16-311-Q Introduction to Robotics Fall'17

Lecture 19: Kalman Filter 2

Instructor:
Gianni A. Di Caro

KF FOR FILTERING MEASUREMENTS FROM ONE SENSOR

- Scenario: The robot does not move, a stream of measures z_K about some quantity $\boldsymbol{\xi}$ of interest is obtained from one if its sensors. The goal is to filter the data stream in order to produce at each time step k the best estimate for the quantity $\boldsymbol{\xi}$ (**Filtering problem**)

- The state-observation equations: the state vector $\boldsymbol{\xi}$ corresponds to the measured quantity of interest. It does not change over time (no system *motion*) apart from small deviations (e.g., because of temperature or robot's vibrations). No controls are issued. For simplicity, we assume that the observation model directly maps the measures into the state vector (i.e., $\boldsymbol{C}$ is the identity matrix and can be therefore removed from the equations). Observations are corrupted by a white Gaussian noise. The resulting system equations are:

$$
\begin{aligned}
\boldsymbol{\xi}_{k+1} &= \boldsymbol{\xi}_k + \boldsymbol{\nu}_k \\
z_{k+1} &= \boldsymbol{\xi}_k + \boldsymbol{w}_k
\end{aligned}
$$

- The KF equations:

At every time step k:

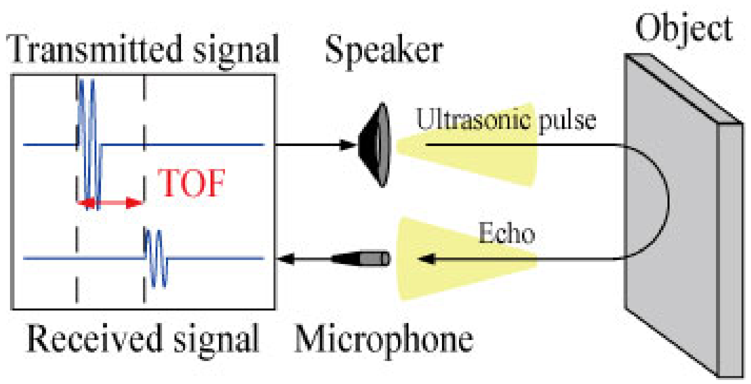
$$
\begin{aligned}
\hat{\boldsymbol{\xi}}_{k+1|k} &= \hat{\boldsymbol{\xi}}_k \\
\boldsymbol{P}_{k+1|k} &= \boldsymbol{P}_k + \boldsymbol{V}_k
\end{aligned}
$$

At every time step $k+1$ when an *observation* z_{k+1} is available:

$$
\begin{aligned}
\hat{\boldsymbol{\xi}}_{k+1} &= \hat{\boldsymbol{\xi}}_{k+1|k} + \boldsymbol{G}_{k+1}(z_{k+1} - \hat{\boldsymbol{\xi}}_{k+1|k}) \\
\boldsymbol{P}_{k+1} &= \boldsymbol{P}_{k+1|k} - \boldsymbol{G}_{k+1}\boldsymbol{P}_{k+1|k} \\
\boldsymbol{G}_{k+1} &= \boldsymbol{P}_{k+1|k}(\boldsymbol{P}_{k+1|k} + \boldsymbol{W}_{k+1})^{-1}
\end{aligned}
$$

RANGE FINDER EXAMPLE

- The robot is not moving and the environment is assumed to be stationary $\rightarrow$ the *true distance*, or the *range and bearing* do not change during the measurement process, apart from maybe environment vibrations, v_k (small white noise)



- The measures from the sensor are affected by Gaussian white noise w_k

- **Proximity sensor** - In case of a simple proximity sensor, the sensor only reports the measured distance d from the closest obstacle, and the state variable has only one component.

- **Range finder** - If a more powerful sensor is available (e.g., a camera), the sensor can measure both the range ρ and bearing angle β of the closest obstacle wrt to the robot. The state variable has two components, which are the same variables measured by the sensor.
  The observation vector is therefore the following, where the $\boldsymbol{C}$ matrix is the identity matrix $\boldsymbol{I}$:

$$z_{k+1} = \begin{bmatrix} \rho_{k+1} & \beta_{k+1} \end{bmatrix}^T = \boldsymbol{\xi}_k + \boldsymbol{w}_k$$

,

VALUES OF THE MODEL MATRICES FOR A DISTANCE SENSOR

- The state transition Matrix $\boldsymbol{A}$: is the transformation factor to obtain the new state from the last state, $\boldsymbol{\xi}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}_k$, but since there is no change dynamics in the system, $\boldsymbol{A} = \begin{bmatrix}1\end{bmatrix}$
- The control matrix $\boldsymbol{B}$: defines how the control inputs affect state changes, $\boldsymbol{\xi}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}_k + \boldsymbol{B}\boldsymbol{u}_k$, however in this case no control actions are executed, therefore $\boldsymbol{B} = \begin{bmatrix}0\end{bmatrix}$
- The observation matrix $\boldsymbol{C}$: multiplies a state vector to translate it to a measurement vector, $\boldsymbol{z}_k = \boldsymbol{C}\boldsymbol{\xi}_k$ but since in this case the measurement d is obtained directly, $\boldsymbol{C} = \begin{bmatrix}1\end{bmatrix}$
- The process covariance matrix $\boldsymbol{V}$: defines how spread state uncertainty is, $\boldsymbol{\xi}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{\nu}_k$. To set it to a reasonable value, some assumptions/knowledge regarding the stability of the distance being measured is needed. In this case, since nothing is moving it's safe to use a small value, such as: $\boldsymbol{V} = \begin{bmatrix}1 \cdot 10^{-5}\end{bmatrix}$
- The measurement covariance matrix $\boldsymbol{W}$: is related to how reliable and stable the sensor is making distance measures, $\boldsymbol{z}_k = \boldsymbol{C}\boldsymbol{\xi}_k + \boldsymbol{w}_k$. We can be conservative, using $\boldsymbol{W} = \begin{bmatrix}1 \cdot 10^{-1}\end{bmatrix}$
- Matrix $\hat{\boldsymbol{\xi}}_0$ is the initial prediction of the distance: This can be based on any a priori knowledge. Let's set it to $\hat{\boldsymbol{\xi}}_0 = \begin{bmatrix}3\end{bmatrix}\mathrm{m}$
- Matrix $\boldsymbol{P}_0$ is the initial prediction of the covariance on the distance estimate: Again, some a priori knowledge would be necessary to set it to a good value. Let's start with a value which corresponds to about 30% of the initial prediction: $\boldsymbol{P}_0 = \begin{bmatrix}1\end{bmatrix}\mathrm{m}$

FILTER'S EQUATIONS FOR SCALAR DISTANCE MEASUREMENT

Blue = Inputs, Red = Outputs, **Black = Constant Parameters**, Gray = Working variables

State prediction
(Predict where the system state will be)

$$\hat{d}_{k+1|k} = \hat{d}_k$$

Covariance prediction
(Predict the amount of error in state prediction)

$$P_{k+1|k} = P_k + 1 \cdot 10^{-5}$$

Innovation
(Compare reality against prediction)

$$\nu_{k+1} = d_{k+1} - \hat{d}_{k+1|k}$$

Innovation covariance
(Compare real error against prediction)

$$\Sigma_{\nu_{k+1}} = P_{k+1|k} + 1 \cdot 10^{-1}$$

Kalman gain
(Rescale/weight the prediction)

$$G_{k+1} = P_{k+1|k} \Sigma_{\nu_{k+1}}^{-1}$$

State update
(New estimate of the system state)

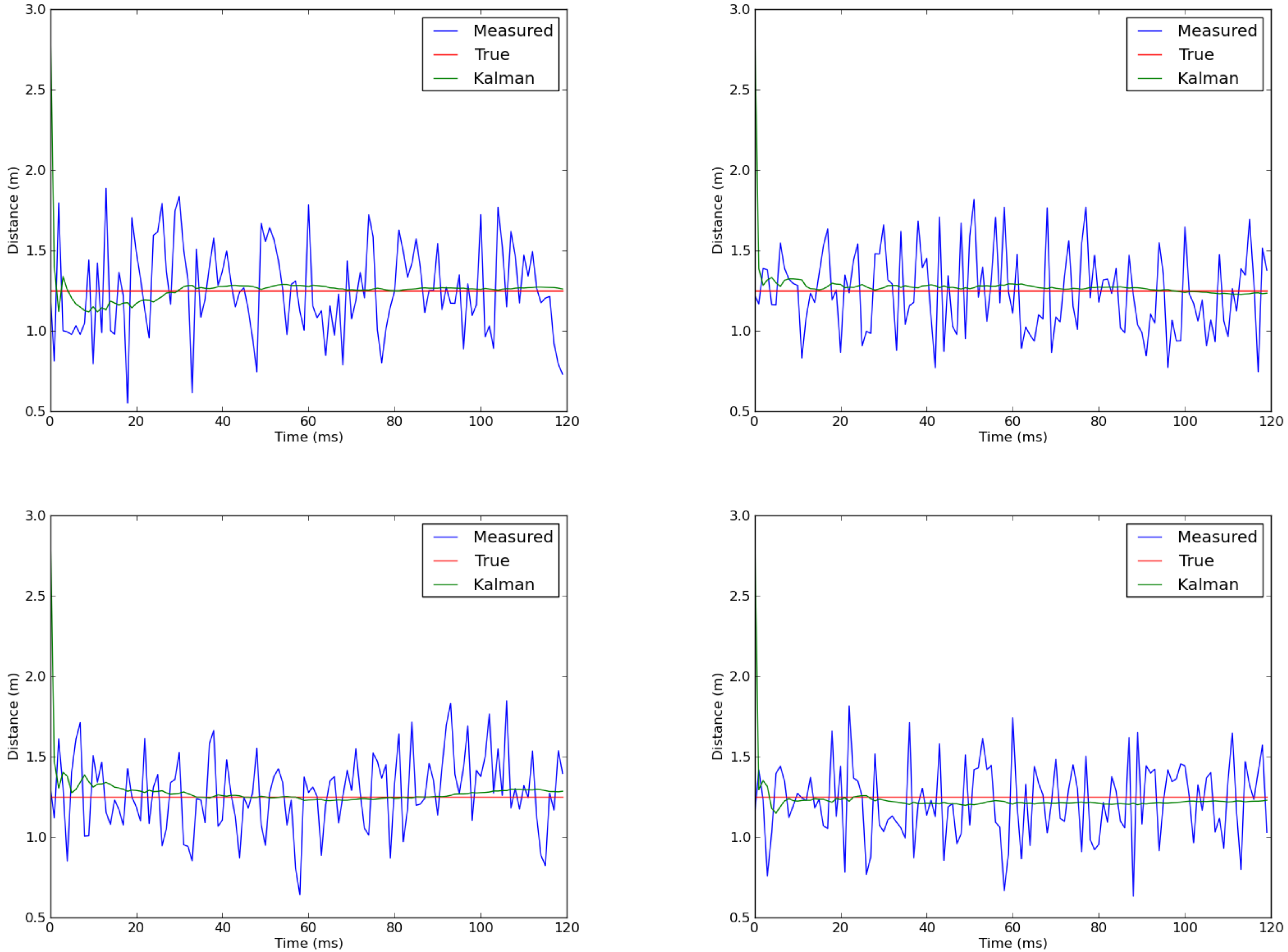
$$\hat{d}_{k+1} = \hat{d}_{k+1|k} + G_{k+1} \nu_{k+1}$$

Covariance update
(New estimate of error)

$$P_{k+1} = (1 - G_{k+1}) P_{k+1|k}$$

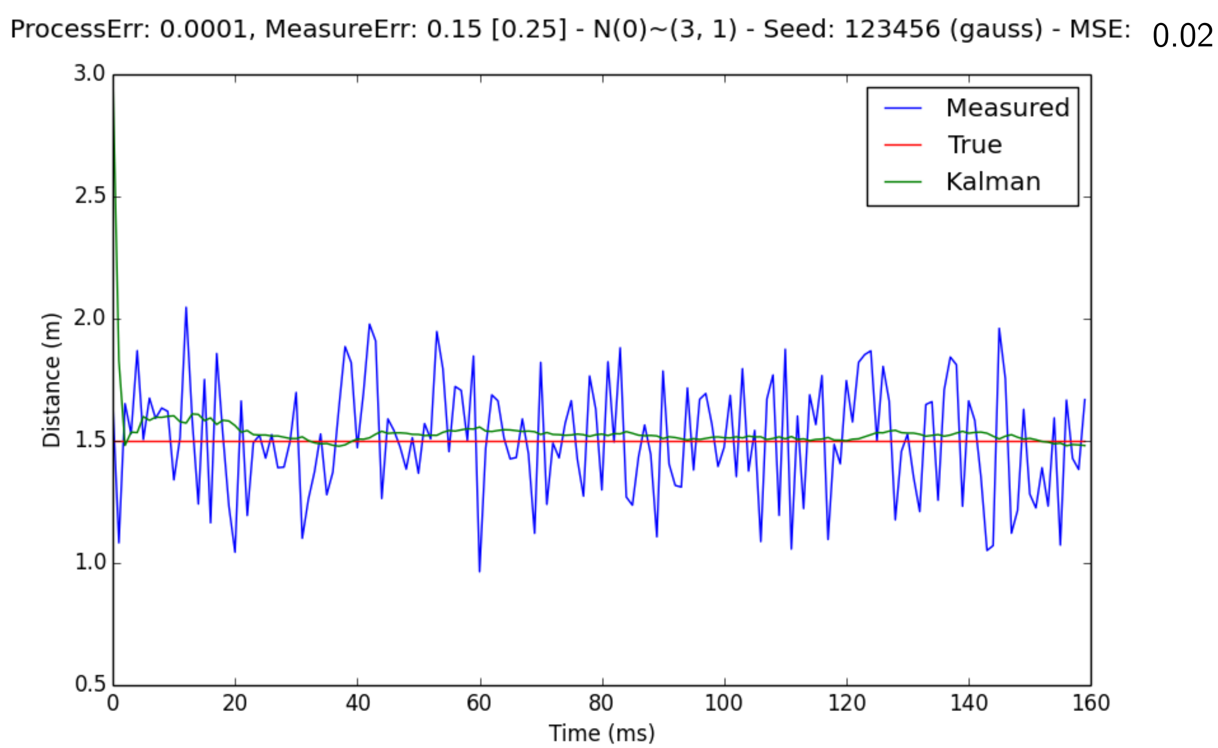
PERFORMANCE OF KF ON SCALAR DISTANCE MEASURES

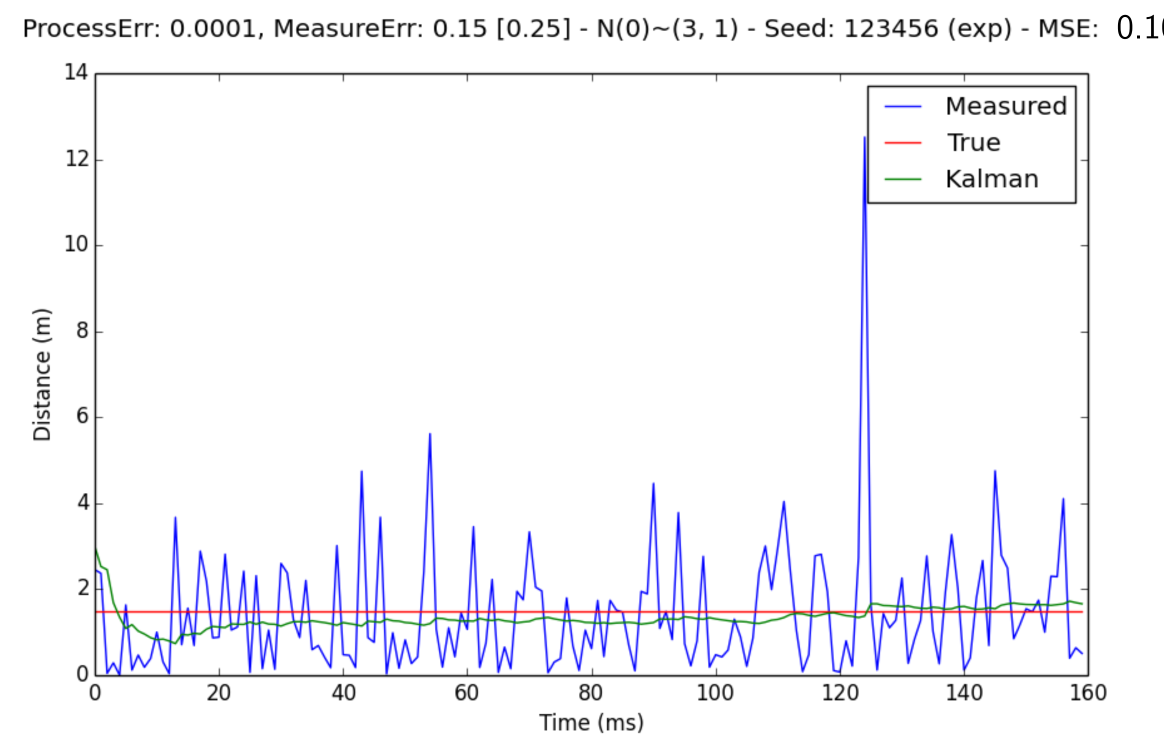
Different random realizations of the same process and filter

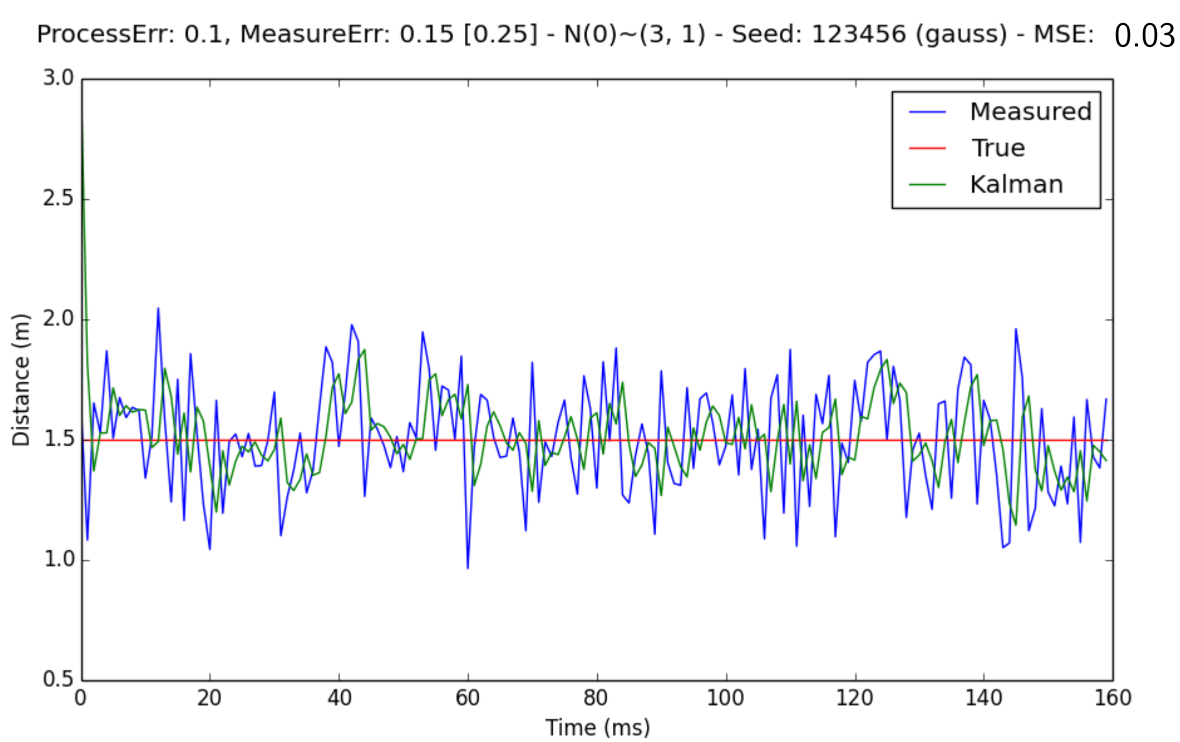


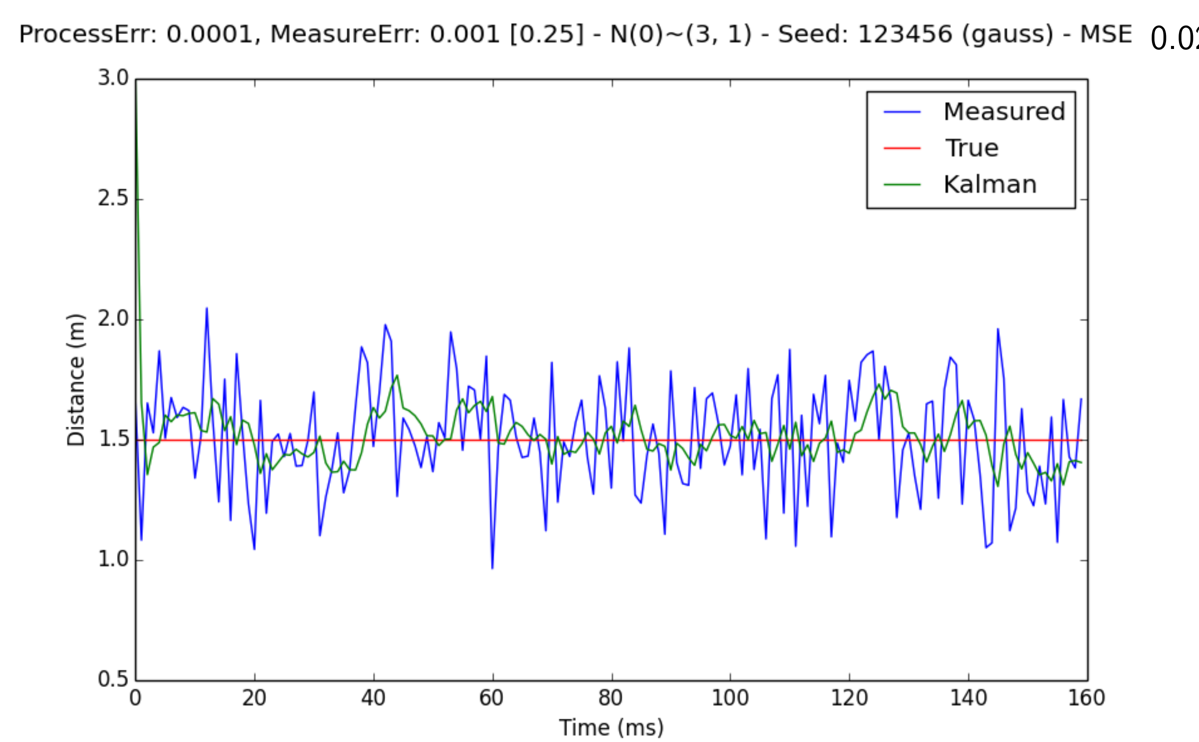
PERFORMANCE OF KF ON SCALAR DISTANCE MEASURES

Different parameters for the process and the filter



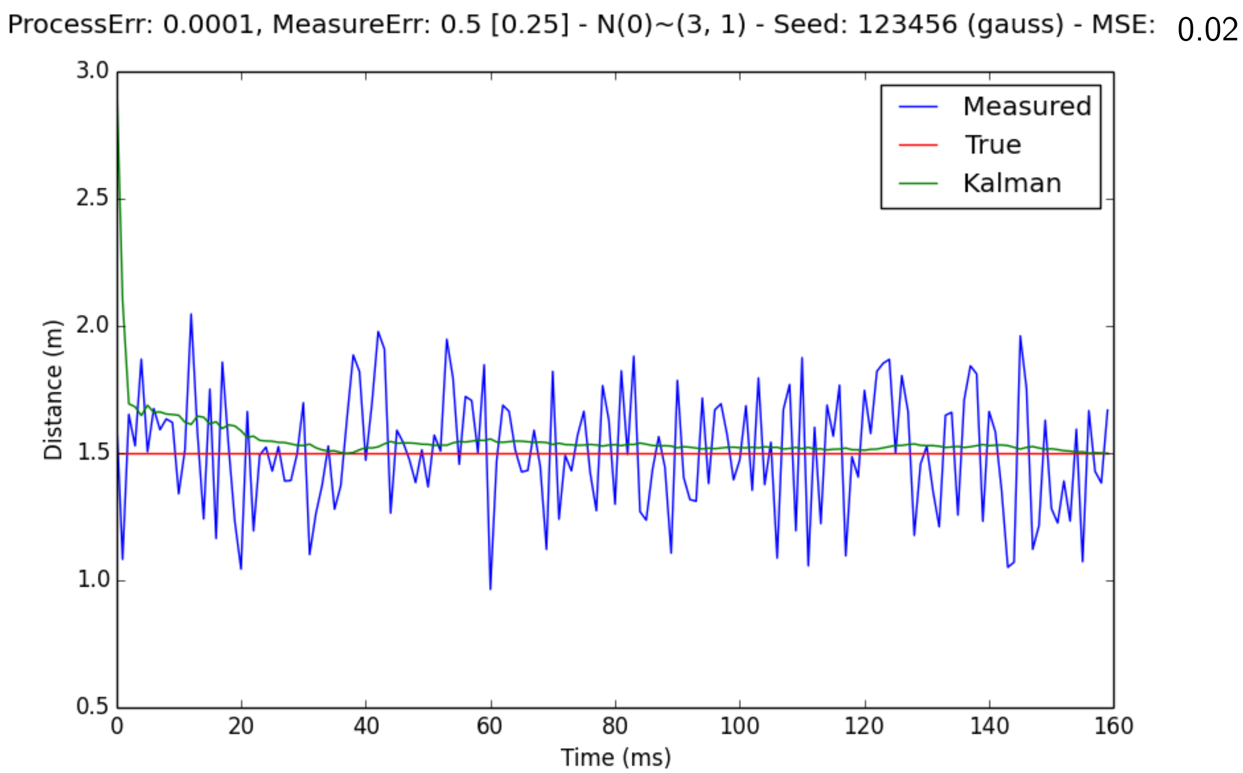


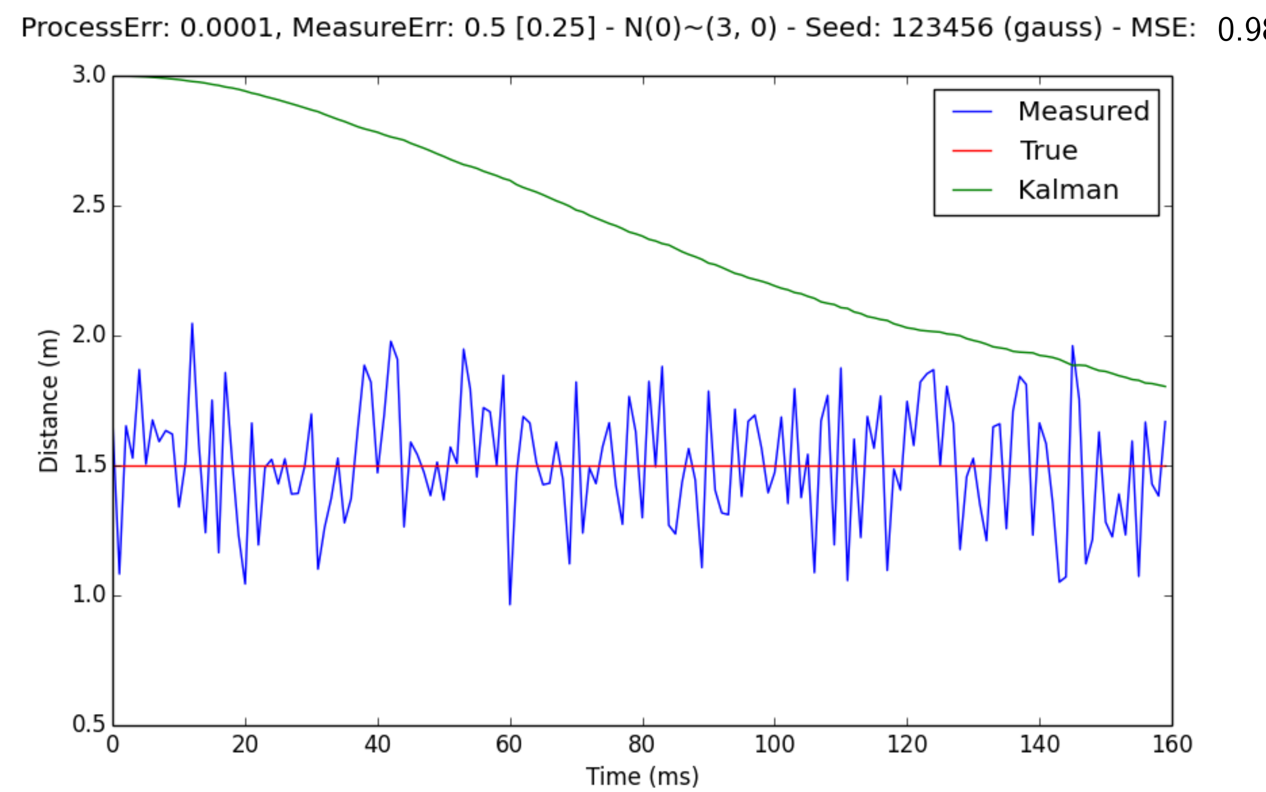


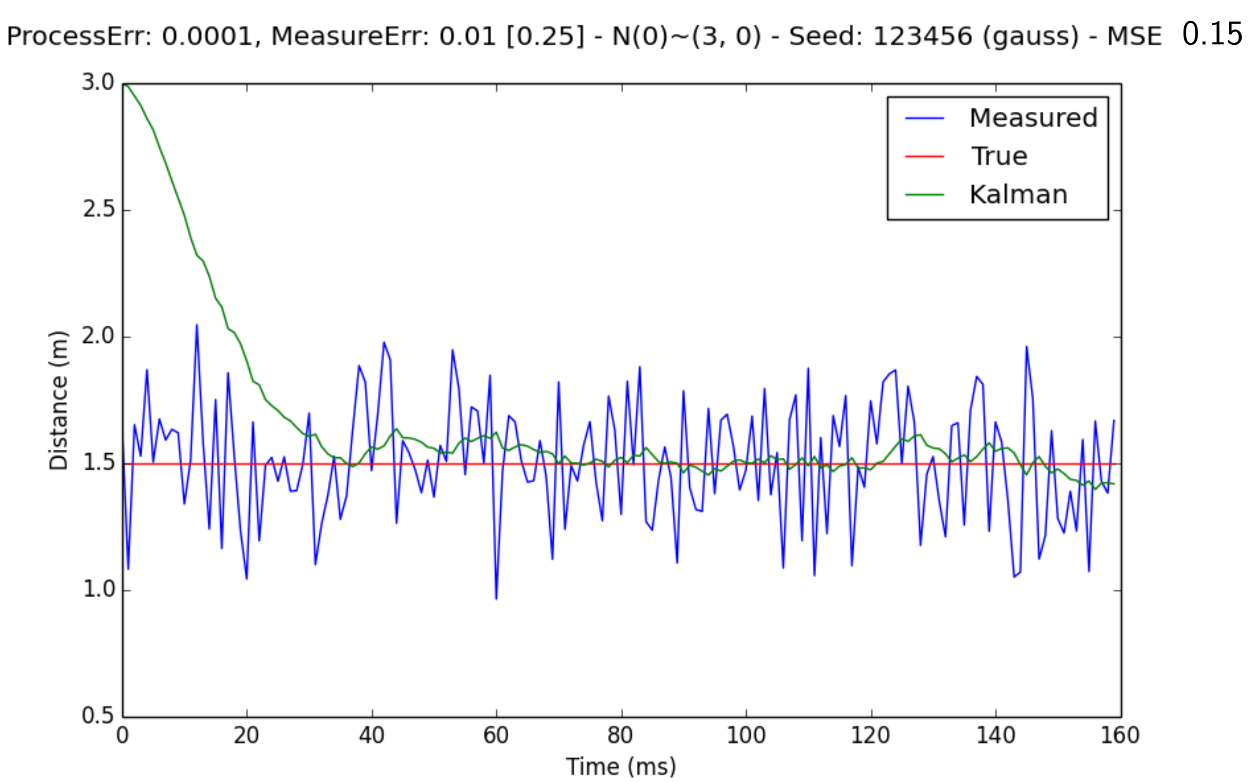


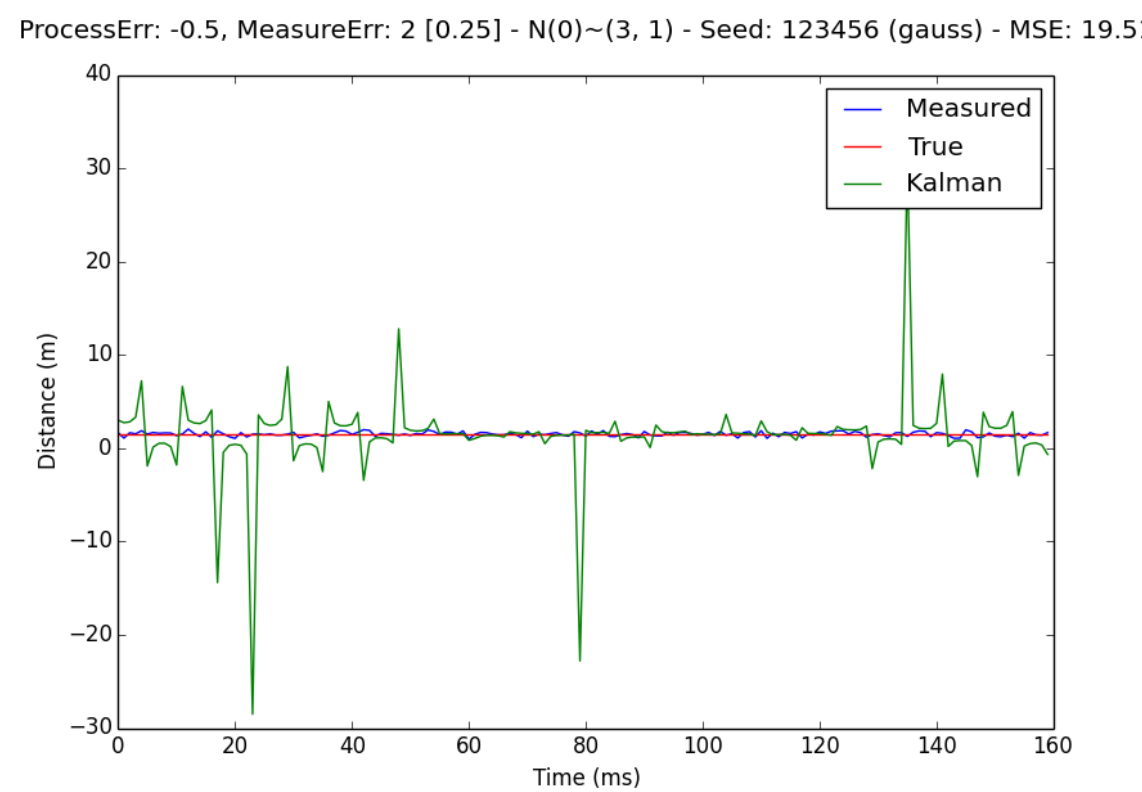
PERFORMANCE OF KF ON SCALAR DISTANCE MEASURES

Different parameters for the process and the filter









EQUATIONS' ANALYSIS FOR THE SCALAR FILTERING PROBLEM

Scenario: The state, the measures, and the controls are all scalars. One scalar measure is obtained when an observation is available. For simplicity, all the linear coefficients are set to 1. If no controls are issued, the scenario is the same as in the case of the scalar filtering of a stream of measures (*Scalar filtering problem*)

The state-observation equations:

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{u}_k + \boldsymbol{\nu}_k$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{w}_k$$

The KF equations:

At every time step k:

$$\widehat{\boldsymbol{\xi}}_{k+1|k} = \widehat{\boldsymbol{\xi}}_k + \boldsymbol{u}_k$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{P}_k + \boldsymbol{V}_k$$

At every time step $k+1$ when an *observation* $\boldsymbol{z}_{k+1}$ is available:

$$\widehat{\boldsymbol{\xi}}_{k+1} = \widehat{\boldsymbol{\xi}}_{k+1|k} + \boldsymbol{G}_{k+1}(\boldsymbol{z}_{k+1} - \widehat{\boldsymbol{\xi}}_{k+1|k})$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{G}_{k+1}\boldsymbol{P}_{k+1|k}$$

$$\boldsymbol{G}_{k+1} = \boldsymbol{P}_{k+1|k}(\boldsymbol{P}_{k+1|k} + \boldsymbol{W}_{k+1})^{-1}$$

In scalar, one-dimensional notation, the equations become:

Prediction update $\begin{cases} \xi \leftarrow \xi + u \\ \sigma_\xi^2 \leftarrow \sigma_\xi^2 + \sigma_\nu^2 \end{cases}$

Measurement correction $\begin{cases} G \leftarrow \dfrac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_w^2} \\ \xi \leftarrow \xi + G(z - \xi) \\ \sigma_\xi^2 \leftarrow (1 - G)\sigma_\xi^2 \end{cases}$

EQUATIONS' ANALYSIS FOR THE SCALAR FILTERING PROBLEM

Prediction update $\begin{cases} \xi \leftarrow \xi + u \\ \sigma_\xi^2 \leftarrow \sigma_\xi^2 + \sigma_\nu^2 \end{cases}$ **Measurement correction** $\begin{cases} \boxed{G \leftarrow \dfrac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_w^2}} \\ \xi \leftarrow \xi + G(z - \xi) \\ \sigma_\xi^2 \leftarrow (1 - G)\sigma_\xi^2 \end{cases}$

Variance of the state estimate:

$$\sigma_\xi^2 = \left(1 - \frac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_w^2}\right)\sigma_\xi^2 = \frac{\sigma_w^2 \sigma_\xi^2}{\sigma_\xi^2 + \sigma_w^2} \quad \Longrightarrow \quad \boxed{\frac{1}{\sigma_\xi^2} \leftarrow \frac{1}{\sigma_\xi^2} + \frac{1}{\sigma_w^2}}$$

(Mean) State estimate:

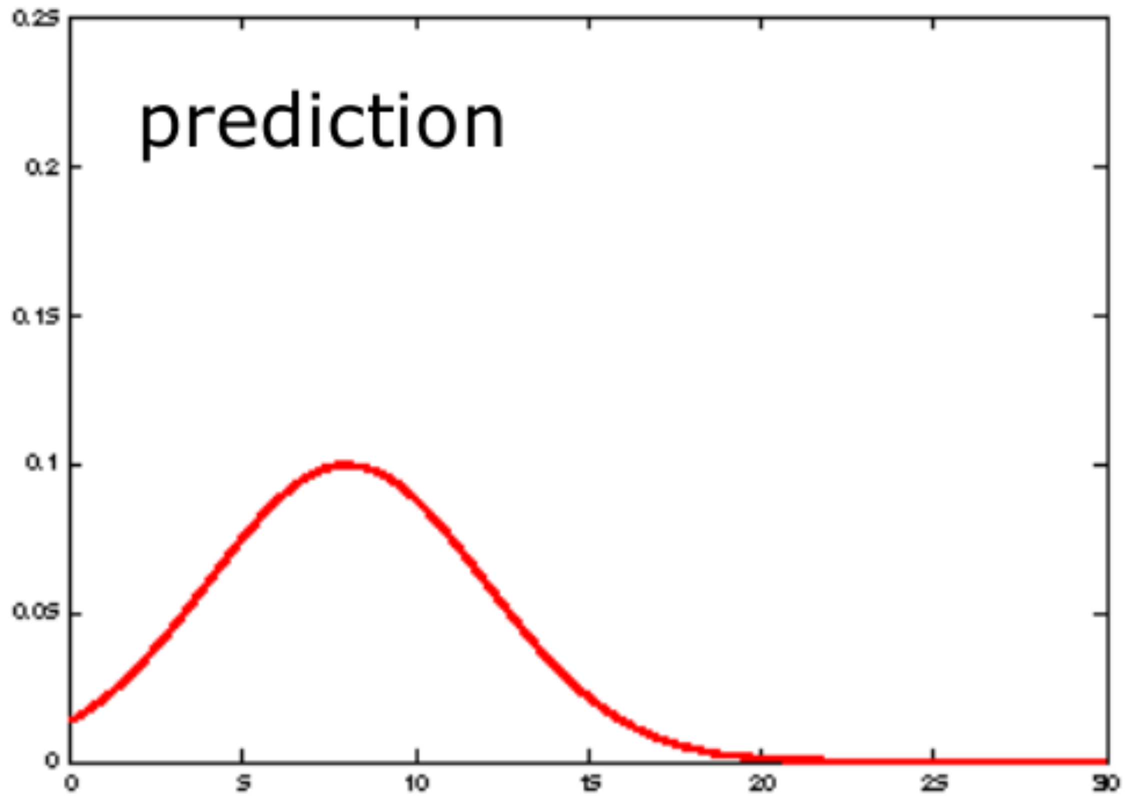
$$\xi \leftarrow (1 - G)\xi \; + \; Gz \quad = \left(\frac{\sigma_w^2}{\sigma_\xi^2 + \sigma_w^2}\right)\xi + \left(\frac{\sigma_\xi^2}{\sigma_\xi^2 + \sigma_w^2}\right)z$$

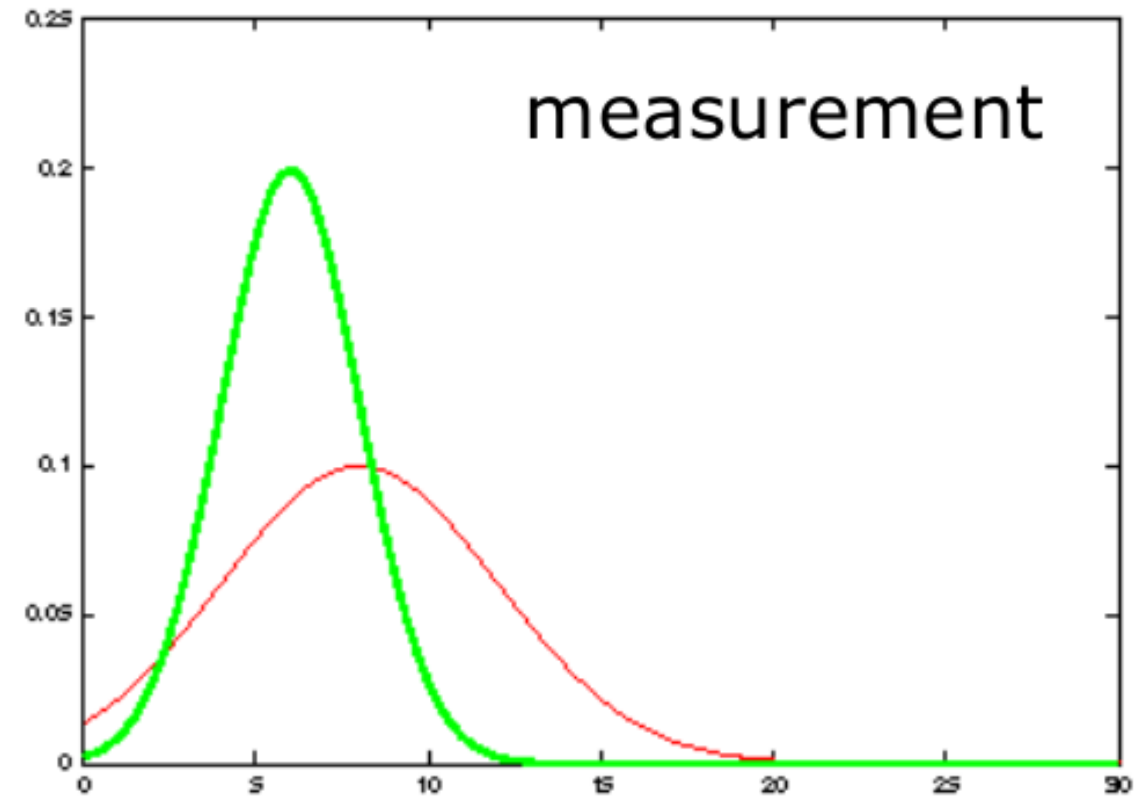
$$= \left(\frac{\sigma_\xi^2 \sigma_w^2}{\sigma_\xi^2 + \sigma_w^2}\right)\left(\frac{\xi}{\sigma_\xi^2} + \frac{z}{\sigma_w^2}\right)$$

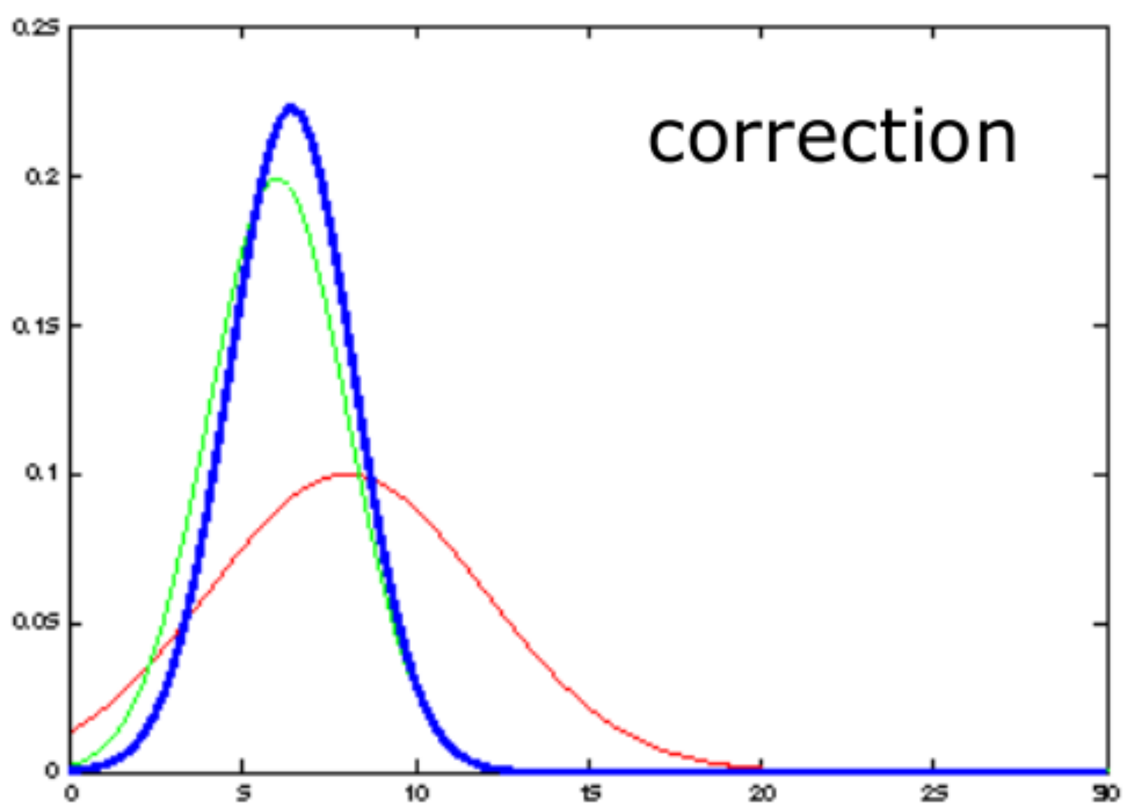
$$\boxed{\xi \leftarrow \left(\frac{1}{\sigma_\xi^2} + \frac{1}{\sigma_w^2}\right)^{-1}\left[\frac{1}{\sigma_\xi^2}\xi + \frac{1}{\sigma_w^2}z\right]}$$

Weighted arithmetic mean, $\dfrac{w_1 x_1 + w_2 x_2}{w_1 + w_2}$, w_i proportional to inverse of variance

EXAMPLE OF KF BELIEF EVOLUTION IN A SCALAR CASE

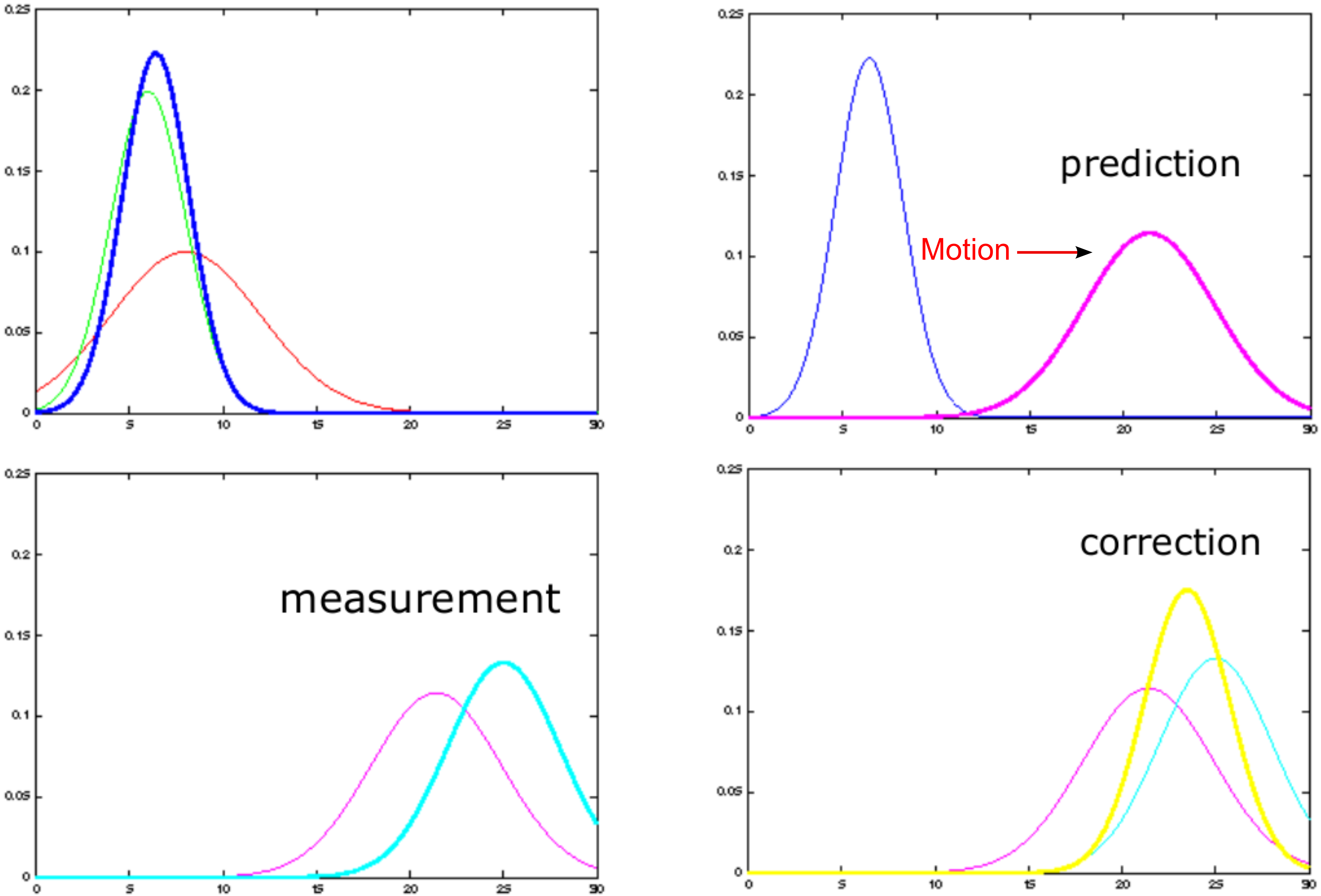






It's a weighted mean!

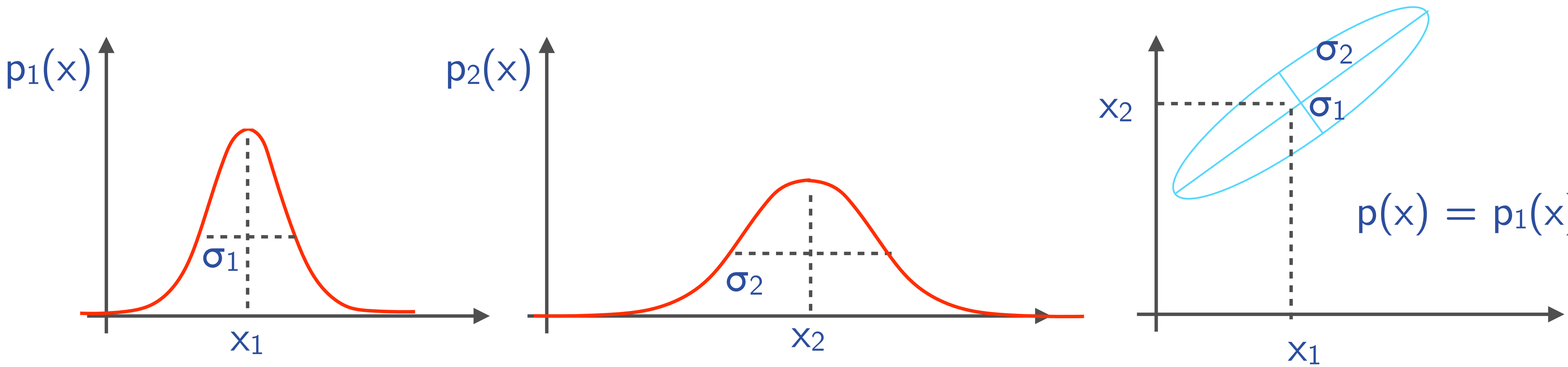
EXAMPLE OF KF BELIEF EVOLUTION IN A SCALAR CASE



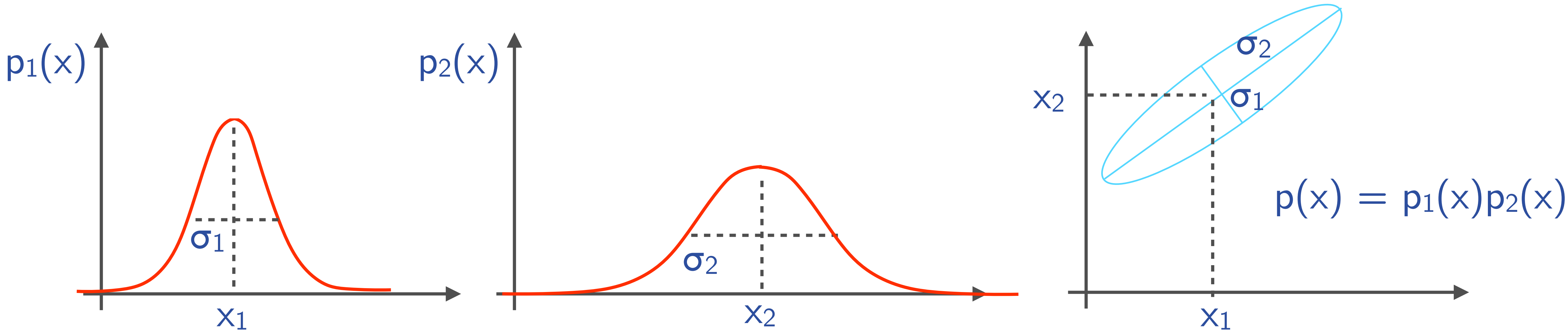
EQUATIONS' ANALYSIS FOR THE SCALAR FILTERING PROBLEM

- $\hat{\xi}_{k+1|k}$ and z_{k+1} (i.e. ξ and z above before state update) are two normal distributed, independent RVs. ξ represents the current state prediction (out of the process model and past history), while z is the current state measure.

- They can be also thought as two readings (x_1, σ_1) and (x_2, σ_2) from two independent instruments with different precision, or as two sequential independent readings made with different precision from the same instrument. All the readings are about the same quantity to be estimated, which is the true state of the system



In any chosen mental or practical representation, the question is how to ***combine*** the ξ and z readings/estimates (and their variances) into a ***new, single state estimation that best represents the information from ξ and z***



$p(x)$ is the probability of a value x given the readings (or the estimates) x_1 and x_2.

$$p(x) = N(x_1,\sigma_1)N(x_2,\sigma_2)$$

$$p(x) = Ce^{-\frac{1}{2}\frac{\sigma_1^2+\sigma_2^2}{\sigma_1^2\sigma_2^2}\left[x-\frac{(x_1\sigma_2^2+x_2\sigma_1^2)}{\sigma_1^2+\sigma_2^2}\right]^2}$$

The most probable result, that best represent the data in the ML sense, corresponds to the distribution center:

$$\widehat{x} = \frac{(x_1\sigma_2^2+x_2\sigma_1^2)}{\sigma_1^2+\sigma_2^2}$$

With variance: $\frac{1}{\widehat{\sigma}^2} = \frac{\sigma_1^2+\sigma_2^2}{\sigma_1^2\sigma_2^2}$

This is precisely the result of the new state estimate produced by the KF!

$$\boxed{\xi \leftarrow \left(\frac{1}{\sigma_\xi^2}+\frac{1}{\sigma_w^2}\right)^{-1}\left[\frac{1}{\sigma_\xi^2}\xi+\frac{1}{\sigma_w^2}z\right] \qquad \frac{1}{\sigma_\xi^2} \leftarrow \frac{1}{\sigma_\xi^2}+\frac{1}{\sigma_w^2}}$$

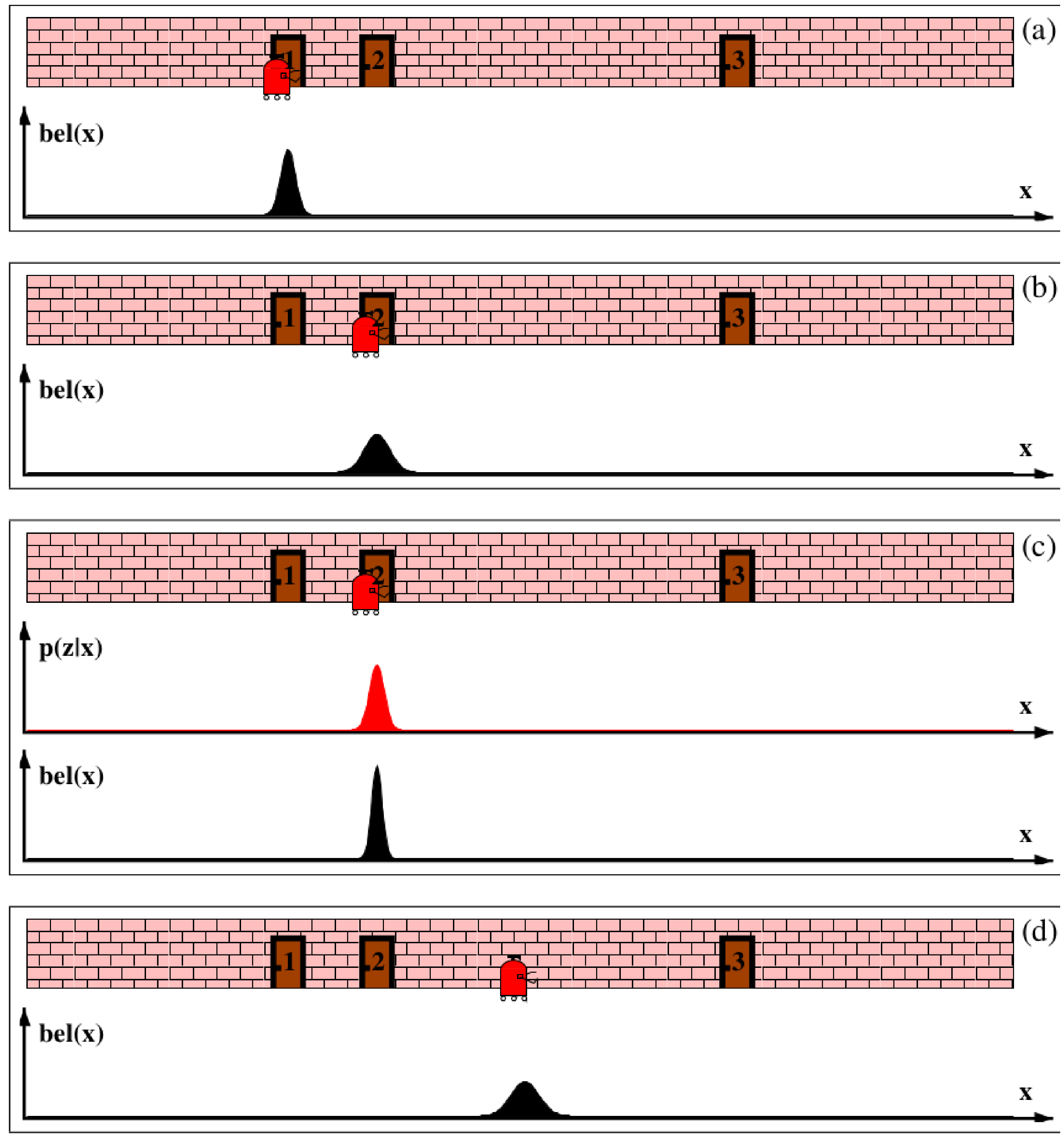
EXAMPLE FOR ROBOT LOCALIZATION

$$bel(x_0) = N(\hat{x}_0, \sigma_0^2)$$

$$\overline{bel}(x_1) = \begin{cases} \hat{x}_{1|0} = & A\hat{x}_0 + Bu_1 \\ \sigma_{1|0}^2 = & A^2\sigma_0^2 + \sigma_{action}^2 \end{cases}$$

$$bel(x_1) = \begin{cases} \hat{x}_1 = \hat{x}_{1|0} + G_1(\hat{x}_{z_1} - \hat{x}_{1|0}) \\ \sigma_1^2 = (1 - G_1)\sigma_{1|0}^2 \end{cases}$$

$$\overline{bel}(x_2) = \begin{cases} \hat{x}_{2|1} = & A\hat{x}_1 + Bu_2 \\ \sigma_{2|1}^2 = & A^2\sigma_1^2 + \sigma_{action}^2 \end{cases}$$



OBSERVATIONS ON LOW-PASS FILTERS

- Without a process dynamics with noise, the Kalman filter implements a recursive ML estimator $\rightarrow$ **Least squares regression filter**, which is a *low-pass filter* with a *variable gain*
- A basic low-pass filter for the estimation of the variable x based on the inputs z, looks like:
$$x_{n+1} = Gz_{n+1} + (1-G)x_n = x_n + G(z_{n+1} - x_n), \qquad G \in [0,1]$$
when the *constant smoothing parameter* G is 1, then no smoothing is performed at all.
- The above discrete-time implementation of a basic low-pass filter is an exponentially-weighted moving average
- The smoothing parameter G determines the weight that a sample input has in the exponential average. That is, G defines the *decay* in the weight (importance) of a sample
- G defines the number of the most recent samples that will really affect the average:
$$x_n = \sum_{i=1}^{n-1} G(1-G)^{n-i} z_i + Gz_n$$
After n samples, the weight of the $i-$th sample, with $n > i$, is: $G(1-G)^{n-i}$. For instance, for $G = 0.1$ approximately only the latest 50 observations will really influence the estimate.
- In the terminology of signal processing, G is related to the *time constant* τ (related to the *cut-off frequency* $f_c = \frac{1}{2\pi\tau}$) of the filter and to the sampling rate $\frac{1}{\Delta T}$ through: $G = \frac{\Delta T}{\tau + \Delta T}$

EXAMPLE: MEASURES FROM AN ACCELEROMETER

- A triple axis accelerometer provides 3D acceleration data. However, the three measures are obtained through independent circuitry, therefore each dimension can be treated independently.

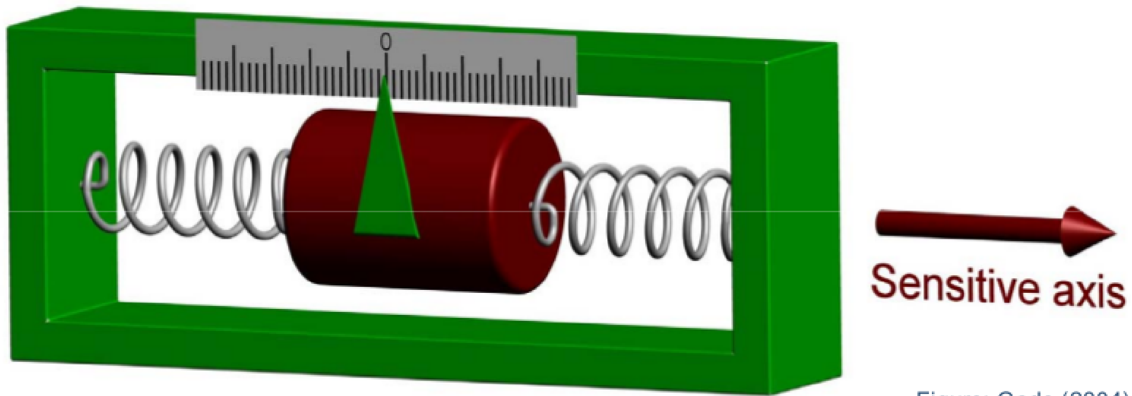


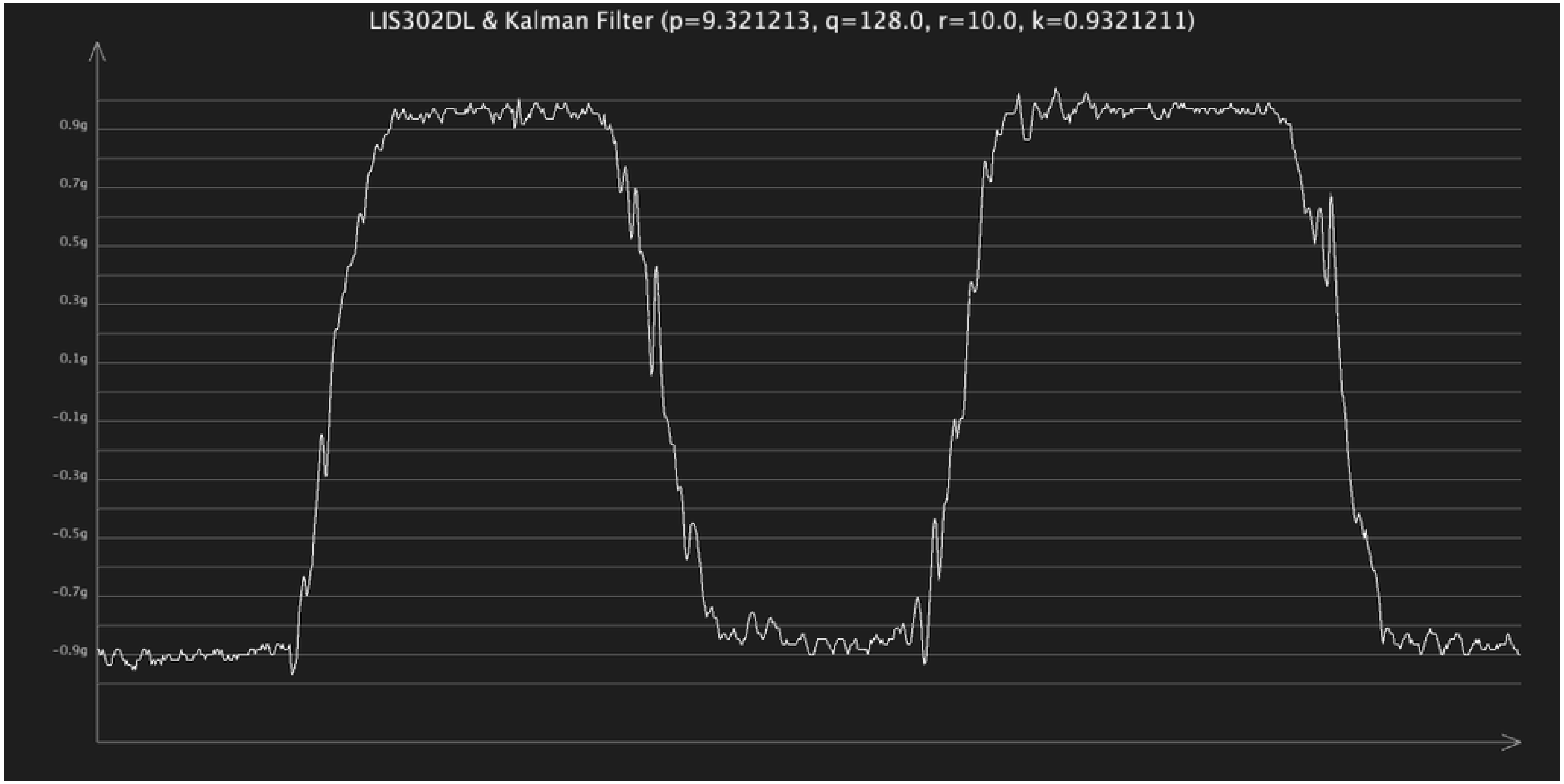
Figure: Gade (2004)

- A 1D Kalman filter is used to estimate the correct acceleration a along a single axis based on the stream of input measures. No control actions are included, $\boldsymbol{Bu} = \mathbf{0}$, such that the state dynamics reduces to: $\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \boldsymbol{\nu}$
- σ_ν^2 and σ_w^2 represent respectively process and sensor noise, which are given **parameters**
- $\sigma_{a_k}^2$, the error in the estimate, must be initialized, but its initial value is not critical since it gets adjusted during the operations, however it needs to be set high enough at the beginning
- The initial value for the estimate a_0 is also not very important for the correct execution of the algorithm

The values for the process and sensor noise are critical to get a good behavior.
How do we set them?

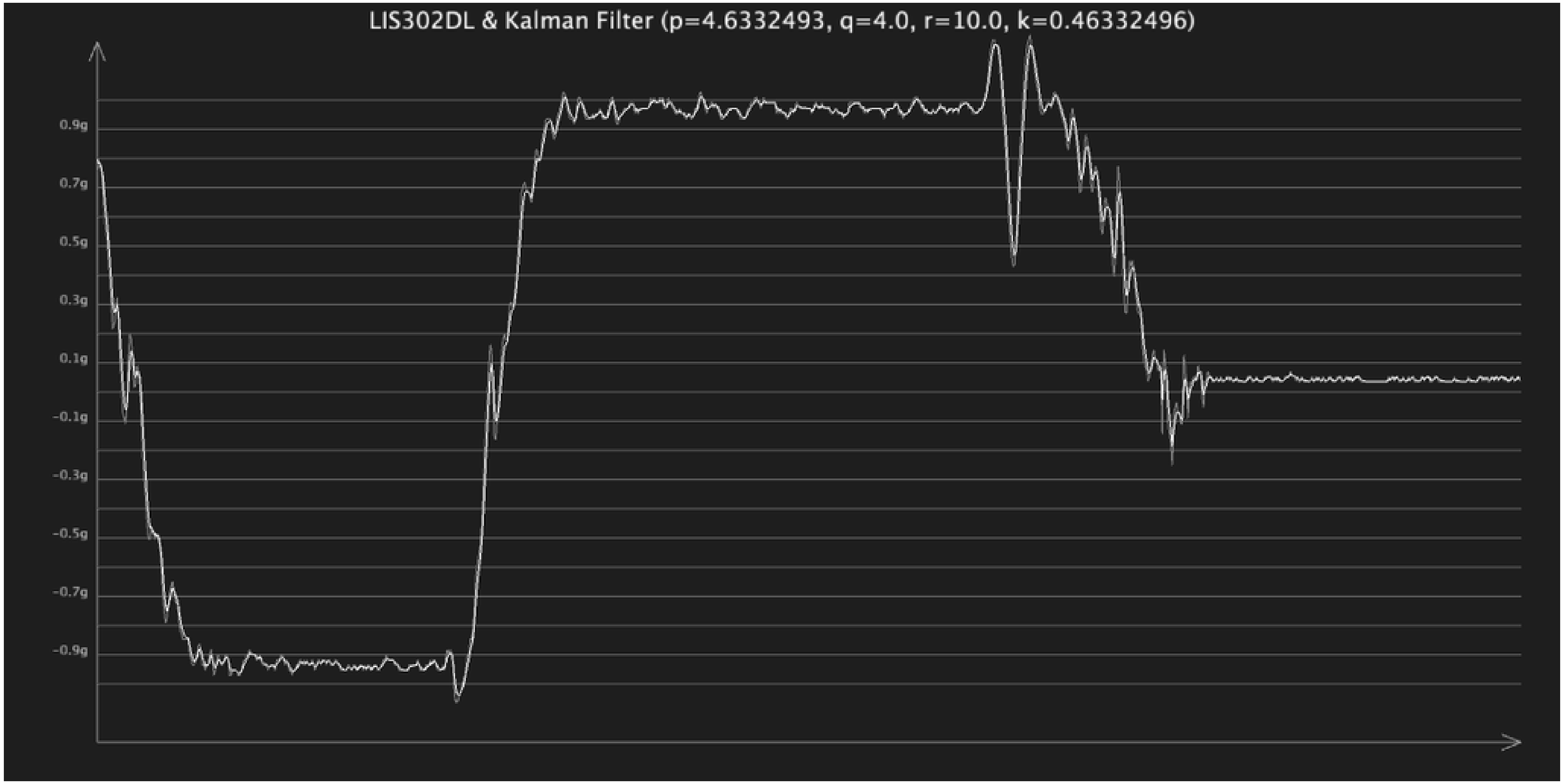
FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_\nu^2 = 128$ (max value) - Sensor noise, $\sigma_w^2 = 10$ ($\approx 8\%$)
- Observations: nearly no difference between filtered data (white) and original data (gray), since the filter cannot smooth the data due to the high process error, it can only follow the data.



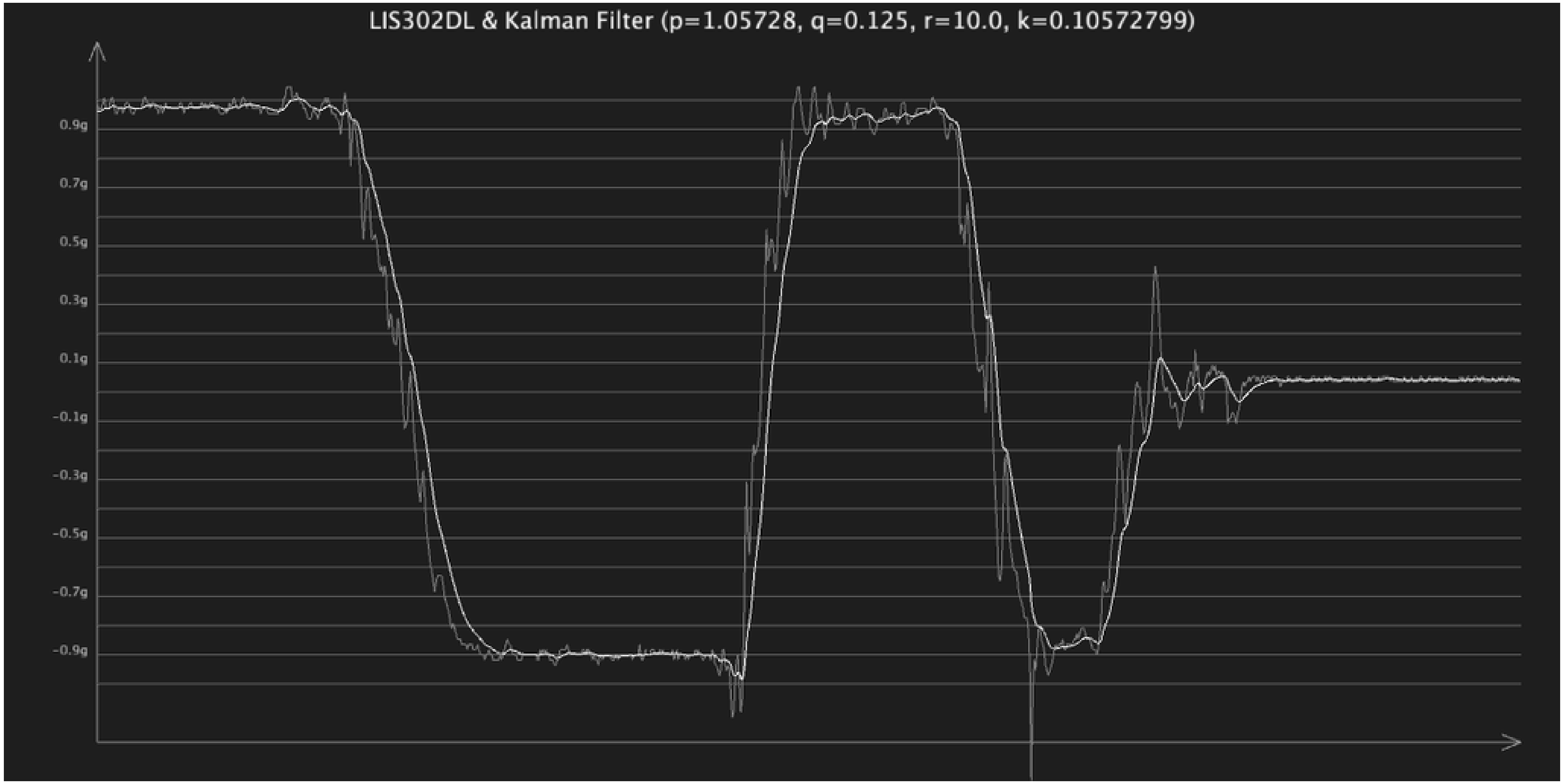
FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_v^2 = 4$ ($\approx 3\%$) - Sensor noise, $\sigma_w^2 = 10$ ($\approx 8\%$)
- Observations: Filtered values are pretty close to real values, but start to show less noise, with the sensor data often overshooting the filtered ones.



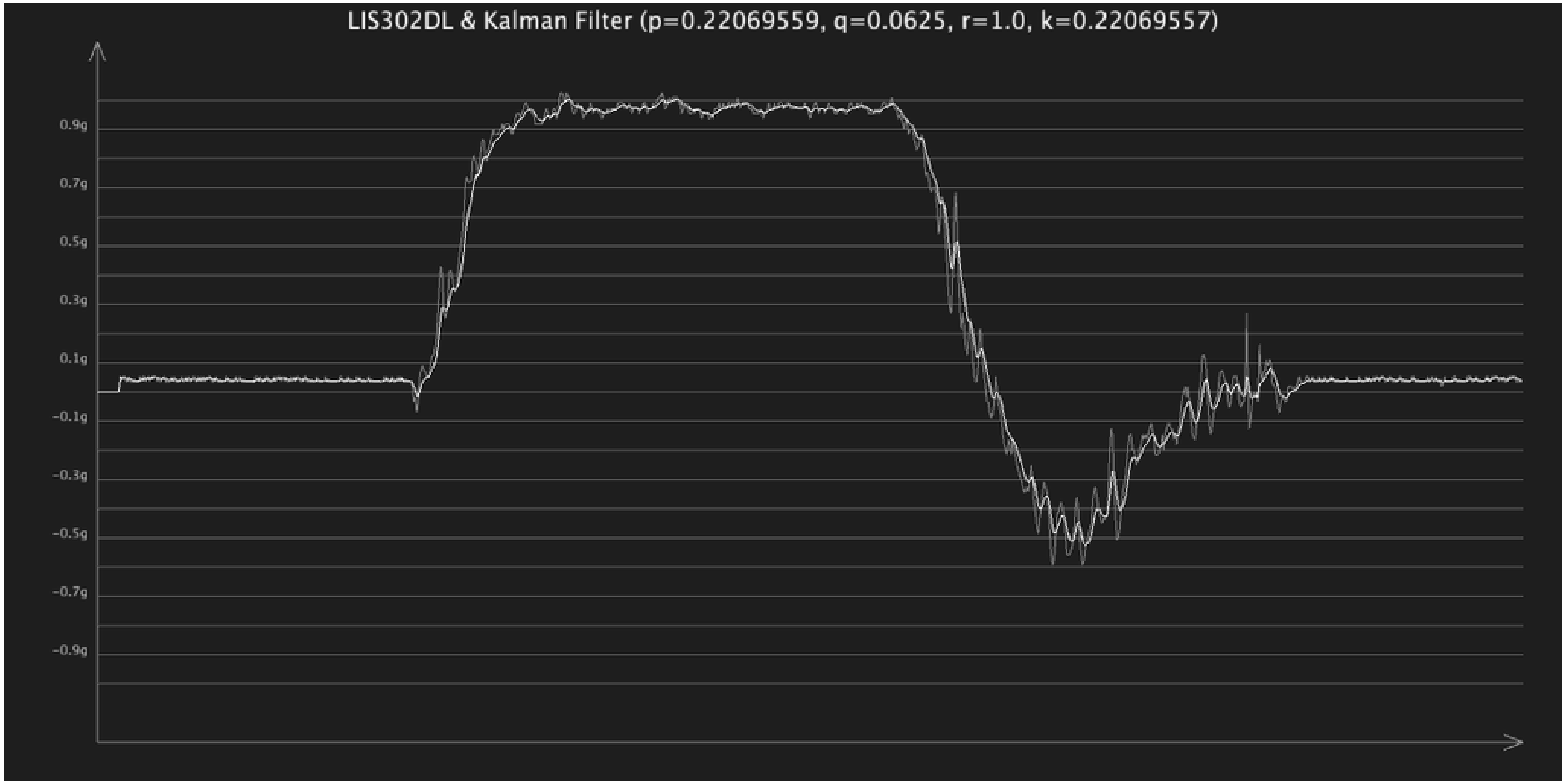
FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_\nu^2 = 0.125$ ($\approx 0.1\%$) - Sensor noise, $\sigma_w^2 = 10$ ($\approx 8\%$)
- Observations: Now the filtered signal is much less noisy than the original, however it lags a bit behind the real data.



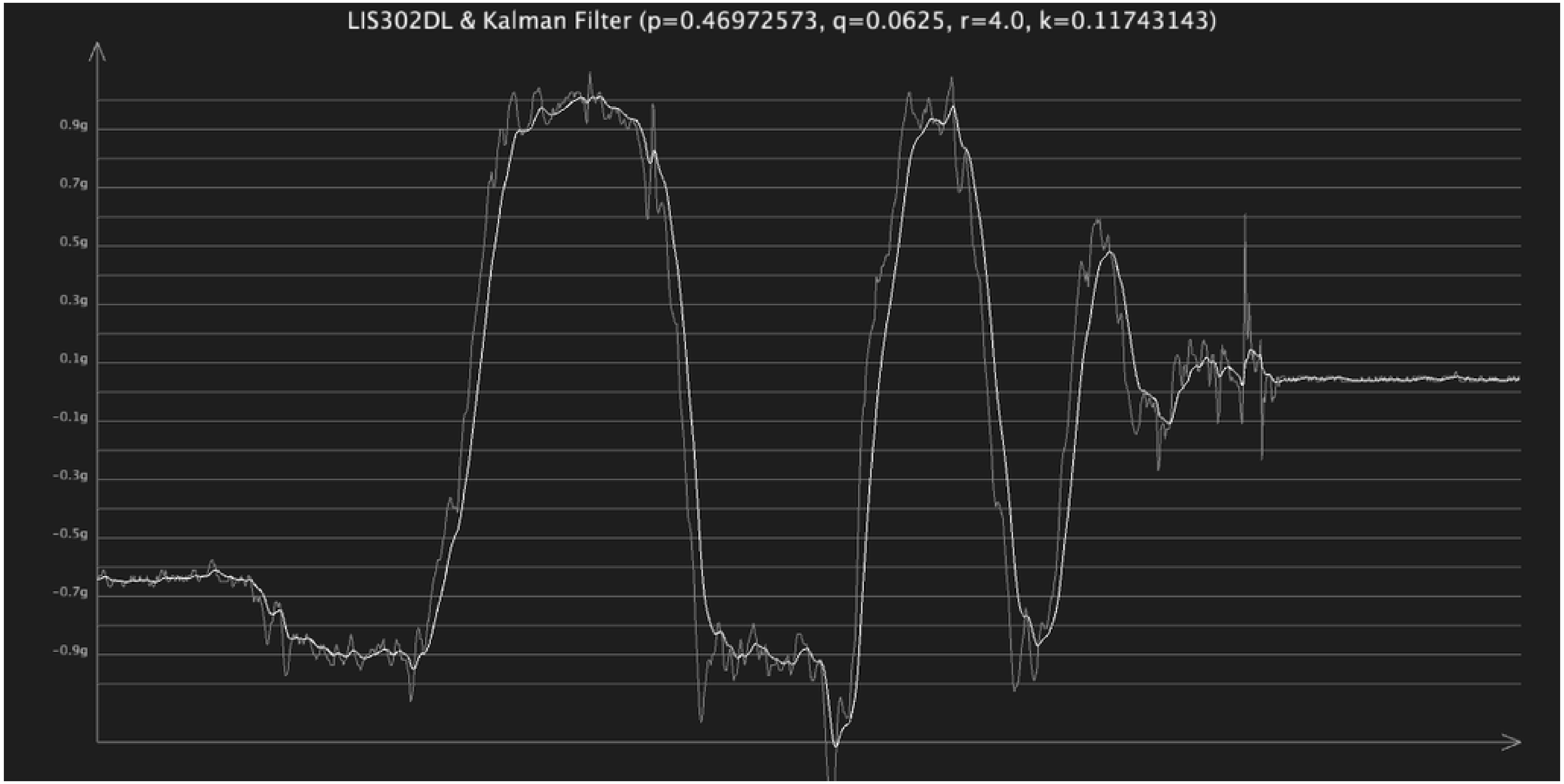
FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_\nu^2 = 0.125/2$ ($\approx 0.05\%$) - Sensor noise, $\sigma_w^2 = 1$ ($\approx 0.8\%$)
- Observations: Decreasing the sensor noise, the filter relies more on the sensor data and as such it follows the data more closely, even if the process noise was further decreased



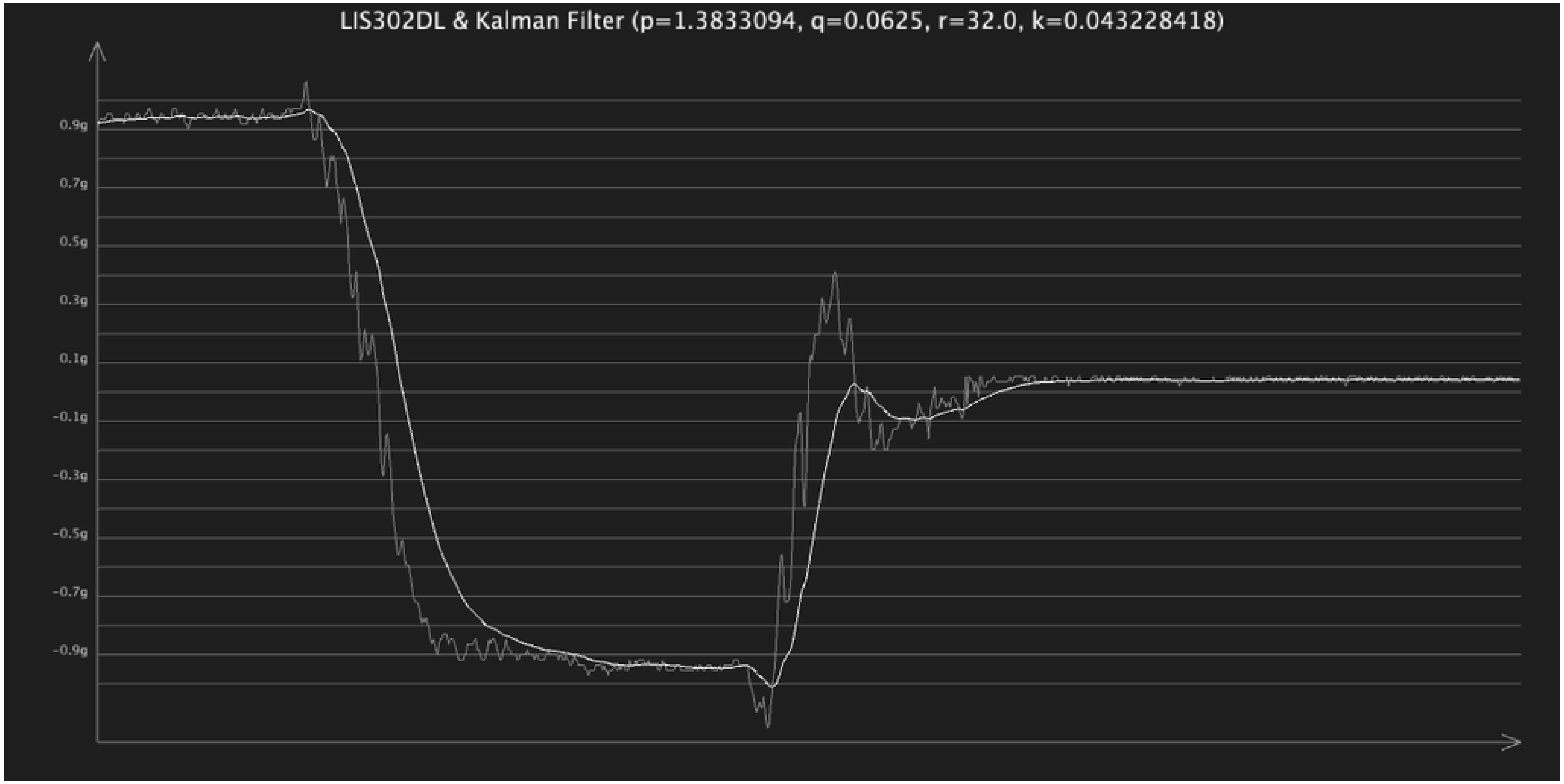
FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_{\nu}^2 = 0.125/2$ ($\approx 0.05\%$) - Sensor noise, $\sigma_w^2 = 4$ ($\approx 3.5\%$)
- Observations: Increasing the noise factor of the sensor a more stable result is obtained, with a smoothed signal, which however systematically lags behind the data



FILTERING MEASURES FROM AN ACCELEROMETER

- Process noise, $\sigma_\nu^2 = 0.125/2$ ($\approx 0.05\%$) - Sensor noise, $\sigma_w^2 = 32$ ($\approx 25\%$)
- Observations: the filtered signal is very smooth but lags significantly behind the real measurements.



N SENSORS: FILTERING AND FUSION PROBLEMS

- Scenario: The robot does not move, a stream of measures z_K about some quantity $\boldsymbol{\xi}$ of interest is obtained from an array of n sensors. Each sensor gives a reading with different precision about the same quantity. The goal is to filter and fuse the data stream in order to produce at each time step k the best estimate for the quantity $\boldsymbol{\xi}$ (*Filtering and fusion problem*)

- The state-observation equations: the state vector $\boldsymbol{\xi}$ corresponds to the measure of interest:

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \boldsymbol{\nu}_k$$
$$\boldsymbol{z}_{k+1} = \boldsymbol{C}\boldsymbol{\xi}_k + \boldsymbol{w}_k$$

- **Array of range finders** - The robot has n devices that all measure the the range ρ and the bearing angle β of the closest obstacle wrt to the robot using n different technologies with different reliability. Therefore, in this case the state vector is $\begin{bmatrix}\rho & \beta\end{bmatrix}^T$, and the (constant) matrix $\boldsymbol{C}$, of dimension $2n \times 2$, is a vector of n identity sub-matrices of dimension 2×2, such that the observation equation becomes:

$$\boldsymbol{z}_{k+1} = \begin{bmatrix} \rho^1_{k+1} \\ \beta^1_{k+1} \\ \dots \\ \rho^n_{k+1} \\ \beta^n_{k+1} \end{bmatrix}_{2n\times 2} = \boldsymbol{C}\boldsymbol{\xi}_k + \boldsymbol{w}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \dots & \dots \\ 1 & 0 \\ 0 & 1 \end{bmatrix}_{2n\times 2} \begin{bmatrix} \rho_k \\ \beta_k \end{bmatrix} + \begin{bmatrix} w^1_{k\rho} \\ w^1_{k\beta} \\ \dots \\ w^n_{k\rho} \\ w^n_{k\beta} \end{bmatrix}_{2n\times 1}$$

N SENSORS: FILTERING AND FUSION PROBLEMS

- The KF equations:

At every time step k:

$$\hat{\xi}_{k+1|k} = \hat{\xi}_k$$
$$P_{k+1|k} = P_k + V_k$$

At every time step $k+1$ when an *observation* z_{k+1} is available:

$$\hat{\xi}_{k+1} = \hat{\xi}_{k+1|k} + G_{k+1}(z_{k+1} - C\hat{\xi}_{k+1|k})$$
$$P_{k+1} = P_{k+1|k} - G_{k+1}CP_{k+1|k}$$
$$G_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + W_{k+1})^{-1}$$

- **The innovation term:**

$$\epsilon_{k+1} = z_{k+1} - C_{k+1}\hat{\xi}_{k+1|k}$$

produces a $2n \times 1$ vector of differences between the new observed set of measures and the current estimate. This difference is weighted by the gain G_{k+1}, that considers the variance associated to each different sensor and measurement, when implementing the correction to the state vector:
$$\hat{\xi}_{k+1} = \hat{\xi}_{k+1|k} + G_{k+1}\epsilon_{k+1}$$

- G_{k+1} has dimensions $2 \times 2n$. If we write it in a compact way as:

$$G_{k+1} = \begin{bmatrix} \frac{1}{\eta^1_{(k+1)\rho}} & 0 & \cdots & \frac{1}{\eta^n_{(k+1)\rho}} & 0 \\ 0 & \frac{1}{\eta^1_{(k+1)\beta}} & \cdots & 0 & \frac{1}{\eta^n_{(k+1)\beta}} \end{bmatrix} = \begin{bmatrix} G^\rho_{k+1} \\ G^\beta_{k+1} \end{bmatrix}$$

where each η^i is related to the current variance estimate associated to the measure of the i-th sensor, the product $G_{k+1}\epsilon_{k+1}$ can be read as the scalar products $G^\rho_{k+1}\epsilon_{k+1}$ and $G^\beta_{k+1}\epsilon_{k+1}$, that result in the (covariance) weighted sum of the inputs from all sensors to compute the correction to the ρ and β components of the state $\rightarrow$ **sensor fusion**