16-311-Q Introduction to Robotics Fall'17

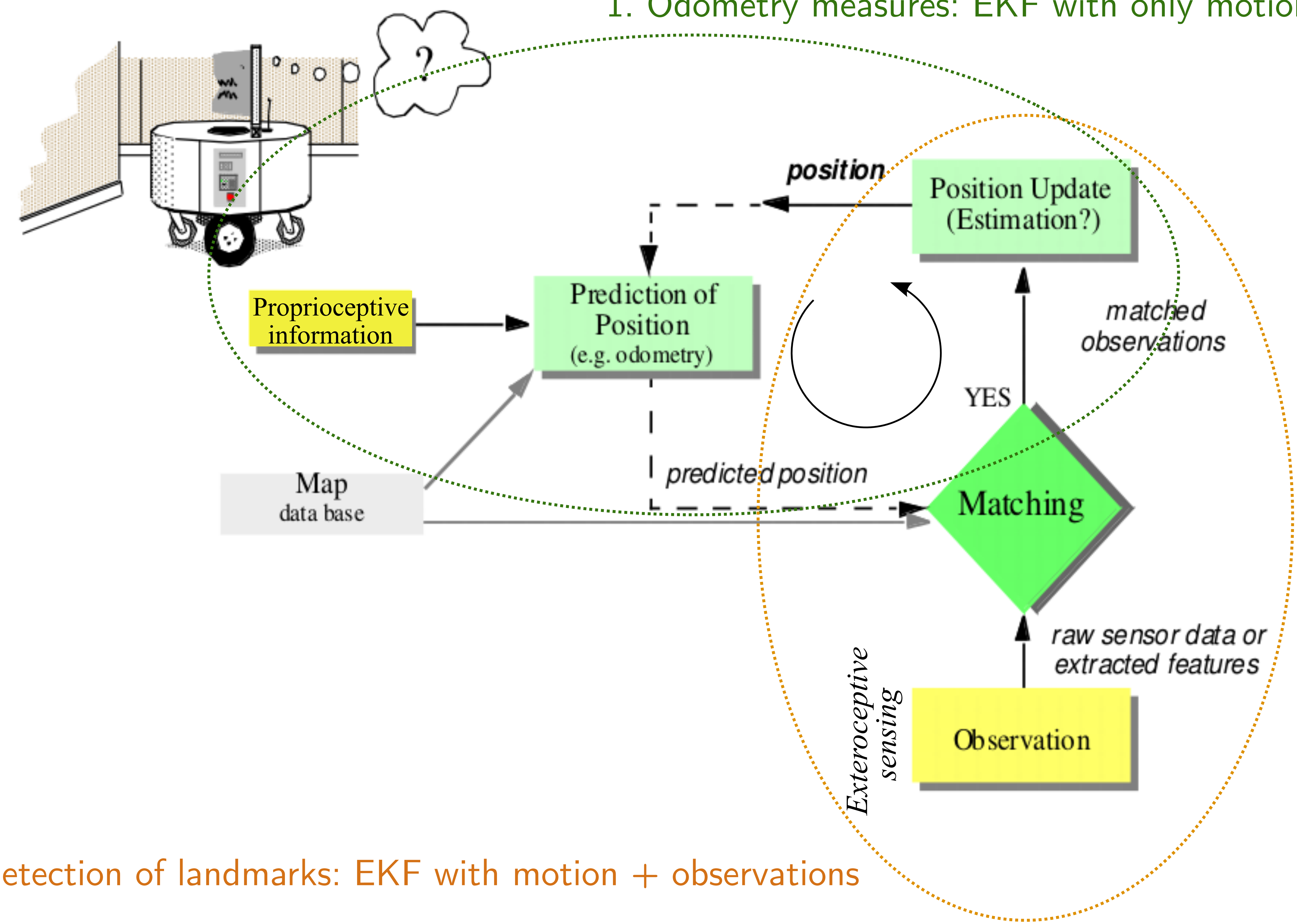
Lecture 20:
Extended Kalman Filter

Instructor:
Gianni A. Di Caro

جامعة كارنيجي ميلون في قطر
Carnegie Mellon University Qatar

EKF FOR MAP-BASED ROBOT LOCALIZATION

1. Odometry measures: EKF with only motion



2. Detection of landmarks: EKF with motion + observations

DISCRETE-TIME MOTION EQUATIONS

$$\xi_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta S_k \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ y_k + \Delta S_k \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ \theta_k + \Delta\theta_k \end{bmatrix}$$

From Runge-Kutta numeric integration of pose evolution kinematic equations.

Assume that the odometry model is perfect, based on measured distance ΔS, and heading variation $\boldsymbol{\Delta\theta}$

Odometry measurements are noisy!

➔ Random noise is added to ΔS and $\boldsymbol{\Delta\theta}$ to model motion's kinematics

Discrete-time process (motion) equations

$$\xi_{k+1} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} (\Delta S_k + \nu_k^s)\cos(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\ (\Delta S_k + \nu_k^s)\sin(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\ \Delta\theta_k + \nu_k^\theta \end{bmatrix}$$

In absence of specific information, motion noise is modeled as ***Gaussian white noise*** (and the two noise components are assumed to be ***uncorrelated)***

Process noise

$$\nu_k = \begin{bmatrix} \nu_k^s & \nu_k^\theta \end{bmatrix}^T \sim N(0, \mathbf{V}_k), \quad \mathbf{V}_k = \begin{bmatrix} \sigma_{ks}^2 & 0 \\ 0 & \sigma_{k\theta}^2 \end{bmatrix}$$

NON LINEARITY OF DISCRETE-TIME MOTION EQUATIONS

Discrete-time process (motion) equations

$$\xi_{k+1} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} (\Delta S_k + \nu_k^s)\cos(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\ (\Delta S_k + \nu_k^s)\sin(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\ \Delta\theta_k + \nu_k^\theta \end{bmatrix}$$

$$\boldsymbol{\xi}_{k+1} = f(\boldsymbol{\xi}_k, \Delta S_k, \Delta\theta_k, \boldsymbol{\nu}_k), \quad \boldsymbol{\nu}_k = \begin{bmatrix}\nu_k^s & \nu_k^\theta\end{bmatrix}^T \sim N(0, \mathbf{V}_k)$$

Process' dynamics function, *f()*, is not linear

➜ Process equations do not meet the *linearity requirement* for using the Kalman filter

Linearize pose evolution f() in the neighborhood of $[\hat{\xi}_{k|k} \;\; u_k \;\; (\nu_k = 0)]$,
the current state estimate, controls (ΔS_k and $\Delta\theta_k$), and mean of process noise

$$f(\boldsymbol{\xi}_k, \boldsymbol{u}_k, \boldsymbol{\nu}_k) = f(\boldsymbol{\xi}, \boldsymbol{u}, \boldsymbol{\nu})|_{\widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, 0} + (\boldsymbol{\xi}_k - \widehat{\boldsymbol{\xi}}_{k|k})\boldsymbol{F}_{\xi}|_{\widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, 0} + (\boldsymbol{\nu}_k - \mathbf{0})\boldsymbol{F}_{\nu}|_{\widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, 0}$$

1st order Taylor series

$$= \boldsymbol{f}_k(\widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}) + (\boldsymbol{\xi}_k - \widehat{\boldsymbol{\xi}}_{k|k})\boldsymbol{F}_{k\xi} + \boldsymbol{\nu}_k \boldsymbol{F}_{k\nu}$$

Linear in $\boldsymbol{\xi}_k$ and ν_k

EXTENDED KALMAN FILTER (EKF): LINEARIZED MOTION MODEL

Scenario (Prediction from motion): The robot *does move* but no external observations are made. Proprioceptive measures from the on-board odometry sensors are used to model robot's motion dynamics avoiding to consider the direct control inputs.

Linear(ized) discrete-time process (motion) equations

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{f}_k(\widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}) + (\boldsymbol{\xi}_k - \widehat{\boldsymbol{\xi}}_{k|k})\boldsymbol{F}_{k\xi} + \boldsymbol{\nu}_k \boldsymbol{F}_{k\nu}$$

Linearization of motion dynamics using the **Jacobians** $\boldsymbol{F}_{k\xi}$ and $\boldsymbol{F}_{k\nu}$, that have to be evaluated in $(\boldsymbol{\xi}_k = \widehat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \boldsymbol{\nu}_k = 0)$

➔ Rules for *linear transformations of mean and (co)variance of Gaussian variables* can be applied

Extended Kalman Filter (EKF) - Motion only

Prediction update

$$\begin{cases} \widehat{\boldsymbol{\xi}}_{k+1|k} = \boldsymbol{f}_k(\widehat{\boldsymbol{\xi}}_{k|k}, \mathbf{0}; \Delta S_k, \Delta\theta_k) + (\widehat{\boldsymbol{\xi}}_{k|k} - \widehat{\boldsymbol{\xi}}_{k|k})\boldsymbol{F}_{\xi}|_{\widehat{\boldsymbol{\xi}}_k, \boldsymbol{u}_k, 0} & \text{(State prediction)} \\ \boldsymbol{P}_{k+1|k} = \boldsymbol{F}_{k\xi}\boldsymbol{P}_k\boldsymbol{F}_{k\xi}^T + \boldsymbol{F}_{k\nu}\boldsymbol{V}_k\boldsymbol{F}_{k\nu}^T & \text{(Covariance prediction)} \end{cases}$$

$= 0$

Measurement correction

$$\begin{cases} \widehat{\boldsymbol{\xi}}_{k+1} = \widehat{\boldsymbol{\xi}}_{k+1|k} + \boldsymbol{G}_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{C}_{k+1}\widehat{\boldsymbol{\xi}}_{k+1|k}) & \text{(State update)} \\ \boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{G}_{k+1}\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k} & \text{(Covariance update)} \\ \boldsymbol{G}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{C}_{k+1}^T(\boldsymbol{C}_{k+1}\boldsymbol{P}_{k+1|k}\boldsymbol{C}_{k+1}^T + \boldsymbol{W}_{k+1})^{-1} & \text{(Kalman gain)} \end{cases}$$

EKF JACOBIANS FOR THE LINEARIZED MOTION MODEL

The *Jacobian* of the non-linear function ***f()*** is computed in $[\hat{\boldsymbol{\xi}}_{\mathbf{k|k}} \;\; \mathbf{u_k} \; (\boldsymbol{\nu}_{\mathbf{k}} = \mathbf{0})]$, the current state estimate (the mean), the current controls, the mean of the Gaussian noise

f() is a *vector function* with three function components:

$$
\begin{aligned}
f_{kx} &= \quad x_k + (\Delta S_k + \nu_k^s)\cos(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\
f_{ky} &= \quad y_k + (\Delta S_k + \nu_k^s)\sin(\theta_k + \frac{\Delta\theta_k}{2} + \nu_k^\theta) \\
f_{k\theta} &= \quad \theta_k + \Delta\theta_k + \nu_k^\theta
\end{aligned}
$$

The Jacobian matrix of f:

$$
\boldsymbol{F}_k(x_k, y_k, \theta_k,\, \nu_k^s, \nu_k^\theta) = [\nabla f_{kx} \quad \nabla f_{ky} \quad \nabla f_{k\theta}]^T = \begin{bmatrix} \frac{\partial f_{kx}}{\partial x_k} & \frac{\partial f_{kx}}{\partial y_k} & \frac{\partial f_{kx}}{\partial \theta_k} & \frac{\partial f_{kx}}{\partial \nu_k^s} & \frac{\partial f_{kx}}{\partial \nu_k^\theta} \\ \frac{\partial f_{ky}}{\partial x_k} & \frac{\partial f_{ky}}{\partial y_k} & \frac{\partial f_{ky}}{\partial \theta_k} & \frac{\partial f_{ky}}{\partial \nu_k^s} & \frac{\partial f_{ky}}{\partial \nu_k^\theta} \\ \frac{\partial f_{k\theta}}{\partial x_k} & \frac{\partial f_{k\theta}}{\partial y_k} & \frac{\partial f_{k\theta}}{\partial \theta_k} & \frac{\partial f_{k\theta}}{\partial \nu_k^s} & \frac{\partial f_{k\theta}}{\partial \nu_k^\theta} \end{bmatrix} = [\boldsymbol{F}_{k\xi} \quad \boldsymbol{F}_{k\nu}]
$$

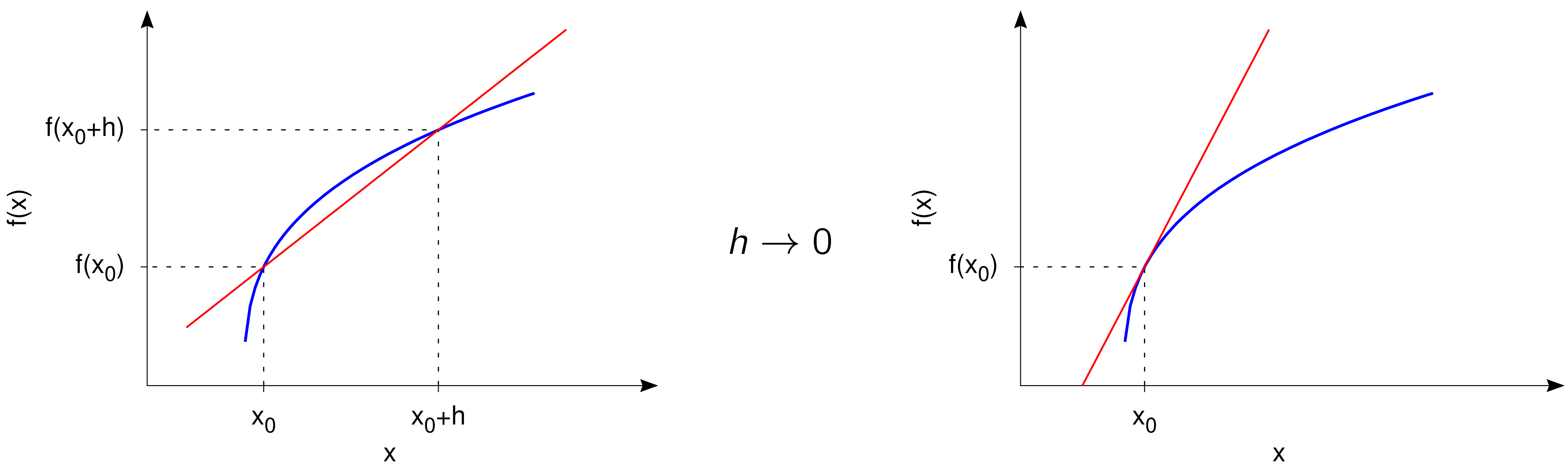
$$
\boldsymbol{F}_{k\xi} = \begin{bmatrix} 1 & 0 & -\Delta S_k \sin(\theta_k + \frac{\Delta\theta_k}{2}) \\ 0 & 1 & \Delta S_k \cos(\theta_k + \frac{\Delta\theta_k}{2}) \\ 0 & 0 & 1 \end{bmatrix}_{\hat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \boldsymbol{\nu}=0}
\qquad
\boldsymbol{F}_{k\nu} = \begin{bmatrix} \cos(\theta_k + \frac{\Delta\theta_k}{2}) & -\Delta S_k \sin(\theta_k + \frac{\Delta\theta_k}{2}) \\ \sin(\theta_k + \frac{\Delta\theta_k}{2}) & \Delta S_k \cos(\theta_k + \frac{\Delta\theta_k}{2}) \\ 0 & 1 \end{bmatrix}_{\hat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \boldsymbol{\nu}=0}
$$

- **Def. Derivative:** Given a scalar function $f : X \subseteq \mathbb{R} \mapsto \mathbb{R}$, if the limit

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

  exists and takes a finite value, f is differentiable in $x_0 \in X$ and the value of the limit is the derivative of the function in x_0, which is also indicated with $f'(x_0) \stackrel{def}{=} \frac{df}{dx}(x_0)$

- Geometric interpretation: the derivative is the slope of the tangent to the graph of f in point $(x_0, f(x_0))$. This can be shown considering that the line passing for two points $(x_0, f(x_0))$ and $((x_0 + h), f(x_0 + h))$ belonging to the graph f, is $y = mx + f(x_0 + h)$, where the slope is $m = \frac{f(x_0+h)-f(x_0)}{(x_0+h)-x_0}$. If $h \to 0$, the secant to the curve overlaps with the tangent in x_0. That is, the equation of the tangent to f in x_0 is: $y = f(x_0) + f'(x_0)(x - x_0)$, which is precisely the first-order Taylor series computed in x_0.



RECAP ON DERIVATIVES, GRADIENTS,JACOBIANS

- **Gradient:** "derivative" for *scalar* functions of multiple variables $\rightarrow$ Normal to the tangent hyperplane to the function graph. Given a scalar, differentiable, multi-variable function $f : \mathbb{R}^n \mapsto \mathbb{R}$, its gradient is the vector of its partial derivatives:

$$\nabla f_{(x_1,x_2,\ldots,x_n)} \stackrel{def}{=} \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right) = \frac{\partial f}{\partial x_1}\boldsymbol{e}_1 + \frac{\partial f}{\partial x_2}\boldsymbol{e}_2 + \ldots + \frac{\partial f}{\partial x_n}\boldsymbol{e}_n$$

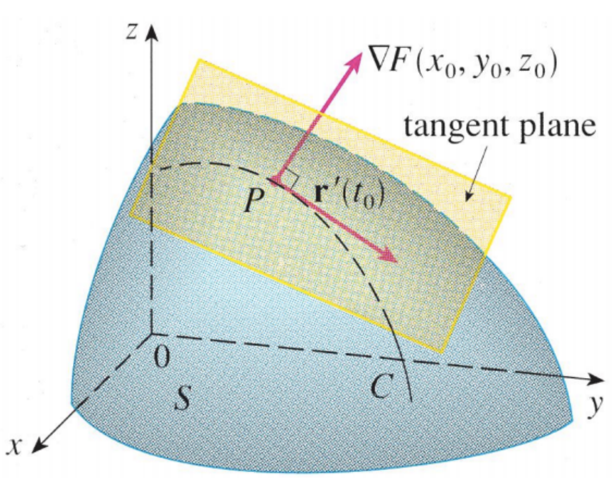
- For $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, the *Taylor series* becomes:

$$f(\boldsymbol{x})_{|\boldsymbol{x}_0} = \sum_{|k|\geq 0} \frac{1}{k!} \partial^k [f(\boldsymbol{x}_0)](\boldsymbol{x} - \boldsymbol{x}_0)^k$$

where k is a multi-index, an integer-valued vector, $k = (k_1, k_2, \ldots, k_n)$, $k_i \in \mathbb{Z}^+$, and $\partial^k f$ means $\partial_1^{k_1} f\ \partial_2^{k_2} f\ \cdots \partial_n^{k_n} f$, where $\partial_j^i f = \frac{\partial^j f}{\partial x_i^j}$. The 2nd order polynomial is:

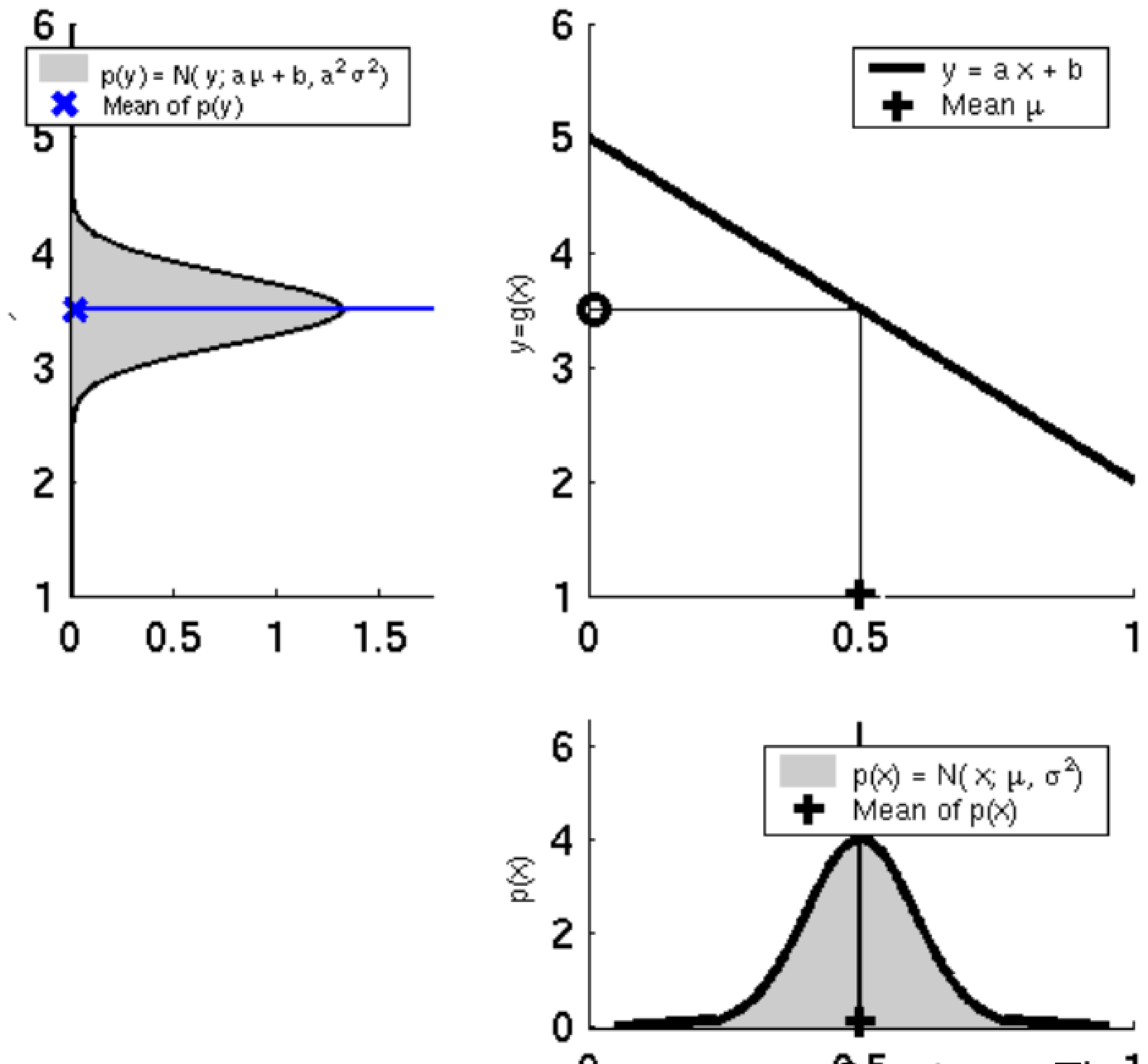
$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0)^T(\boldsymbol{x} - \boldsymbol{x}_0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_0)^T \boldsymbol{H}(f(\boldsymbol{x}_0))(\boldsymbol{x} - \boldsymbol{x}_0)$$

Removing the quadratic part, the linear approximation is obtained, that is, the equation of the tangent hyperplane in $\boldsymbol{x}_0$, where the gradient is normal to the tangent hyperplane



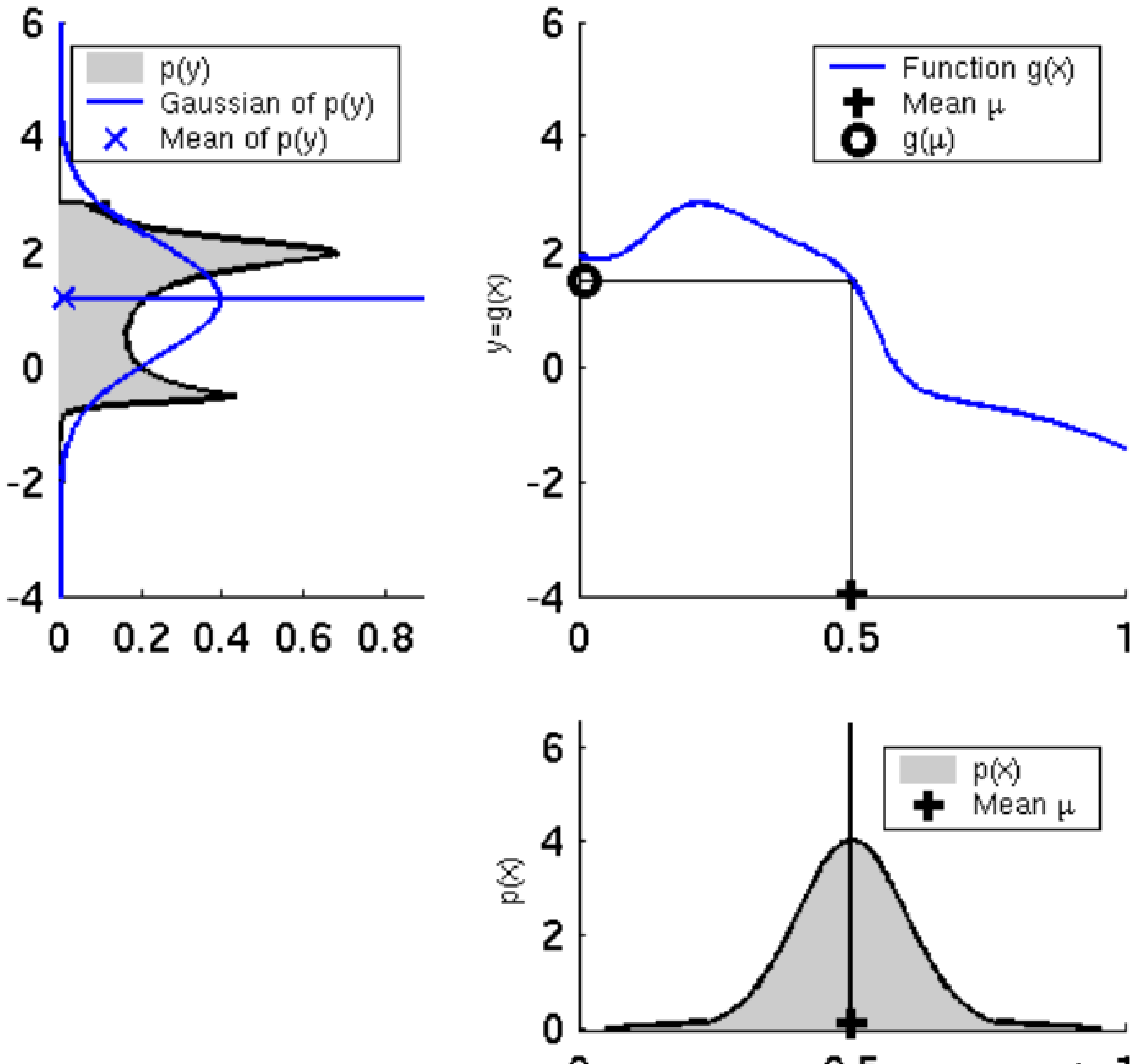
Jacobian: "gradient" for *vector* functions of multiple variables $\rightarrow$ Each function component has a tangent hyperplane to the function graph $\rightarrow$ Map of tangent hyperplanes

EFFECT OF LINEARIZATION: LINEAR CASE



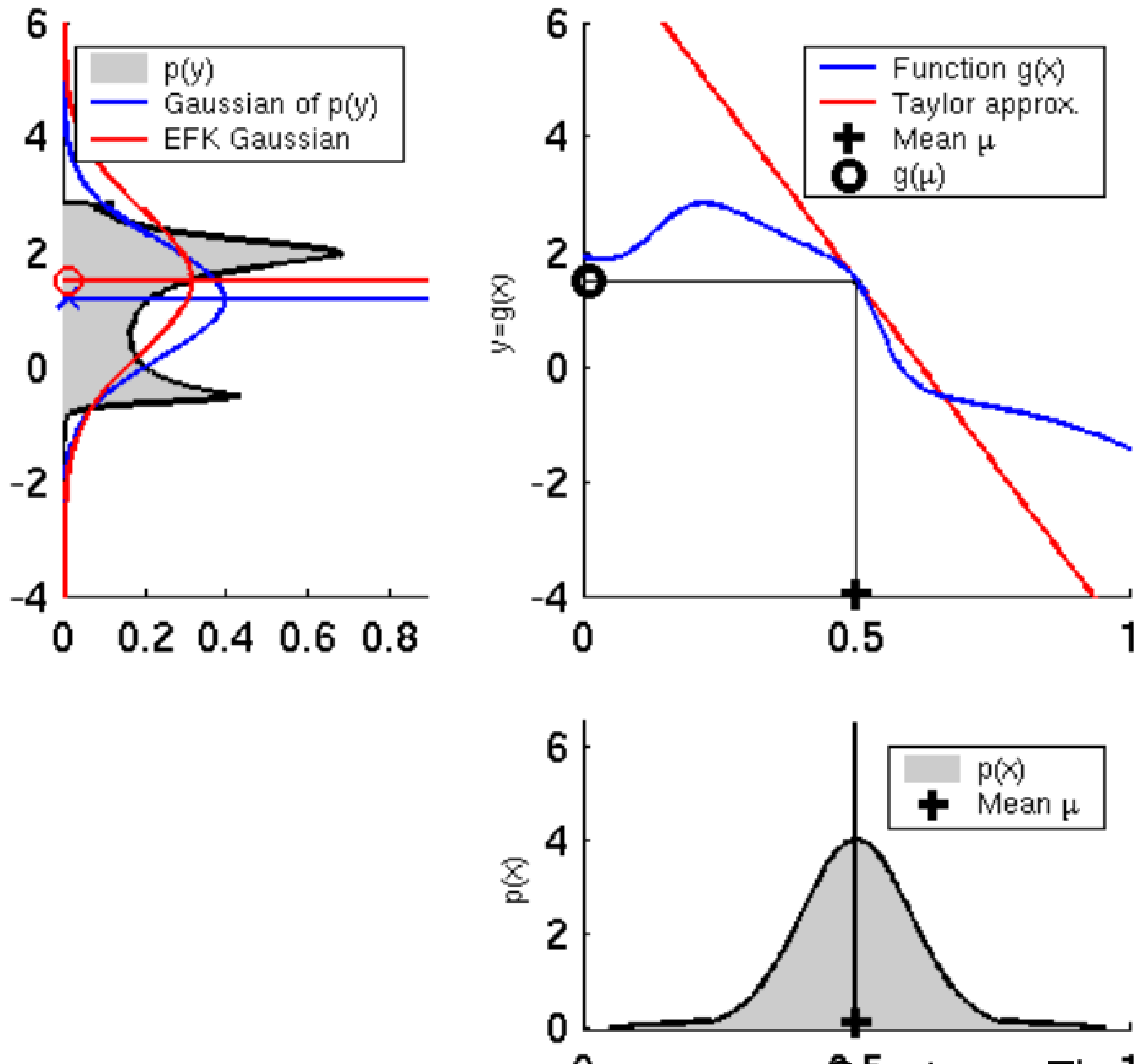
Courtesy: Thrun, Burgard, Fox

EFFECT OF LINEARIZATION: NON LINEAR CASE



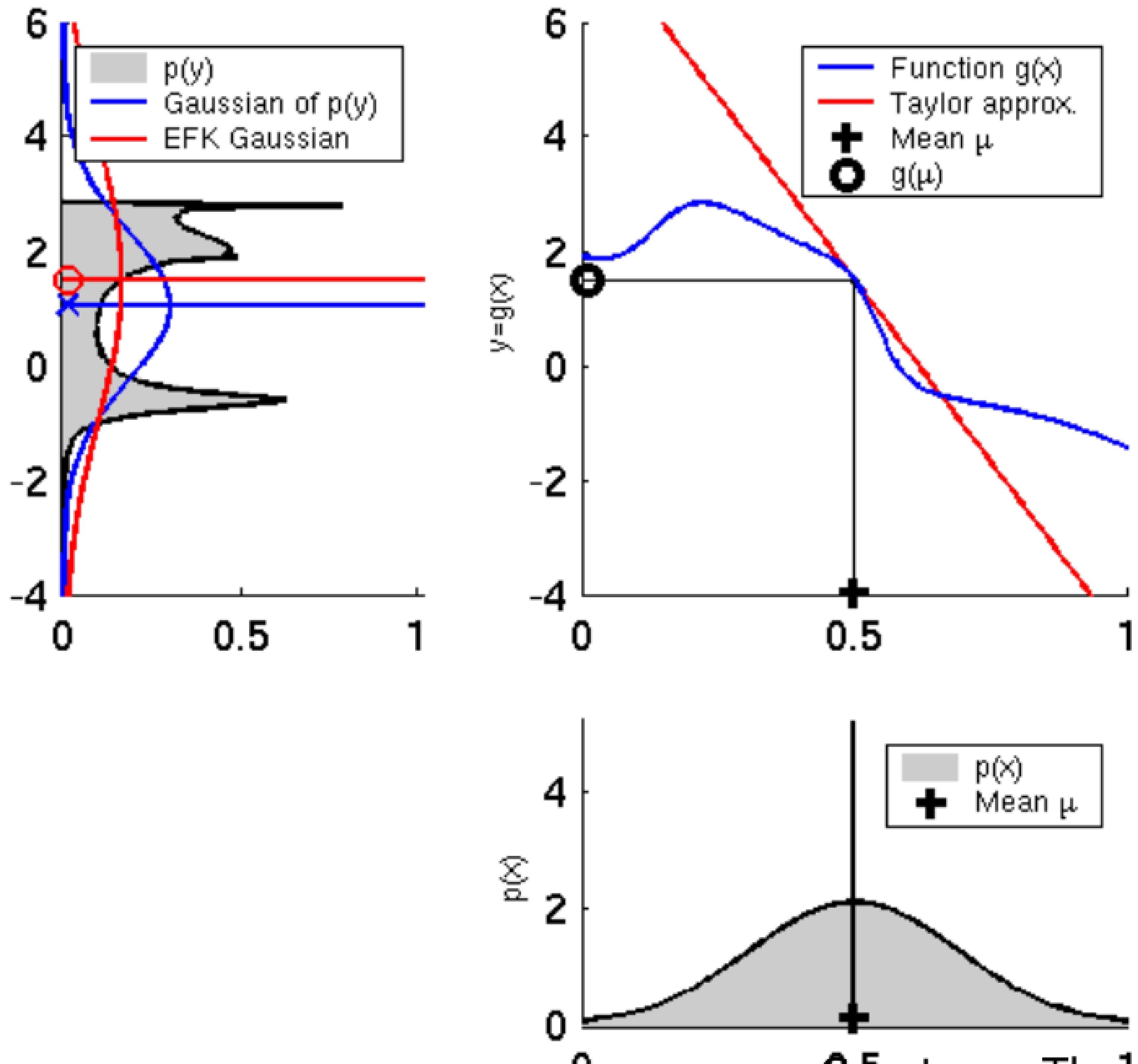
Courtesy: Thrun, Burgard, Fox

EFFECT OF LINEARIZATION: NON LINEAR CASE



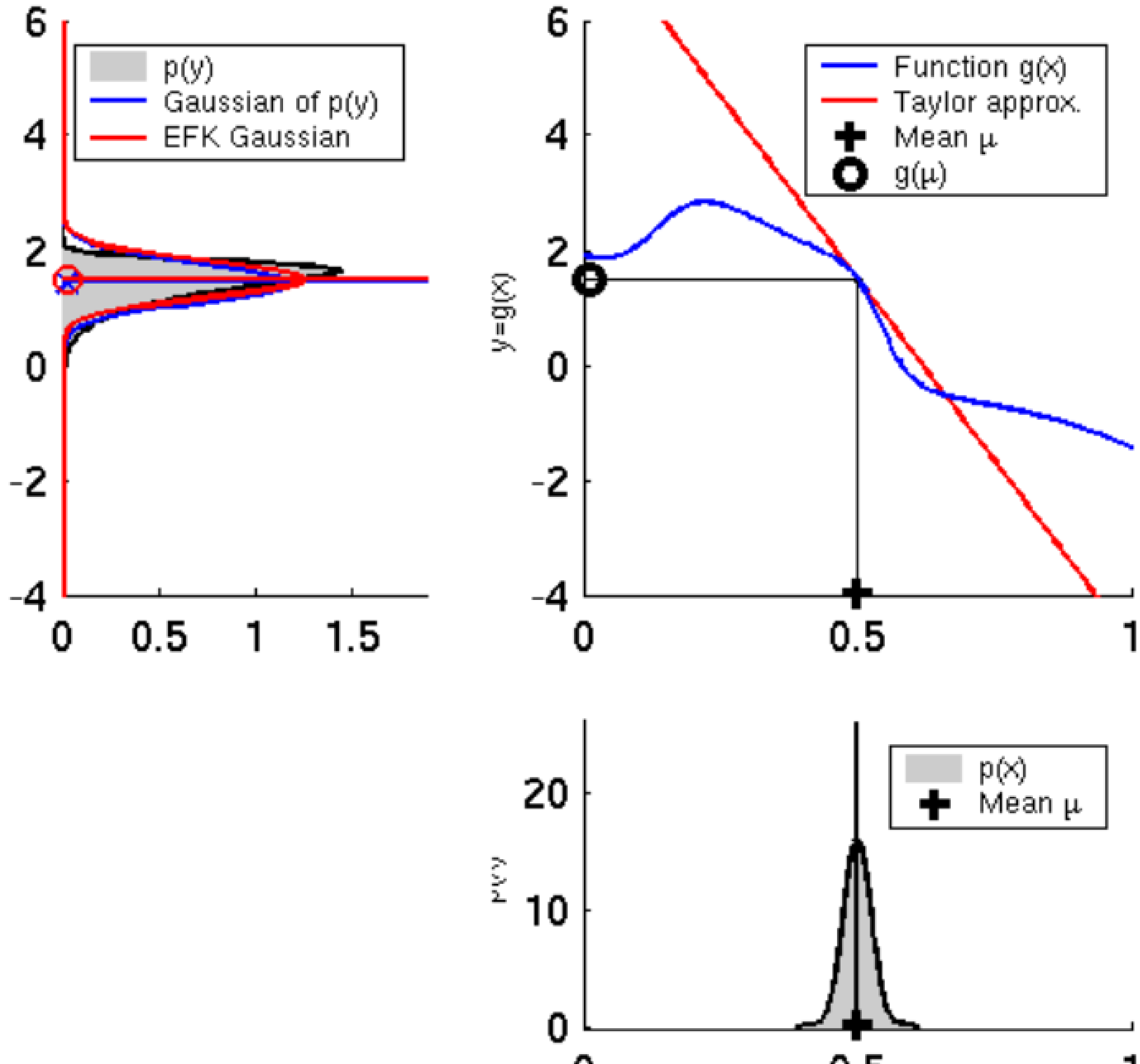
Courtesy: Thrun, Burgard, Fox

EFFECT OF LINEARIZATION: NON LINEAR CASE



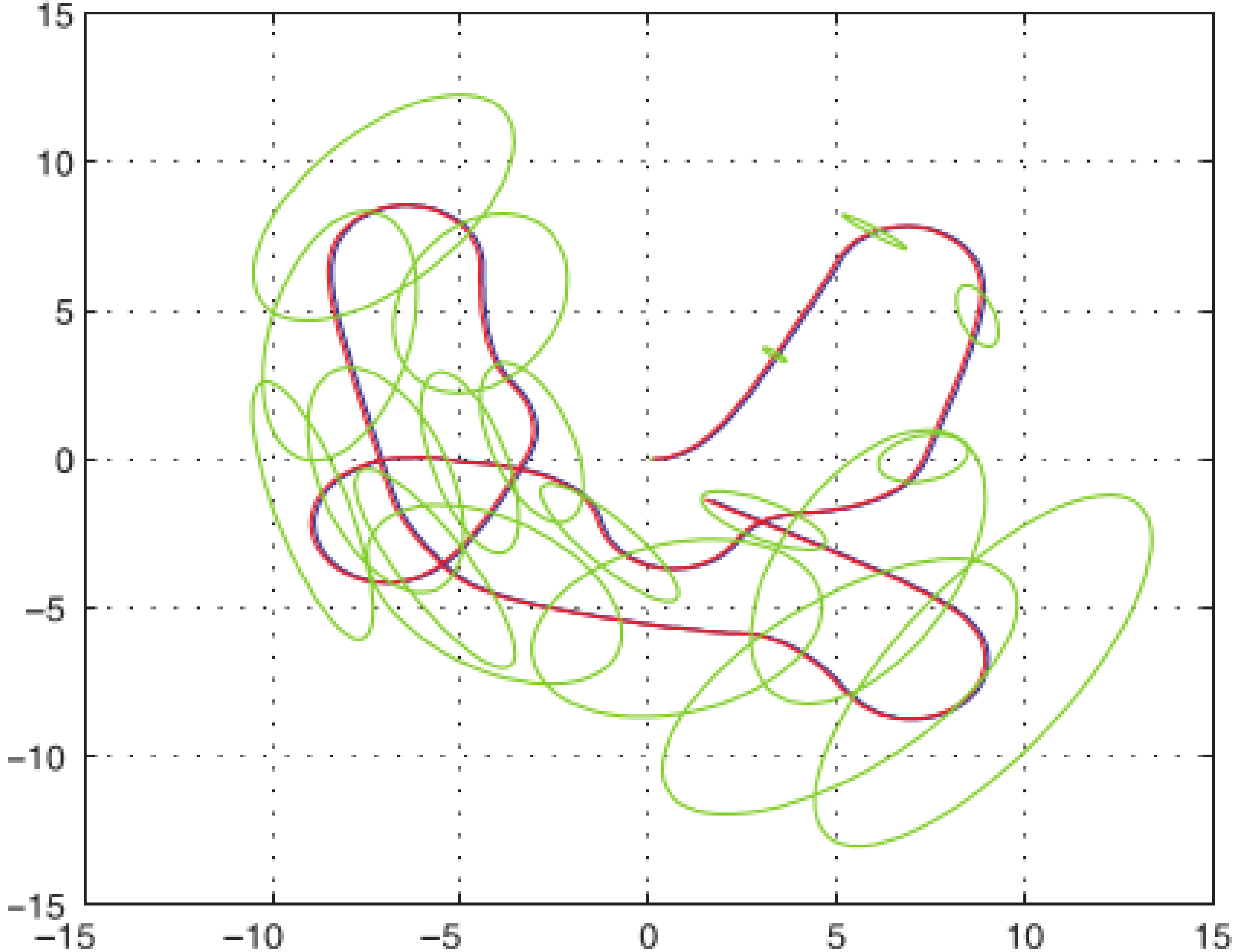
Courtesy: Thrun, Burgard, Fox

EFFECT OF LINEARIZATION: NON LINEAR CASE



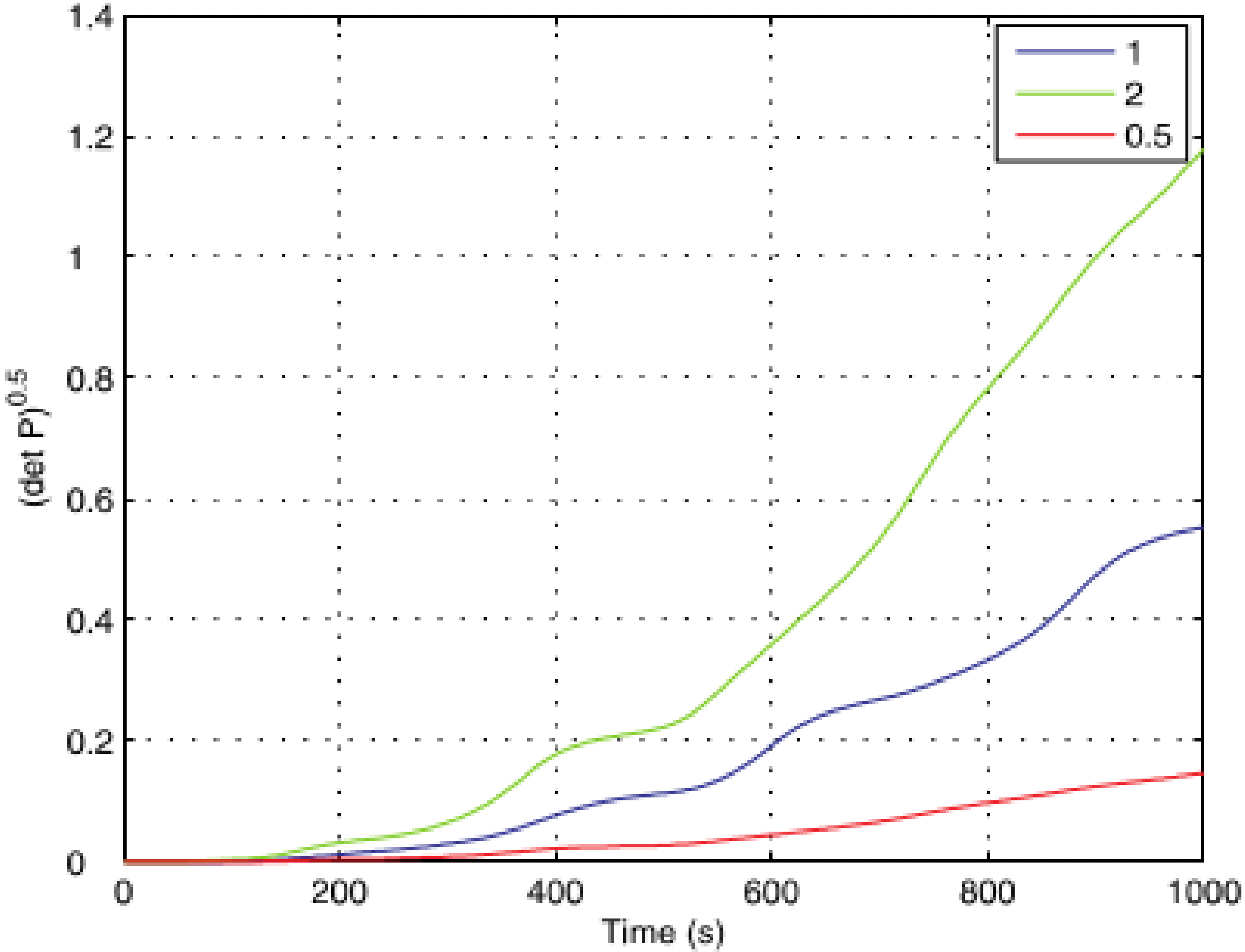
Courtesy: Thrun, Burgard, Fox

ERROR IN LOCALIZATION KEEPS GROWING



- The ellipses in the plot show the error in (x, y), but also the error in θ (the third component of the covariance matrix) grows (but usually less than that in (x, y))

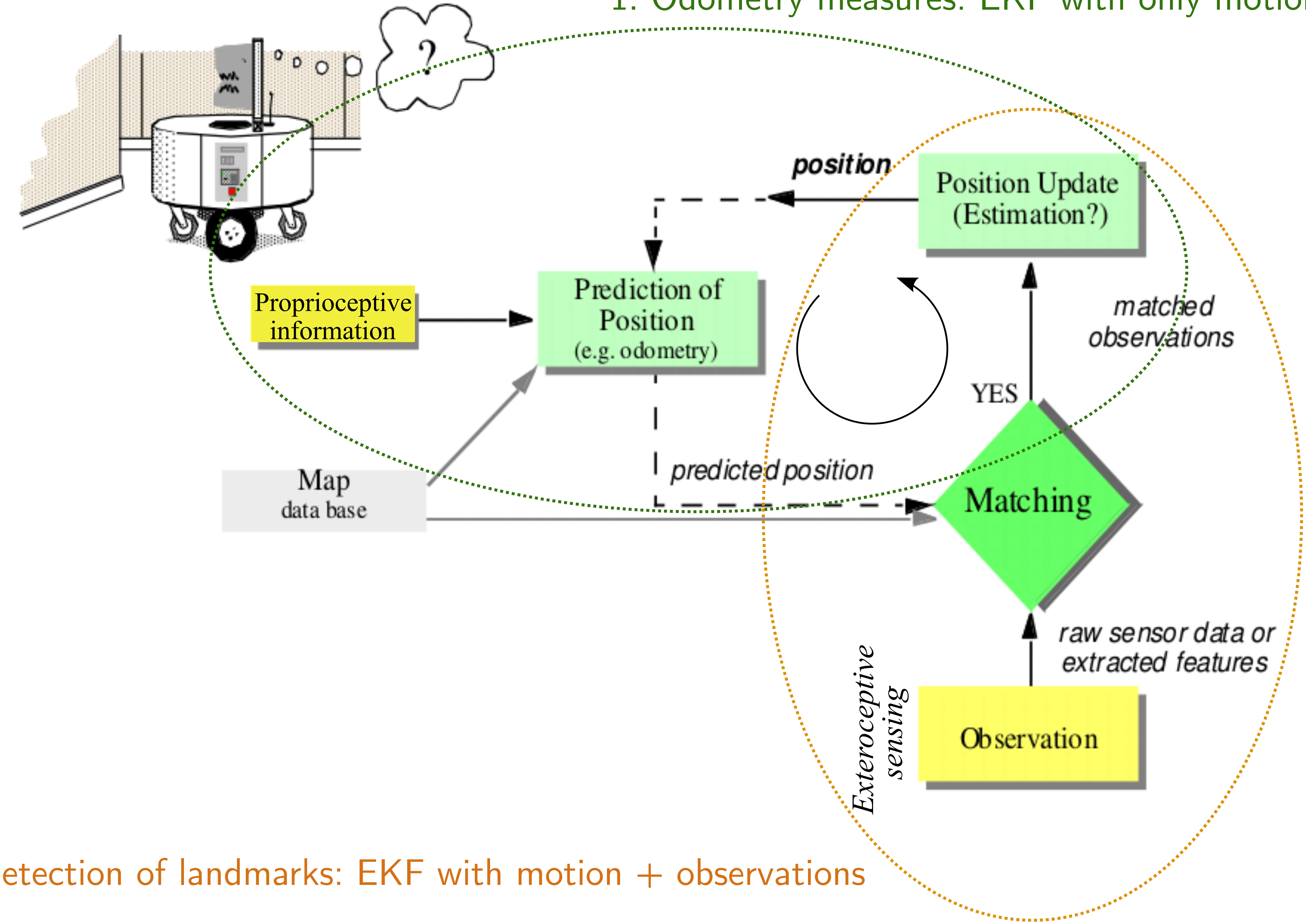
UNCERTAINTY AS PROCESS VARIANCE



- The magnitude of the total uncertainty, including both position and heading, is quantified by the $\sqrt{\det(\hat{P})}$, shown in the plot for different values of $V = \alpha V',\ \alpha = \{0.5, 1, 2\}$

EKF FOR MAP-BASED ROBOT LOCALIZATION

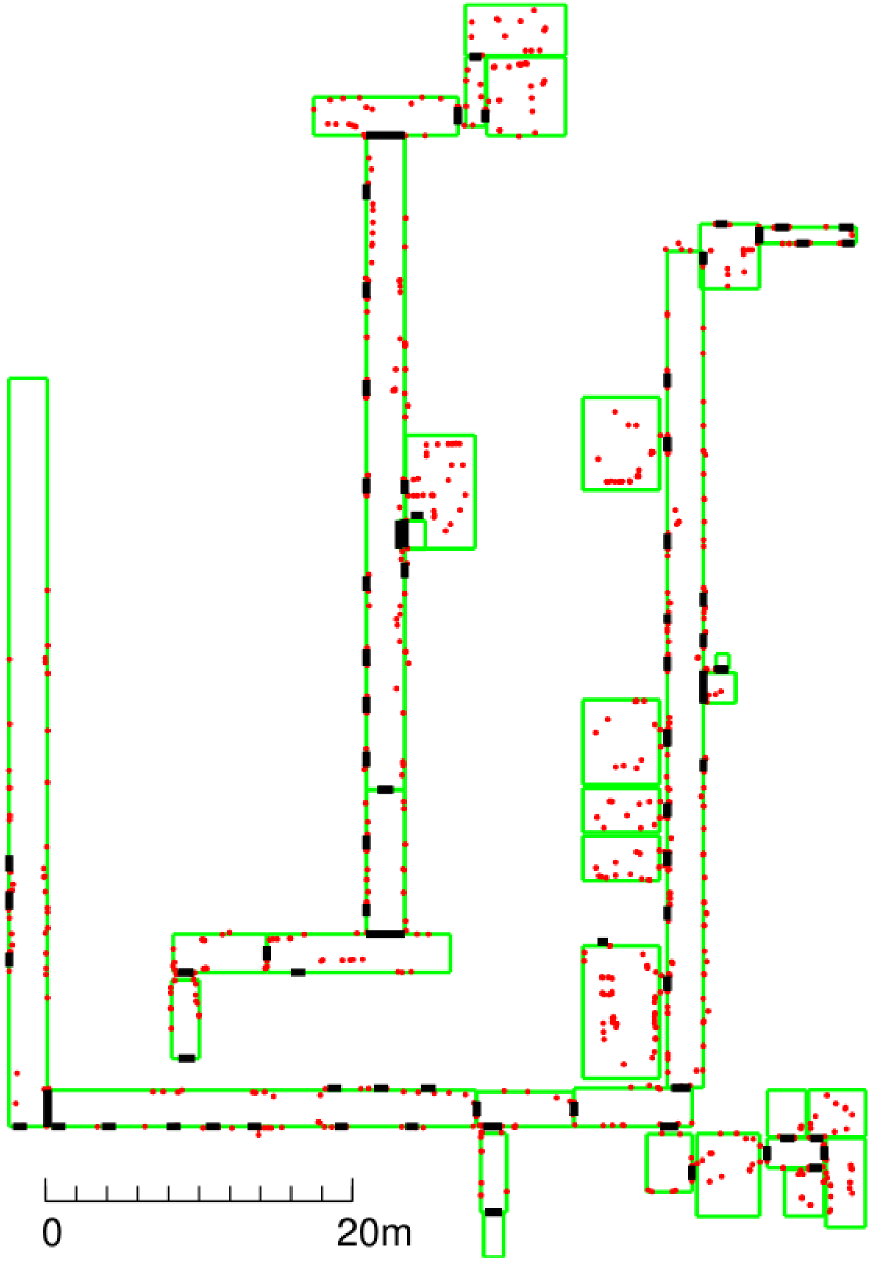
1. Odometry measures: EKF with only motion



2. Detection of landmarks: EKF with motion + observations

USING MAPS TO REDUCE THE ERROR

- Exteroceptive measures are needed in the filter to reduce pose uncertainty
- A map is provided to the robot: a list of objects in the environment along with their properties
- Let's consider the case in which the map contains n fixed landmarks with their position. **Each landmark is identifiable by the robot through a set of detectable features**



LANDMARK-BASED MAPS

The robot is equipped with (range finder) sensors that provide observations of the landmarks **with respect to the robot** as described by the observation model:

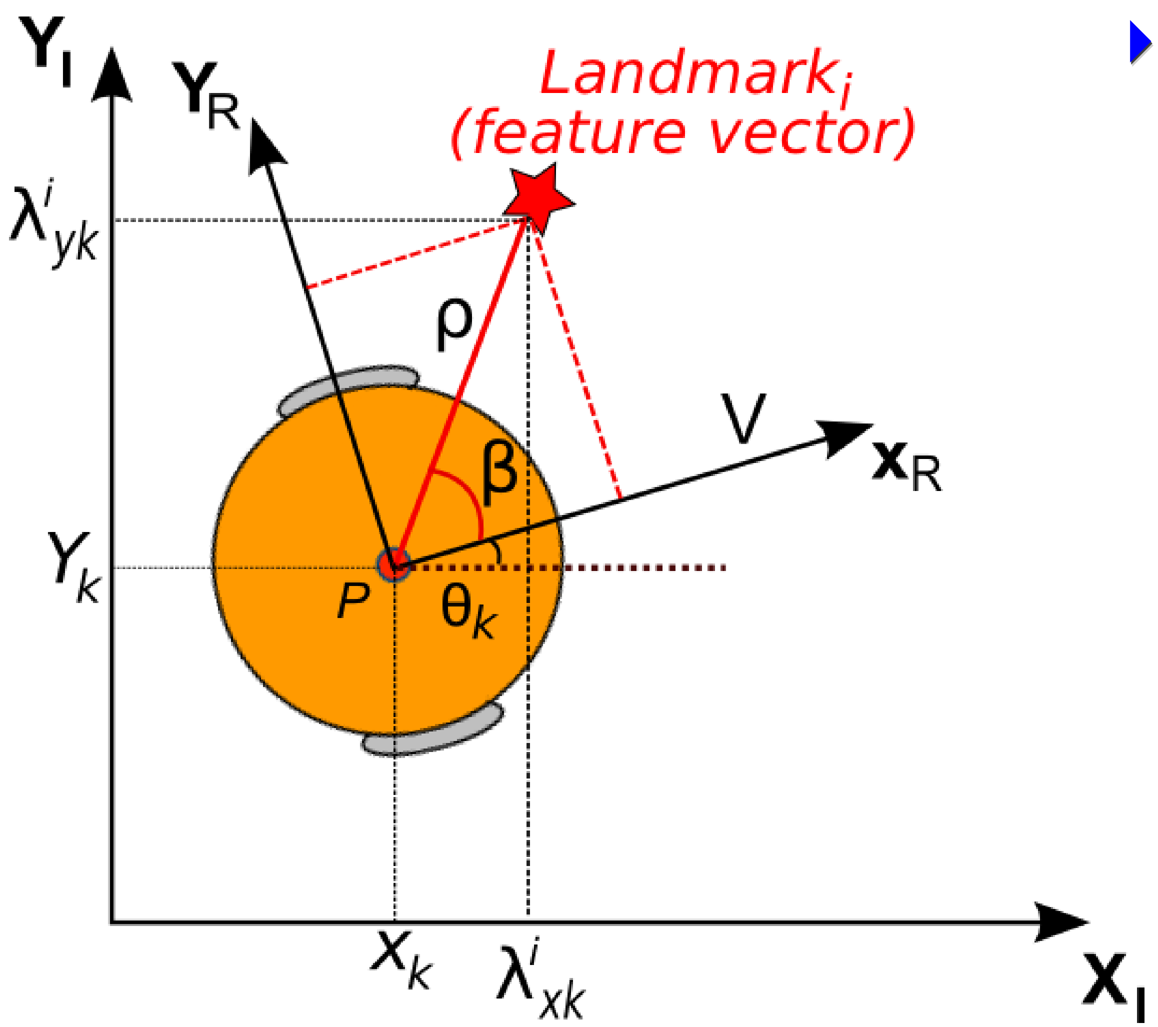
$$z_{k+1} = \boldsymbol{h}_k(\boldsymbol{\xi}_k, \boldsymbol{w}_k; \boldsymbol{\lambda}_k^i)$$

$\boldsymbol{\lambda}_k^i = \begin{bmatrix} \lambda_{kx}^i & \lambda_{ky}^i \end{bmatrix}^T$ is the known (from map) location in the world frame of the landmark observed at time step k, $\boldsymbol{w}_k$ models sensing errors, $\boldsymbol{\xi}_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$

- Using its range sensor, the robot performs the measure $z_{k+1} = \begin{bmatrix} \rho_k & \beta_k \end{bmatrix}^T$ relative to landmark i detected at step k: ρ_k is the range, β_k is the bearing angle of the landmark with respect to the robot (i.e., landmark's position expressed in polar coordinates in the robot's local frame)
- In the considered scenario, an observation also returns the **identity i of the sensed landmark**
- In more general terms, the observation of the landmarks is performed through the observation of a *feature vector* (e.g., a set of geometric features like line or arc segments), that in turn need to be associated to a specific landmark $\rightarrow$ **data association** problem, to distinguish among different landmarks as well as to discard pure noise, which is not considered here
- The knowledge of the identity i of the landmark allows the robot to retrieve from the map the **Cartesian coordinates** $(\lambda_{kx}^i, \lambda_{ky}^i)$ **of the landmark**
- In absence of specific information, the sensor noise is modeled as Gaussian white noise and the two noise components of the sensing are assumed to be uncorrelated:

$$\boldsymbol{w}_k = \begin{bmatrix} w_k^\rho & w_k^\beta \end{bmatrix}^T \sim N(0, \boldsymbol{W}_k), \quad \boldsymbol{W}_k = \begin{bmatrix} \sigma_{k\rho}^2 & 0 \\ 0 & \sigma_{k\beta}^2 \end{bmatrix}$$

LANDMARK DETECTION AND OBSERVATION MODEL



- Function $\boldsymbol{h}_k$ plays the role of $\boldsymbol{f}$ for the observations: it allows to compute the predicted measurement from the predicted state $\hat{\boldsymbol{\xi}}_{k+1|k}$. **It maps the state vector into the observation vector** z_{k+1}

> At time k, the observation model
> $\boldsymbol{h}_k(\boldsymbol{\xi}_k, \boldsymbol{w}_k; \boldsymbol{\lambda})$
> returns the observation z_{k+1}
> that the robot is expected to make in state $\boldsymbol{\xi}_k$
> accounting for sensor noise

In the scenario, at pose $\boldsymbol{\xi}_k$ the robot is expected to detect landmark i at a defined range ρ and bearing β, that is, through the measure $z_{k+1} = (\rho, \beta)$ that can be possibly corrupted by white Gaussian noise

- Since $\boldsymbol{h}_k$ maps the state (robot coordinates in the world reference frame) into the observation vector (polar coordinates of the landmark in the robot's reference frame), the observation model is:

$$z_{k+1} = \begin{bmatrix} \sqrt{(\lambda^i_{kx} - x_k)^2 + (\lambda^i_{ky} - y_k)^2} \\ \arctan\left((\lambda^i_{yx} - y_k)/(\lambda^i_{kx} - x_k)\right) - \theta_k \end{bmatrix} + \begin{bmatrix} w^\rho_k \\ w^\beta_k \end{bmatrix}$$

WHAT MEASUREMENTS TELL

- $\boldsymbol{h}_k$ **potentially changes at each time step**, being parametrized by the coordinates $\boldsymbol{\lambda}_k^i = (\lambda_{kx}^i, \lambda_{ky}^i)$ of the specific landmark detected, whose identity i is assumed to be known/acquired

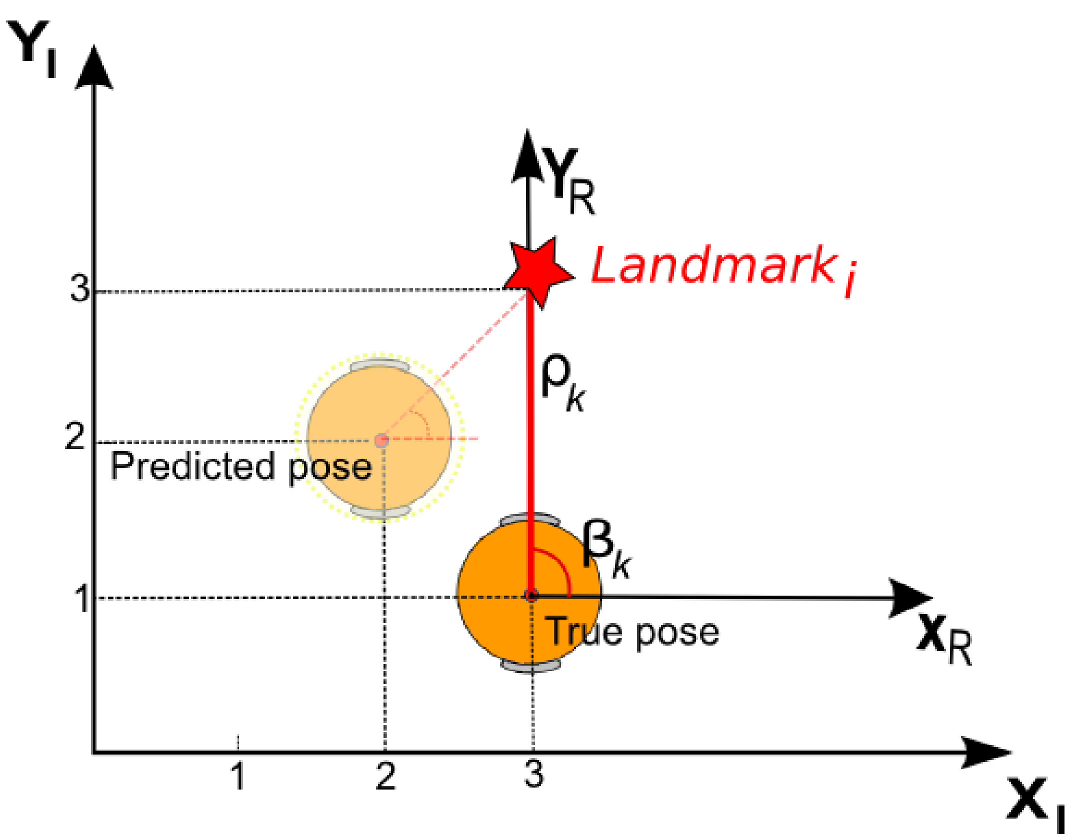
- Using the observation model $\boldsymbol{h}_k$, the robot computes the expected range and the bearing angle to the detected feature based on its own *predicted pose* $\hat{\boldsymbol{\xi}}_{k+1|k}$ and the *known* position of the landmark from the input map

> Any difference between the actual observation $\boldsymbol{z}_{k+1} = (\rho_k, \beta_k)$
> and the estimated observation/position $\boldsymbol{h}_k(\hat{\boldsymbol{\xi}}_{k+1|k}; \boldsymbol{\lambda}_k^i)$
> indicates an error in the robot's position estimate:
> *the robot isn't where it thought it was!*

> The difference is quantified in the Kalman filter by the **innovation** term:
> $$\boldsymbol{\epsilon}_{k+1} = \boldsymbol{z}_{k+1} - \boldsymbol{h}_k(\hat{\boldsymbol{\xi}}_{k+1|k}, \boldsymbol{0}; \boldsymbol{\lambda}_{k+1}^i)$$

- Same problem as before: $\boldsymbol{h}$ is a non linear function of the state!

- Example: at step $k+1$ the robot detects landmark i at a relative range of 2m and a relative angle of 90^o, that is, $z_{k+1} = \begin{bmatrix} 2 & 90 \end{bmatrix}^T$; from the input map, position of landmark i is $\boldsymbol{\lambda}^i = (3, 3)$; robot's predicted pose according to the current state of the Kalman filter is $\hat{\boldsymbol{\xi}}_{k+1|k} = \begin{bmatrix} 2 & 2 & 0 \end{bmatrix}^T$, while its correct pose is $\boldsymbol{\xi}_{k+1} = \begin{bmatrix} 3 & 1 & 0 \end{bmatrix}^T$ (i.e., there is no sensing error, as it can be seen from the figure)



- The innovation is:

$$
\begin{aligned}
\boldsymbol{\epsilon}_{k+1} &= z_{k+1} - \boldsymbol{h}_k(\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}; \boldsymbol{\lambda}^i_{k+1}) \\
&= \begin{bmatrix} 2 \\ 90 \end{bmatrix} - \begin{bmatrix} \sqrt{1^2 + 1^2} \\ \arctan(1/1) - 0 \end{bmatrix} = \begin{bmatrix} 2 - \sqrt{2} \\ 45^o \end{bmatrix}
\end{aligned}
$$

In [m, rad] units, the Euclidean norm of the innovation is: $\begin{bmatrix} 2 - \sqrt{2} & 0.79 \end{bmatrix}^T$

$\Rightarrow \|\boldsymbol{\epsilon}_{k+1}\| = \sqrt{(2 - \sqrt{2})^2 + 0.79^2} \approx 0.98$

LINEARIZATION OF THE OBSERVATION MODEL

Linearized observation model in the EKF:

1st order Taylor expansion for $\boldsymbol{h}_k()$ in the neighborhood of the current state estimate, and parametrized by the coordinates $\boldsymbol{\lambda}_k$, results in:

$$\begin{aligned}\boldsymbol{h}_k(\boldsymbol{\xi}_k, \boldsymbol{w}_k; \boldsymbol{\lambda}_k) &= \boldsymbol{h}_k(\boldsymbol{\xi}, \boldsymbol{w}; \boldsymbol{\lambda}_k)|_{\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}} + (\boldsymbol{\xi}_k - \hat{\boldsymbol{\xi}}_{k+1|k}) \boldsymbol{H}_{\boldsymbol{\xi}}|_{\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}} + (\boldsymbol{w}_k - \mathbf{0}) \boldsymbol{H}_{\boldsymbol{w}}|_{\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}} \\ &= \boldsymbol{h}_k(\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}; \boldsymbol{\lambda}_k) + (\boldsymbol{\xi}_k - \hat{\boldsymbol{\xi}}_{k+1|k}) \boldsymbol{H}_{k\xi} + \boldsymbol{w}_k \boldsymbol{H}_{kw}\end{aligned}$$

Therefore, observation predictions return linear and can be used in the EKF equations below by using $\boldsymbol{H}$, the Jacobian of $\boldsymbol{h}$, to play the role of matrix $\boldsymbol{C}$

$$\text{Prediction update} \begin{cases} \hat{\boldsymbol{\xi}}_{k+1|k} = \boldsymbol{f}_k(\hat{\boldsymbol{\xi}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}) & \text{(State prediction)} \\ \boldsymbol{P}_{k+1|k} = \boldsymbol{F}_{k\xi} \boldsymbol{P}_k \boldsymbol{F}_{k\xi}^T + \boldsymbol{F}_{k\nu} \boldsymbol{V}_k \boldsymbol{F}_{k\nu}^T & \text{(Covariance prediction)} \end{cases}$$

$$\text{Measurement correction} \begin{cases} \hat{\boldsymbol{\xi}}_{k+1} = \hat{\boldsymbol{\xi}}_{k+1|k} + \boldsymbol{G}_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{h}_k(\hat{\boldsymbol{\xi}}_{k+1|k}, \mathbf{0}; \boldsymbol{\lambda}_k^i)) & \text{(State update)} \\ \boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{G}_{k+1} \boldsymbol{H}_{k\xi} \boldsymbol{P}_{k+1|k} & \text{(Covariance update)} \\ \boldsymbol{G}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k\xi}{}^T \boldsymbol{S}_{k+1}^{-1} & \text{(Kalman gain)} \\ \boldsymbol{S}_{k+1} = \boldsymbol{H}_{k\xi} \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k\xi}{}^T + \boldsymbol{H}_{kw} \boldsymbol{W}_{k+1} \boldsymbol{H}_{kw}{}^T & \end{cases}$$

JACOBIANS FOR THE LINEARIZED OBSERVATION MODEL

- The Jacobian of the non-linear function $\boldsymbol{h}_k$ is computed at the mean of the Gaussian measurement noise ($\boldsymbol{w} = \boldsymbol{0}$) and at the current state estimate $\hat{\boldsymbol{\xi}}_{k+1|k}$ (which corresponds to the estimated mean of the Gaussian distribution of the state variable):
- Let's adopt a notation similar to the one used before for $\boldsymbol{f}$ to express the function $\boldsymbol{h}_k$, defining $\boldsymbol{h}_k = \begin{bmatrix} h_{k\rho} & h_{k\beta} \end{bmatrix}^T$ and including sensor noise:

$$\begin{aligned} h_{k\rho} &= \sqrt{(\lambda^i_{kx} - x_k)^2 + (\lambda^i_{ky} - y_k)^2} + w^\rho_k \\ h_{k\beta} &= \arctan\Big((\lambda^i_{yx} - y_k)/(\lambda^i_{kx} - x_k)\Big) - \theta_k + w^\beta_k \end{aligned}$$

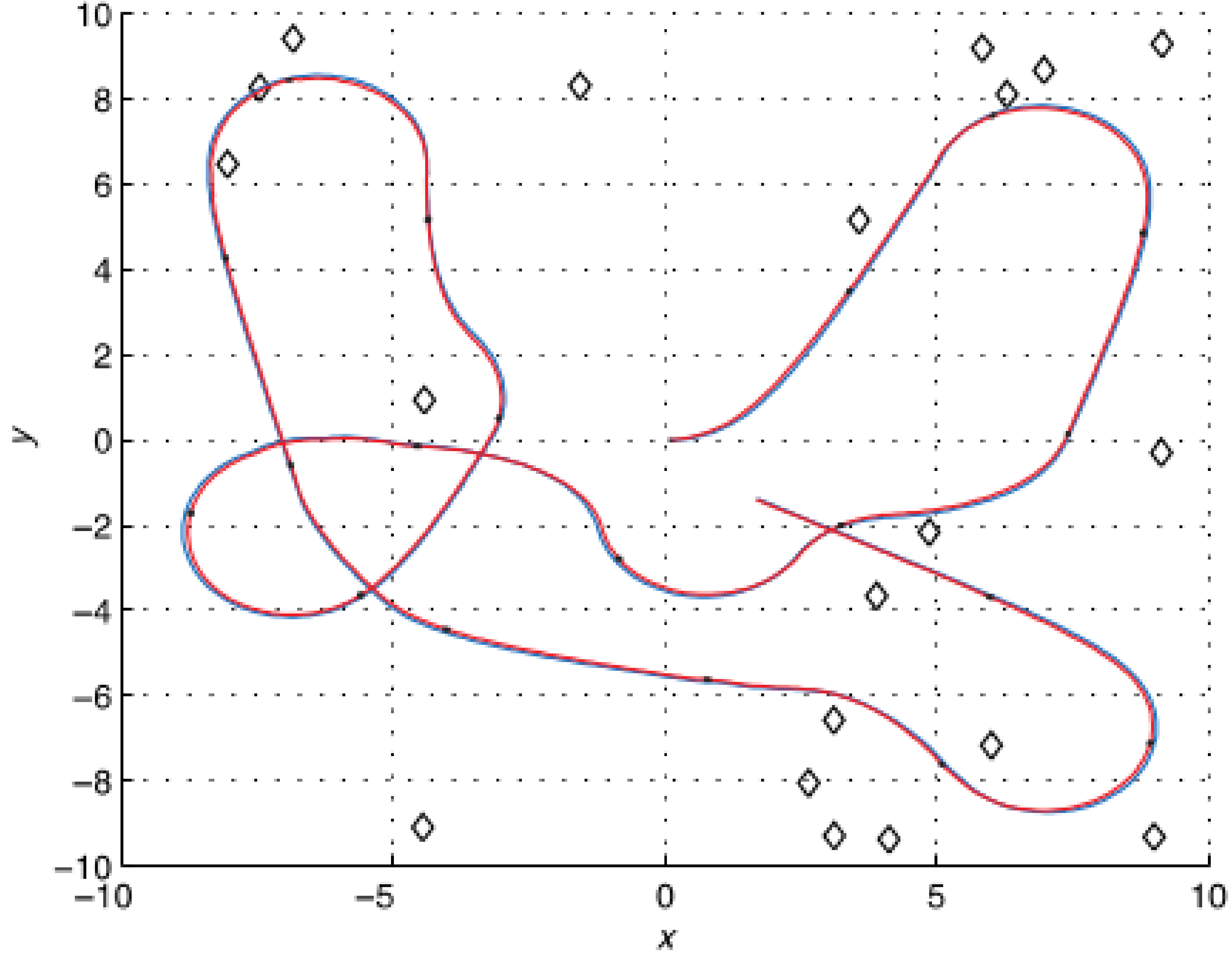
The Jacobian matrix of $\boldsymbol{h}_k$ is therefore:

$$\boldsymbol{H}_k(x_k, y_k, \theta_k,\ w^\rho_k, w^\beta_k) = \begin{bmatrix} \nabla h_{k\rho} & \nabla h_{k\beta} \end{bmatrix}^T = \begin{bmatrix} \frac{\partial h_{k\rho}}{\partial x_k} & \frac{\partial h_{k\rho}}{\partial y_k} & \frac{\partial h_{k\rho}}{\partial \theta_k} & \frac{\partial h_{k\rho}}{\partial w^\rho_k} & \frac{\partial h_{k\rho}}{\partial w^\beta_k} \\ \frac{\partial h_{k\beta}}{\partial x_k} & \frac{\partial h_{k\beta}}{\partial y_k} & \frac{\partial h_{k\beta}}{\partial \theta_k} & \frac{\partial h_{k\beta}}{\partial w^\rho_k} & \frac{\partial h_{k\beta}}{\partial w^\beta_k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_{k\boldsymbol{\xi}} & \boldsymbol{H}_{k\boldsymbol{w}} \end{bmatrix}$$

$$\boldsymbol{H}_{k\boldsymbol{\xi}} = \begin{bmatrix} -\frac{\lambda^i_{kx} - x_k}{r^i_k} & -\frac{\lambda^i_{ky} - y_k}{r^i_k} & 0 \\ \frac{\lambda^i_{ky} - y_k}{(r^i_k)^2} & -\frac{\lambda^i_{kx} - x_k}{(r^i_k)^2} & -1 \end{bmatrix}_{\hat{\boldsymbol{\xi}}_{k+1|k}, \boldsymbol{w}=0} \qquad \boldsymbol{H}_{k\boldsymbol{w}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

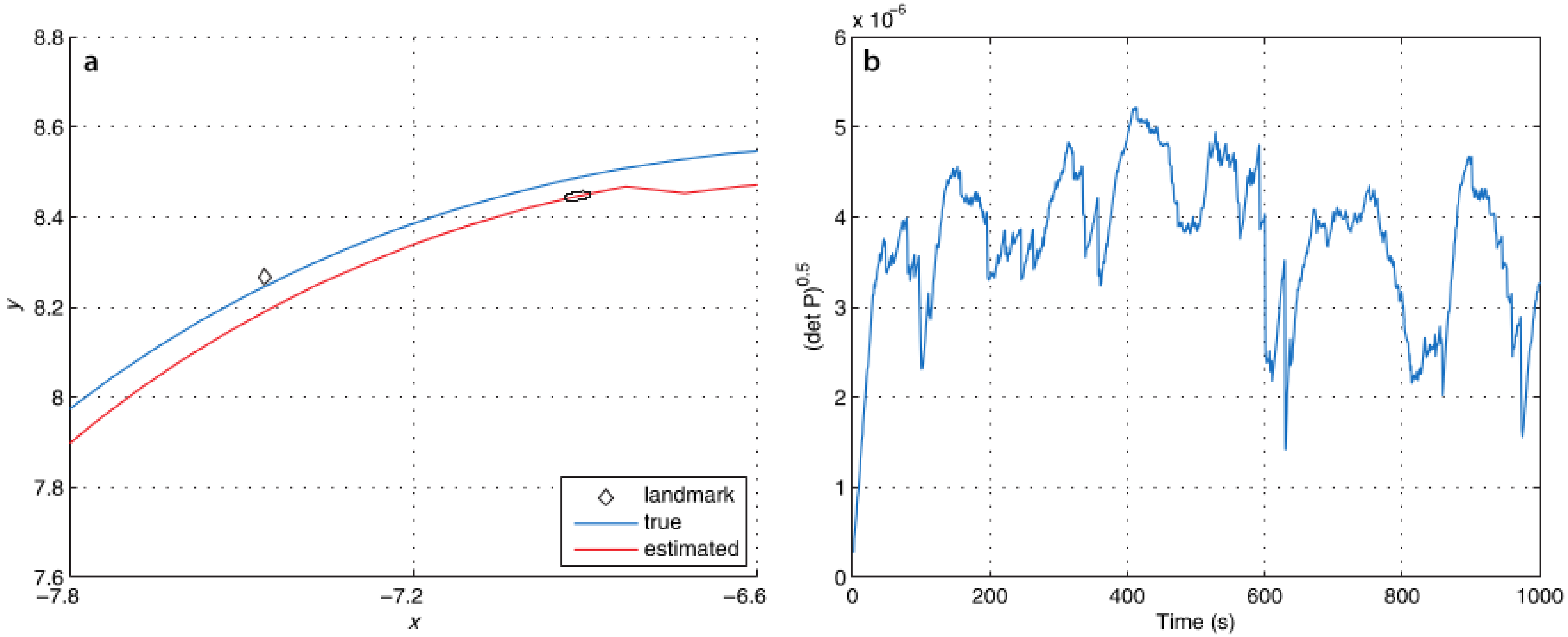
where r^i_k is the distance of landmark i from the predicted state: $r^i_k = \sqrt{(\lambda^i_{kx} - x_k)^2 + (\lambda^i_{ky} - y_k)^2}$

EXPERIMENTAL RESULTS: AN ALMOST PERFECT TRACKING

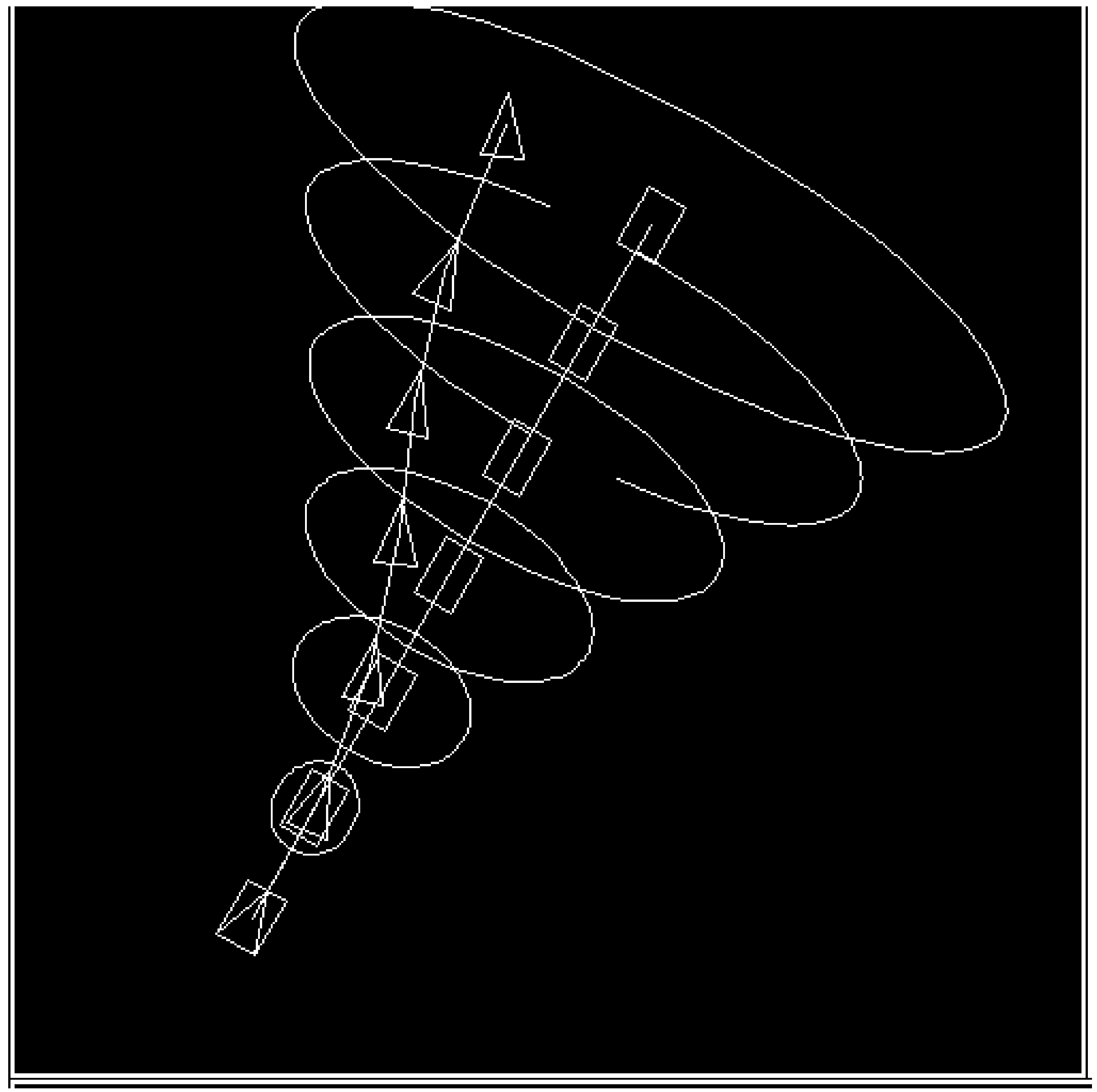


- $n = 20$ landmarks are randomly deployed in a squared environment of 20×20 m^2
- $\sigma_\rho = 0.1$ m, $\sigma_\beta = 1^o$
- Every n steps, a reading is performed, returning the measured range and bearing to a randomly selected landmark
- This is a quite favorable scenario for the EKF

EVOLUTION OF THE ERROR: NO SYSTEMATIC GROWTH

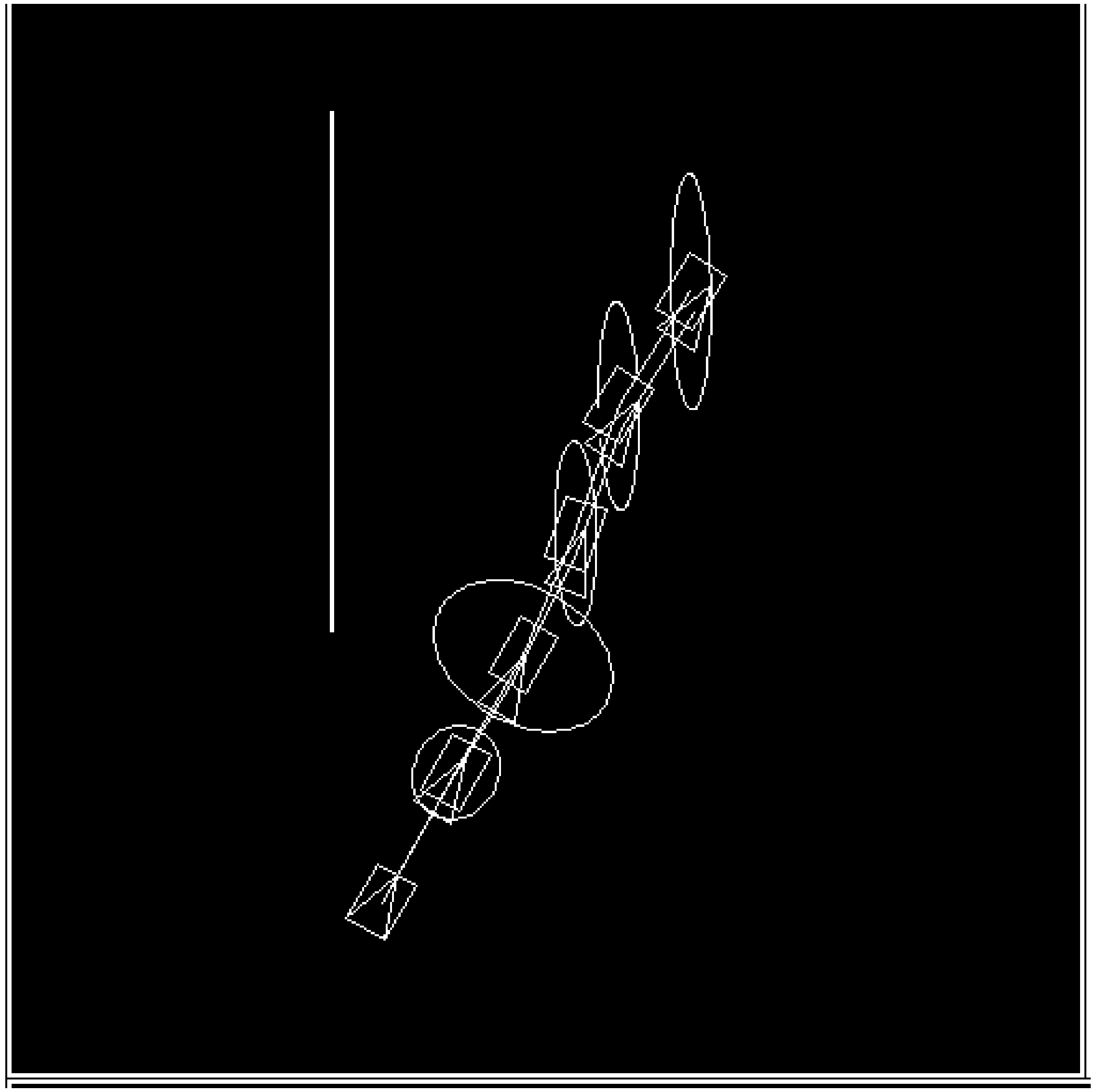


EKF WITH ENVIRONMENT BEACONS



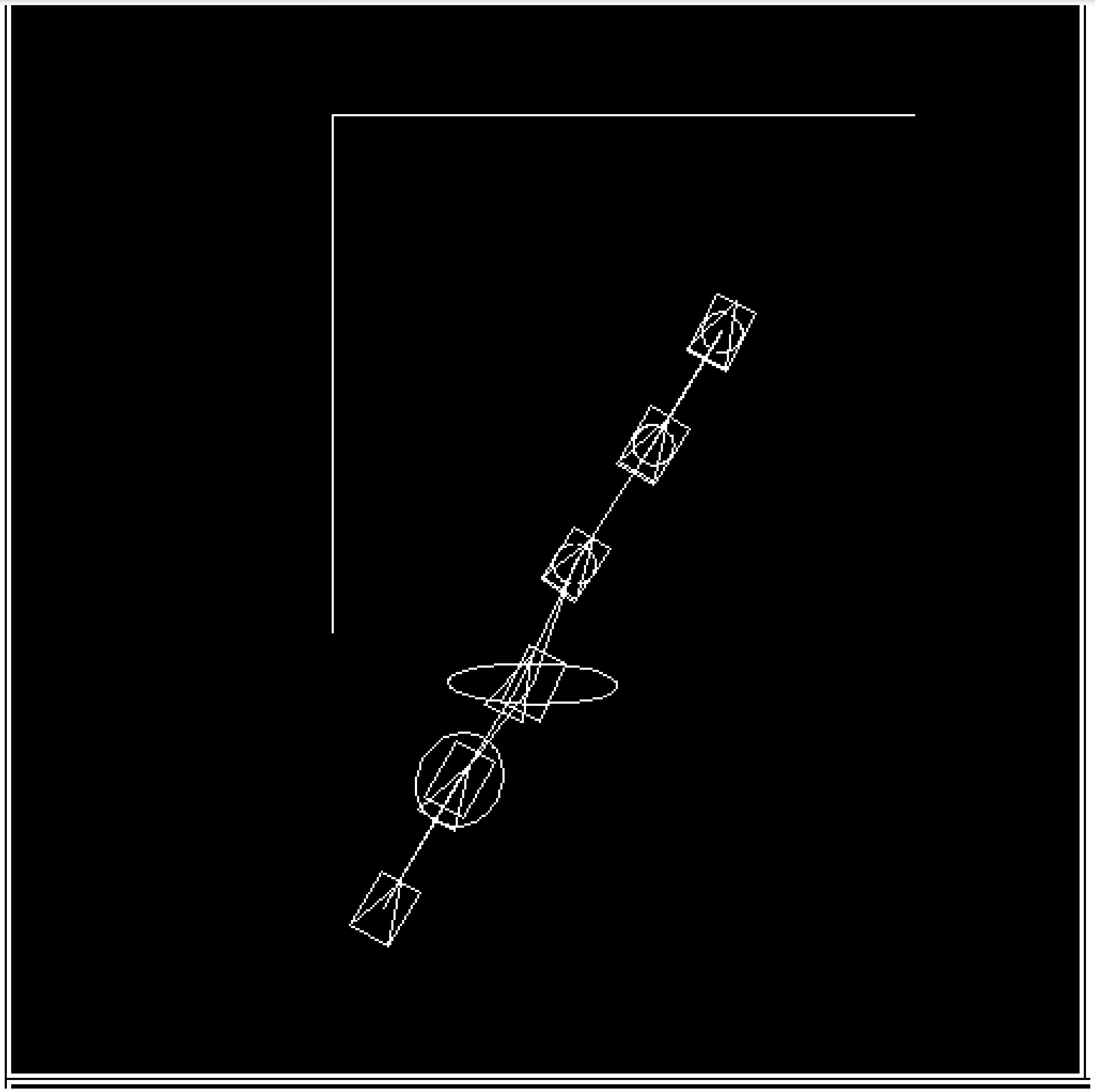
- Simulated run with no visible beacons.
- The triangles represent the actual robot position and orientation $\left[x(k), y(k), \theta(k)\right]^T$, the rectangles represent the estimated robot pose, the ellipses represent the confidence in the estimates of $x(k)$ and $y(k)$

EKF WITH ENVIRONMENT BEACONS



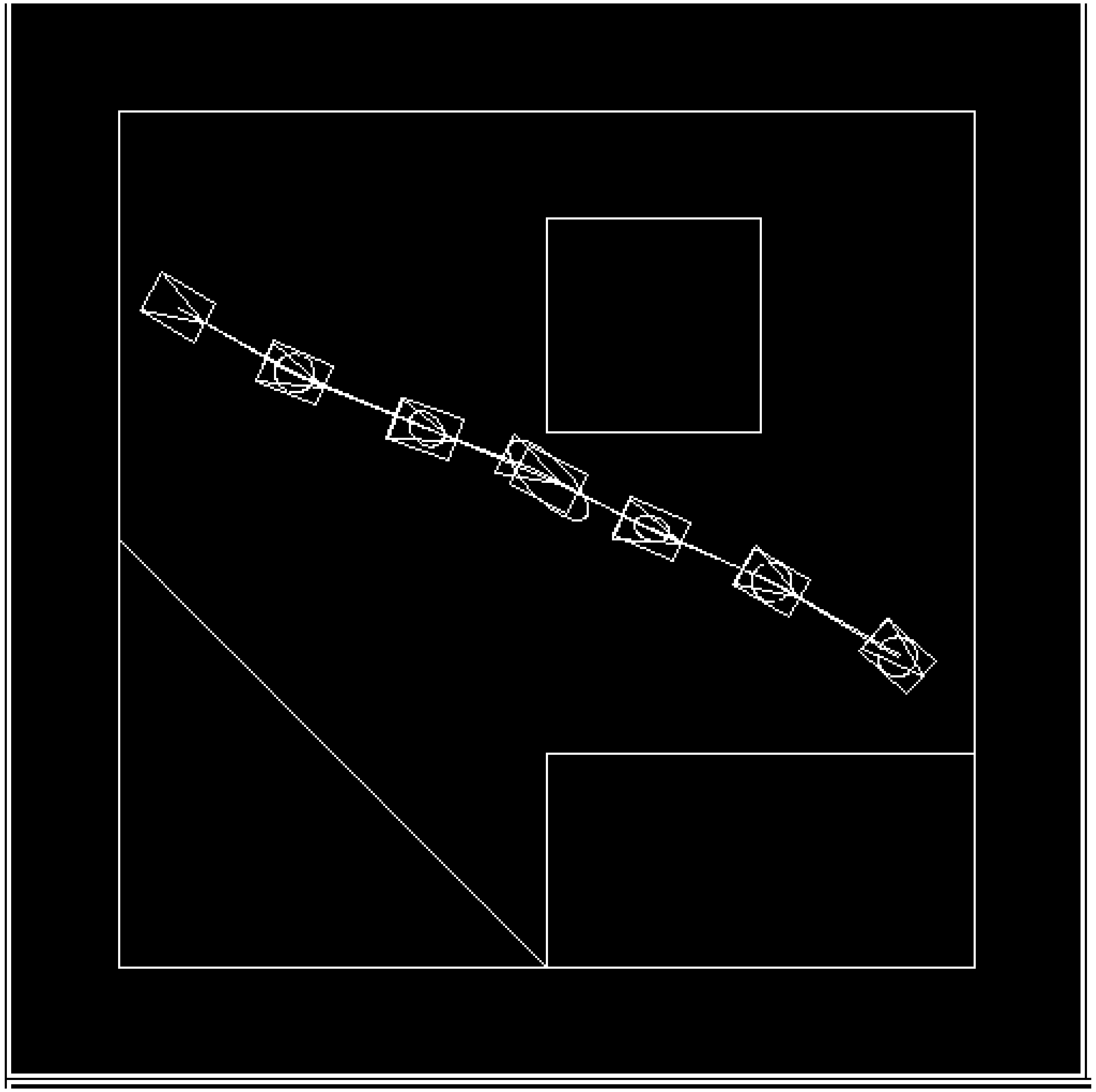
- Simulated run taking observations of a single wall beacon using a sonar sensors.
- After the wall comes into view, the **error ellipse shrinks perpendicular to the wall** as a posteriori confidence in the estimate of $x(k)$ and $y(k)$ increases.
- Note that the only part of a smooth wall that can be "seen" by a **sonar sensor** is the portion of the wall that is perpendicular to the incident sonar beam.

EKF WITH ENVIRONMENT BEACONS



- Simulated run with localization from first one, then two wall beacons.
- After the first wall comes into view, the error ellipse shrinks perpendicular to the wall as a posteriori confidence in the estimate of $x(k)$ and $y(k)$ increases. The same happens with the view of the second wall, overall reducing estimate uncertainty.

EKF WITH ENVIRONMENT BEACONS



- Simulated run with localization from a sequence of wall beacons
- The presence of multiple wall beacons allows to always keep uncertainty estimation very low.