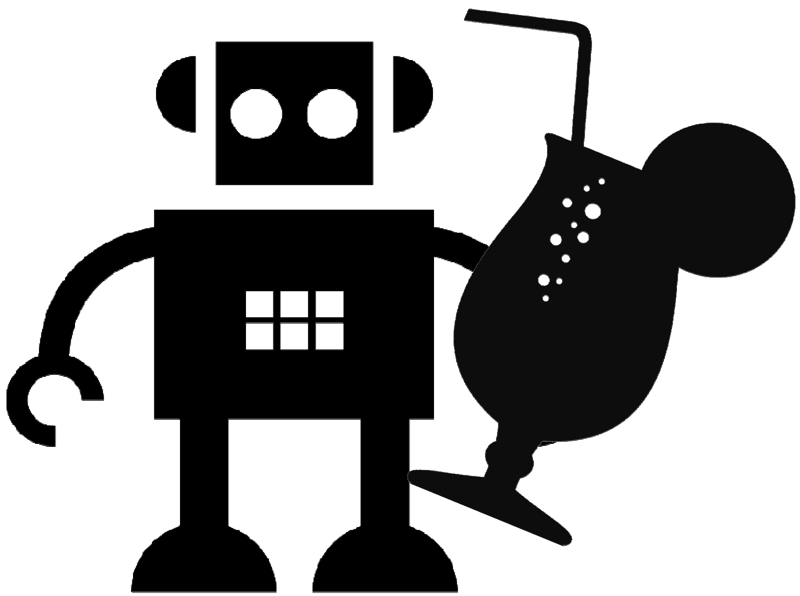
# Kalman Filter for Robotic Data Fusion



# References

https://home.wlu.edu/~levys/kalman_tutorial/

**Courses :**
*Probabilistic Robotics: Occupancy Grid Maps*
Sebastian Thrun & Alex Teichman Stanford Artificial Intelligence Lab
Slide credits: Wolfram Burgard, Dieter Fox, Cyrill Stachniss, Giorgio Grisetti, Maren Bennewitz, Christian Plagemann, Dirk Haehnel, Mike Montemerlo, Nick Roy, Kai Arras, Patrick Pfaff and others

***Mobile Robot Localization and Mapping using the Kalman Filter : CMRoboBits: Creating an Intelligent AIBO Robot***
**Paul E. Rybski**

*Bayesian Filtering* Peter Cox

*Introduction to AI Robotics (MIT Press), Chapter 11: Localization and Map Making,*
Robin Murphy

***Robotics : Vision and control fundamental*** **Peter Corke**

# Sensors to States

- Global Positionning (GPS, other RF)
- IMU, gyroscope,
- Odometry
- Features/Landmarks detection and local localization

# how to merge data ?

- Good way to merge data ?
- Do we need extra information ?
- Could we have information on error localization ?

# Ideal World

## Measurement equation (sensor model)

Sensor reading

State

$$z_{t+1} = H_{t+1} x_{t+1}$$

Sensor function

If we know $H_{t+1}$, Its a good start If $H_t$ is constant it's simpler !!!!

$$\boxed{x_{t+1} = H_{t+1}^{-1} \; z_{t+1}}$$

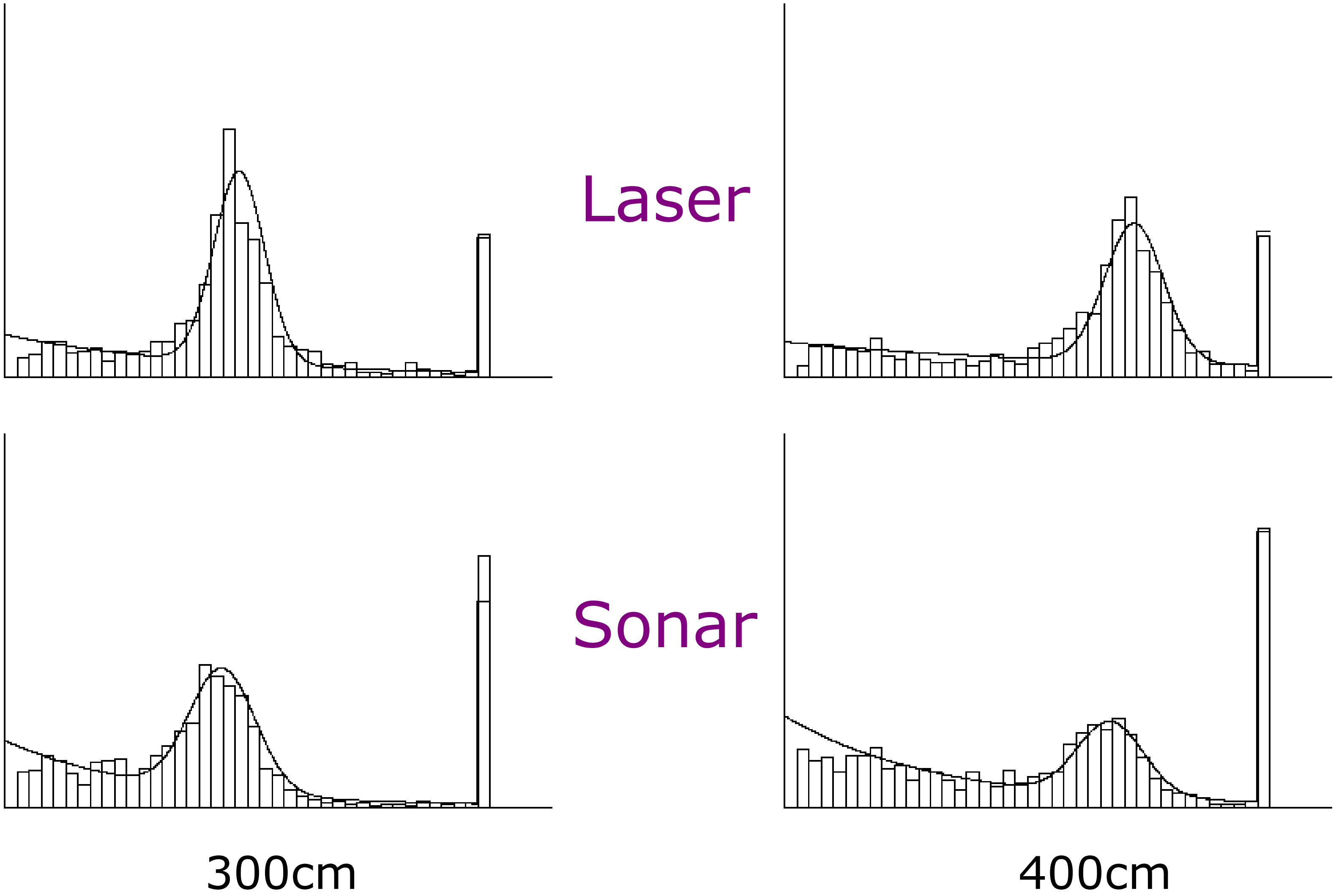
Is H « inversible » ?
We try to have the sensors for !!!
But there is so many incertitude

With for example for localization this type of state :

$$x = (\mathrm{x}, \mathrm{y}, \theta)$$

$$z_{t+1} = Hx_{t+1} + \varepsilon_{t+1(observation)}$$

# Gaussians are good candidates for real life

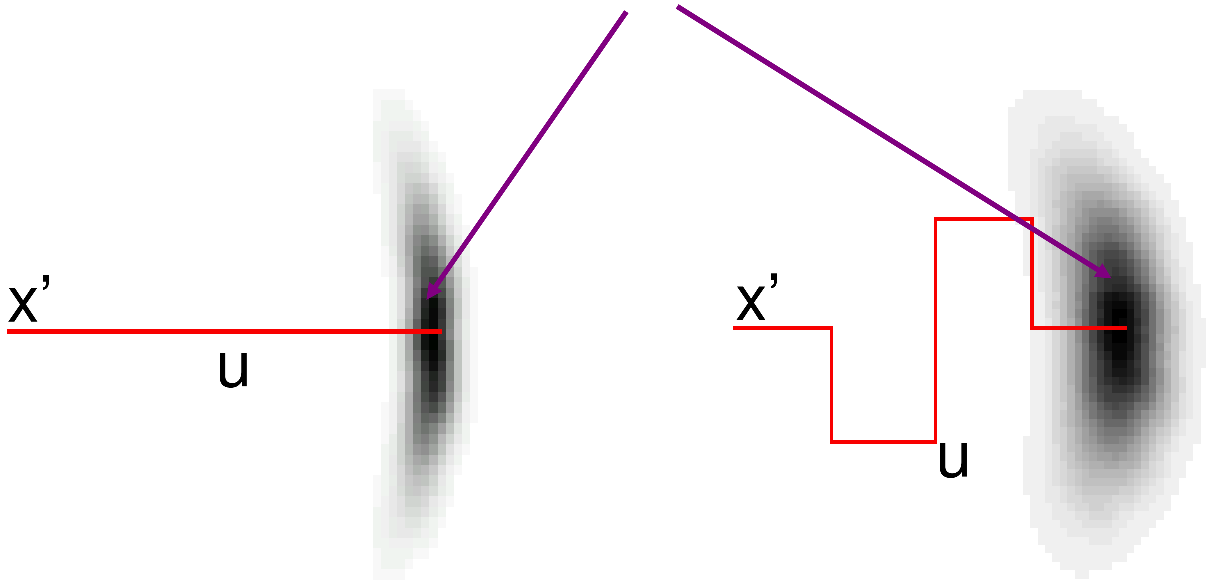


# Probabilistic Kinematics

- Odometry information is inherently noisy.

Due to actions incertitude arriving in a common point could be different

$p(x|a,x')$ $p(x|u,x')$



A lot of solutions still exist.

We needs :

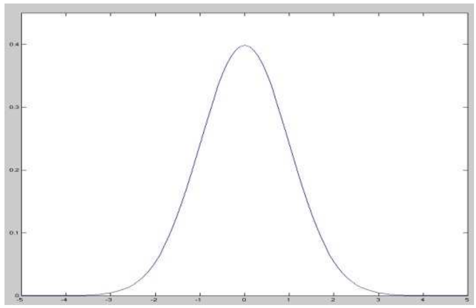
- Representation for the belief function
- Update equations
- Motion model
- Sensor model
- Initial belief state

# Kalman filter was the first proposed solution

Asumptions are strong :
Models are linear and noise is modeled as gaussian distribution

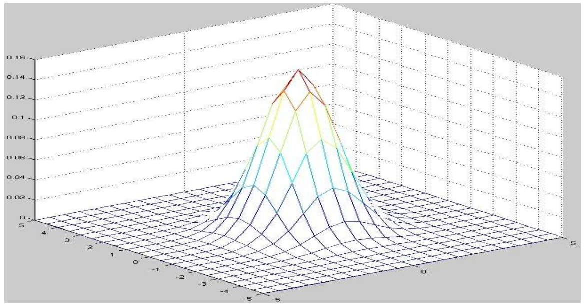
A 1-d Gaussian distribution is given by:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



An n-d Gaussian distribution is given by:

$$P(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

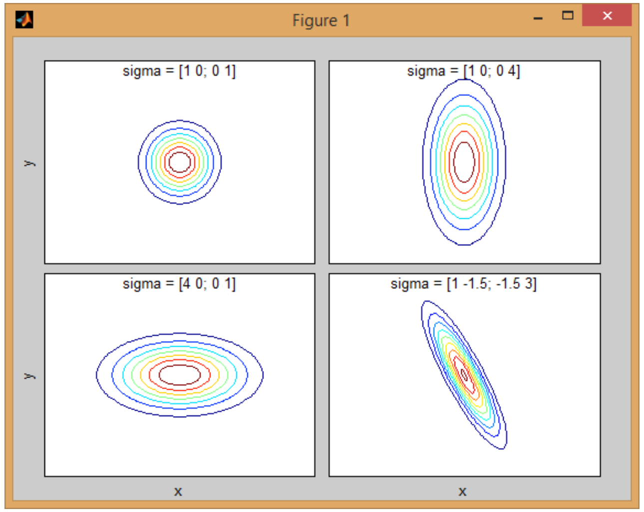


But Belief is a gaussian distribution too and the filter is prove to be optimal !!!
(calculus of the kalman gain)

$$E[(x-\hat{x})^2 \mid Z_t]$$

# Belief representation with Gaussian Distribution.

Mean matrix  and covariance Matrix

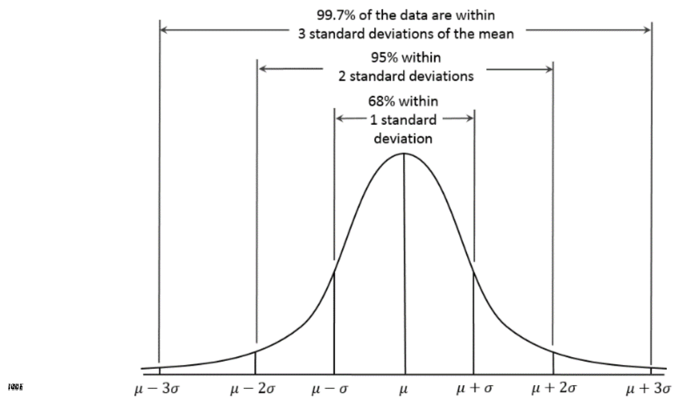


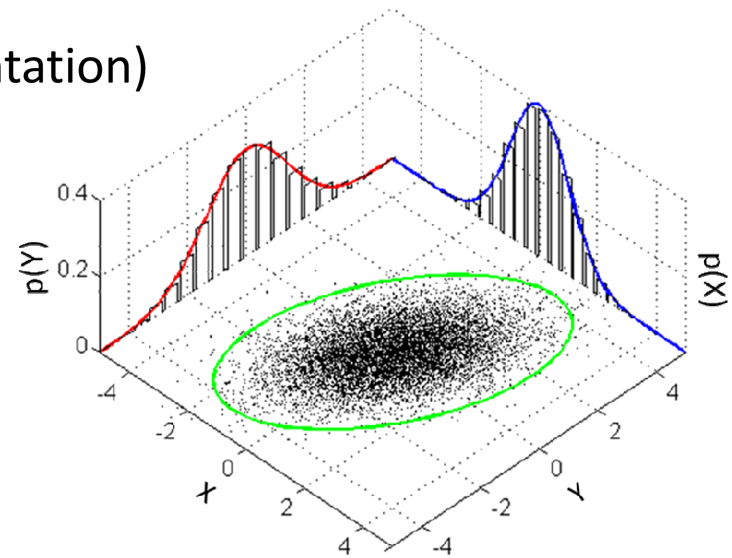

$$\boldsymbol{\mu}_{X,Y} = \begin{bmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_Y \end{bmatrix}, \qquad \boldsymbol{\Sigma}_{X,Y} = \begin{bmatrix} \boldsymbol{\Sigma}_{XX} & \boldsymbol{\Sigma}_{XY} \\ \boldsymbol{\Sigma}_{YX} & \boldsymbol{\Sigma}_{YY} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

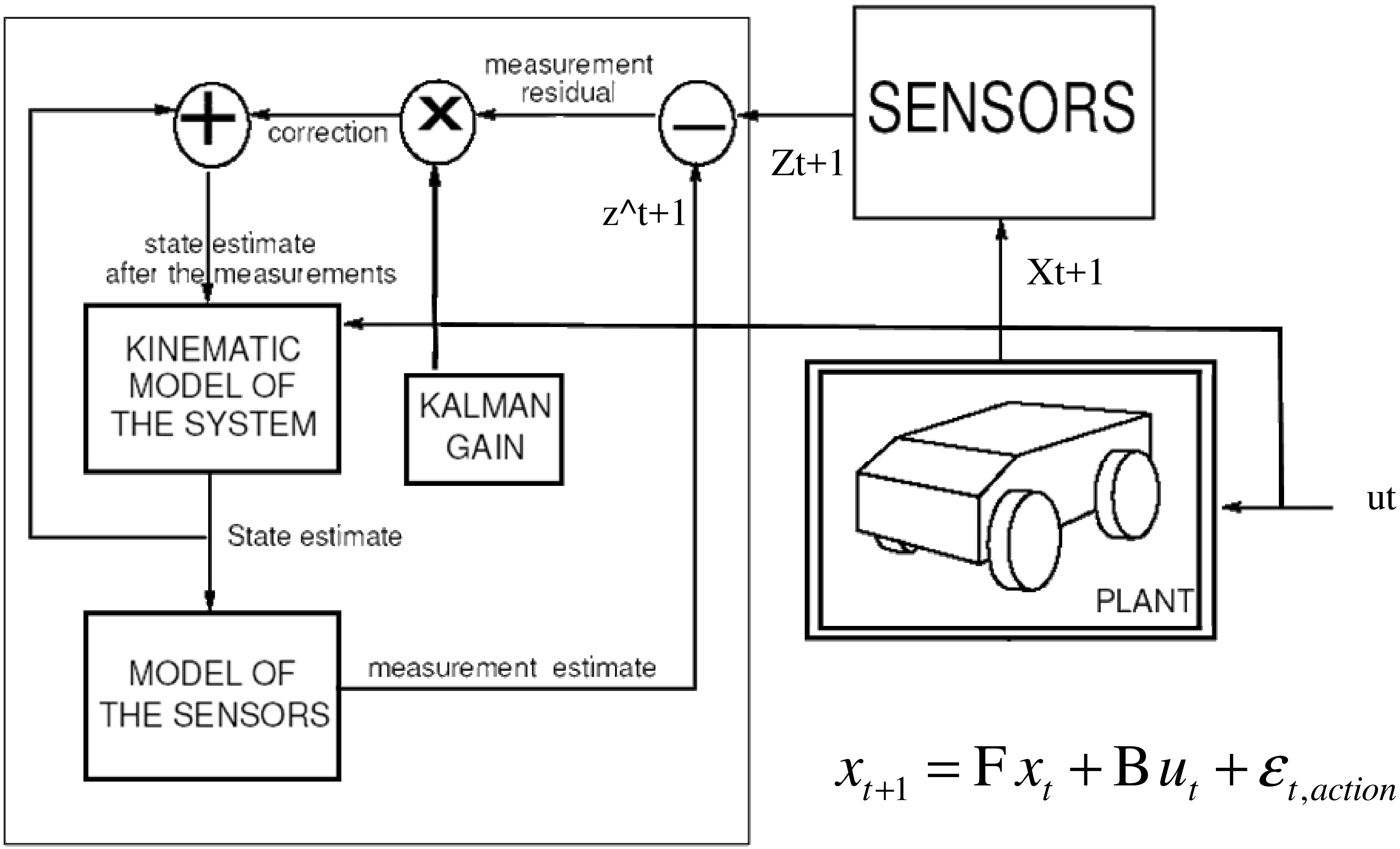
Ellipse representation (typical 3σ representation)



## Kalman Filter

z




$$x_{t+1} = \mathrm{F}\, x_t + \mathrm{B}\, u_t + \varepsilon_{t,action}$$

Kalman Filter

## Linear discrete time dynamic system (motion model)

State — Control input — Process noise

$$x_{t+1} = F_t x_t + B_t u_t + G_t w_t$$

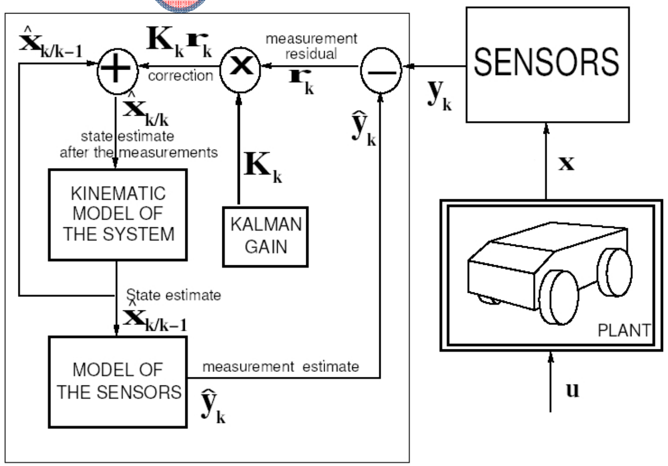
State transition function — Control input function — Noise input function with covariance Q

## Measurement equation (sensor model)

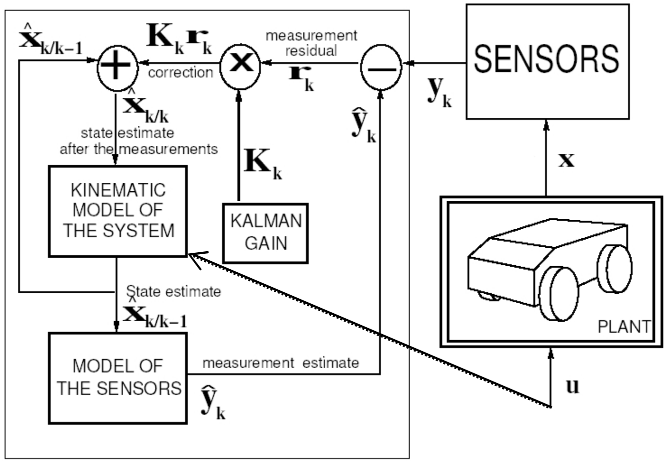
Sensor reading — State — Sensor noise with covariance R

$$z_{t+1} = H_{t+1} x_{t+1} + n_{t+1}$$

Sensor function



## Kalman Filter




Propagation (motion model):
« Prediction part »

$$\hat{x}_{t+1/t} = F_t \hat{x}_{t/t} + B_t u_t$$

- State estimate is updated from system dynamics

$$P_{t+1/t} = F_t P_{t/t} F_t^T + G_t Q_t G_t^T$$

- Uncertainty estimate *GROWS*

We need :
2 dynamics models F and B

Update (sensor model):

$$\hat{z}_{t+1} = H_{t+1} \hat{x}_{t+1/t}$$

- Compute expected value of sensor reading

$$r_{t+1} = z_{t+1} - \hat{z}_{t+1}$$

- Compute the difference between expected and “true”

$$S_{t+1} = H_{t+1} P_{t+1/t} H_{t+1}{}^T + R_{t+1}$$

- Compute covariance of sensor reading

$$K_{t+1} = P_{t+1/t} H_{t+1}{}^T S_{t+1}{}^{-1}$$

- Compute the Kalman Gain (how much to correct est.)

$$\hat{x}_{t+1/t+1} = \hat{x}_{t+1/t} + K_{t+1} r_{t+1}$$

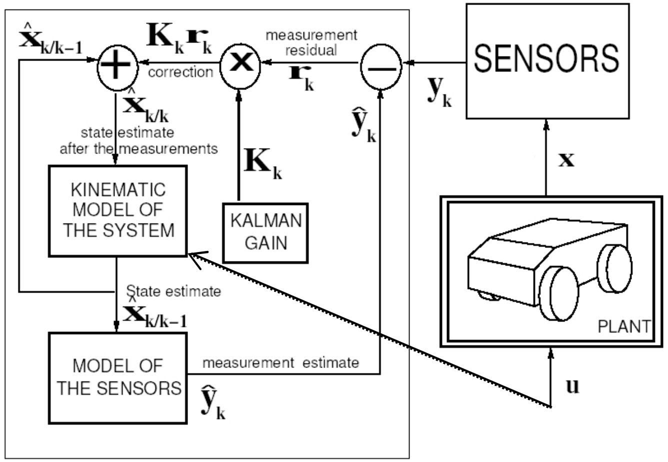
- Multiply residual times gain to correct state estimate

$$P_{t+1/t+1} = P_{t+1/t} - P_{t+1/t} H_{t+1}{}^T S_{t+1}{}^{-1} H_{t+1} P_{t+1/t}$$

- Uncertainty estimate SHRINKS

Rq: cov(F$x$) = F.cov($x$)F$^T$

Kalman Filter as Algorythm




($\mu_t$, $\Sigma_t$ )=Algorithm **Kalman_filter**( $\mu_{t-1}$, $\Sigma_{t-1}$, $u_t$, $z_t$):

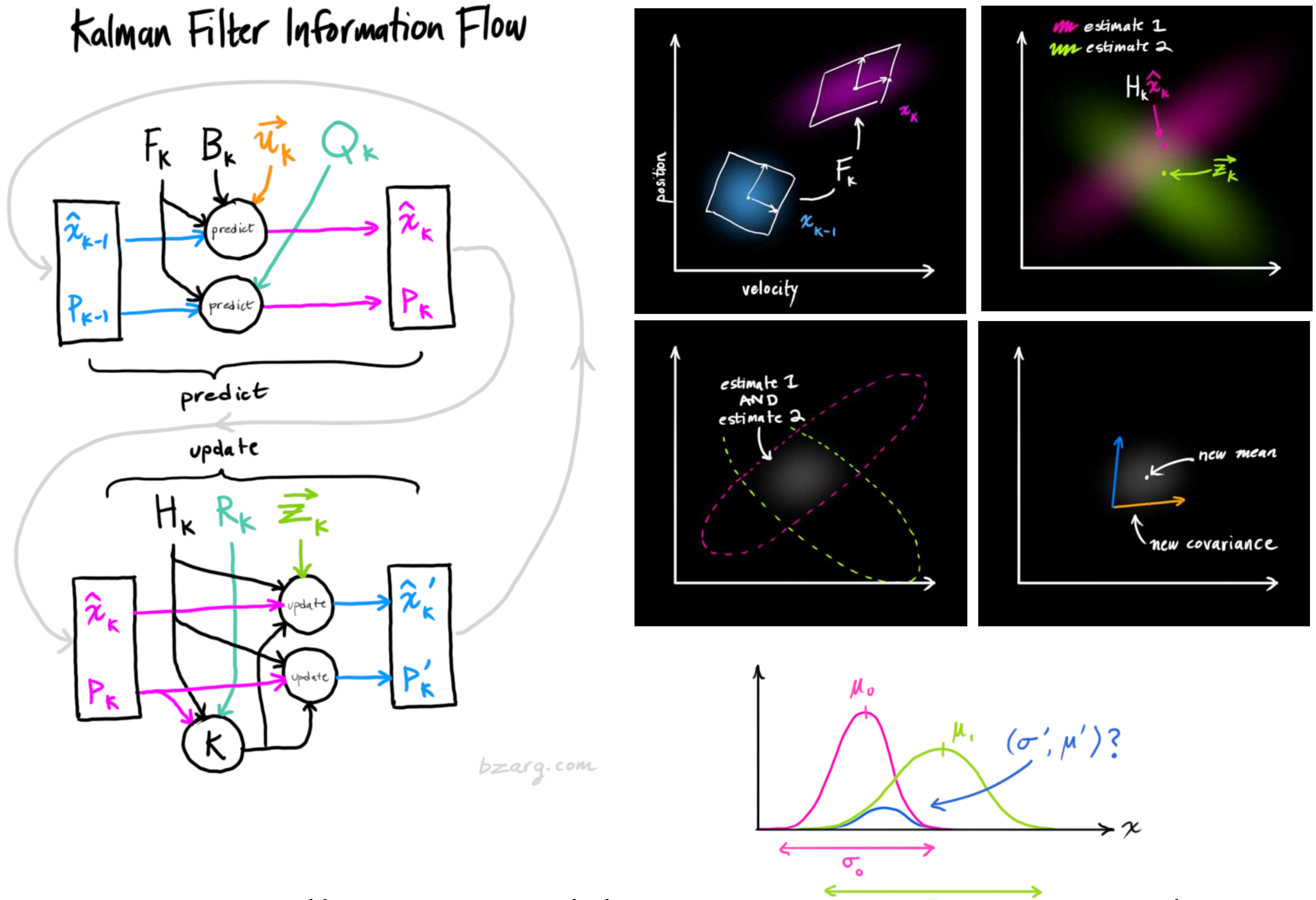
Prediction

$$\bar{\mu}_t = F_t \mu_{t-1} + B_t u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

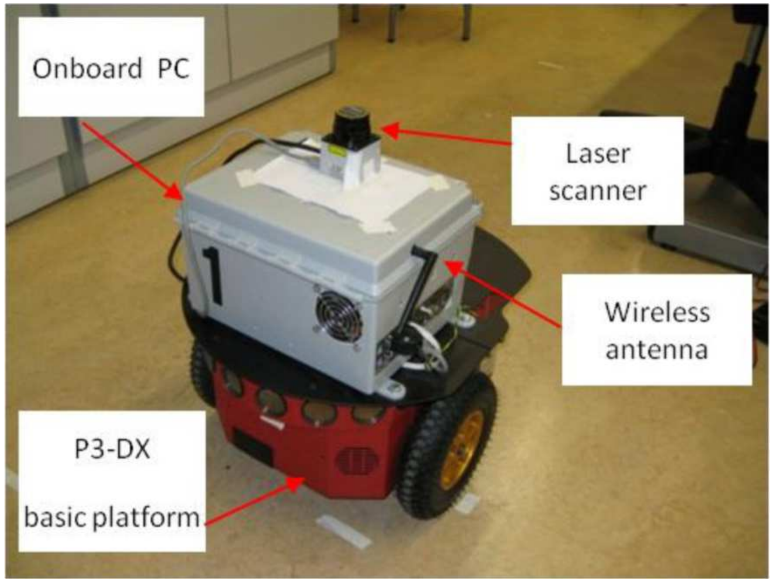
Correction

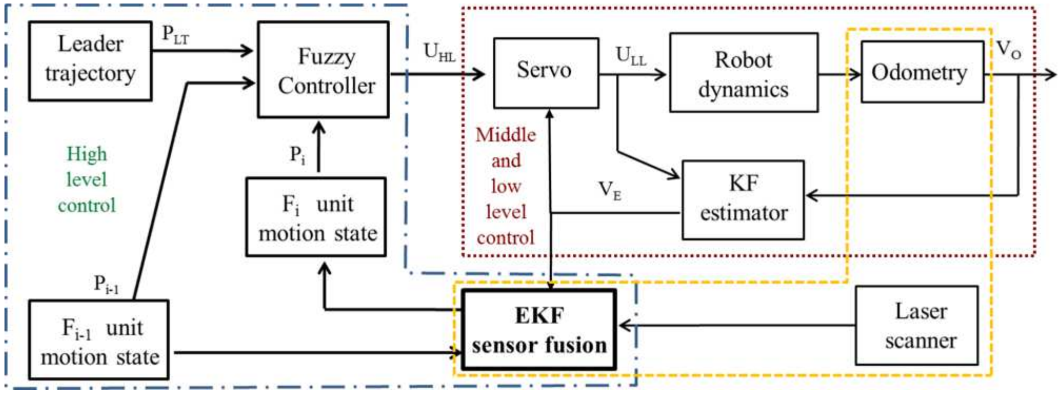
$$K_t = \bar{\Sigma}_t C_t (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

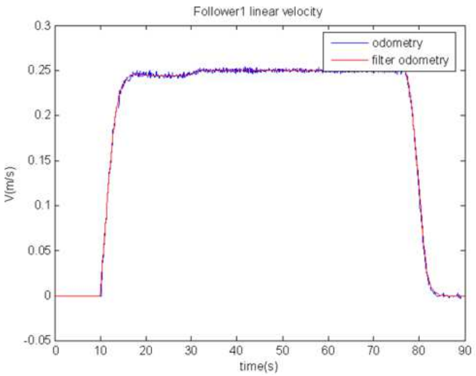
Return $\mu_t$, $\Sigma_t$

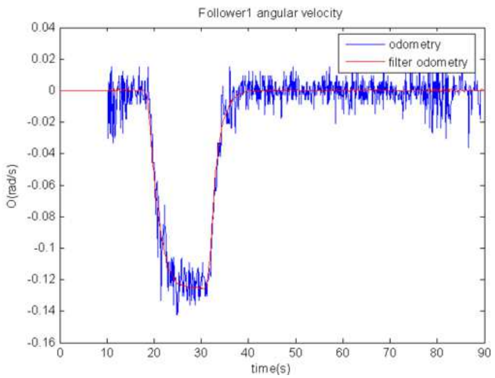


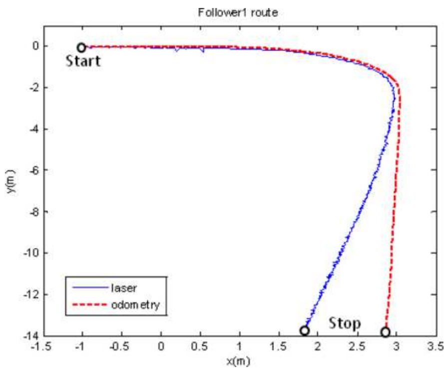

http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

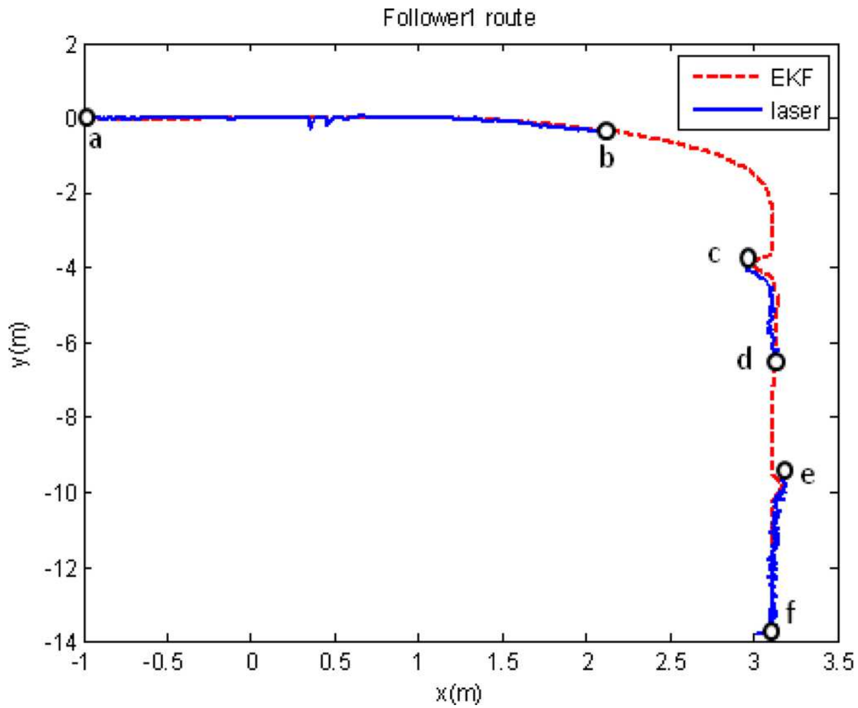

Onboard PC
Laser scanner
Wireless antenna
P3-DX basic platform


Leader trajectory
$P_{LT}$
Fuzzy Controller
$U_{HL}$
Servo
$U_{LL}$
Robot dynamics
Odometry
$V_O$
High level control
$P_i$
$F_i$ unit motion state
Middle and low level control
$V_E$
KF estimator
$P_{i-1}$
$F_{i-1}$ unit motion state
EKF sensor fusion
Laser scanner


Follower1 linear velocity
odometry
filter odometry
V(m/s)
time(s)


Follower1 angular velocity
odometry
filter odometry
O(rad/s)
time(s)


Follower1 route
Start
Stop
laser
odometry
y(m)
x(m)


Follower1 route
EKF
laser
a
b
c
d
e
f
y(m)
x(m)

# Extended Kalman Filter

$$x_{t+1} = \mathrm{f}(x_t, u_t) + \varepsilon_t \qquad x_{t+1} = \mathrm{F}\,x_t + \mathrm{B}\,u_t + \varepsilon_{t(action)}$$

$$z_{t+1} = \mathrm{h}(x_{t+1}) + \varepsilon_{t+1(observation)} \qquad z_{t+1} = Hx_{t+1} + \varepsilon_{t+1(observation)}$$

Fonction is non linear, we need to approximate one which is linear. It can be done by hand by « 1st order approximation » or more automaticily calculating at each step the Jacobian on one point (if no derivative parametric representation is available)

$$f_i(X) = f_i(X_0) + \nabla f_i(X_0)(X - X_0) + o(\|X - X_0\|).$$

Each of the gradient (row) vectors $\nabla f_i(X_0)$ can be made the $i$-th row of an $m \times n$ matrix

$$\nabla F(X_0) \equiv \begin{pmatrix} \nabla f_1(X_0) \\ \nabla f_2(X_0) \\ \vdots \\ \nabla f_m(X_0) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(X_0) & \frac{\partial f_1}{\partial x_2}(X_0) & \cdots & \frac{\partial f_1}{\partial x_n}(X_0) \\ \frac{\partial f_2}{\partial x_1}(X_0) & \frac{\partial f_2}{\partial x_2}(X_0) & \cdots & \frac{\partial f_2}{\partial x_n}(X_0) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1}(X_0) & \frac{\partial f_m}{\partial x_2}(X_0) & \cdots & \frac{\partial f_m}{\partial x_n}(X_0) \end{pmatrix}.$$

Combining the equations above involving the linear approximation $f_i(X_0) + \nabla f_i(X_0)(X - X_0)$ to each $f_i$ at $X_0$, we obtain, again for $X$ near $X_0$,

$$F(X) = F(X_0) + \nabla F(X_0)(X - X_0) + o(\|X - X_0\|),$$

# Extended Kalman Filter

*P. Corke Notations*

Now consider the case where the system is not linear

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\big(\boldsymbol{x}\langle k\rangle, \boldsymbol{u}\langle k\rangle\big) + \boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{h}\big(\boldsymbol{x}\langle k\rangle\big) + \boldsymbol{w}\langle k\rangle$$

where $\boldsymbol{f}$ and $\boldsymbol{h}$ are now functions instead of constant matrices. $\boldsymbol{f}: \mathbb{R}^n, \mathbb{R}^m \to \mathbb{R}^n$ is a function that describes the new state in terms of the previous state and the input to the system. The function $\boldsymbol{h}: \mathbb{R}^n \to \mathbb{R}^p$ maps the state vector to the sensor measurements.

To use the linear Kalman filter with a non-linear system we first make a local linear approximation

$$\boldsymbol{x}\langle k+1\rangle = \boldsymbol{f}\big(\hat{\boldsymbol{x}}\langle k\rangle, \boldsymbol{u}\langle k\rangle\big) + \boldsymbol{F}_x\big(\boldsymbol{x}\langle k\rangle - \hat{\boldsymbol{x}}\langle k|k\rangle\big) + \boldsymbol{F}_u\boldsymbol{u}\langle k\rangle + \boldsymbol{F}_v\boldsymbol{v}\langle k\rangle$$
$$\boldsymbol{z}\langle k+1\rangle = \boldsymbol{h}\big(\hat{\boldsymbol{x}}\langle k\rangle\big) + \boldsymbol{H}_x\big(\hat{\boldsymbol{x}}\langle k+1|k\rangle - \boldsymbol{x}\langle k\rangle\big) + \boldsymbol{H}_w\boldsymbol{w}\langle k\rangle$$

where $\boldsymbol{F}_x \in \mathbb{R}^{n\times n}, \boldsymbol{F}_u \in \mathbb{R}^{n\times m}, \boldsymbol{F}_v \in \mathbb{R}^{n\times n}, \boldsymbol{H}_x \in \mathbb{R}^{p\times n}$ and $\boldsymbol{H}_w \in \mathbb{R}^{p\times p}$ are Jacobians of the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ and are evaluated at each time step.

# Extended Kalman Filter

*P. Corke Notations*

We define a prediction error

$$\begin{aligned}\tilde{\boldsymbol{x}}\langle k+1|k\rangle &= \boldsymbol{x}\langle k\rangle - \hat{\boldsymbol{x}}\langle k+1|k\rangle \\ &= \boldsymbol{F}_x\tilde{\boldsymbol{x}}\langle k|k\rangle + \boldsymbol{F}_u\boldsymbol{u}\langle k\rangle + \boldsymbol{F}_v\boldsymbol{v}\langle k\rangle\end{aligned}$$

and a measurement residual

$$\begin{aligned}\tilde{\boldsymbol{z}}\langle k+1|k\rangle &= \boldsymbol{z}\langle k+1\rangle - \boldsymbol{h}\langle k+1|k\rangle \\ &= \boldsymbol{H}_x\tilde{\boldsymbol{x}} + \boldsymbol{H}_w\boldsymbol{w}\langle k\rangle\end{aligned}$$

which are linear and the Kalman filter equations above can be applied. The prediction step of the extended Kalman filter is

$$\begin{aligned}\hat{\boldsymbol{x}}\langle k+1|k\rangle &= \boldsymbol{f}\big(\hat{\boldsymbol{x}}\langle k\rangle, \boldsymbol{u}\langle k\rangle\big) \\ \hat{\boldsymbol{P}}\langle k+1|k\rangle &= \boldsymbol{F}_x\hat{\boldsymbol{P}}\langle k|k\rangle\boldsymbol{F}_x^T + \boldsymbol{F}_v\hat{\boldsymbol{V}}\langle k\rangle\boldsymbol{F}_v^T\end{aligned}$$

and the update step is

$$\begin{aligned}\hat{\boldsymbol{x}}\langle k+1|k+1\rangle &= \hat{\boldsymbol{x}}\langle k+1|k\rangle + \boldsymbol{K}\langle k+1\rangle\boldsymbol{\nu}\langle k+1\rangle \\ \hat{\boldsymbol{P}}\langle k+1|k+1\rangle &= \hat{\boldsymbol{P}}\langle k+1|k\rangle - \boldsymbol{K}\langle k+1\rangle\boldsymbol{H}_x\hat{\boldsymbol{P}}\langle k+1|k\rangle\end{aligned}$$

where the innovation is

$$\boldsymbol{\nu}\langle k+1\rangle = \boldsymbol{z}\langle k+1\rangle - \boldsymbol{h}\big(\hat{\boldsymbol{x}}\langle k+1|k\rangle\big)$$

and the Kalman gain is

$$\boldsymbol{K}\langle k+1\rangle = \hat{\boldsymbol{P}}\langle k+1|k\rangle\boldsymbol{H}_x^T\left(\boldsymbol{H}_x\hat{\boldsymbol{P}}\langle k+1|k\rangle\boldsymbol{H}_x^T + \boldsymbol{H}_w\hat{\boldsymbol{W}}\boldsymbol{H}_w^T\right)^{-1}$$

A fundamental problem with the extended Kalman filter is that PDFs of the random variables are no longer Gaussian after being operated on by the non-linear

# Extended Kalman Filter

*Extensions*

- Multiple hypothesis tracking
  - Multiple Kalman filters are used to track the data
  - Multi-Gaussian approach allows for representation of arbitrary probability densities
  - Consistent hypothesis are tracked while highly inconsistent hypotheses are dropped
  - Similar in spirit to particle filter, but orders of magnitude fewer filters are tracked as compared to the particle filter

# Ideal World

Measurement equation (sensor model)

Sensor reading

State

$$z_{t+1} = H_{t+1} x_{t+1}$$

Sensor function

If we know $H_{t+1}$, Its a good start If $H_t$ is constant it's simpler !!!!

$$\boxed{x_{t+1} = H_{t+1}^{-1} \; z_{t+1}}$$

Is H « inversible » ?
We try to have the sensors for !!!
But there is so many incertitude

With for example for localization this type of state :

$$x = (\mathrm{x}, \mathrm{y}, \theta)$$

$$z_{t+1} = Hx_{t+1} + \varepsilon_{t+1(observation)}$$

# Other bayes filters

$$Bel(x_t) = \eta \ P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \ Bel(x_{t-1}) \ dx_{t-1}$$

- **Kalman filters**
- Particle filters
- Hidden Markov models
- Dynamic Bayesian networks
- Partially Observable Markov Decision Processes (POMDPs)

# Tracking Techniques

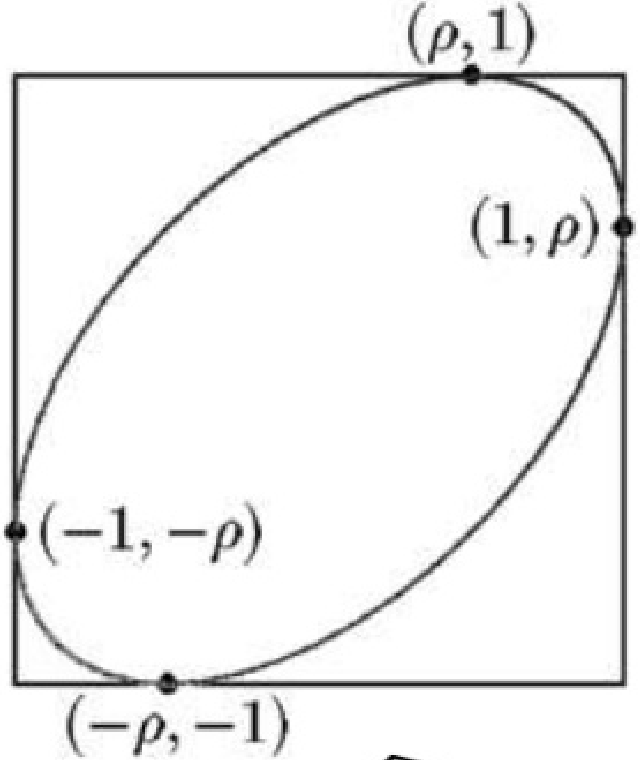
- Kalman Filter
  - Highly efficient, robust (even for nonlinear)
  - Uni-modal, limited handling of nonlinearities
- Particle Filter
  - Less efficient, highly robust
  - Multi-modal, nonlinear, non-Gaussian
- Rao-Blackwellised Particle Filter, MHT
  - Combines PF with KF
  - Multi-modal, highly efficient

# The shapes of correlation ellipses (2)

The density function of 2-dim Gauss-distribution with standardizations.

$$f_\rho(x,y) = \frac{\exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right)}{2\pi\sqrt{1-\rho^2}}$$

$$x^2 - 2\rho xy + y^2 = 1 - \rho^2$$




Note: for higher dimensions,

$$f_{\mu,\Sigma}(x) = \frac{\exp\left(-\frac{(x-\mu)'\Sigma^{-1}(x-\mu)}{2}\right)}{\left((2\pi)^k\,|\Sigma|\right)^{1/2}}$$

The ellipse inscribes the unit square at 4 points ($\pm 1, \pm\rho$) and ($\pm\rho, \pm 1$).