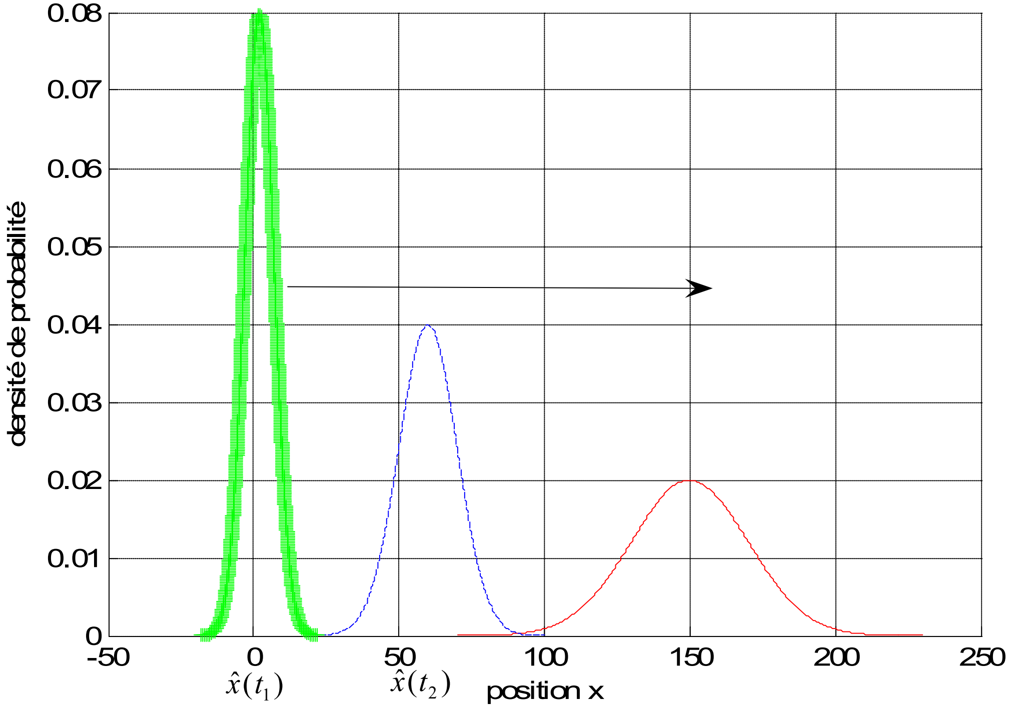
$$1/\hat{\sigma}^2(t_2) = 1/\sigma_{z_1}^2 + 1/\sigma_{z_2}^2 \quad \Rightarrow \quad \hat{\sigma}^2(t_2) = \sigma_{z_1}^2 - K(t_2)\sigma_{z_1}^2$$

- *The best estimate at time t_2 is equal to the* ***best prediction*** *at time t_1 plus a correction term weighted by a gain* ***K****.*
- The gain ***K*** is a weighting function of uncertainties
- Whatever the variance of integrated measures, the resulting variance is lower.

Introductive example:

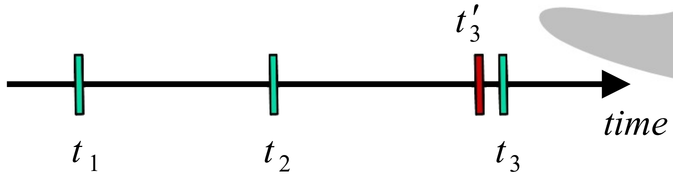
- **Dynamic Estimation**: Suppose now that the ship moves at a constant velocity *u* with disturbances (gusts of wind and currents) of *zero mean*, and *known variance*
- During time evolution, the conditional density propagates as shown below:





Mechbal N.

Introductive example:



- The kinematic model of displacement is given by

$$\frac{dx(t)}{dt} = u + w$$

- At the time t'_3 just before taking into account a new measure at t_3, the density is Gaussian with known mean and variance given by,

$$\hat{x}(t'_3) = \hat{x}(t_2) + u(t'_3 - t_2) \ et \ \ \hat{\sigma}^2_{t'_3} = \hat{\sigma}^2_{t_2} + \hat{\sigma}^2_w (t'_3 - t_2)$$

- At t_3 a new measurement is performed,

Corrected by measure

$$\hat{x}(t_3) = \hat{x}(t'_3) + K(t_3)\left(z_3 - \hat{x}(t'_3)\right) \quad with \quad K(t_3) = \left[\sigma^2_{t'_3} \Big/ \left(\sigma^2_{t'_3} + \sigma^2_{z_3}\right)\right]$$

$$\hat{\sigma}^2_{t_3} = \hat{\sigma}^2_{t'_3} - K(t_3)\sigma^2_{t'_3}$$

Predicted by the model

It 's a Kalman structure based

Mechbal N.

Introduction

- In the early 60's, Kalman (1960) and Kalman & Bucy (1961) considered the problem state by Wiener but with a state approach. They have developed a new filter called a Kalman filter. This filter is based on the fact that a random process can be modeled as the output of a linear system governed by a white noise, while the Wiener filter systems are represented by equations of covariance.

- This method of filtering and estimation has outstanding performances. It also benefited from technological advances which contributed largely to its use in many applications: robotics (localization, estimation of bias); Automatic (control, identification, fault detection) and economy (market forecast).

Rudolph E. Kalman (1960), " A new approach to linear filtering and prediction problems". Transactions of the ASME–Journal of Basic Engineering, 82 (Series D): 35-45.

Research Institute for Advanced Study, Baltimore

Problem statement:

- **Model**: we consider a **discrete linear <u>Gaussian</u> state space** representation

Noise process $\underline{\varsigma}_k$

Measurement noise $\underline{\omega}_k$

input $\underline{u}_k$

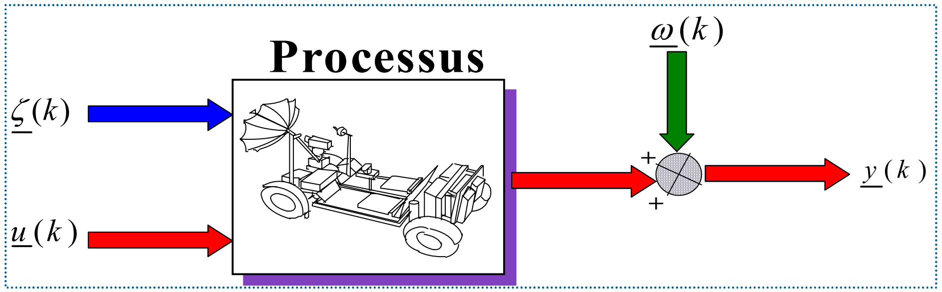
$$\underline{x}_{k+1} = F_k \underline{x}_k + G_k \underline{u}_k + H_k \underline{\varsigma}_k \quad\longrightarrow\quad \underline{y}_k = C_k \underline{x}_k + \underline{\omega}_k \quad\longrightarrow\quad \underline{y}_k$$

sensor

$$\begin{cases} \underline{\mathbf{x}}_{k+1} = F_k \underline{\mathbf{x}}_k + G_k \underline{u}_k + H_k \underline{\varsigma}_k & \textbf{(state equation)} \\ \underline{\mathbf{y}}_k = C_k \underline{\mathbf{x}}_k + \underline{\boldsymbol{\omega}}_k & \textbf{(observation equation)} \end{cases}$$

Notation

$$\underline{y}_1^k = \left[\underline{y}_1, \underline{y}_2 \cdots, \underline{y}_k\right]$$



✷ Problem statement:

- Assumption: The noise are supposed to be Gaussian independent with zero mean and having the following covariance matrix:

$$E\left\{\underline{\zeta}(k)\right\} = 0 \qquad E\left\{\underline{\omega}(k)\right\} = 0$$

$$E\left\{\underline{\zeta}(k)\underline{\zeta}^T(j)\right\} = Q\delta_{kj} \qquad E\left\{\underline{\omega}(k)\underline{\omega}^T(j)\right\} = R\delta_{kj}$$

$$E\left\{\underline{\zeta}(k)\underline{\omega}^T(j)\right\} = 0$$

- Initial state: It is unknown since only the observation vector is known. It is assumed to be a Gaussian random vector independent from the other random vector and having the following mean and cross-covariance:

$$E\left\{\underline{x}(0)\right\} = \underline{x}_0$$

$$E\left\{\underline{x}_0(k)\underline{\zeta}^T(j)\right\} = 0 \qquad E\left\{\underline{x}_0(k)\underline{\omega}^T(j)\right\} = 0$$

Mechbal N.

Problem statement:

- State Estimation: The goal is to determine an estimation of the state vector $\underline{x}(k)$ given the information available at time ***n***.

  As we are in a Bayesian Gaussian "world", we have seen that the optimal estimator is the MAP (BMS) estimator:

$$\boxed{\underline{\hat{x}}(k/n) = E\left\{\underline{x}(k)/\underline{y}(1), \dots, \underline{y}(n)\right\}}$$

- We can consider three estimation problems :

  - $k = n$: **filtering**
  - $k < n$: **smoothing**
  - $k > n$: **Prediction**

✷ Problem statement:

- <u>State Estimation</u>: Our goal is to develop a recursive estimation of the state vector which, from an estimate at time $k-1$ we provide a new estimate given the available measurements at time k. we use the linear estimator

Prediction of the output:

$$\hat{y}(k) = C\,\underline{\hat{x}}(k/k-1)$$

$$\boxed{\underline{\hat{x}}(k/k) = \underline{\hat{x}}(k/k-1) + K\,\underbrace{\left[\underline{y}(k) - C\,\underline{\hat{x}}(k/k-1)\right]}_{\text{Correction term}}}$$

Kalman gain

- When you observe an error in trial k, the amount that you should change $\hat{x}$ should depend on <u>how certain you are about ω.</u>
- *The more certain you are, the less you should be influenced by the error. The less certain you are, the more you should "pay attention" to the error*

Mechbal N.

Problem statement:

- Criterion: In the case of filtering problem, the objective is to calculate the gain K in order to minimize a cost function. Here we will use the LS criterion on the estimation error.
  - Estimation error:

$$\underline{\tilde{x}}(k) = \underline{x}(k) - \underline{\hat{x}}(k/k)$$

  - Criterion

$$\boxed{J = E\left\{\underline{\tilde{x}}^T(k/k)\underline{\tilde{x}}(k/k)\right\}}$$

- Minimum variance filter: The problem of Kalman filter is also a minimum variance problem, *i.e.*,
  - Introduce the estimation error covariance matrix

$$P(k/k) = E\left\{[\underline{x}(k) - \underline{\hat{x}}(k/k)][\underline{x}(k) - \underline{\hat{x}}(k/k)]^T\right\} = E\left\{\underline{\tilde{x}}(k)\underline{\tilde{x}}^T(k)\right\}$$

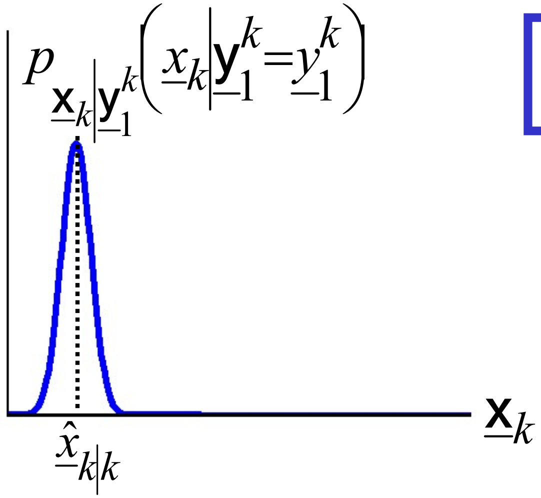
  - We can then verify that

$$trace\ P(k/k) = J(k)$$

  - Hence the optimization procedure will also lead to a **minimum variance filter**.

✷ Intuitive approach:

✦ Principle



$\boxed{\hat{\underline{x}}_{k|k} \text{ and } P_{k|k}}$

Recursive method :

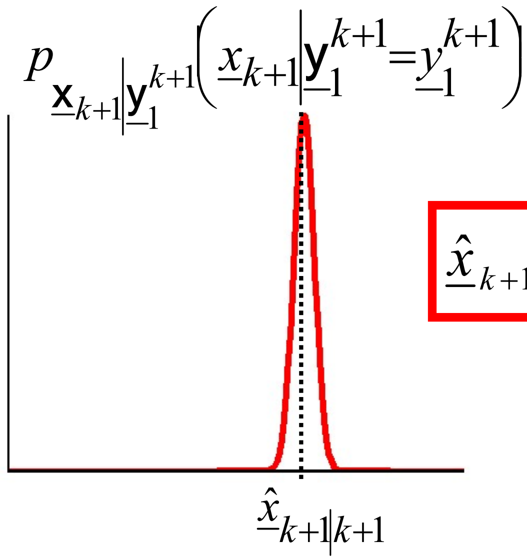
We propagate $\hat{\underline{x}}_{k|k}$ and $P_{k|k}$

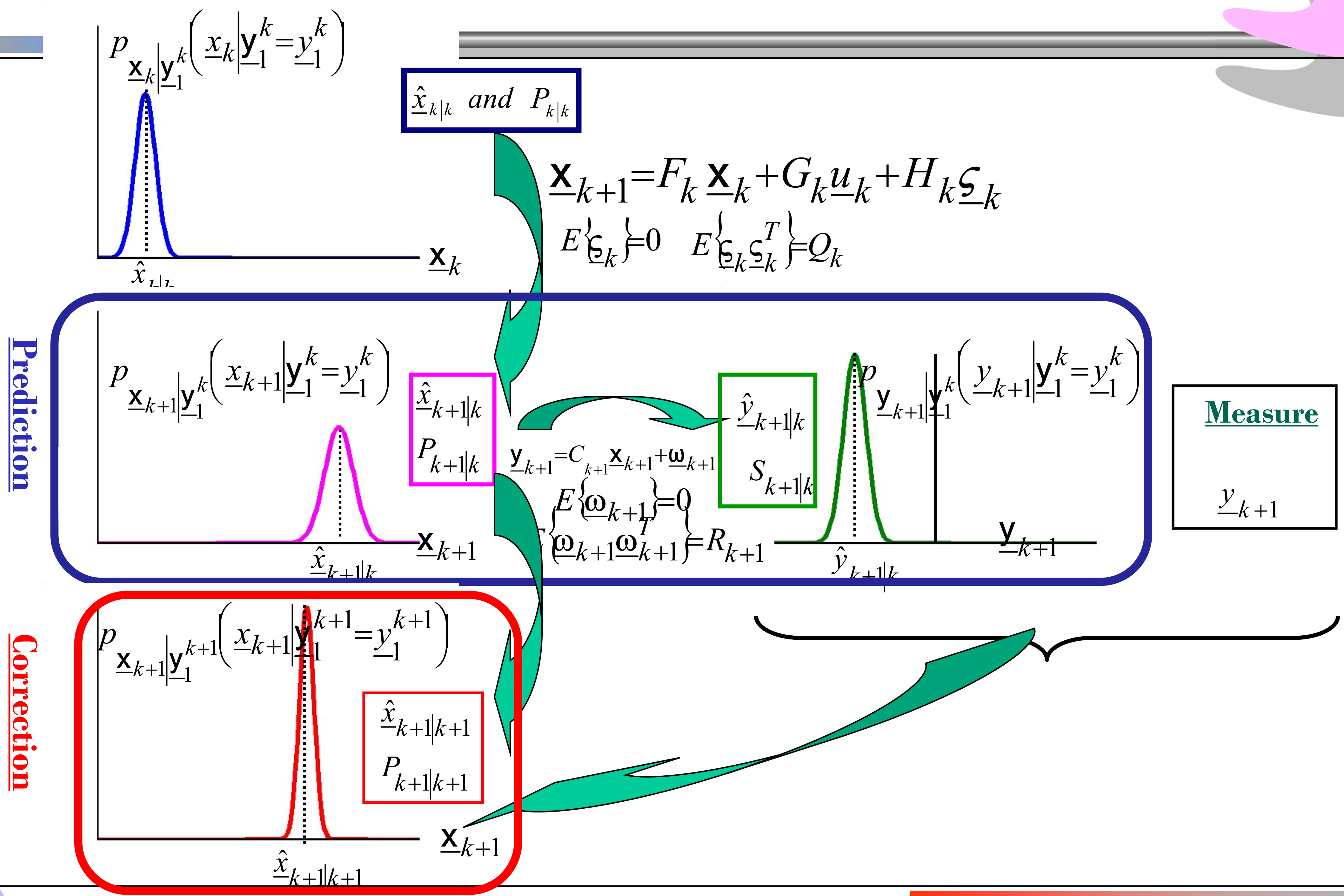
To have $\hat{\underline{x}}_{k+1|k+1}$ and $P_{k+1|k+1}$

This is done in 2 steps :

1. *Prediction*
2. *Correction*

$\boxed{\hat{\underline{x}}_{k+1|k+1} \text{ and } P_{k+1|k+1}}$





$\hat{\underline{x}}_{k|k}$ and $P_{k|k}$

$$\underline{\mathbf{x}}_{k+1}=F_k\,\underline{\mathbf{x}}_k+G_k\underline{u}_k+H_k\underline{\varsigma}_k$$

$$E\{\underline{\varsigma}_k\}=0 \quad E\{\underline{\varsigma}_k\underline{\varsigma}_k^T\}=Q_k$$

$p_{\underline{\mathbf{x}}_k|\underline{\mathbf{y}}_1^k}\left(\underline{x}_k\middle|\underline{\mathbf{y}}_1^k=\underline{y}_1^k\right)$

$p_{\underline{\mathbf{x}}_{k+1}|\underline{\mathbf{y}}_1^k}\left(\underline{x}_{k+1}\middle|\underline{\mathbf{y}}_1^k=\underline{y}_1^k\right)$

$\hat{\underline{x}}_{k+1|k}$, $P_{k+1|k}$

$$\underline{\mathbf{y}}_{k+1}=C_{k+1}\underline{\mathbf{x}}_{k+1}+\underline{\boldsymbol{\omega}}_{k+1}$$

$$E\{\underline{\boldsymbol{\omega}}_{k+1}\}=0 \quad E\{\underline{\boldsymbol{\omega}}_{k+1}\underline{\boldsymbol{\omega}}_{k+1}^T\}=R_{k+1}$$

$\hat{\underline{y}}_{k+1|k}$, $S_{k+1|k}$

$p_{\underline{\mathbf{y}}_{k+1}|\underline{\mathbf{y}}_1^k}\left(\underline{y}_{k+1}\middle|\underline{\mathbf{y}}_1^k=\underline{y}_1^k\right)$

Measure: $\underline{y}_{k+1}$

$p_{\underline{\mathbf{x}}_{k+1}|\underline{\mathbf{y}}_1^{k+1}}\left(\underline{x}_{k+1}\middle|\underline{\mathbf{y}}_1^{k+1}=\underline{y}_1^{k+1}\right)$

$\hat{\underline{x}}_{k+1|k+1}$, $P_{k+1|k+1}$

Procedure:

- **Prediction step: we use the state equation (the model !)**

We have $\underline{\hat{x}}_{k|k}$ and $P_{k|k}$

We look for $\underline{\hat{x}}_{k+1|k}$ and $P_{k+1|k}$

$$\underline{\hat{x}}_{k+1|k} = E\left\{\underline{x}_{k+1} \,\middle|\, \underline{y}_1^k = y_1^k\right\} \qquad \underline{x}_{k+1} = F_k \underline{x}_k + G_k \underline{u}_k + H_k \underline{\varsigma}_k$$

$$\Rightarrow \quad \underline{\hat{x}}_{k+1|k} = E\left\{\underline{x}_{k+1} \,\middle|\, \underline{y}_1^k = \underline{y}_1^k\right\} = F_k\, E\left\{\underline{x}_k \,\middle|\, \underline{\boldsymbol{y}}_1^k = \boldsymbol{y}_1^k\right\} + G_k E\{\underline{u}_k\} + H_k E\left\{\underline{\varsigma}_k\right\}$$

$$\Rightarrow \quad \boxed{\underline{\hat{x}}_{k+1|k} = F_k\, \underline{\hat{x}}_{k|k} + G_k\, \underline{u}_k}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{E}\left\{\left(\underline{x}_{k+1} - \underline{\hat{x}}_{k+1|k}\right)\left(\underline{x}_{k+1} - \underline{\hat{x}}_{k+1|k}\right)^{\boldsymbol{T}} \,\middle|\, \underline{\boldsymbol{y}}_1^k = \boldsymbol{y}_1^k\right\}$$

$$\Rightarrow \quad \boxed{\boldsymbol{P}_{k+1|k} = F_k\, \boldsymbol{P}_{k|k} F^T + H_k\, \boldsymbol{Q}_k H_k^T}$$

Procedure:

- **Prediction Step:** We use the observation equation

$$\underline{y}_{k+1} = C_{k+1}\,\underline{x}_{k+1} + \underline{\omega}_{k+1}$$

$$\Rightarrow \underline{\hat{y}}_{k+1|k} = E\left\{\underline{y}_{k+1} \,\middle|\, \underline{y}_1^k = y_1^k\right\} = C_k\,\underline{\hat{x}}_{k+1|k}$$

⇨ *Covariance matrix of* $\underline{\boldsymbol{y}}_{k+1}$ *conditioned on* $\underline{\mathbf{y}}_1^k = \boldsymbol{y}_1^k$

$$\boldsymbol{S}_{k+1|k} = \boldsymbol{E}\left\{\left(\underline{\boldsymbol{y}}_{k+1} - \underline{\hat{\boldsymbol{y}}}_{k+1|k}\right)\left(\underline{\boldsymbol{y}}_{k+1} - \underline{\hat{\boldsymbol{y}}}_{k+1|k}\right)^{T} \,\middle|\, \underline{\boldsymbol{y}}_1^k = \boldsymbol{y}_1^k\right\} = \boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^{T} + \boldsymbol{R}_{k+1}$$

⇨ *Cross correlation matrix of* $\underline{x}_{k+1}$ *and* $\underline{\boldsymbol{y}}_{k+1}$ *conditioned on* $\underline{\mathbf{y}}_1^k = \boldsymbol{y}_1^k$

$$\boldsymbol{T}_{k+1|k} = \boldsymbol{E}\left\{\left(\underline{x}_{k+1} - \underline{\hat{x}}_{k+1|k}\right)\left(\underline{\boldsymbol{y}}_{k+1} - \underline{\hat{\boldsymbol{y}}}_{k+1|k}\right)^{T} \,\middle|\, \underline{\boldsymbol{y}}_1^k = \boldsymbol{y}_1^k\right\} = \boldsymbol{P}_{k+1/k}\boldsymbol{C}_{k+1}^{T}$$

$$\Rightarrow \boldsymbol{S}_{k+1|k} = \boldsymbol{C}_{k+1}\boldsymbol{T}_{k+1/k} + \boldsymbol{R}_{k+1}$$

Procedure:

- <u>Prediction Step:</u> We wish to consider the measure $y(k+1)$ to derive a posteriori knowledge:

$$\underline{\hat{x}}_{k+1|k+1} \; and \;\; P_{k+1|k+1}$$

- Return to the **linear static case** where we define x and z two Gaussian random vectors with $m_x\, P_{xx}$; $m_z\, P_{zz}$ as respective mean and covariance. We had also defined the cross covariance: $P_{xz} = P_{xz}^T$
- The solution is then straightforward by performing the analogy (**slide 80-83**)

$\underline{z}^N$	$\underline{y}_{k+1}$
m_x	$\underline{\hat{x}}_{k+1\|k}$
m_z	$\underline{\hat{y}}_{k+1\|k} = C_{k+1}\, \underline{\hat{x}}_{k+1\|k}$
P_{xx}	$P_{k+1\|k}$
P_{zz}	$S_{k+1\|k} = C_{k+1}\, P_{k+1\|k}\, C_{k+1}^T + R_{k+1}$
P_{xz}	$T_{k+1\|k} = P_{k+1\|k}\, C_{k+1}^T$
P_{zx}	$T_{k+1\|k}^T = C_{k+1}\, P_{k+1\|k}$
H	C_{k+1}
R	R_k
$m_{x\|z}$	$\underline{\hat{x}}_{k+1\|k+1}$
$P_{x\|z}$	$P_{k+1\|k+1}$

- **Procedure:**
  - **Prediction Step:** From the static result (slide 80-83)

$$m_{x|z} = m_x + K(\underline{z}^N - m_z)$$
$$P_{x|z} = P_{xx} - K P_{zx}$$
$$K = P_{xz}\,(P_{zz})^{-1}$$

  - By analogy we have

$$\underline{\hat{x}}_{k+1|k+1} = \underline{\hat{x}}_{k+1|k} + K_{k+1}\,\left(\underline{y}_{k+1} - C_{k+1}\,\underline{\hat{x}}_{k+1|k}\right)$$
$$P_{k+1|k+1} = (I - K_{k+1}C_{k+1})\;P_{k+1|k}$$
$$\text{avec}\quad K_{k+1} = P_{k+1|k}\,C_{k+1}^T\,(C_{k+1}\,P_{k+1|k}\,C_{k+1}^T + R_{k+1})^{-1}$$

K_{k+1} : ***Kalman gain***

Minimum variance solution: *we propose here to drive the gain filter K from:* $\boxed{traceP(k/k) = J(k)}$

Perform the following difference between the state and it estimate

$$\underline{x}(k) - \hat{\underline{x}}(k/k-1) = F\left[\underline{x}(k-1) - \hat{\underline{x}}(k-1/k-1)\right] + H\underline{\zeta}(k-1)$$

But we have:

$$P(k/k-1) = E\left\{[\underline{x}(k) - \hat{\underline{x}}(k/k-1)][\underline{x}(k) - \hat{\underline{x}}(k/k-1)]^T\right\}$$

Using expectation, one have:

$$P(k/k-1) = F.E\left\{[\underline{x}(k-1) - \hat{\underline{x}}(k-1/k-1)][\underline{x}(k-1) - \hat{\underline{x}}(k-1/k-1)]^T\right\}.F^T + H.E\left\{\underline{\zeta}(k-1)\underline{\zeta}^T(k-1)\right\}$$

With our assumptions, we can derive the following recursive expression:

$$\boxed{P(k/k-1) = FP(k-1/k-1).F^T + H.Q.H^T}$$

with

$$P(k-1/k-1) = E\left\{[\underline{x}(k-1) - \hat{\underline{x}}(k-1/k-1)][\underline{x}(k-1) - \hat{\underline{x}}(k-1/k-1)]^T\right\}$$

Let us now express $P(k/k)$ in function of $P(k/k-1)$. Let us take for that the estimation error and replace $\hat{\underline{x}}(k/k)$ by its expression

$$\tilde{\underline{x}}(k) = \underline{x}(k) - \hat{\underline{x}}(k/k) = \underline{x}(k) - \hat{\underline{x}}(k/k-1) - K\left[\underline{y}(k) - C\hat{\underline{x}}(k/k-1)\right]$$

$$= \tilde{\underline{x}}(k/k-1) - K\left[\underline{y}(k) - C\hat{\underline{x}}(k/k-1)\right]$$

hence

$$\tilde{\underline{x}}(k/k) = \tilde{\underline{x}}(k/k-1) - K\left[C\underline{x}(k) + \underline{\omega}(k) - C\hat{\underline{x}}(k/k-1)\right]$$

$$= [I - KC]\tilde{\underline{x}}(k/k-1) - K\underline{\omega}(k)$$

We can now drive the expression of $P(k/k)$, *i.e.*,:

$$\boxed{P(k/k) = [I - KC]P(k/k-1)[I - KC]^T - K.R.K^T}$$

Minimum variance solution

We seek now to calculate de gain K that minimise the criterium

$$\arg\min_{K} \left(traceP(k/k) \right):$$

Therefore, we need to calculate the derivative :

$$\frac{\partial J(k)}{\partial K} = 0$$

using:

$$\frac{\partial (traceWTW^{T})}{\partial W} = 2WT \quad \Rightarrow \quad K.R. - [I - KC]P(k/k-1)C^{T} = 0$$

By letting $D = \left[CP(k/k-1)C^{T} + R \right]$, we have the optimal expression of the gain

$$\boxed{K(k) = P(k/k-1)C^{T}D^{-1}}$$

We report now this expression of the gain in the covariance to have a simplified formula :

$$\begin{aligned} P(k/k) &= [I - KC]P(k/k-1) - P(k/k-1)C^{T}K^{T} + KCP(k/k-1)C^{T}K^{T} - K.R.K^{T} \\ &= [I - KC]P(k/k-1) - P(k/k-1)C^{T}K^{T} + K\left[CP(k/k-1)C^{T} - R\right]K^{T} \\ &= [I - KC]P(k/k-1) - P(k/k-1)C^{T}K^{T} + KD.K^{T} \\ &= [I - KC]P(k/k-1) - P(k/k-1)C^{T}K^{T} + P(k/k-1)C^{T}D^{-1}D.K^{T} \\ &= [I - KC]P(k/k-1) \end{aligned}$$

Convergence: from the expression of the covariance we can note that as $K.C.P(k/k-1)$ is positive definite (semi-positive), it follows that all diagonal elements are either positive or zero. Therefore, we can say

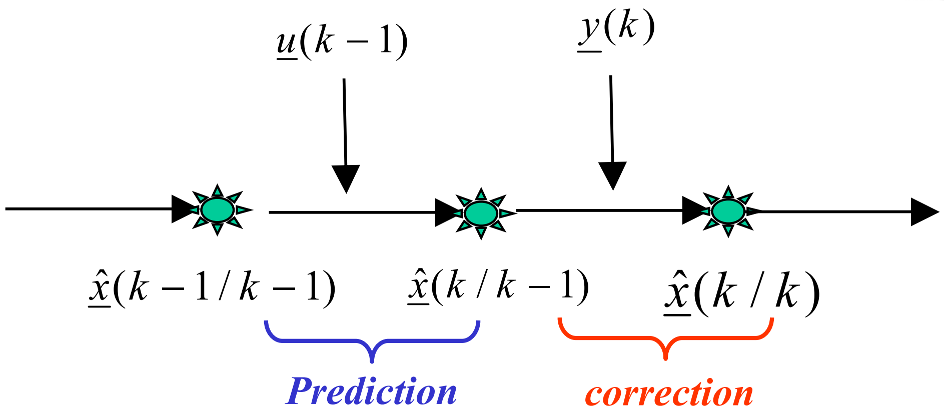
$$traceP(k/k) < traceP(k/k-1)$$

Implementation:

- Consider the system:

$$\underline{x}(k+1) = F\,\underline{x}(k) + G\underline{u}(k) + H\underline{\zeta}(k)$$

$$\underline{y}(k) = C\,\underline{x}(k) + \underline{\omega}(k)$$



- ***Initialization :***

$$\hat{\underline{x}}(0) = E\{\underline{x}(0)\} = \underline{x}_0$$

$$P(0/0) = E\left\{[\underline{x}(0) - \hat{\underline{x}}(0)][\underline{x}(0) - \hat{\underline{x}}(0)]^T\right\} = P(0)$$

- ***Prediction or a priori estimation :***

$$\hat{\underline{x}}(k/k-1) = F\,\hat{\underline{x}}(k-1/k-1) + G\underline{u}(k-1)$$

$$P(k/k-1) = FP(k-1/k-1).F^T + H.Q.H^T$$

- ***Correction or a posteriori estimation :***

$$P(k/k) = [I - K(k)C]P(k/k-1)$$

$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + K(k)\left[\underline{y}(k) - C\hat{\underline{x}}(k/k-1)\right]$$

$$K(k) = P(k/k-1)C^T\left[CP(k/k-1)C^T + R\right]^{-1}$$

Kalman Algorithm

Initialization:

Choose $\underline{\hat{x}}_{0|0}$ et $P_{0|0}$.

$k = 0$

Prediction:

$$\underline{\hat{x}}_{k+1|k} = F_k\, \underline{\hat{x}}_{k|k} + G_k\, u_k$$

$$P_{k+1|k} = F_k\, P_{k|k}\, F_k^T + H_k\, Q_k\, H_k^T$$

Measure $\underline{y}_{k+1}$

Correction:

$$K_{k+1} = P_{k+1|k}\, C_{k+1}^T \left(C_{k+1}\, P_{k+1|k}\, C_{k+1}^T + R_k\right)^{-1}$$

$$\underline{\hat{x}}_{k+1|k+1} = \underline{\hat{x}}_{k+1|k} + K_{k+1} \left(\underline{y}_{k+1} - C_{k+1}\, \underline{\hat{x}}_{k+1|k}\right)$$

$$P_{k+1|k+1} = \left(I - K_{k+1} C_{k+1}\right) P_{k+1|k}$$

$k = k + 1$

Innovation signal :

- The innovation signal is defined by

$$\boxed{\underline{\upsilon}(k) = \underline{y}(k) - C\,\underline{\hat{x}}(k/k-1)}$$

- If the prediction is "perfect" the innovation is zero !
- If the estimation is **optimal** then this signal is a **white noise** with zero mean. *That is, it doesn't contain further information that can enhance the update of the estimate.*
- Therefore, one way to test the optimality of the filter is to test its whiteness
- This signal is widely used as a residual in *Fault detection and Isolation* (FDI) process.

Parameterization

- The choice of tuning parameters is not ease: $x(0), P(0), Q$ and R
- For the initial state vector, it is common to use other estimators on a small set of data to estimate it. For example, we can use **LS estimator** to come up with an estimation of $x(0)$ and $P(0)$. If we cannot do that, we choose **a large value of $P(0)$** to inform the estimator that we are uncertain about this initial value, e.g., $P(0) = 10^6$.
- *For the covariance:* Q is related to the confidence we have on the model. It is an image of the model uncertainties. R is related to measurement uncertainties.

  To better understand the influence of these choices, we use another expression of the gain

$$\boxed{K(k) = P(k/k)C^T R(k)^{-1}}$$

and we consider then the scalar case:

$$x(k) = x(k-1) + K(k)[\,correction \quad term\,]$$

What's happened then if we freeze R and we increase and decrease Q ? and vis versa ?

- *If R is constant and Q is low then K is low which show that we thrust more estimation from the model than the one from measurements. If now Q is high then K will be high which make the correction term weighted by K higher that mean that we don't have much confidence in the model.*
- *If Q (or P) is constant and R is low which means that we have measurement that are weakly noised. The high value of the gain K, will give more importance to the correction term in regard to the model estimation term.*

Simple Example : scalar case

- *Consider now the scalar state space model with the following values:*

$$F = 1, \quad C = 1, \quad P(0/0) = 10, \quad Q = 20, \quad R = 10, \quad G = 1 \;\; et \;\; H = 1$$

- *Give the filter equations*

$$P(k/k-1)\ ,\ K(k),\ P(k/k)$$

- *Give a table of the evolution (for each k) of the following parameters:*

- *Calculate the solution in the permanent regime, $\boldsymbol{P}_\infty$, i.e.,*

$$\boldsymbol{P}_\infty = P(k/k) = P(k+1/k+1)$$

Simple example: Estimation of a constant

- ***Model*** :
  - *The system is described by the following discrete state space model:*

$$x(k+1) = x(k) + \zeta(k)$$
$$y(k) = x(k) + \omega(k)$$

  - *The input u is null and noises are of zero mean and described by their covariance matrices Q et R.*

- ***Filter equation:***

Prediction:

$$\underline{x}(k/k-1) = \hat{\underline{x}}(k-1/k-1)$$
$$P(k/k-1) = P(k-1/k-1).+Q.$$

Correction:

$$K(k) = \frac{P(k/k-1)}{P(k/k-1)+R}$$
$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + K\left[\underline{y}(k) - \hat{\underline{x}}(k/k-1)\right]$$
$$P(k/k) = [I-KC]P(k/k-1)$$

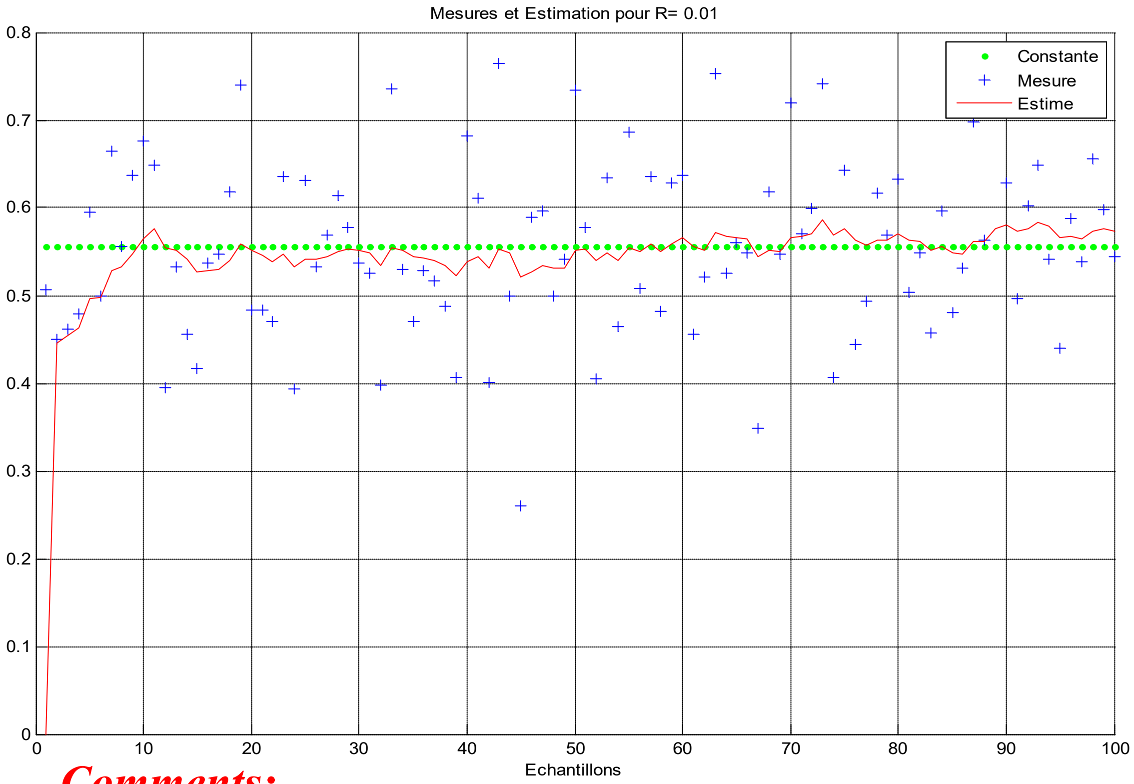
- ***Simulation:*** We perform simulations for different values of ***R*** we also assume that:

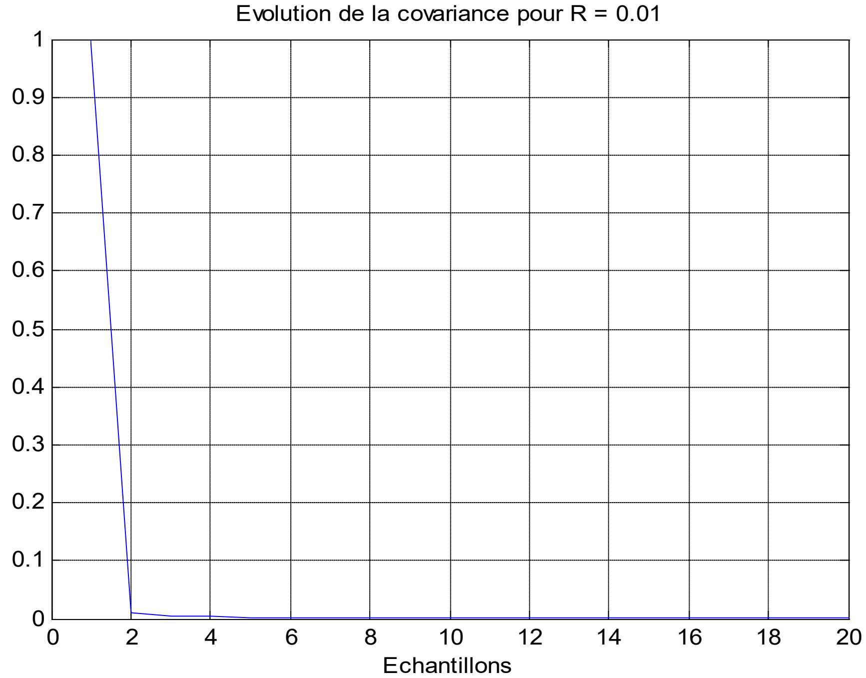
  - ✓ The target value is: $x(k) = 0.5555$,
  - ✓ The process noise covariance is: $Q = 1e^{-4}$
  - ✓ Initialed values are: $\hat{x}(0) = 0$ et $P(0) = 1$,

Simple example: Estimation of a constant

Simulation :

$\boxed{R = 0.01}$



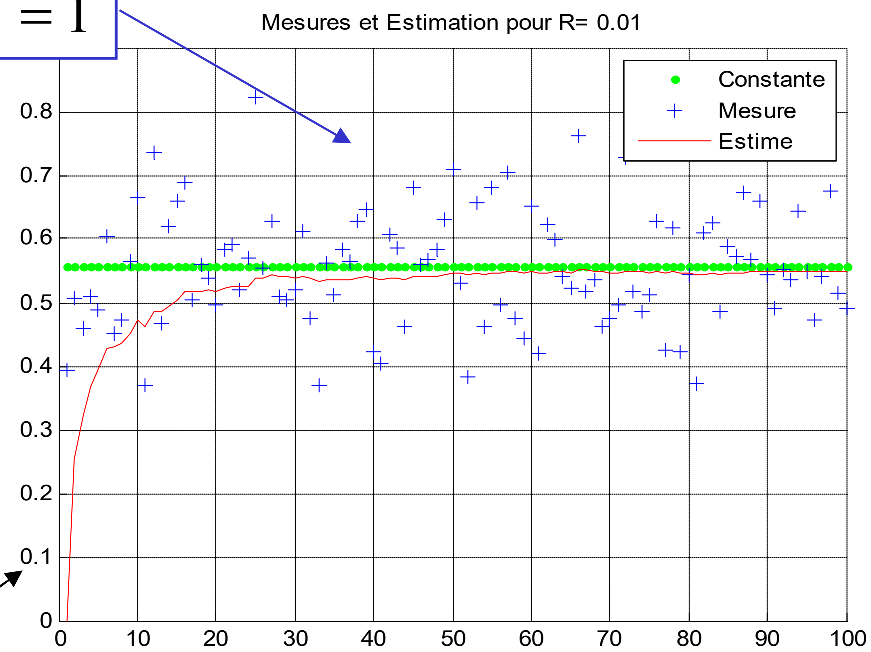


Comments:

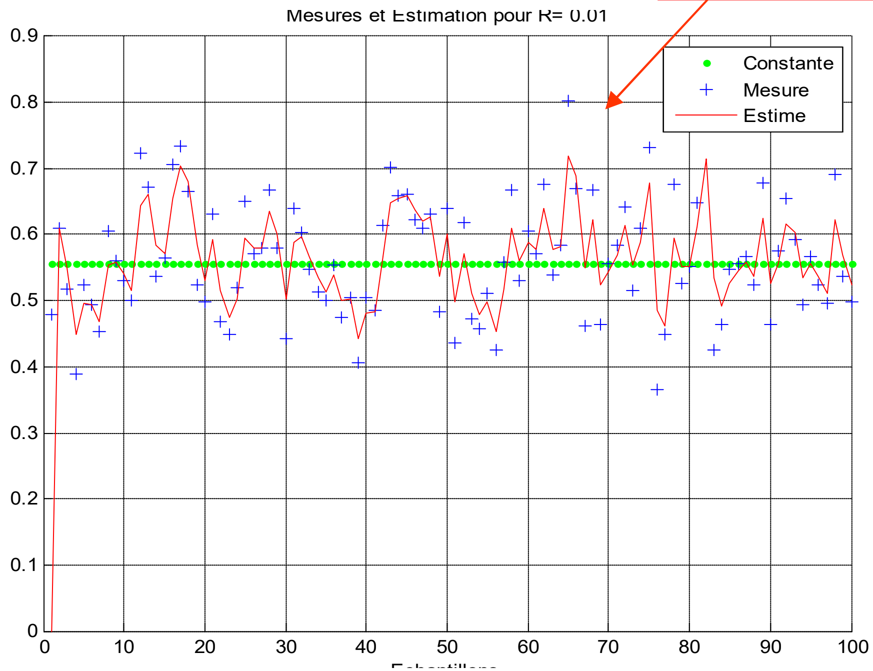
In this first simulation, we simulate the system with a variance of measurement noise representing the true variance of noise. We then expect the filter to give us the best estimate in terms of tradeoff between confidence in the measurement and the model. We then observe that the filter converges to the desired value and the variance decreases rapidly to a value around zero

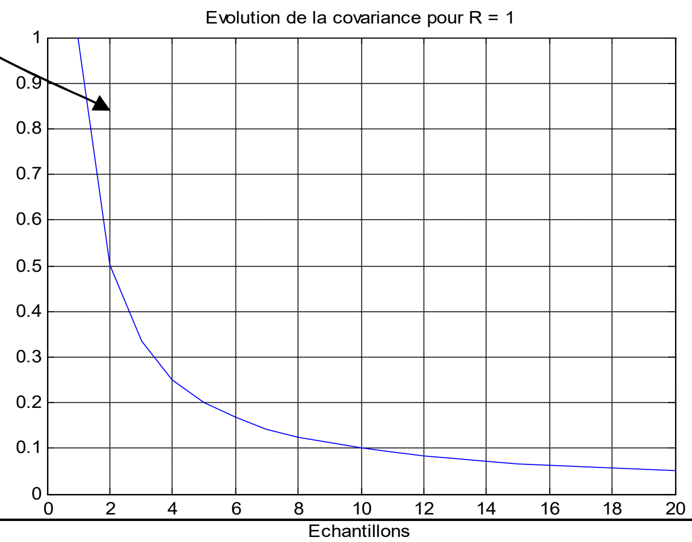
Simple example: Estimation of a constant

$R = 1$



$R = 0.0001$





Comments:

✓*In the first case (R=1), we expect large variations in the shape of the signal to be estimated. We then gives great confidence to the equation of state. The filter will ignore the rapid changes of the signal and perform a smoothing of the data. But provides an estimate for which the variance of the error takes longer to decrease*

✓*In the second case (R=0.0001), the filter quickly adapts to changes. It relies on measurements. The variance of the estimation error will converge faster but without filtering the noise*

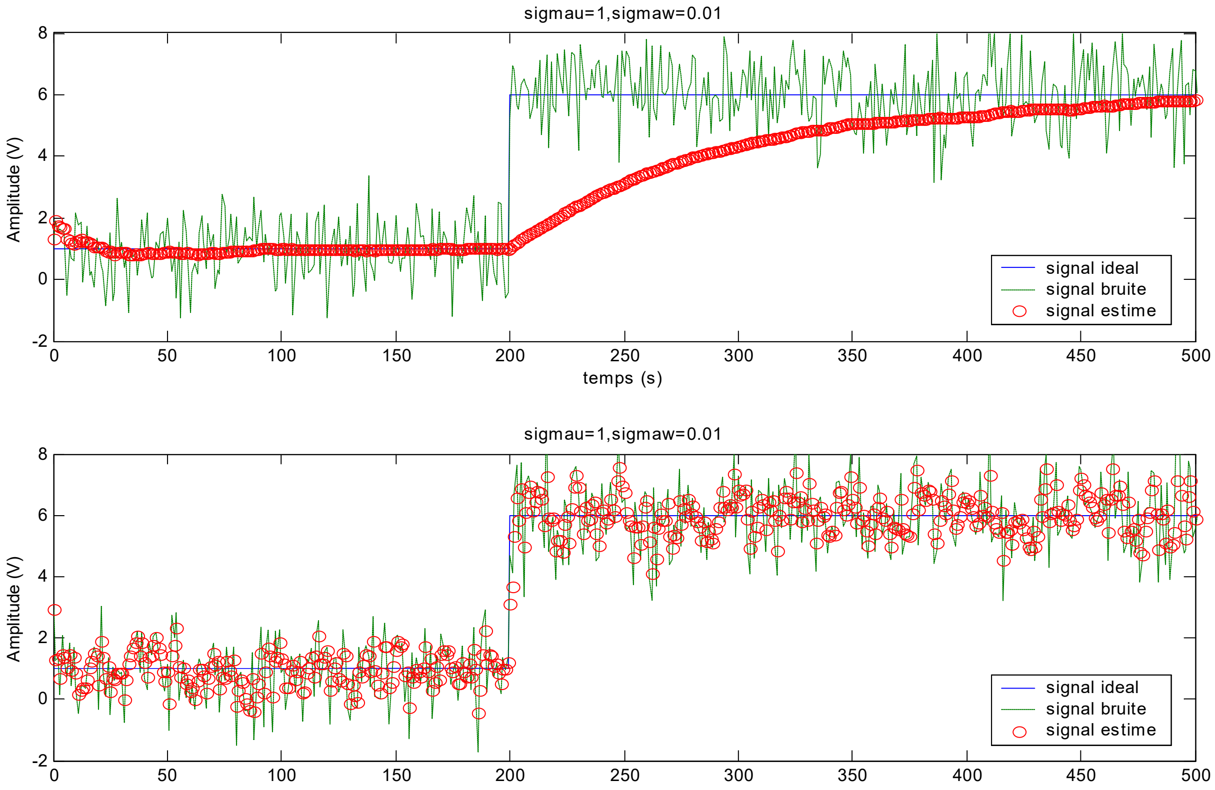
Mechbal N.

- **Example:**

  - Consider the following system having unit measurement covariance, R=1.

$$y(k)=\begin{cases}1+\omega(k) & if \;\; k\in[1,300]\\ 5+\omega(k) & if \;\; k\in[301\;,1000]\end{cases}$$

  - We have performed Two simulations:
    - Great confidence in the model: Q=0.01
    - Uncertain model: Q=0.8



Exercise:

- We consider a system represented by the following discrete model:

$$x_{k+1} = 0.5\, x_k + \zeta_k$$
$$y_k = x_k + \omega_k$$

- The process and measurement noise are Gaussian, independent with zero mean and covariance

$$E\{\zeta_k^2\} = Q_k = 2$$
$$E\{\omega_k^2\} = R_k = \frac{1}{2^{k-1}}$$

- The initial state is supposed Gaussian and independent from noise

1. Express the equations of the Kalman filter based on the data of the problem.
2. Calculate the value of

$$\hat{x}_{k/k} \; and \; P_{k/k}$$

for $k = 1, 2$ when the initial values are

$$\hat{x}_{0/0} = 0 \;\; and \;\; P_{0/0} = 1$$

Monte-Carlo Simulations

- It was shown that the Kalman filter is an unbiased estimator with minimum variance. However, this proof assumes that the matrices F_k, G_k, H_k, C_k, Q_k and R_k are known, and the assumptions made about the noise are met. These theoretical results do not guarantee the behavior of the filter when these assumptions are violated. To assess the behavior of the filter when all the assumptions are not met, we can proceed to **Monte-Carlo simulations**

- Notations: The quantities involved in the model simulating the "**reality**" are marked with the index **"V"** those related to the **filter** with index "F".

- Principle: we perform Ns simulations

Monte-Carlo Simulations

Simulation number *i* of *Ns*:

- ***Generate the realization of the initial state:***

$$\underline{x}_0^V(i) \sim N(\underline{x}_0^{V\,nb}, P_{0/0}^V)$$

From $k = 1$ until time end simulation t_{end}, do:

- ***Generate the realization of the true state vector :***

$$\underline{x}_{k+1}^V(i) = F_k \underline{x}_k^V(i) + G\underline{u}_k + H_k \underline{\zeta}_k(i)$$

$$\underline{\zeta}_k(i) \sim N(0, Q_k^V)$$

- ***Generate the realization of the measure vector:***

$$\underline{y}_k^V(i) = C_k \underline{x}_k^V(i) + \underline{\omega}_k(i) \qquad \underline{\omega}_k(i) \sim N(0, R_k^V)$$

- ***Calculate the estimate and covariance***

$$\underline{\hat{x}}_{k+1/k}^F(i) \ , P_{k+1/k}^F(i) \longrightarrow \underline{\hat{x}}_{k+1/k+1}^F(i) \ , P_{k+1/k+1}^F(i)$$

Using Kalman equations with : Q_k^V and R_k^V

Monte-Carlo Simulations

- Evaluate the bias by calculating the empirical mean of the estimation error:

$$\overline{\underline{\tilde{x}}_{k/k}} = \frac{1}{N_s}\sum_{i=1}^{N_s} \underline{\tilde{x}}_{k/k}(i)$$

With

$$\underline{\tilde{x}}_k(i) = \underline{x}_k^V(i) - \underline{\hat{x}}_{k/k}^F(i)$$

- Compare the empirical covariance (*it's the **real** precision of the estimation*) of the estimation error to the one given by the filter (*it's the **thrust** level of the filter on its estimation*), i.e.,

$$\overline{P_{k/k}^V} = \frac{1}{N_s}\sum_{i=1}^{N_s}\left(\underline{\tilde{x}}_k(i) - \overline{\underline{\tilde{x}}_k}\right)\left(\underline{\tilde{x}}_k(i) - \overline{\underline{\tilde{x}}_k}\right)^T$$

is compared to

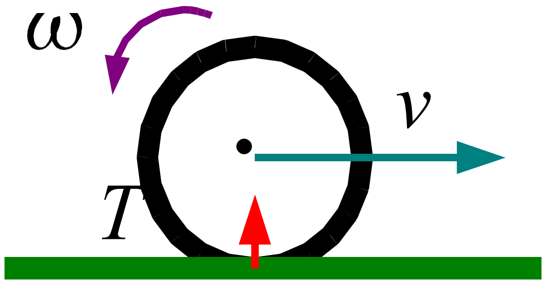
$$\overline{P_{k/k}^F}$$

Monte-Carlo Simulations

- It is desired that the two matrices $\overline{P^F_{k/k}}$ and $\overline{P^V_{k/k}}$ are substantially equal because this means that the filter evaluates well its estimate.

- If the diagonal components of $\overline{P^F_{k/k}}$ are superior to those of $\overline{P^V_{k/k}}$:
  - This means that the filter is **conservative (underestimated its capabilities).**
  - The contrary is **not acceptable**

✴ Example MC simulations: Estimation of wheel-soil interaction force



$$\dot{v} = \frac{1}{m} T$$

$$\dot{\omega} = \frac{-r}{J} T + \frac{1}{J} u(t)$$

- where m is the mass of the mobile, J, inertia, r is the radius of the wheel, T is the force of wheel-soil interaction $u(t)$ is the torque applied to the wheel.
- There is several complicate model (Pacejka, de Lure, …) that give $T(v,\omega)$, here, we have considered ***T*** as a state variable and we fixed it dynamics

$$\ddot{T} = 0$$

- We get then a continuous linear state representation

$$\dot{x} = Ax + Bu$$

$$x = \begin{bmatrix} v \\ \omega \\ T \\ \dot{T} \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{-r}{J} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{J} \\ 0 \\ 0 \end{bmatrix}$$

Example MC simulations: Estimation of wheel-soil interaction force

- The model is then discretised with T_e as sampling period:

$$\underline{\mathbf{x}}_{k+1} = F\,\underline{\mathbf{x}}_k + G\,\underline{u}_k + \underline{\zeta}_k$$

$$y_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} x_k$$

- Kalman filter and MC simulations:

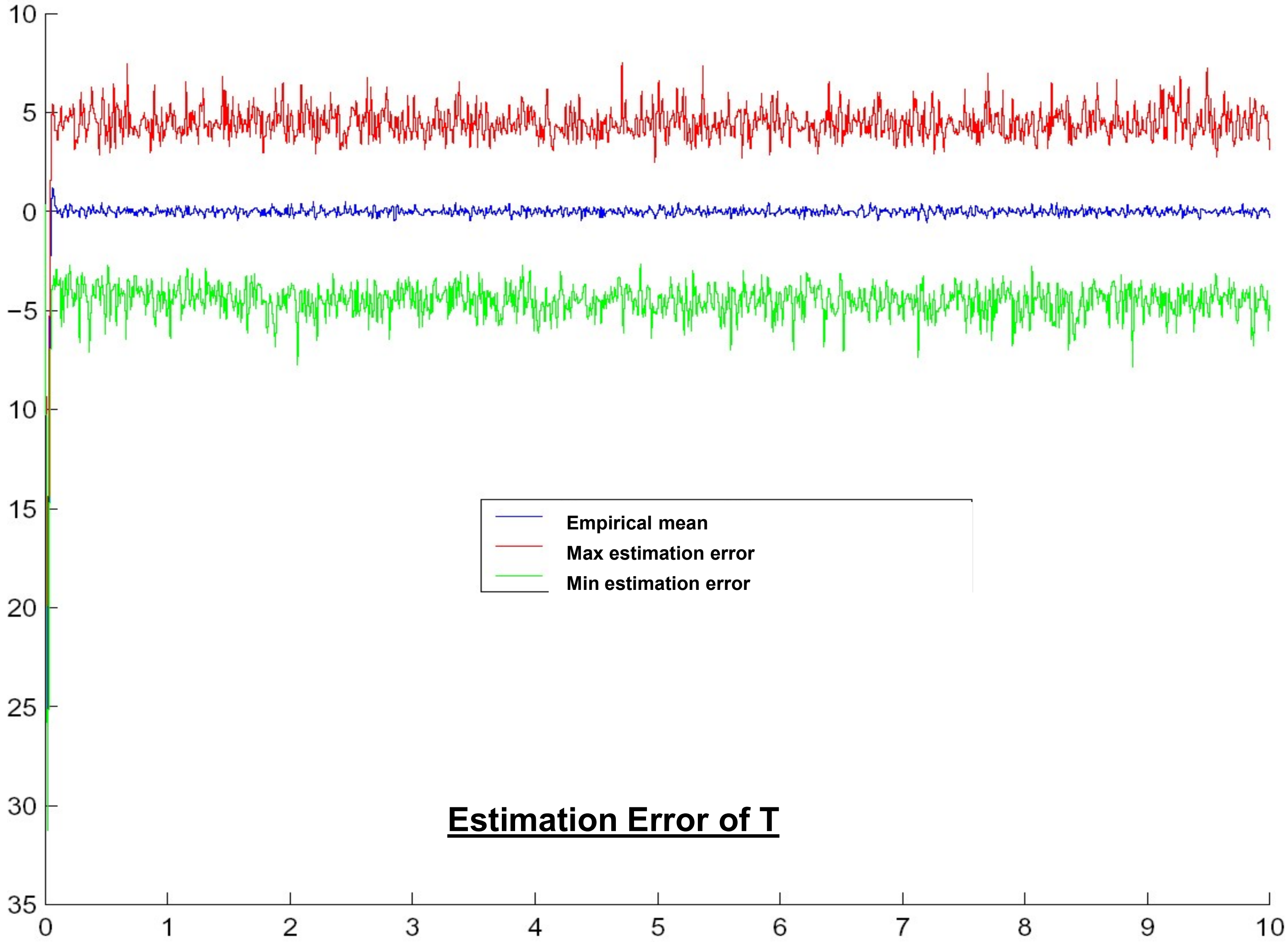
$$N_s = 100$$

$$m = 5,\ \bar{J} = 0.23,\ r = 0.25,$$

$$\underline{x}_0^{V_{\mathrm{nb}}} = [4\,2\,0\,0]^T,\ Q^V = Q^F \text{ et } R^V = R^F$$

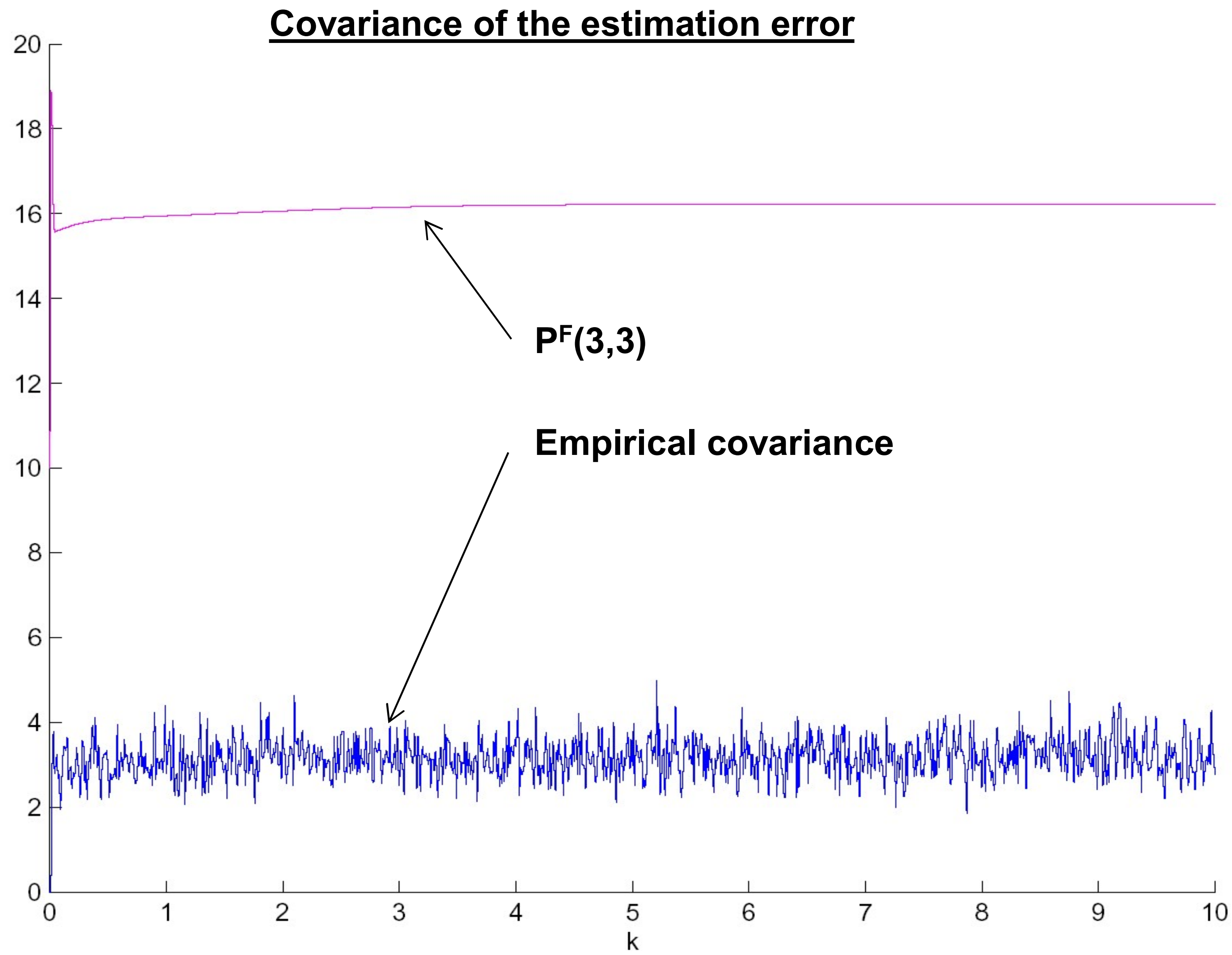
- For the simulation of the real system the force T is calculated using the model of Pacejka

✷ Example MC simulations: Estimation of wheel-soil interaction force



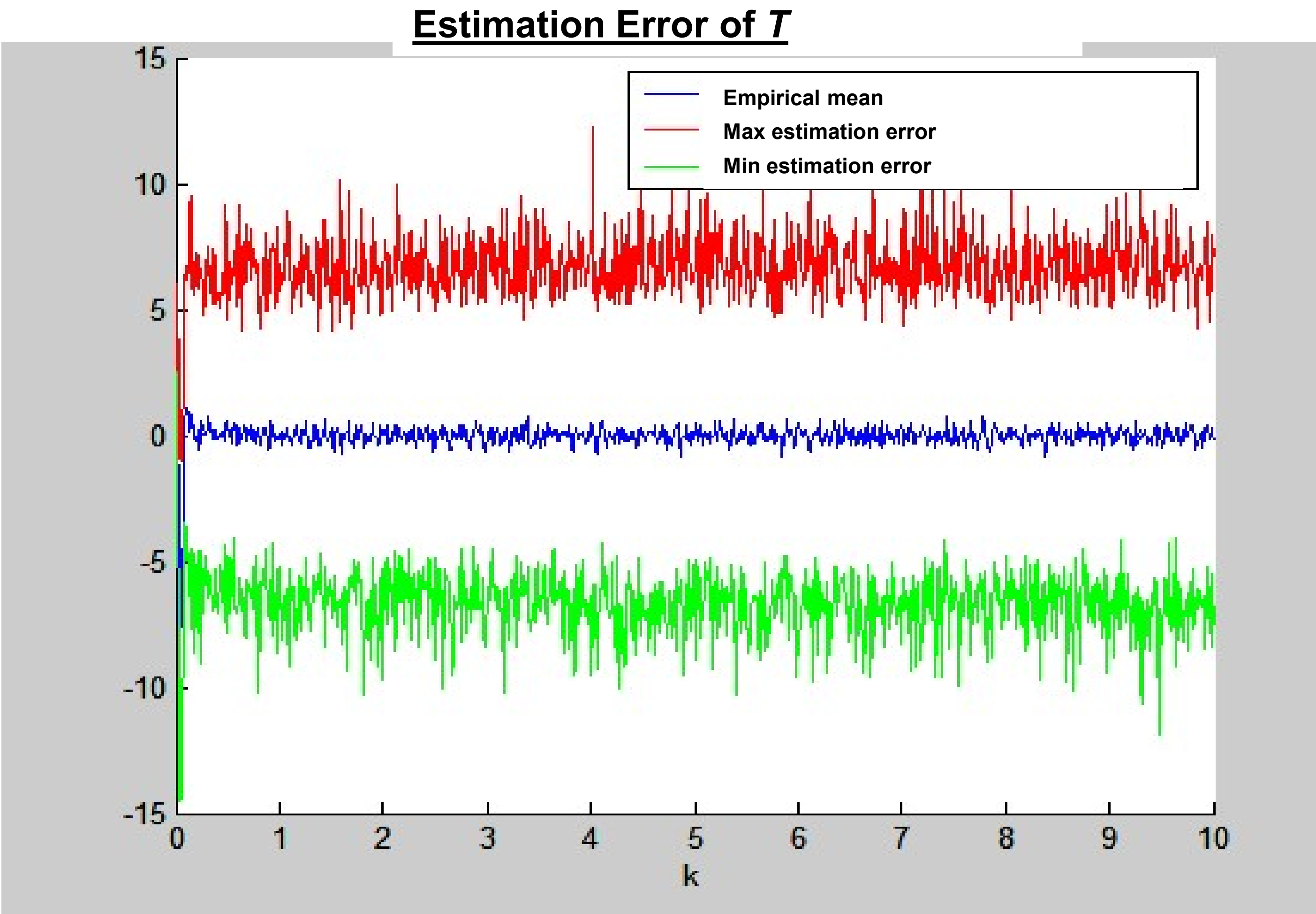
Estimation Error of T

✷ Example MC simulations: Estimation of wheel-soil interaction force

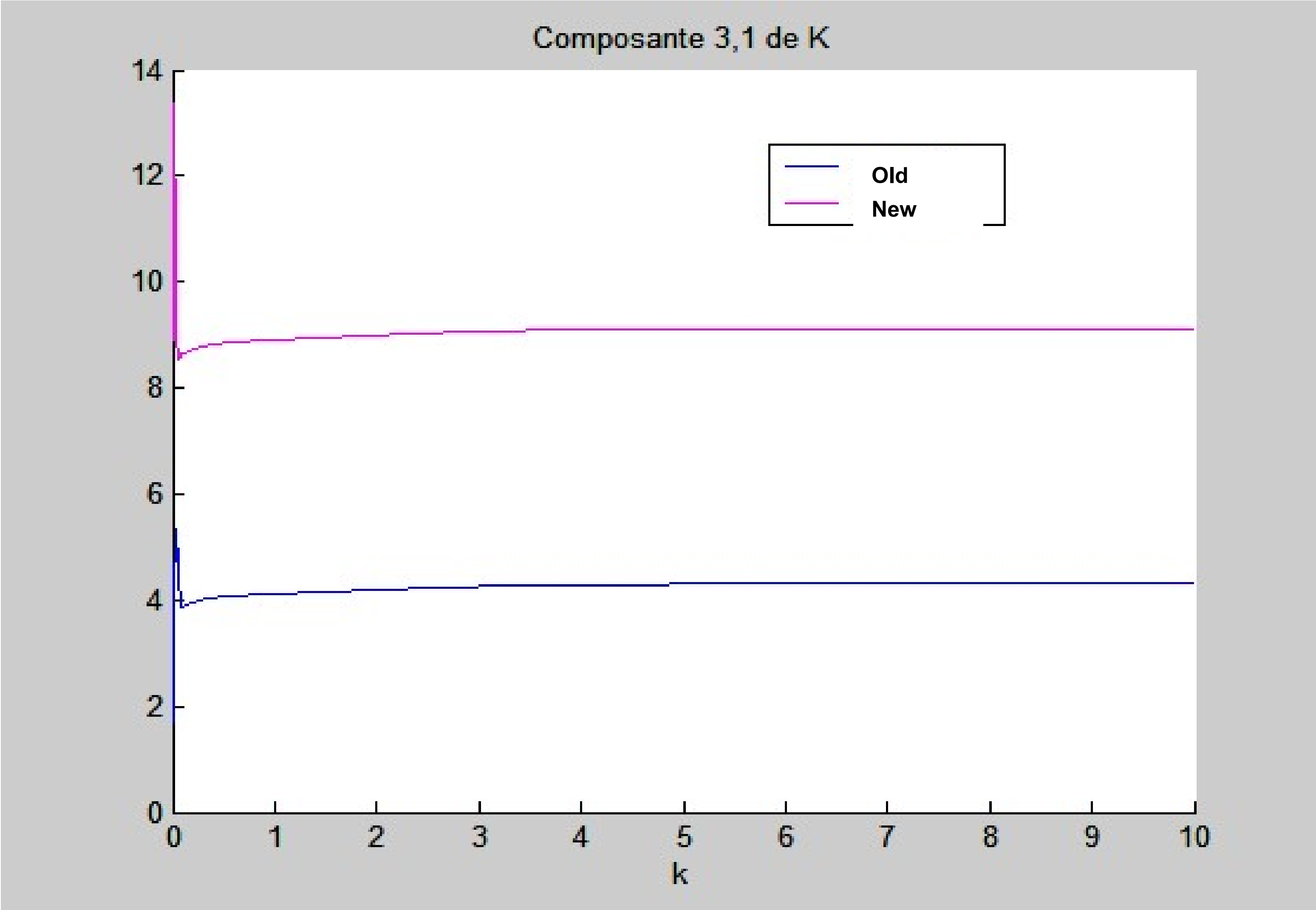


- **Example MC simulations: Estimation of wheel-soil interaction force**
  - Robustness : we modify the noise covariance ***R*new=10*R*old**

$$\underline{x}_0^{V_{\rm nb}} = [4\,2\,0\,0]^T,\ Q^V = Q^F \text{ et } R^V = R^F$$



- Example MC simulations: Estimation of wheel-soil interaction force



Matlab Exercise

- We consider the estimation by a **Kalman filter** the position, velocity and acceleration of mobile autonomous robot. The kinematic model of the robot is given by the following discrete SS representation:

$$\underline{x}(k+1) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0.9 & 1 \\ 0 & 0 & 1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \zeta(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \underline{x}(k) + \omega(k)$$

- The state space vector to be estimated is $\underline{X}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} Position \\ Velocity \\ Accelerati\ on \end{bmatrix}$

- The noise are considered Gaussian with zero mean. Covariances and initial values are given by:

$$\text{Initial } : P(0) = \begin{bmatrix} 10^8 & 0 & 0 \\ 0 & 2.5\times 10^3 & 0 \\ 0 & 0 & 10^2 \end{bmatrix}, \ \underline{x}(0) = \begin{bmatrix} 10^2 \\ 50 \\ 5 \end{bmatrix},$$

$$\text{Process noise} : Q^V = 1$$

$$\text{Measuremen t noise} : R^V = 10^2$$

Simulations

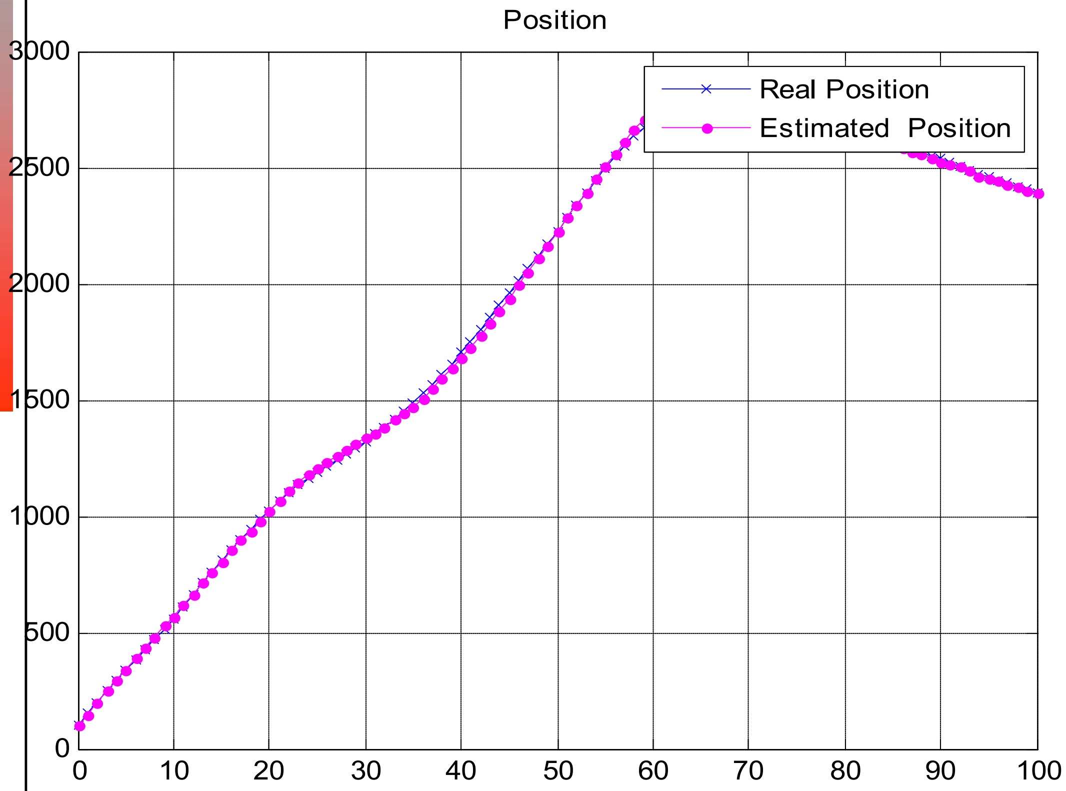
1. The first step is to simulate the system in order to have input-output data.

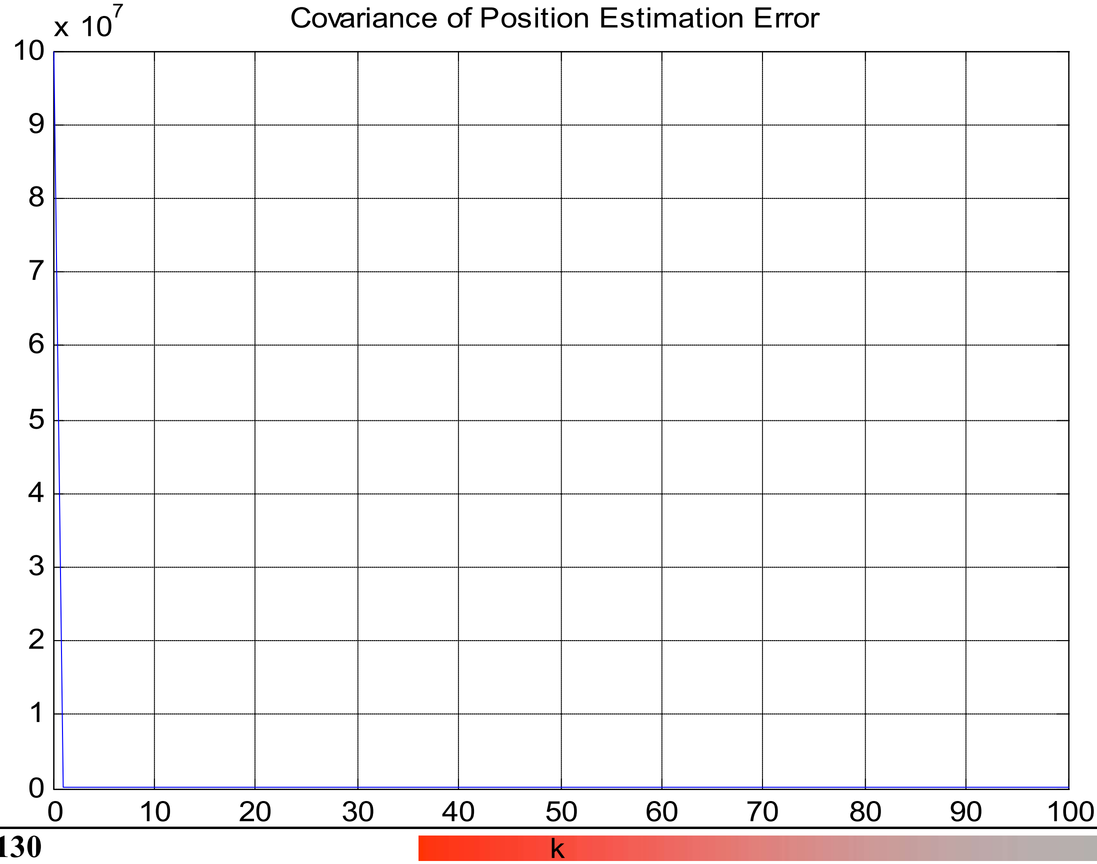
2. Simulate the Kalman filter

3. Perform Monte-Carlo simulations

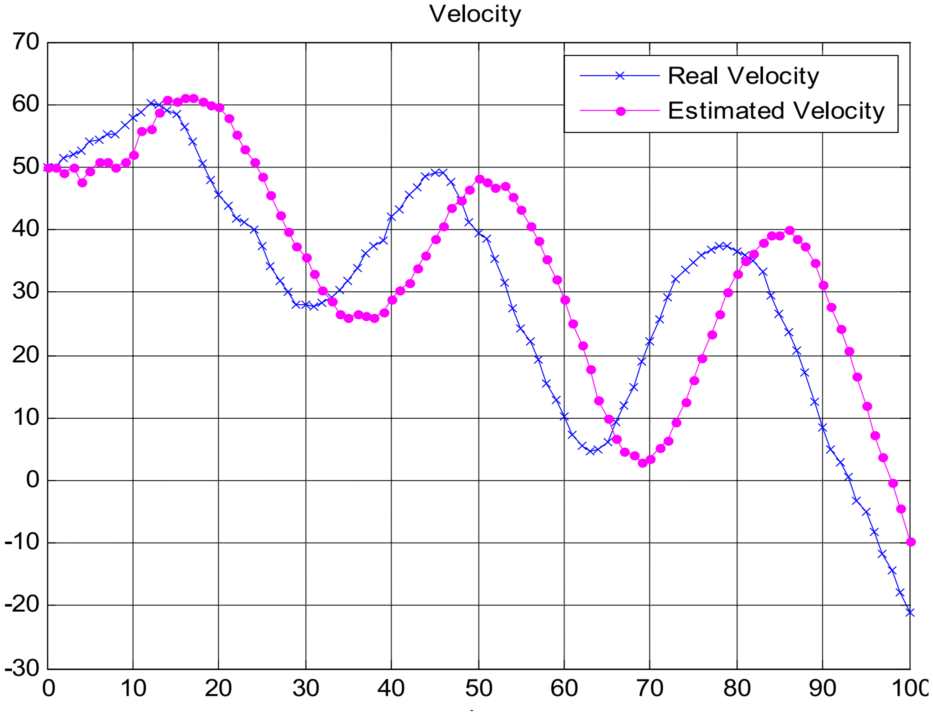
4. Evaluate the performances of the filter

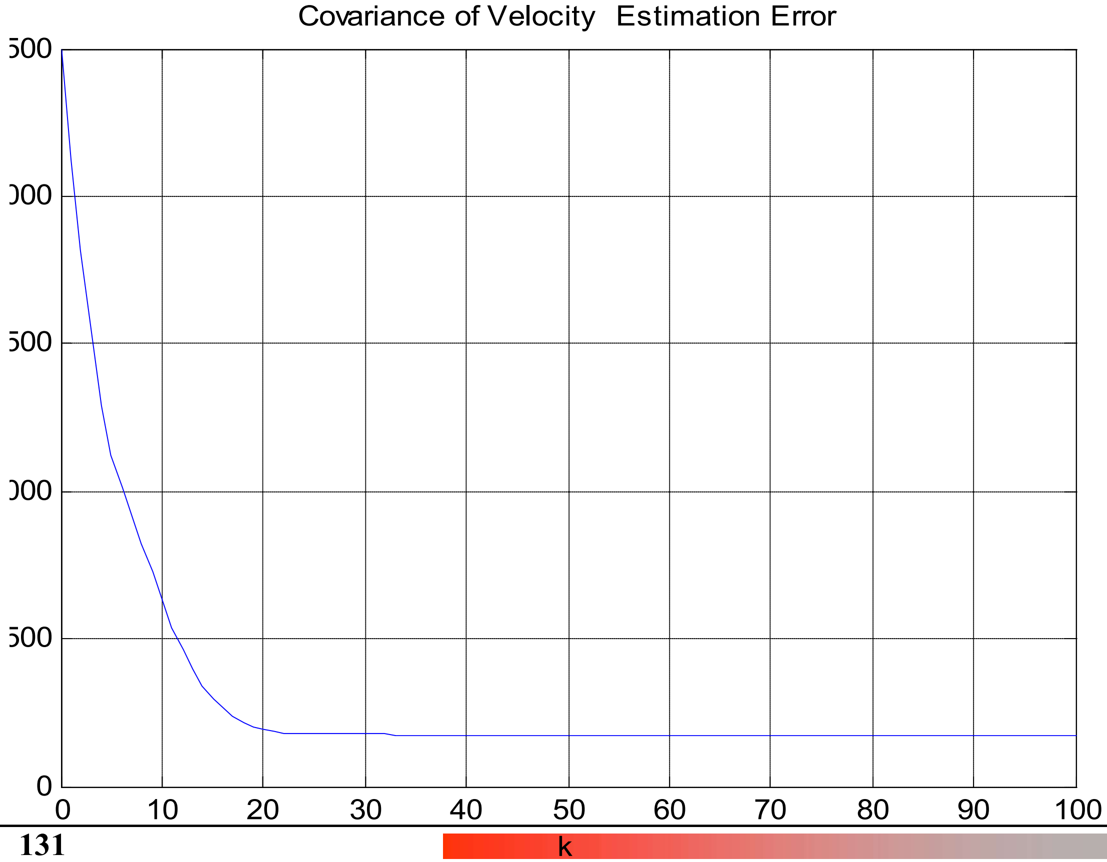
Simulations results: Position



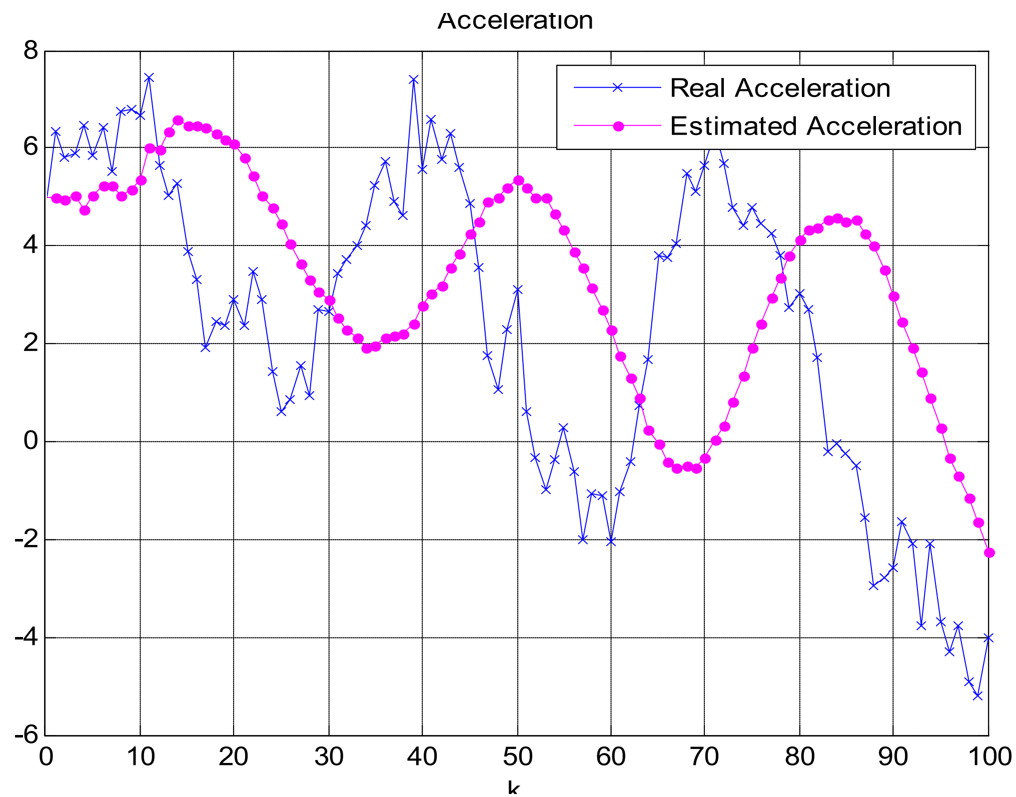


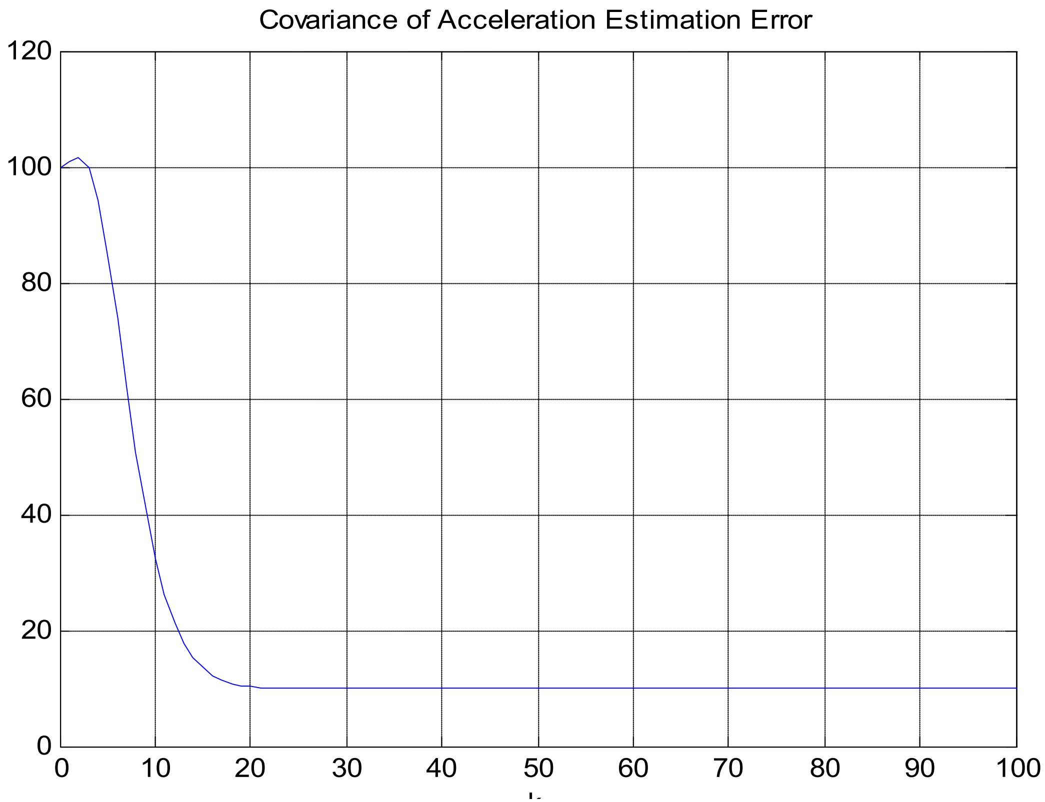
Simulations results: Velocity





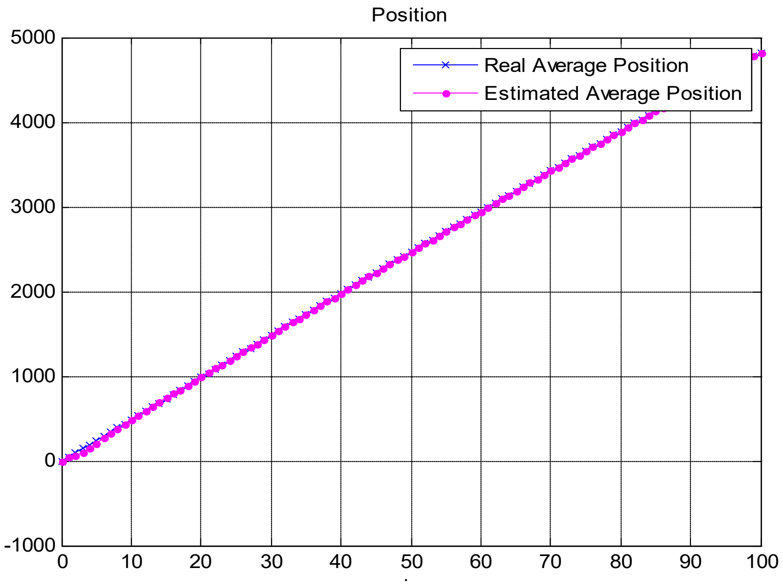
Simulations results: Acceleration

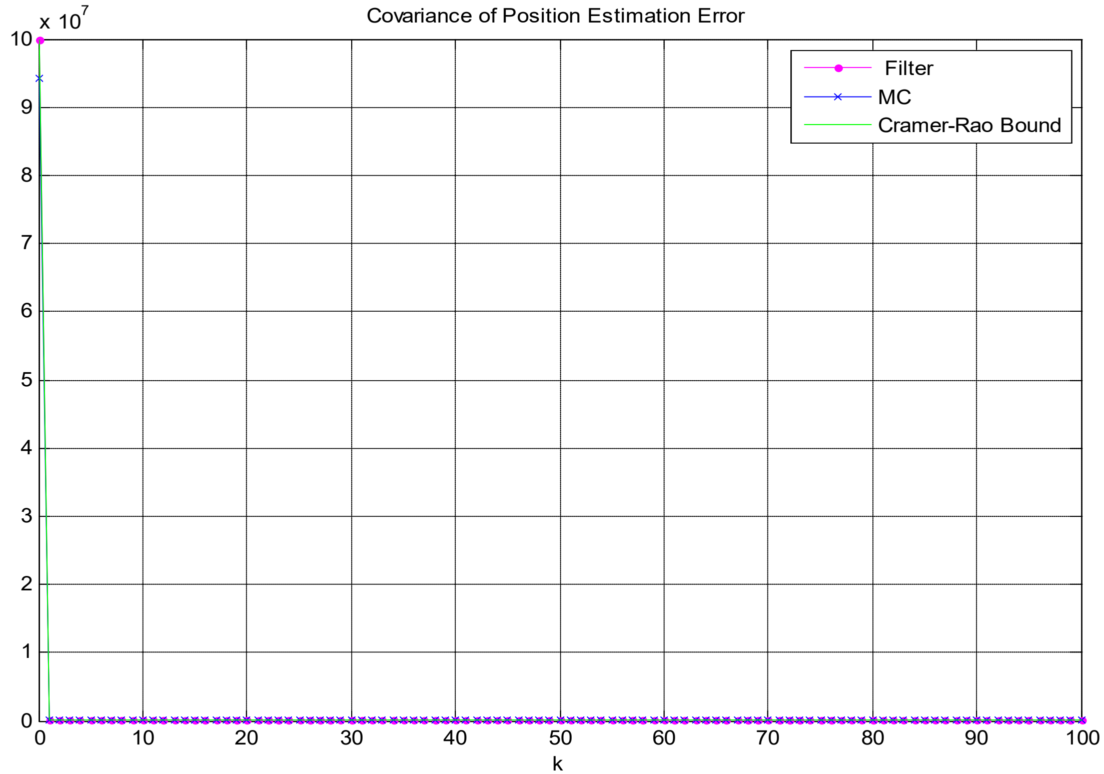


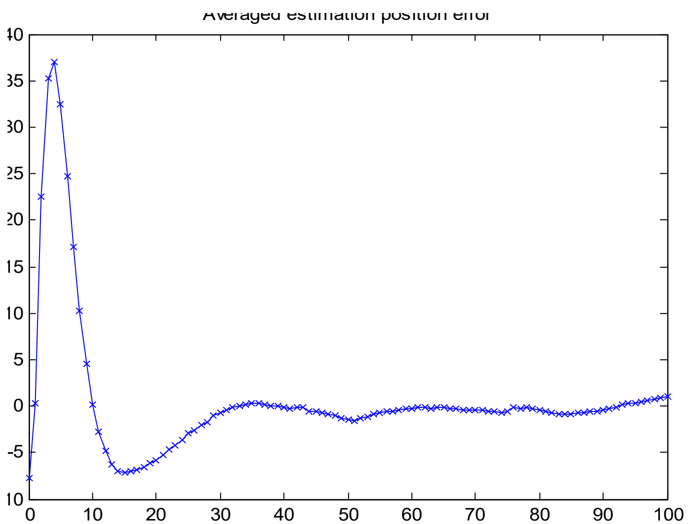


✷ Monte-Carlo simulations: Position

with N_s=1000 and R^F=100R^V



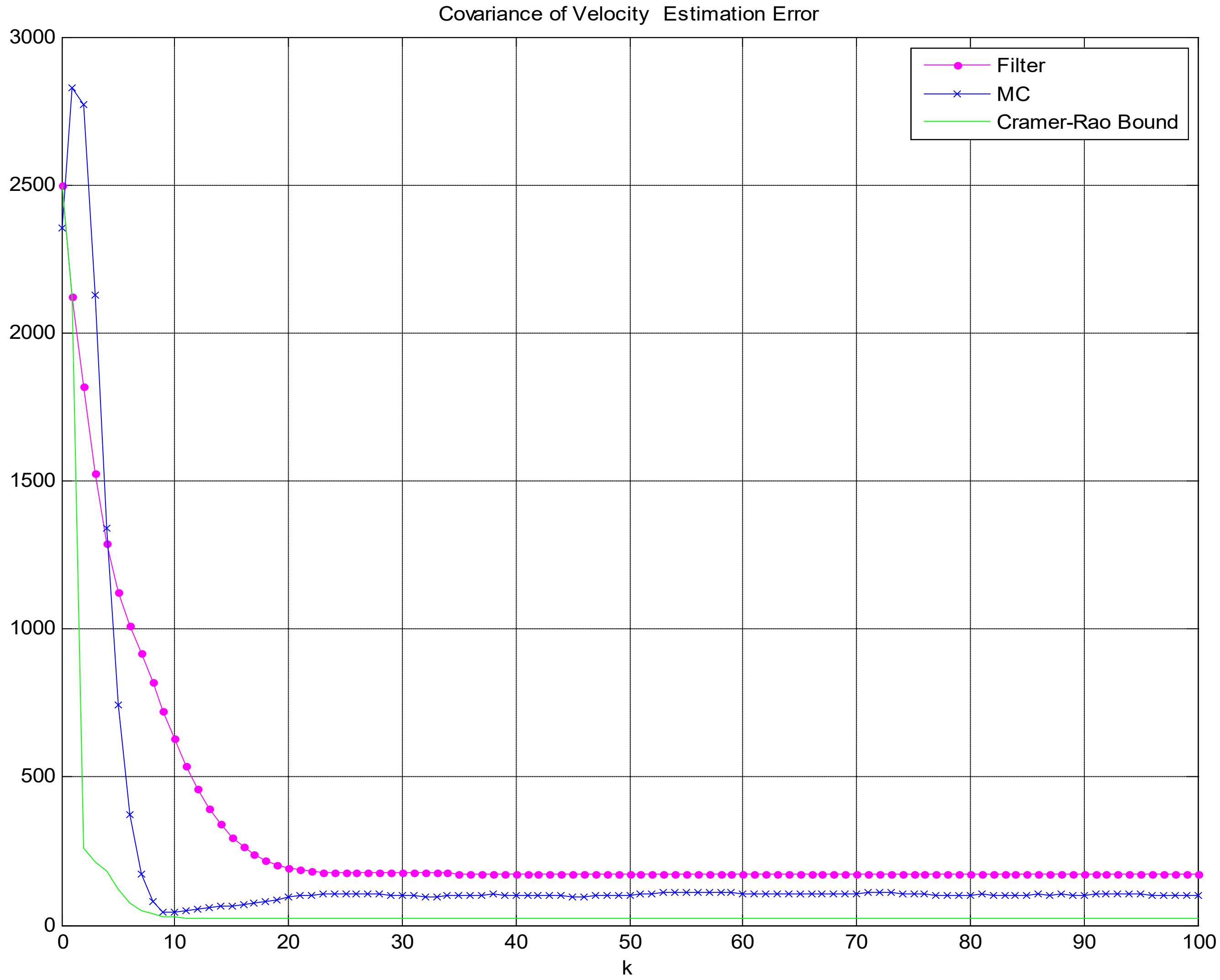




Mechbal N.

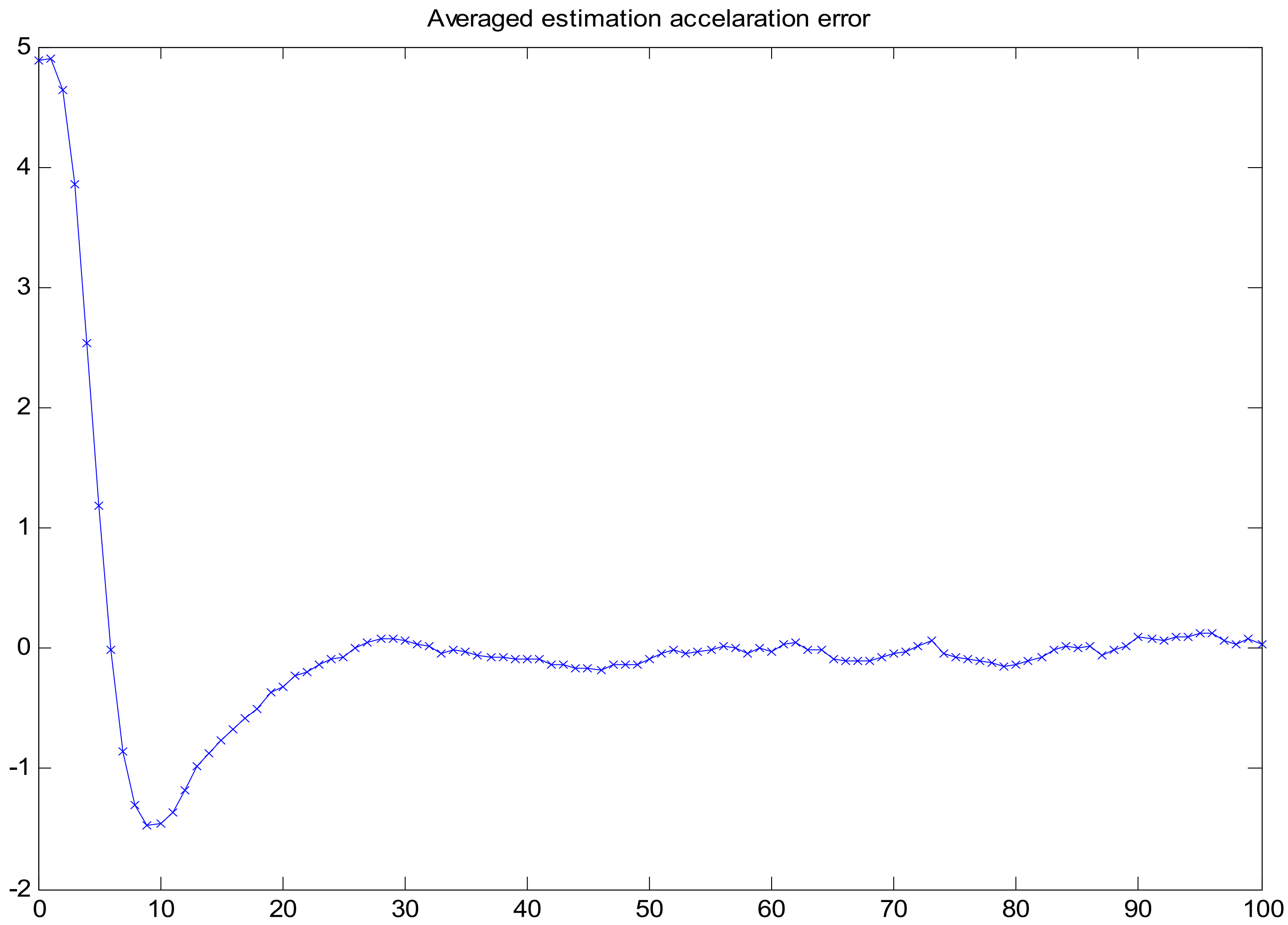
✷ Monte-Carlo simulations: Velocity

with N_s=1000 and R^F=100R



Monte-Carlo simulations: Accelaration

with N_s=1000 and R^F=100R



✷ Monte-Carlo simulations: Accelaration

with N_s=1000 and R^F=100R