Introduction to Mobile Robotics

Error Propagation

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Kai Arras

Slides by Kai Arras

Last update: June 2010

Error Propagation: Motivation

- Probabilistic robotics is
  - **Representation**
  - **Propagation**
  - **Reduction**

  of uncertainty

- **First-order error propagation** is fundamental for:
  Kalman filter (KF), landmark extraction, KF-based localization and SLAM

Gaussian Distribution

Why is the Gaussian distribution everywhere?

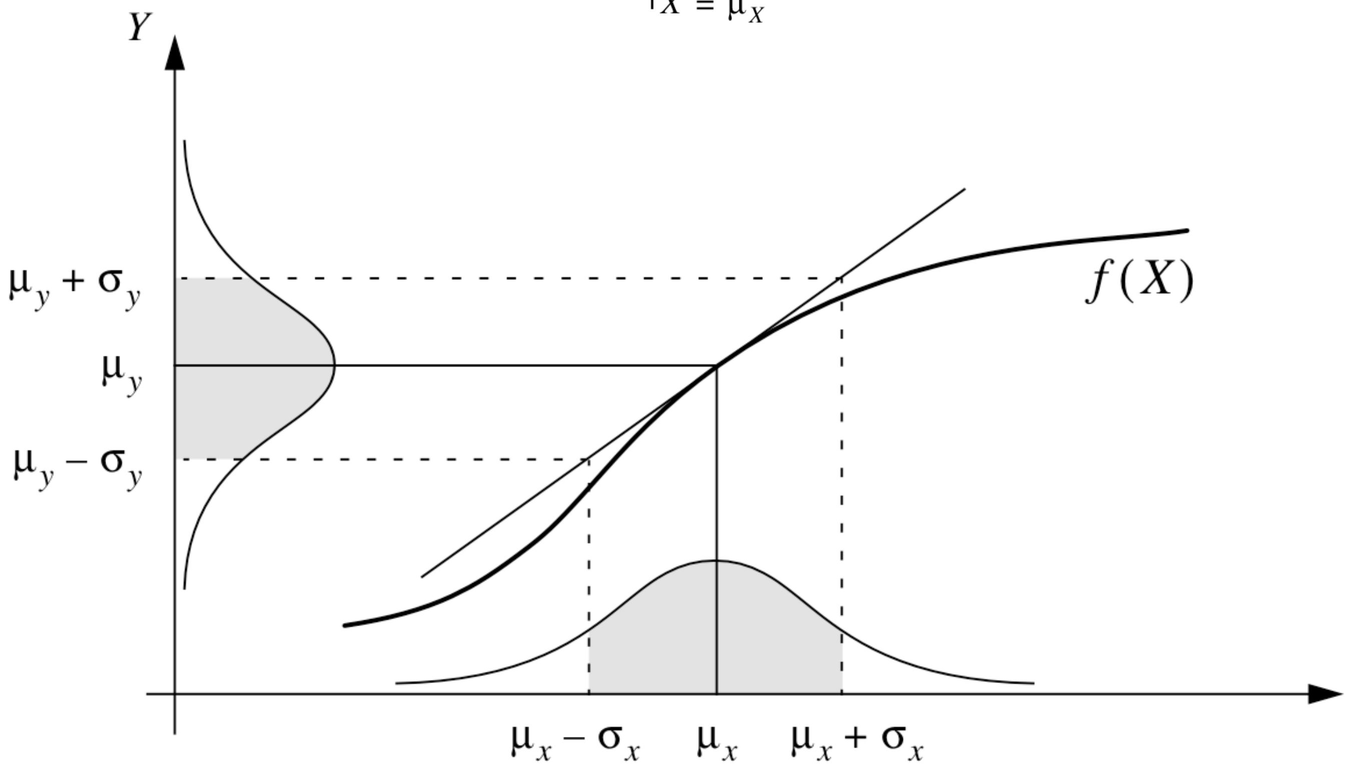
The importance of the normal distribution follows mainly from the **Central Limit Theorem**:

- The mean/sum of a large number of independent RVs, each with finite mean and variance (ergo not e.g. uniformally distributed RVs), will be approximately **normally distributed**.
- The more RVs the better the approximation.

First-Order Error Propagation

Approximating *f(X)* by a **first-order** Taylor series expansion about the point $X = \mu_X$

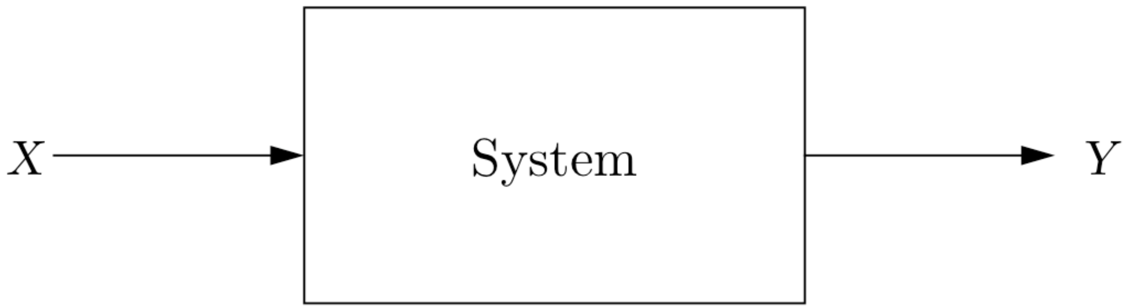
$$Y \approx f(\mu_X) + \left.\frac{\partial f}{\partial X}\right|_{X = \mu_X} (X - \mu_X)$$



First-Order Error Propagation

X,Y assumed to be Gaussian

$Y = f(X)$



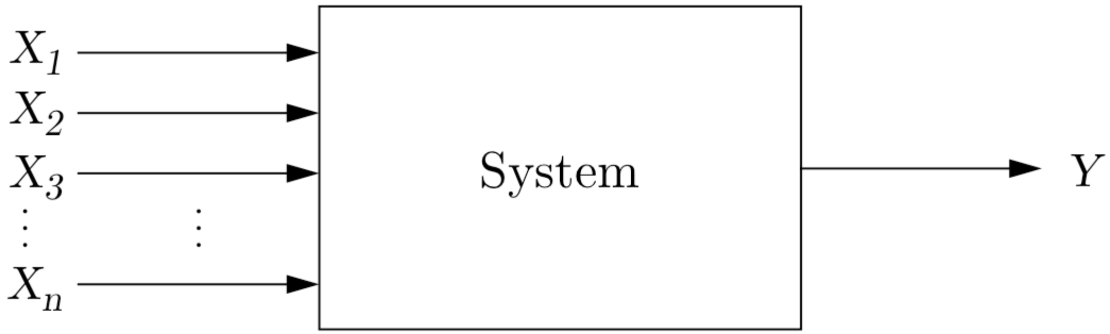
Taylor series expansion

$$Y \approx f(\mu_X) + \left.\frac{\partial f}{\partial X}\right|_{X = \mu_X} (X - \mu_X)$$

Wanted: $\mu_Y\,,\ \sigma_Y^2$ (Solution on blackboard)

First-Order Error Propagation

$$Y = f(X_1, X_2, \ldots, X_n)$$



Taylor series expansion

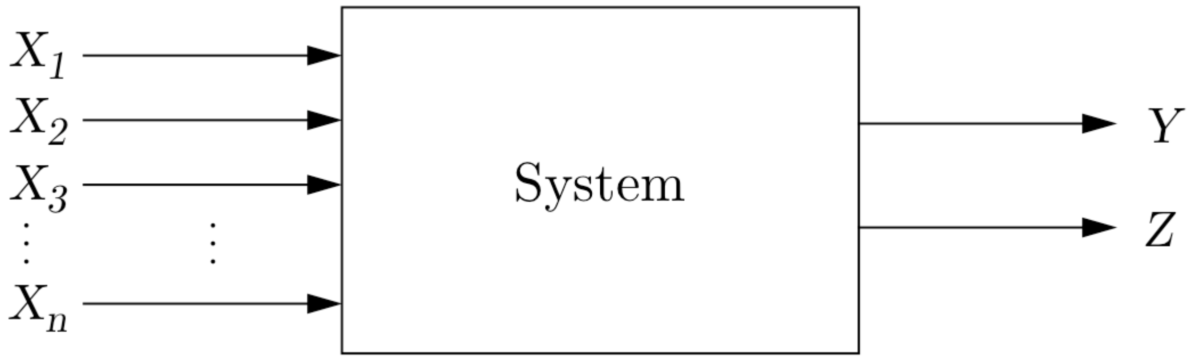
$$Y \approx f(\mu_1, \mu_2, \ldots, \mu_n) + \sum_{i=1}^{n} \left[\frac{\partial f}{\partial X_i}(\mu_1, \mu_2, \ldots, \mu_n)\right][X_i - \mu_i]$$

Wanted: μ_Y , σ_Y^2 (Solution on blackboard)

First-Order Error Propagation

$$Y = f(X_1, X_2, \dots, X_n)$$

$$Z = g(X_1, X_2, \dots, X_n)$$



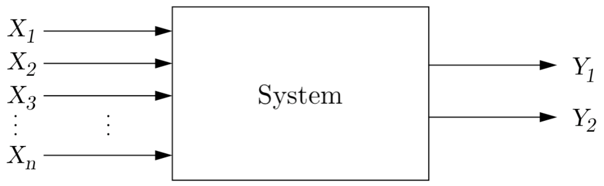
Wanted: σ_{YZ}

(Exercise)

First-Order Error Propagation

Putting things together...

$$C_X = \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_n} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_n} \\ \vdots & \vdots & & \vdots \\ \sigma_{X_nX_1} & \sigma_{X_nX_2} & \cdots & \sigma^2_{X_n} \end{bmatrix}$$



$$C_Y = \begin{bmatrix} \sigma^2_{Y_1} & \sigma_{Y_1Y_2} \\ \sigma_{Y_2Y_1} & \sigma^2_{Y_2} \end{bmatrix}$$

with

$$\sigma_Y^2 = \sum_i \left(\frac{\partial f}{\partial X_i}\right)^2 \sigma_i^2 + \sum_{i \neq j}\sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial f}{\partial X_j}\right)\sigma_{ij}$$

$$\sigma_{YZ} = \sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial g}{\partial X_i}\right)\sigma_i^2 + \sum_{i \neq j}\sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial g}{\partial X_j}\right)\sigma_{ij}$$

→ "Is there a **compact form?...**"

Jacobian Matrix

- It's a **non-square matrix** $n \times m$ in general
- Suppose you have a vector-valued function $f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix}$
- Let the **gradient operator** be the vector of (first-order) partial derivatives

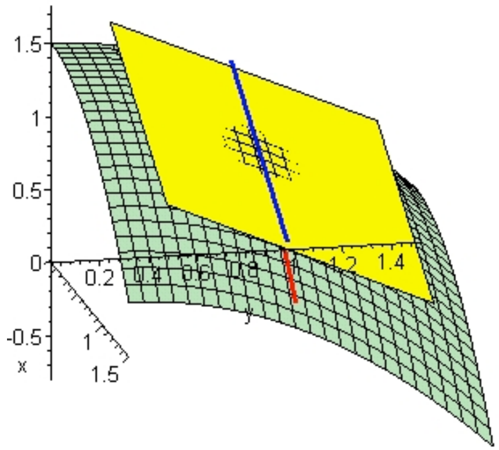
$$\nabla_{\mathbf{x}} = \left[\begin{array}{cccc} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \cdots & \frac{\partial}{\partial x_n} \end{array} \right]^T$$

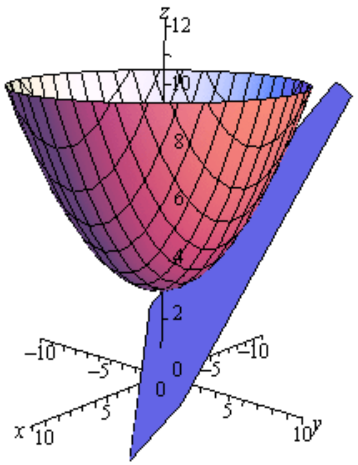
- Then, the **Jacobian matrix** is defined as

$$\mathbf{F}_{\mathbf{x}} = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix} \cdot \left[\begin{array}{ccc} \frac{\partial}{\partial x_1} & \cdots & \frac{\partial}{\partial x_n} \end{array} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \end{bmatrix}$$

Jacobian Matrix

- It's the orientation of the **tangent plane** to the vector-valued function at a given point



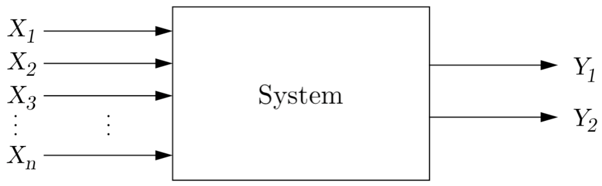


- **Generalizes the gradient** of a scalar valued function
- Heavily used for **first-order error propagation...**

First-Order Error Propagation

Putting things together...

$$C_X = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_n} \\ \sigma_{X_2X_1} & \sigma_{X_2}^2 & \cdots & \sigma_{X_2X_n} \\ \vdots & \vdots & & \vdots \\ \sigma_{X_nX_1} & \sigma_{X_nX_2} & \cdots & \sigma_{X_n}^2 \end{bmatrix}$$



$$C_Y = \begin{bmatrix} \sigma_{Y_1}^2 & \sigma_{Y_1Y_2} \\ \sigma_{Y_2Y_1} & \sigma_{Y_2}^2 \end{bmatrix}$$

with

$$\sigma_Y^2 = \sum_i \left(\frac{\partial f}{\partial X_i}\right)^2 \sigma_i^2 + \sum_{i \neq j}\sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial f}{\partial X_j}\right)\sigma_{ij}$$

$$\sigma_{YZ} = \sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial g}{\partial X_i}\right)\sigma_i^2 + \sum_{i \neq j}\sum \left(\frac{\partial f}{\partial X_i}\right)\left(\frac{\partial g}{\partial X_j}\right)\sigma_{ij}$$

→ "Is there a **compact form?...**"

First-Order Error Propagation

...Yes! Given

- Input covariance matrix C_X
- Jacobian matrix F_X

the **Error Propagation Law**

$$\boxed{C_Y = F_X C_X F_X^T}$$

computes the output covariance matrix C_Y

First-Order Error Propagation

Alternative Derivation in Matrix Notation

$$
\begin{aligned}
\mu_x &= E(x) \\
&= E(Au + b) \\
&= AE(u) + b \\
&= A\mu_u + b
\end{aligned}
$$

$$
\begin{aligned}
\Sigma_x &= E((x - E(x))(x - E(x))^T) \\
&= E((Au + b - AE(u) - b)(Au + b - AE(u) - b)^T) \\
&= E((A(u - E(u)))(A(u - E(u)))^T) \\
&= E((A(u - E(u)))((u - E(u))^T A^T)) \\
&= AE((u - E(u))(u - E(u))^T)A^T \\
&= A\Sigma_u A^T
\end{aligned}
$$

Example: Line Extraction

Wanted: Parameter Covariance Matrix

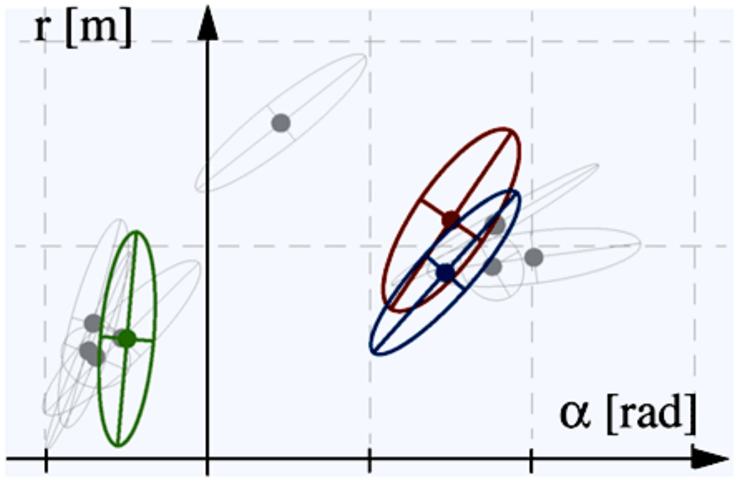
$$C_{AR} = \begin{bmatrix} \sigma_A^2 & \sigma_{AR} \\ \sigma_{AR} & \sigma_R^2 \end{bmatrix}$$

$$C_X = \begin{bmatrix} \sigma_{\rho_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{\rho_2}^2 & \dots & 0 \\ : & : & & : \\ 0 & 0 & \dots & \sigma_{\rho_n}^2 \end{bmatrix}$$

Simplified sensor model:
all $\sigma_{\theta_i}^2 = 0$, independence

$$C_{AR} = F_X C_X F_X^T$$

Result: Gaussians in the model space



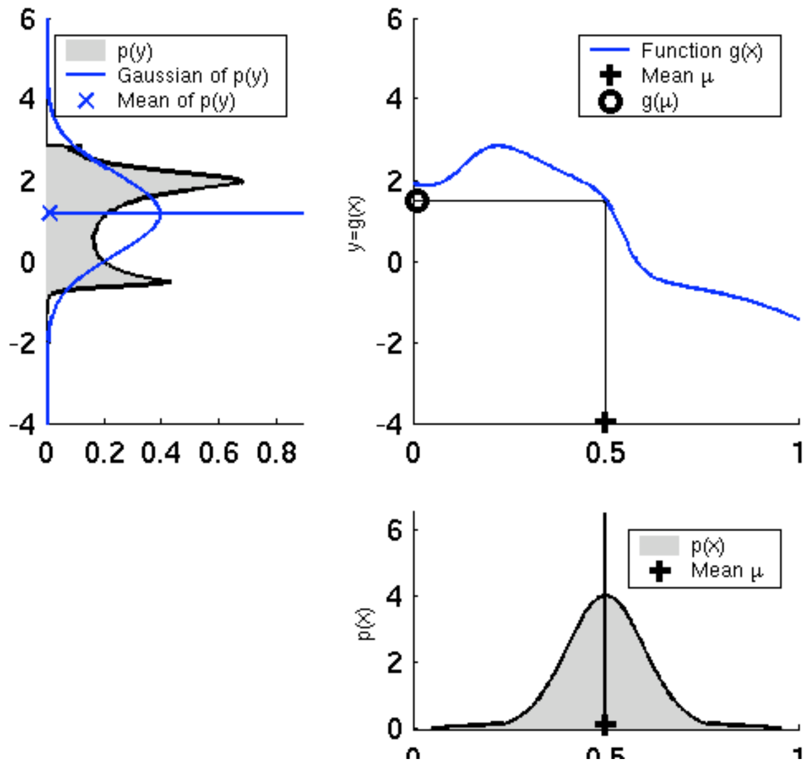
Other Error Prop. Techniques

- **Second-Order Error Propagation**

Rarely used (complex expressions)

- **Monte-Carlo**

Non-parametric representation of uncertainties



1. Sampling from $p(X)$
2. Propagation of samples
3. Histogramming
4. Normalization