CS491/691: Introduction to Aerial Robotics

Topic: Extended Kalman Filter

Dr. Kostas Alexis (CSE)

These slides relied on the lectures from C. Stachniss, J. Sturm and the book "Probabilistic Robotics" from Thurn et al.

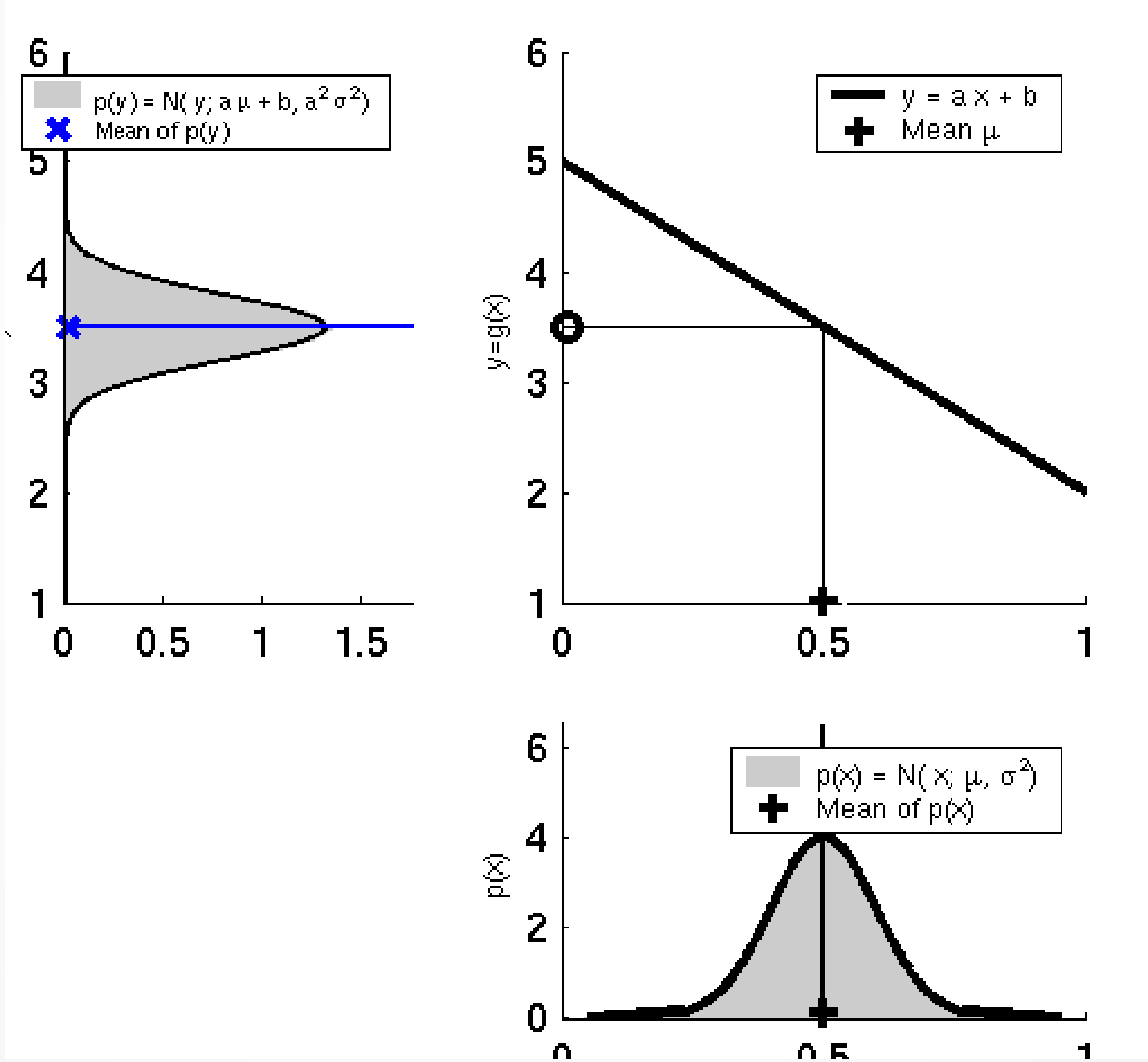
Kalman Filter Assumptions

- Gaussian distributions and noise
- Linear motion and observation model
  - **What if this is not the case?**

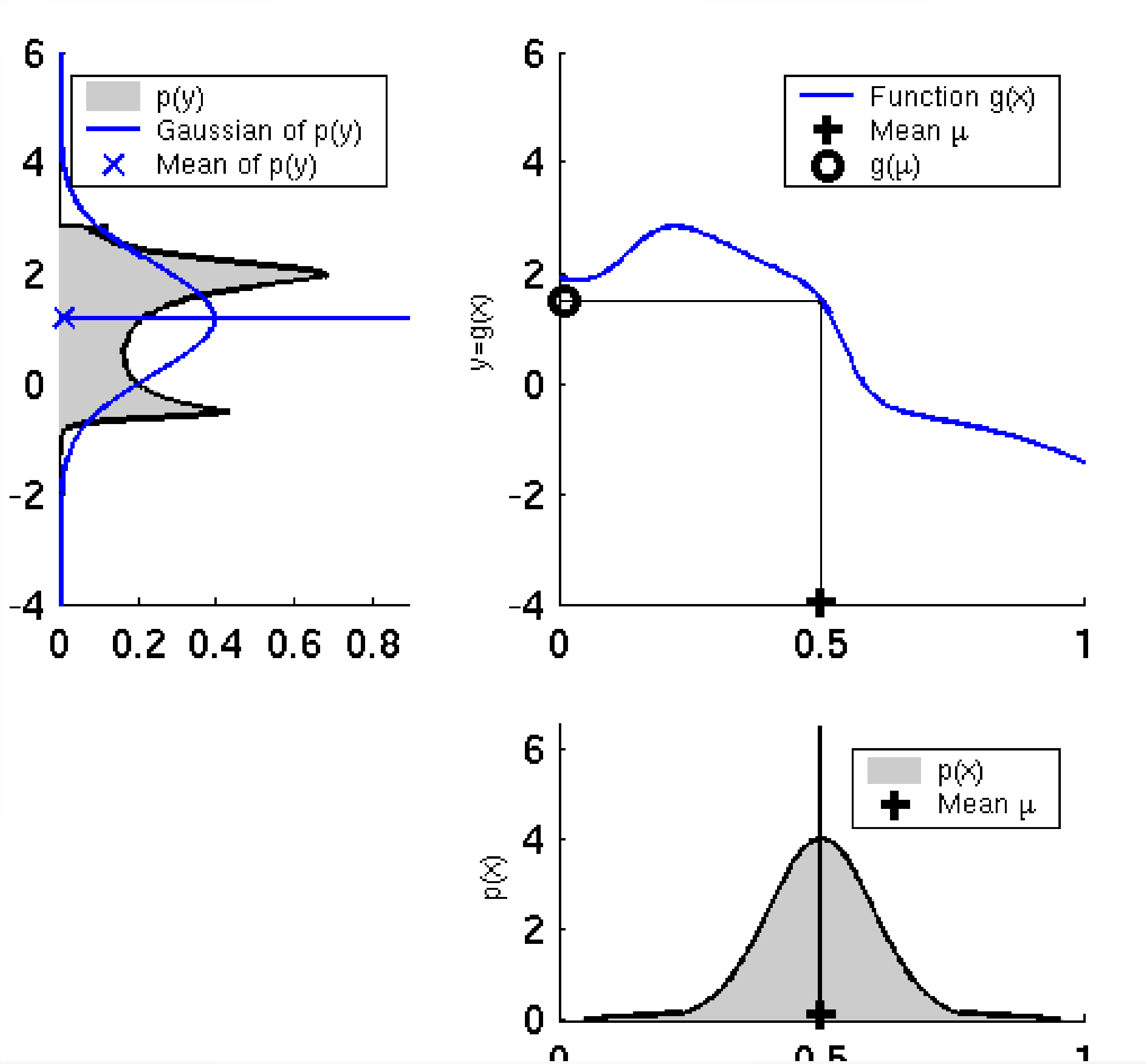
$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$$

$$z_t = C_t x_t + \delta_t$$

Linearity Assumption Revisited



Nonlinear Function



Nonlinear Dynamical Systems

- Real-life robots are mostly nonlinear systems.
- The **motion equations** are expressed as **nonlinear differential (or difference) equations**:

$$x_t = g(u_t, x_{t-1})$$

- Also leading to a **nonlinear observation function:**

$$z_t = h(x_t)$$

Taylor Expansion

- Solution: approximate via linearization of both functions
- **Motion Function:**

$$g(x_{t-1}, u_t) \approx g(\mu_{t-1}, u_t) + \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$
$$= g(\mu_{t-1}, u_t) + G_t(x_{t-1} - \mu_{t-1})$$

- **Observation Function:**

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t}(x_t - \mu_t)$$
$$= h(\bar{\mu}_t) + H_t(x_t - \mu_t)$$

Reminder: Jacobian Matrix

- It is a non-square matrix *mxn* in general
- Given a vector-valued function:

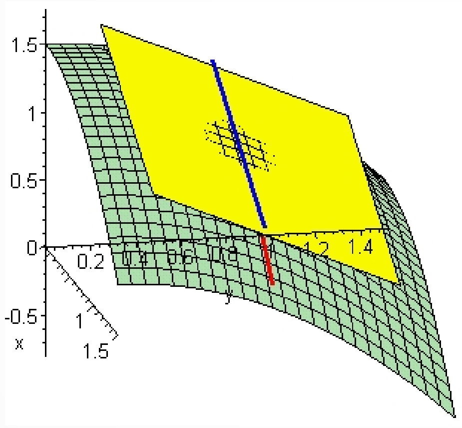
$$g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix}$$

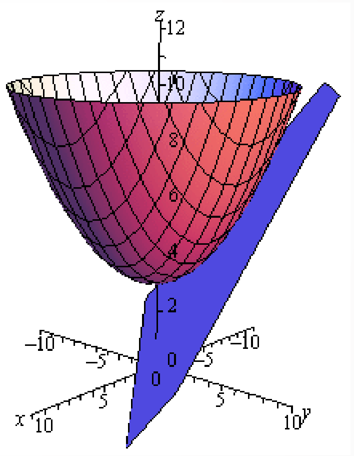
- The **Jacobian matrix** is defined as:

$$G_x = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}$$

Reminder: Jacobian Matrix

- It is the orientation of the tangent plane to the vector-valued function at a given point





Courtesy: K. Arras

- Generalizes the gradient of a scaled-valued function.

Extended Kalman Filter

- For each time step, do:
- **Apply Motion Model:**

$$\bar{\mu}_t = g(\mu_{t-1}, u_t)$$

$$\bar{\Sigma}_t = G_t \Sigma G_t^\top + Q \quad \text{with} \quad G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial x_{t-1}}$$

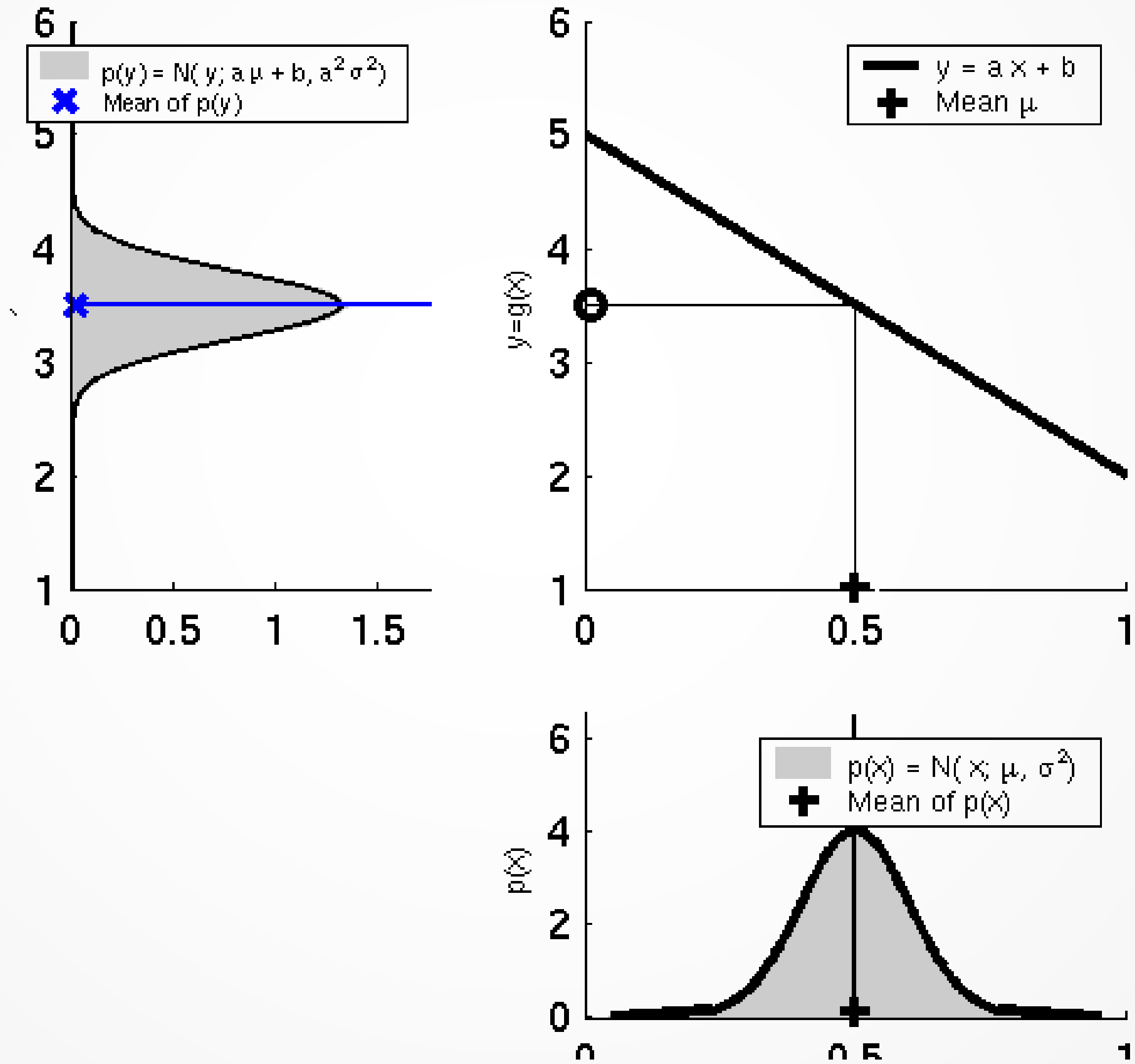
- **Apply Sensor Model:**

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$$

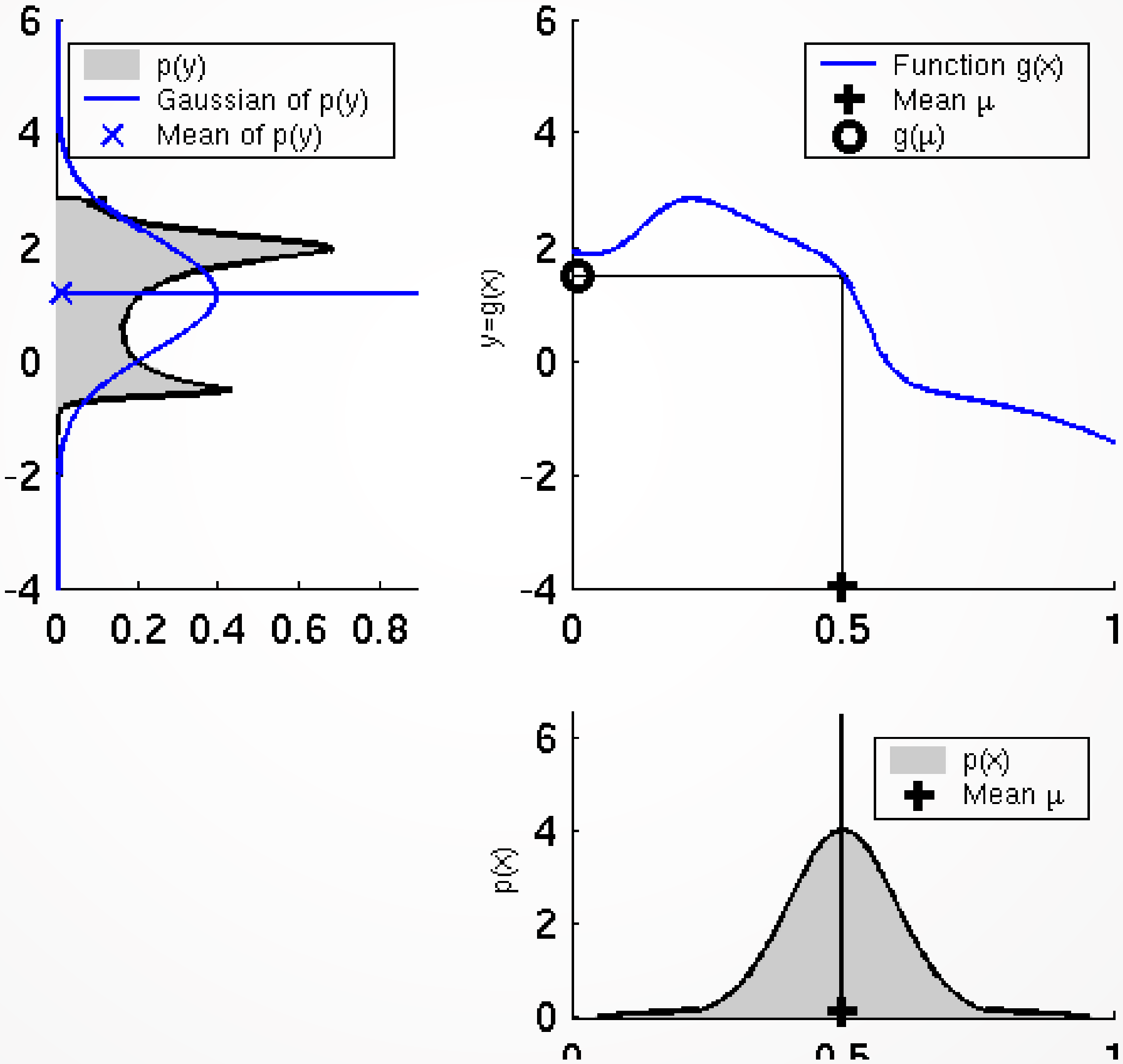
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

where $K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + R)^{-1}$ and $H_t = \dfrac{\partial h(\bar{\mu}_t)}{\partial x_t}$

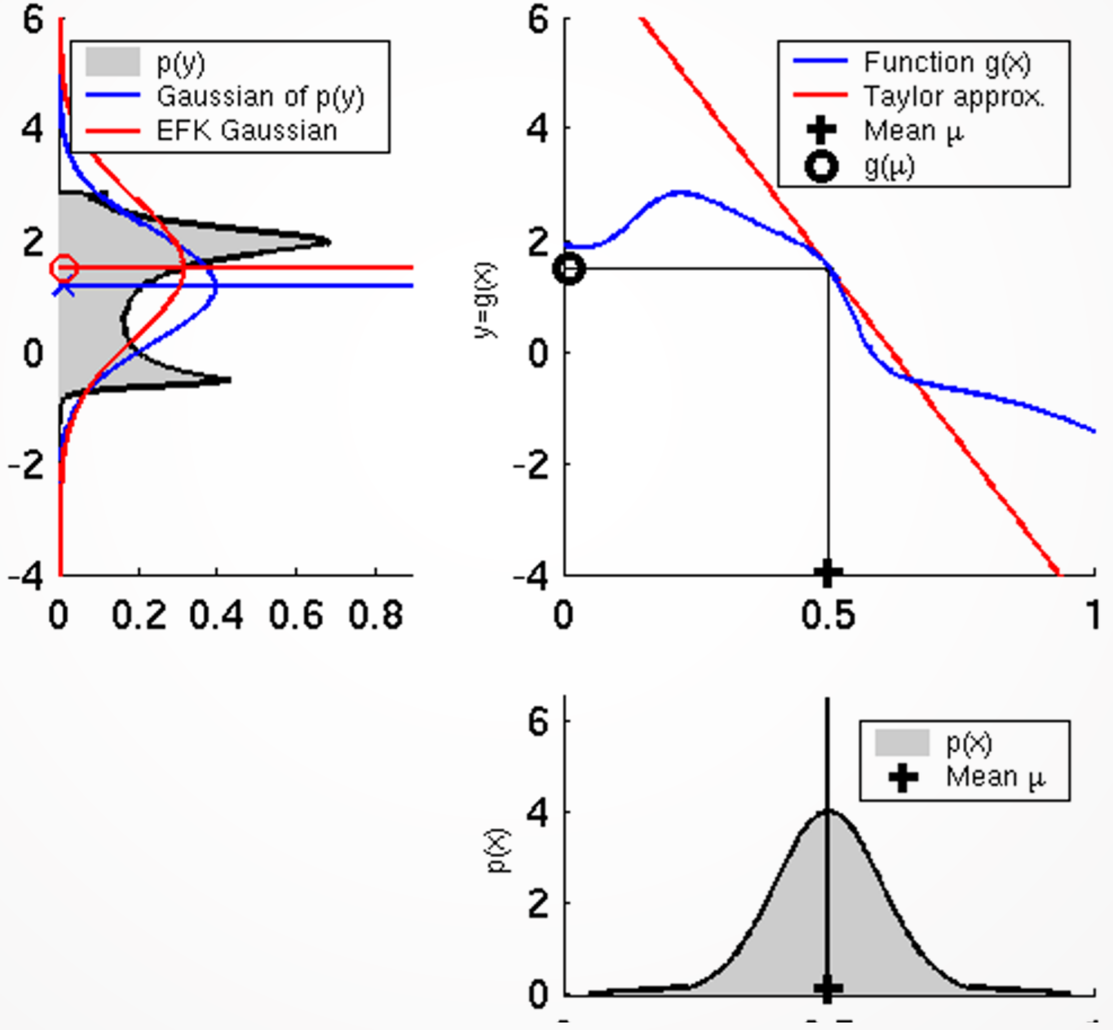
Linearity Assumption Revisited



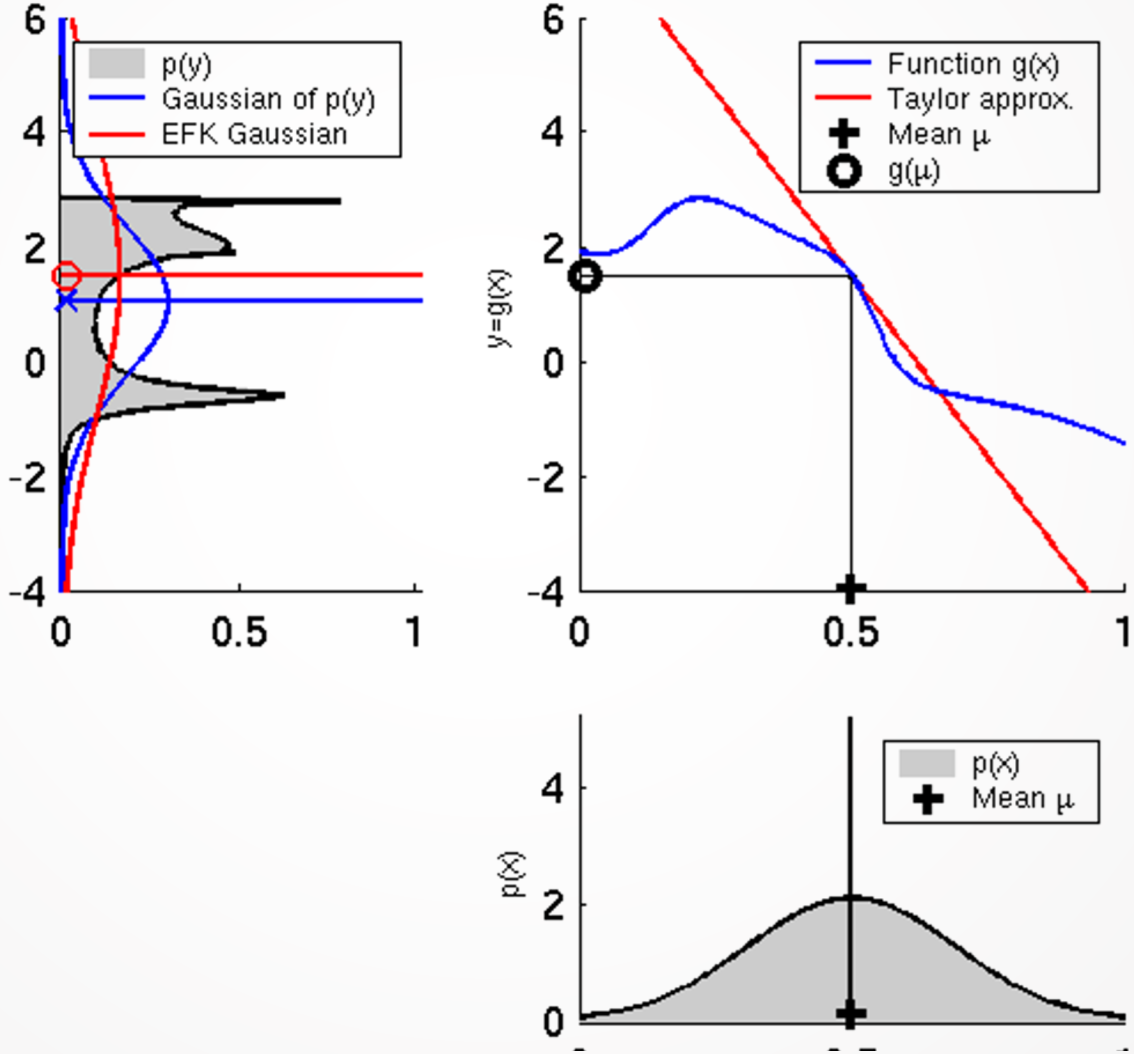
Nonlinear Function



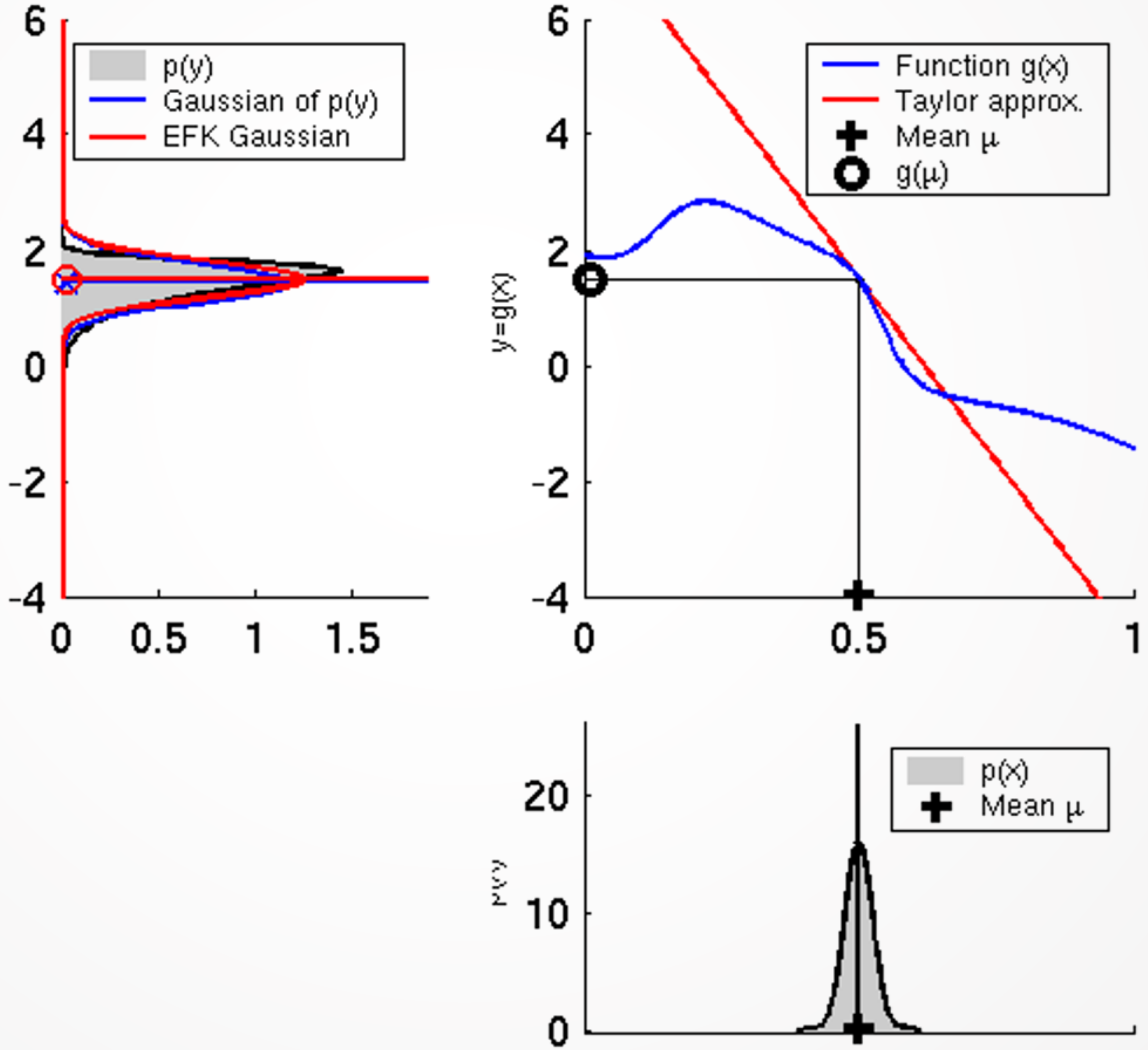
EKF Linearization (1)



EKF Linearization (2)



EKF Linearization (3)



Linearized Motion Model

- The linearized model leads to:

$$p(x_t \mid u_t, x_{t-1}) \approx \det\left(2\pi R_t\right)^{-\frac{1}{2}}$$
$$\exp\Big(-\frac{1}{2}\left(x_t - g(u_t, \mu_{t-1}) - G_t\left(x_{t-1} - \mu_{t-1}\right)\right)^T$$
$$R_t^{-1}\left(x_t - \underbrace{g(u_t, \mu_{t-1}) - G_t\left(x_{t-1} - \mu_{t-1}\right)}_{\text{linearized model}}\right)\Big)$$

- R_t describes the noise of the motion.

Linearized Observation Model

- The linearized model leads to:

$$p(z_t \mid x_t) = \det\left(2\pi Q_t\right)^{-\frac{1}{2}}$$
$$\exp\left(-\frac{1}{2}\left(z_t - h(\bar{\mu}_t) - H_t\left(x_t - \bar{\mu}_t\right)\right)^T\right.$$
$$\left. Q_t^{-1}\left(z_t - \underbrace{h(\bar{\mu}_t) - H_t\left(x_t - \bar{\mu}_t\right)}_{\text{linearized model}}\right)\right)$$

- Q_t describes the noise of the motion.

EKF Algorithm

1: $\textbf{Extended\_Kalman\_filter}(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)\textbf{:}$

2: $\bar{\mu}_t = \underline{g(u_t, \mu_{t-1})}$

3: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$

4: $K_t = \bar{\Sigma}_t \, H_t^T (H_t \, \bar{\Sigma}_t \, H_t^T + Q_t)^{-1}$

5: $\mu_t = \bar{\mu}_t + K_t(z_t - \underline{h(\bar{\mu}_t)})$

6: $\Sigma_t = (I - K_t \, H_t) \, \bar{\Sigma}_t$

7: return μ_t, Σ_t

$A_t \leftrightarrow G_t$

$C_t \leftrightarrow H_t$

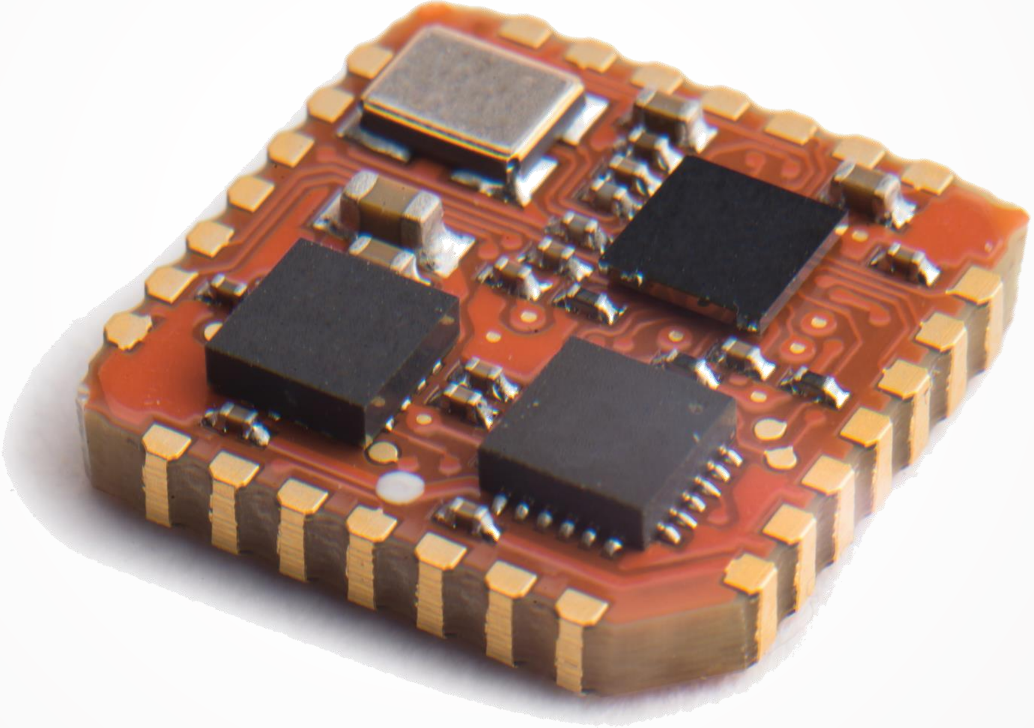
KF vs EKF

EKF Summary

- Extension of the Kalman Filter.
- One way to deal with nonlinearities.
- Performs local linearizations.
- Works well in practice for moderate nonlinearities.
- Large uncertainty leads to increased approximation error.

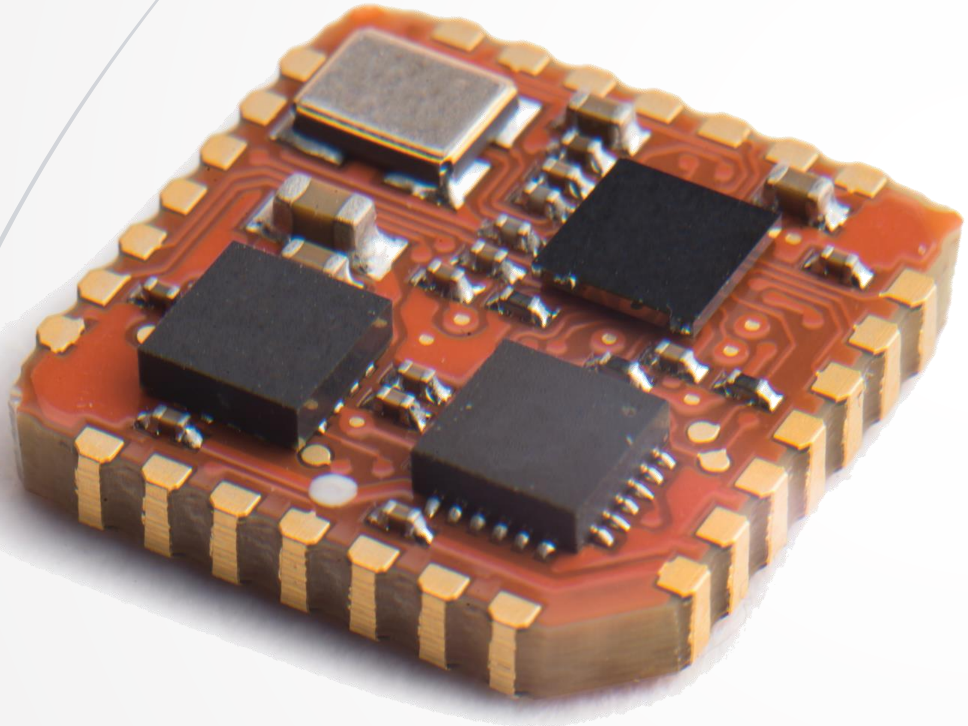
EKF Discussion

IMU



EKF Discussion

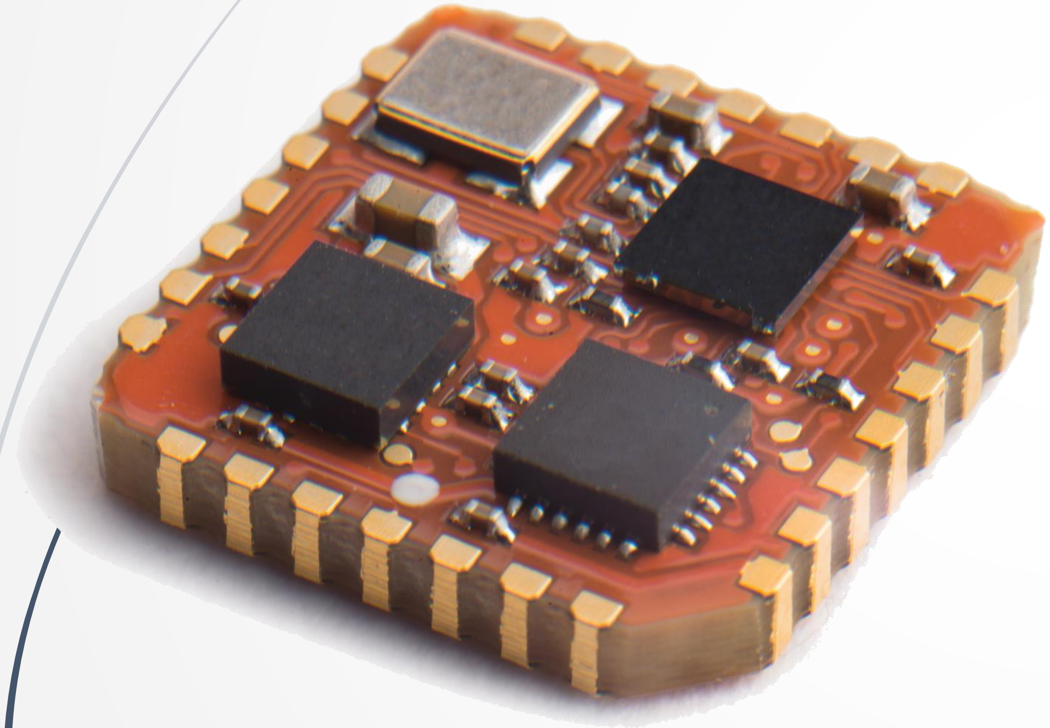
IMU + Compass



GPS

EKF Discussion

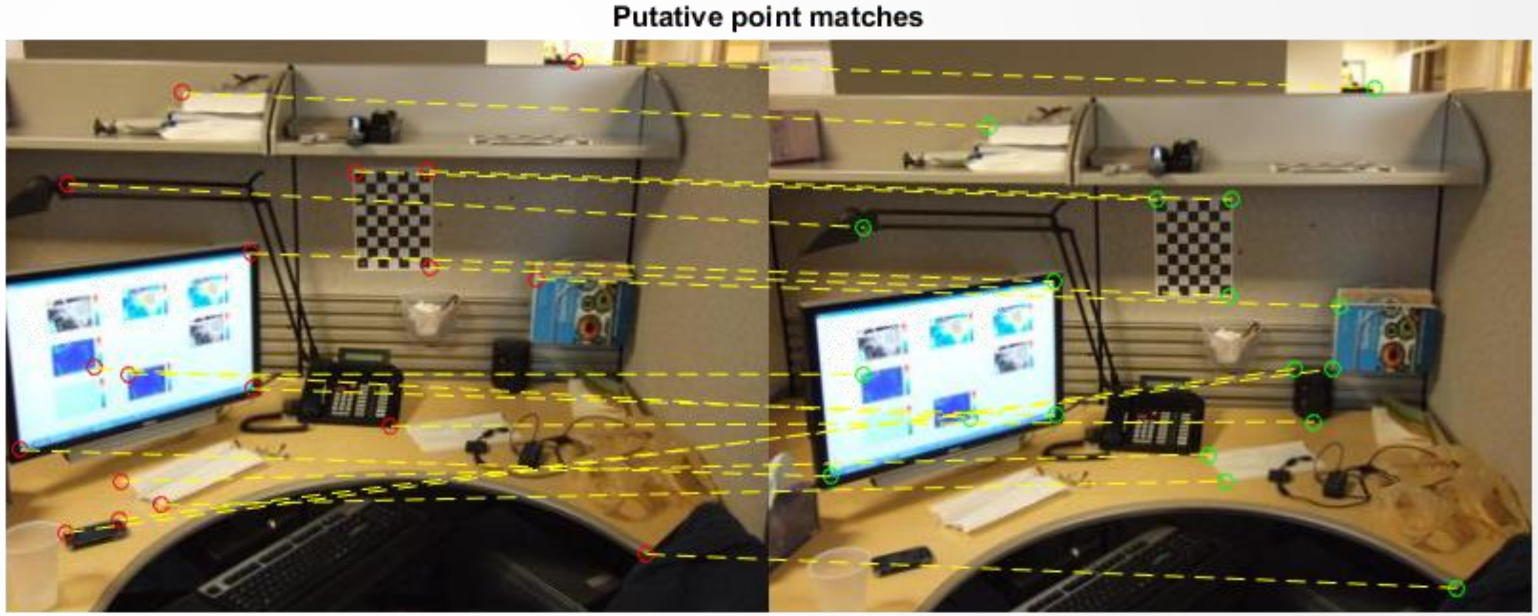
IMU + Compass



Camera

Code Examples and Tasks

- https://www.mathworks.com/help/vision/ref/estimatefundamentalmatrix.html



How does this apply to my project?

- State estimation is the way to use robot sensors to infer the robot state. You will use it for estimating your robot pose or its map, to track and object and be able to follow it etc.

Find out more

- Thrun et al.: “Probabilistic Robotics”, Chapter 3
- Schön and Lindsten: “Manipulating the Multivariate Gaussian Density”
- “A New Extension of the Kalman Filter to Nonlinear Systems” by Julier and Uhlmann, 1995
- http://www.cs.unc.edu/~welch/kalman/
- http://home.wlu.edu/~levys/kalman_tutorial/
- https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python
- http://www.kostasalexis.com/literature-and-links.html

Thank you!

Please ask your question!