Autonomous Robotics

Extended Kalman Filters

Josiah Hanna
University of Wisconsin — Madison

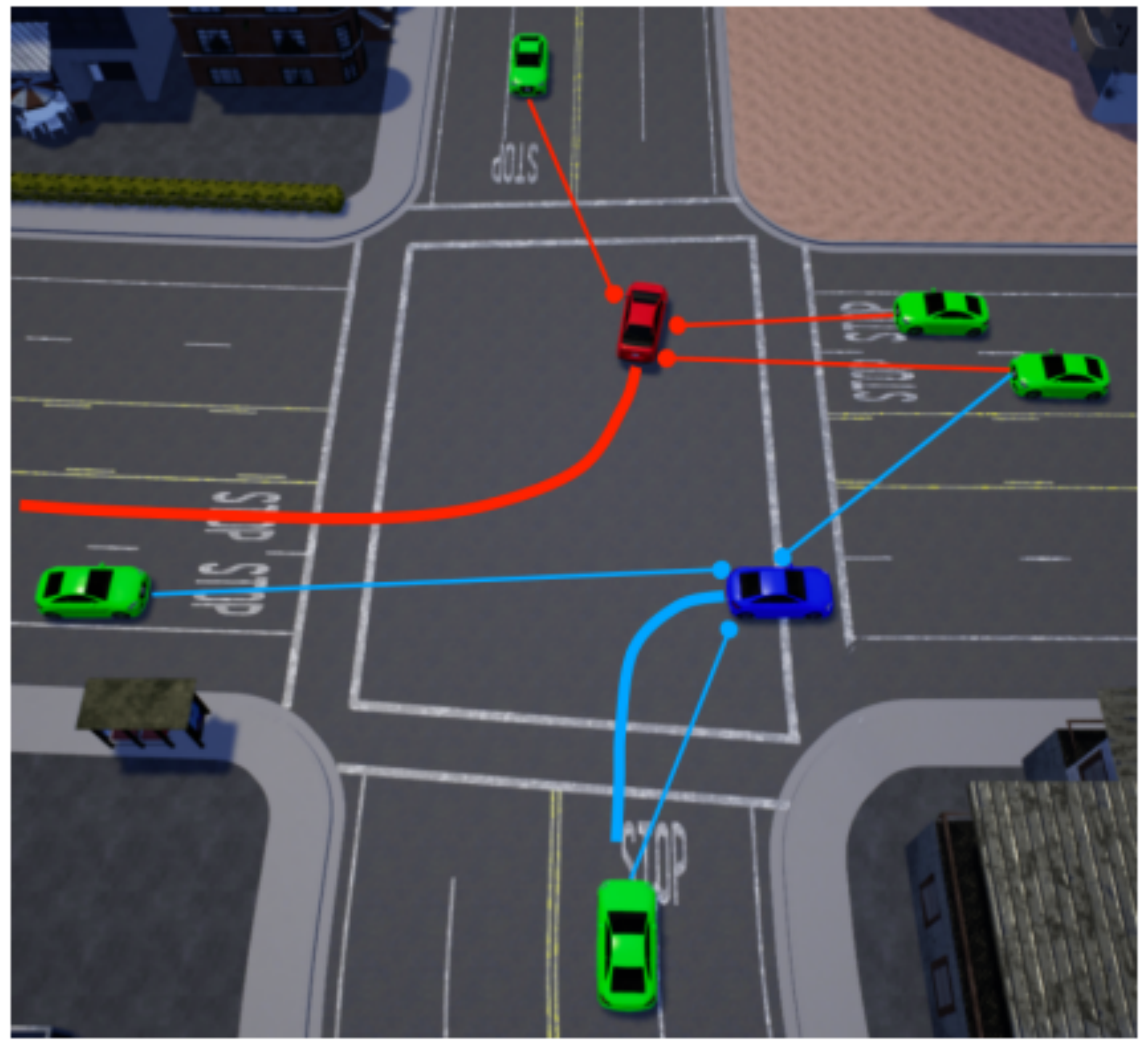
Learning Outcomes

After today's lecture, you will:

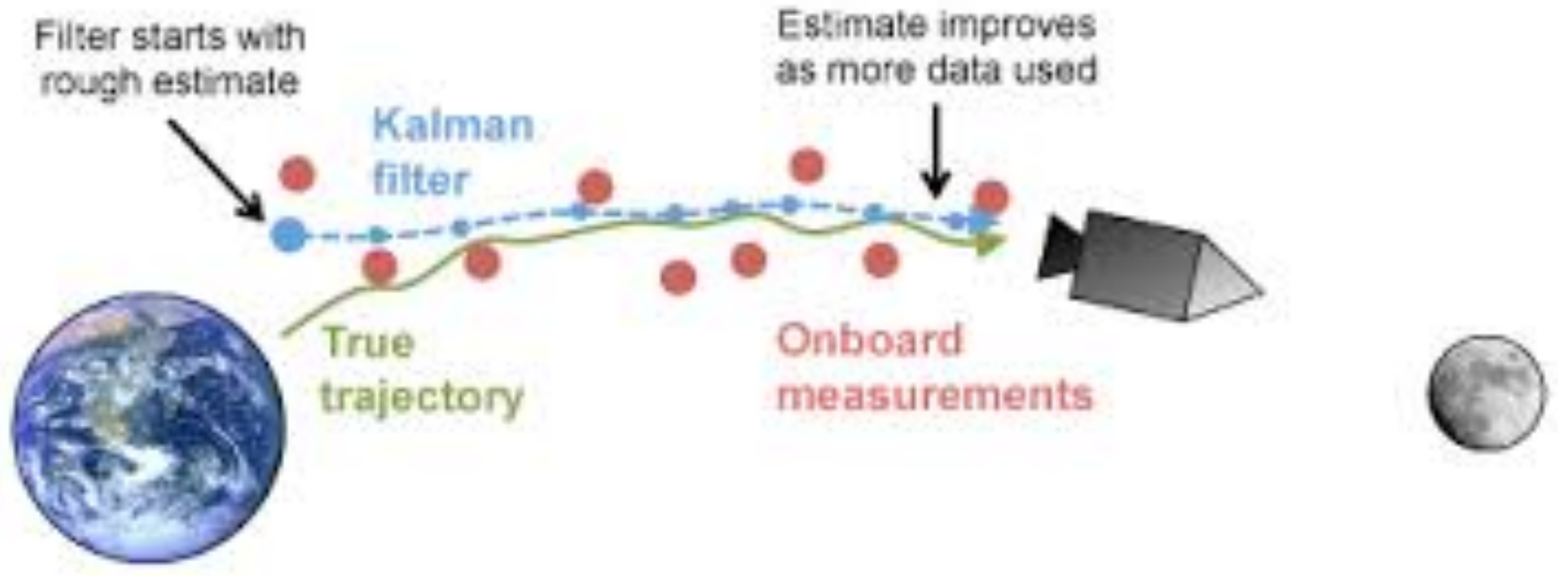
- Be able to specify the key assumptions underlying the extended Kalman filters.
- Understand the extended Kalman filter as an approximation of the Kalman filter.
- Understand the strengths and limitations of extended Kalman filters.

Kalman Filter Applications

Robot Localization



Autonomous driving [e.g., 1]



Object Tracking

[1] https://arxiv.org/pdf/2004.05965

Linear Gaussian Systems

We make the following assumptions on the robot's environment:

- States, controls, and observations are vectors: $x \in \mathbf{R}^d$ and $u \in \mathbf{R}^k$ and $z \in \mathbf{R}^m$.

- State transition and observation function are linear Gaussians:

  - $x_t = Ax_{t-1} + Bu_t + w_t$ where $w_t \sim \mathcal{N}(0,Q)$, $A \in \mathbf{R}^{d\times d}$, $B \in \mathbf{R}^{d\times k}$ and $Q \in \mathbf{R}^{d\times d}$.
    $$\implies p(x_t \mid x_{t-1}, u_t) = \mathcal{N}(x; Ax_{t-1} + Bu_t, Q)$$

  - $z_t = Hx_t + v_t$ where $v_t \sim \mathcal{N}(0,R)$, $H \in \mathbf{R}^{m\times d}$, and $R \in \mathbf{R}^{m\times m}$.
    $$\implies g(z_t \mid x_t) = \mathcal{N}(z; Hx_t, R)$$

Kalman Filter

- The Kalman filter is a Bayes filter that represents $\texttt{bel}(x_t)$ with a Gaussian distribution, $\mathcal{N}(\mu_t, \Sigma_t)$.
- The initial belief is Gaussian: $\texttt{bel}(x_0) = \mathcal{N}(x_0; \mu_0, \Sigma_0)$.
- Under our assumptions, the posterior remains a Gaussian distribution using the updates from the Bayes filter:

$$p(x_t \,|\, z_{1:t}, u_{1:t}) = \mathcal{N}(x_t; \mu_t, \Sigma_t)$$

- Intuition for correctness: plug Gaussian beliefs and linear Gaussian system state transitions and observations into Bayes filter updates.

The Kalman Filter as a Bayes Filter

- Initialize belief:

$$\texttt{bel}(x_0) = \mathcal{N}(x_0, \mu_0, \Sigma_0)$$

- Prediction:

$$\overline{\texttt{bel}}(x_t) = \int p(x_t \mid x_{t-1}, u_t)\texttt{bel}(x_{t-1})dx_{t-1}$$

$$\bar{\mu}_t = A\mu_{t-1} + Bu_t$$

$$\bar{\Sigma}_t = A^T \Sigma A + R$$

- Correction:

$$\texttt{bel}(x_t) = \eta g(z_t \mid x_t)\overline{\texttt{bel}}(x_t)$$

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

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

The Kalman Gain

$$K_t = \bar{\Sigma}_t H^\top (H \bar{\Sigma}_t H^\top + R)^{-1}$$

- K_t is called the Kalman gain at time-step t.

- Use univariate case with $H = 1$ to build intuition:

$$K_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + R}$$

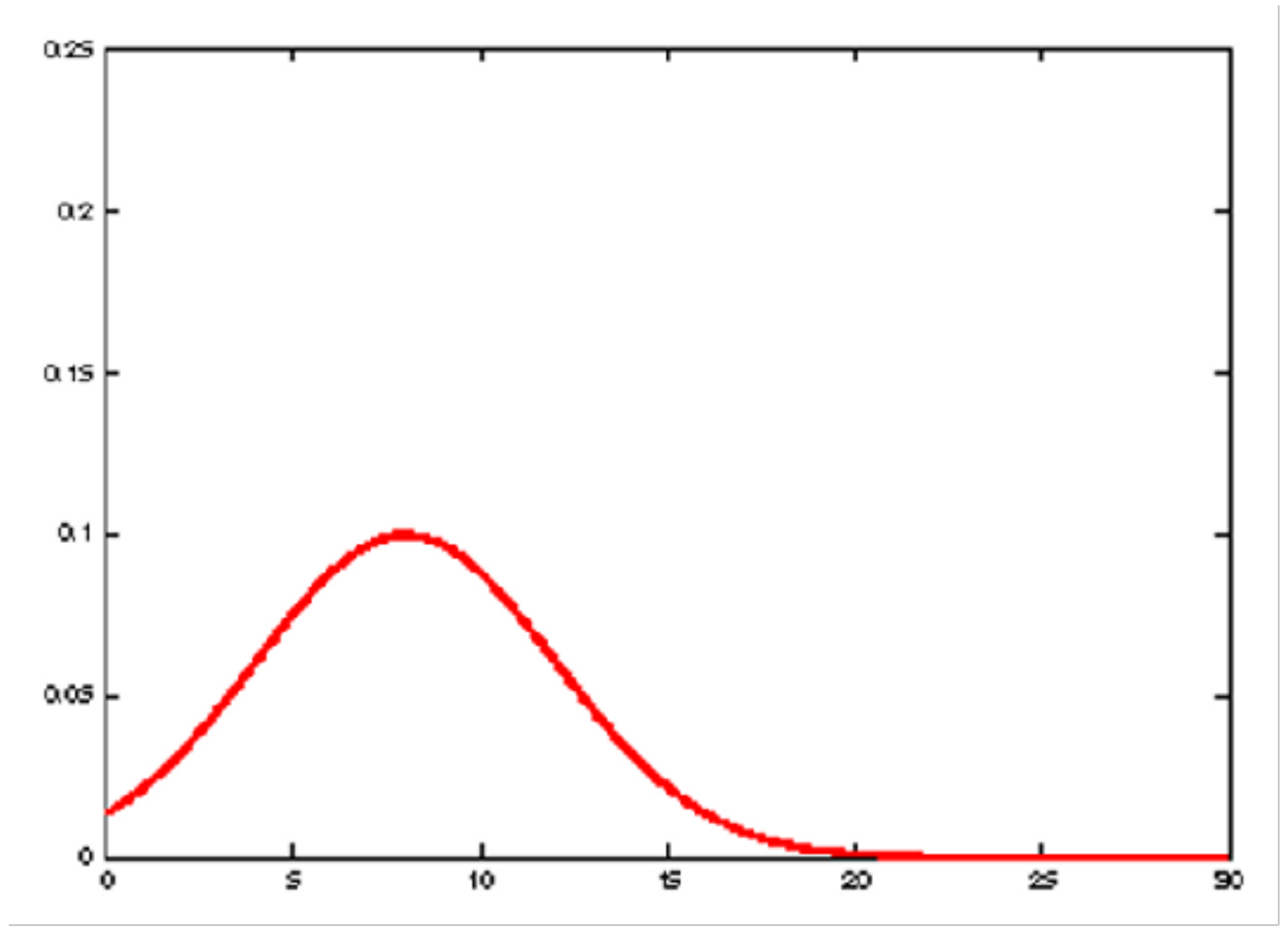
Uncertainty from prediction step

Total uncertainty

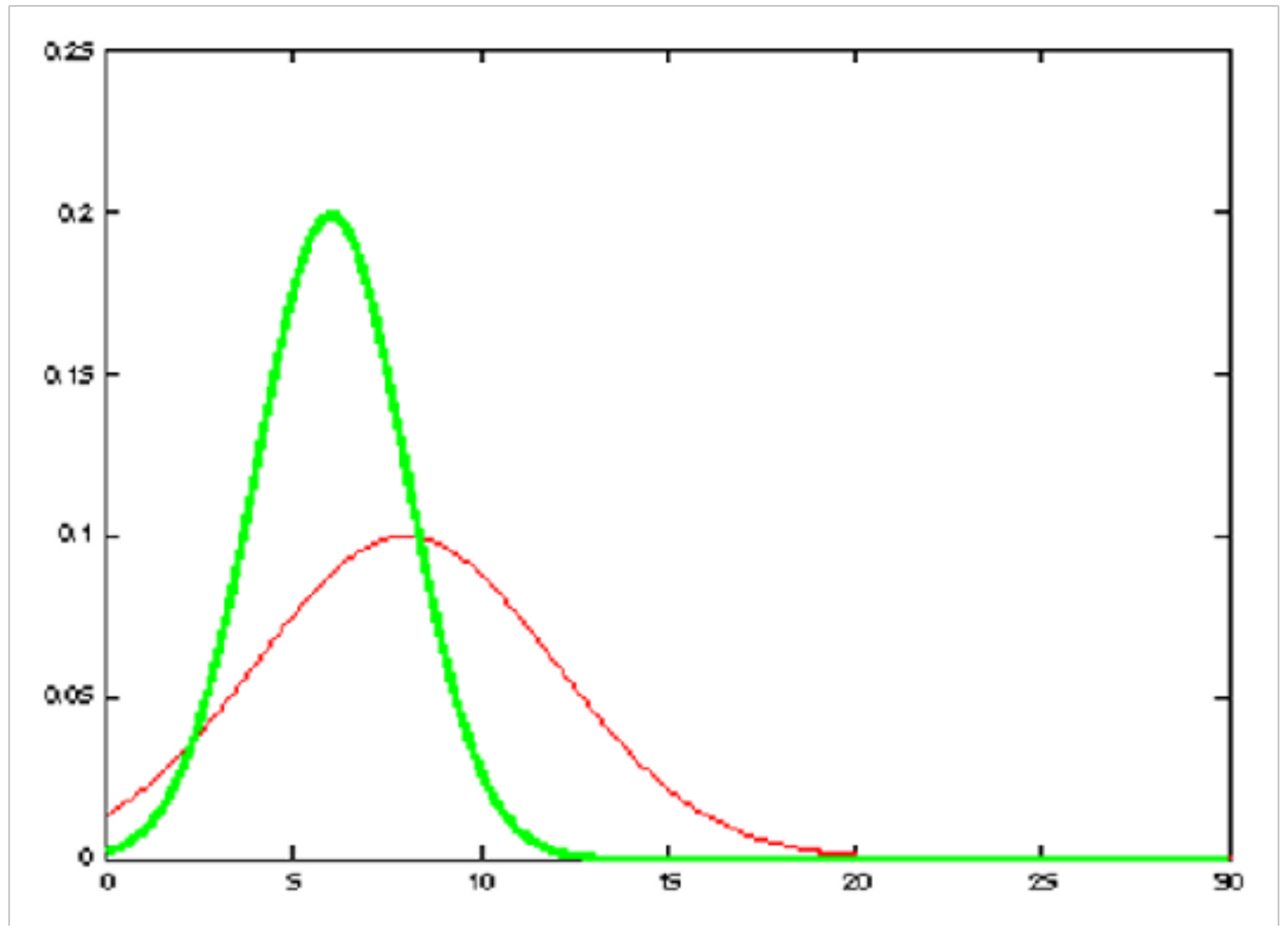
- The Kalman gain tells you how much to trust the prediction vs the observation.

- Small gain implies the measurement is less reliable and the belief is updated less from the prediction belief.

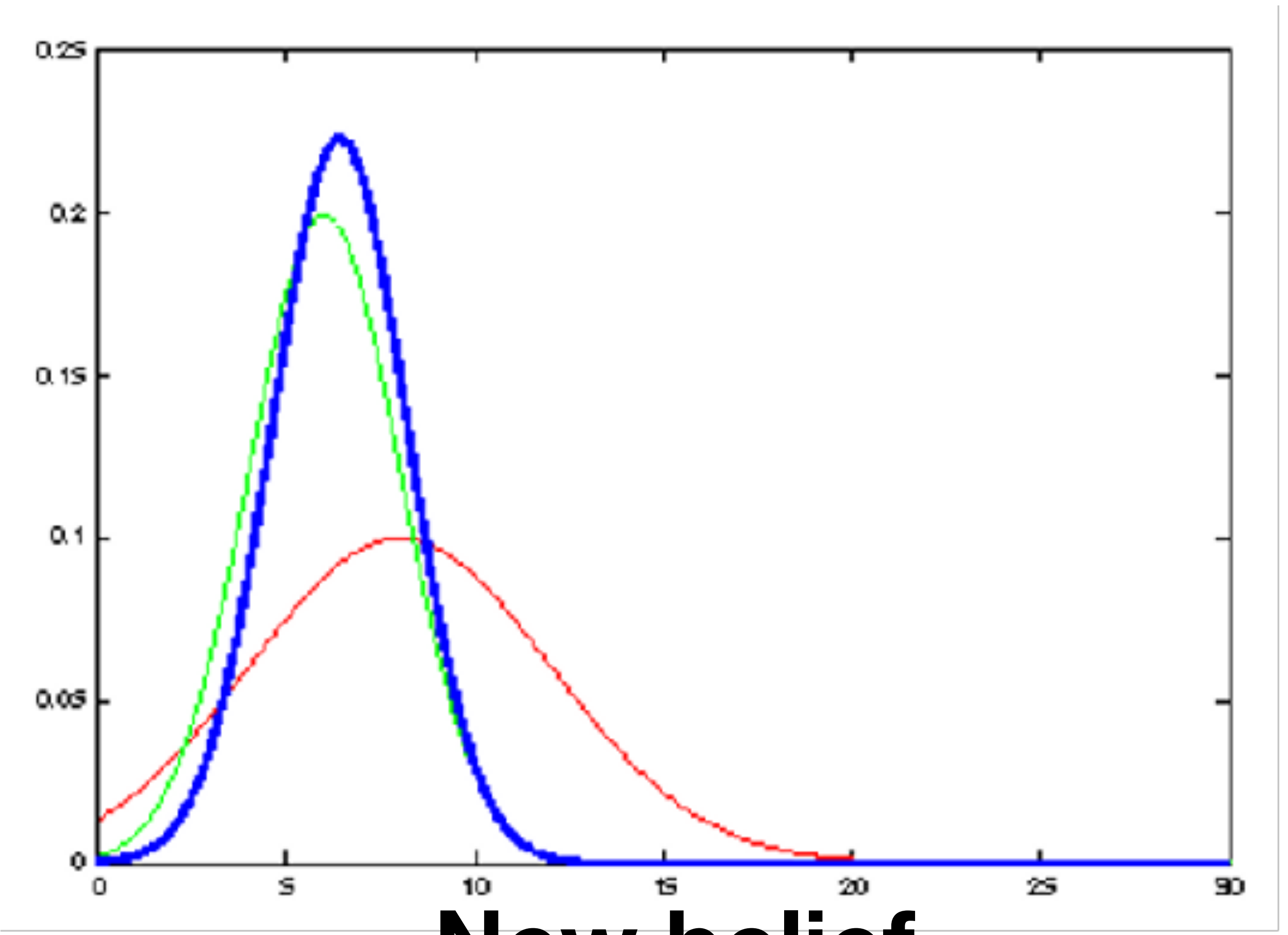
Illustration of Kalman Filter Updates



Belief after motion



Observation Probability



New belief

Advantages / Disadvantages

- Kalman filters:
  - Can be used for continuous state spaces.
  - Are optimal filters if our assumptions hold.
  - Are very efficient; polynomial in state and observation dimensionality.
- But…
  - Randomness may not be Gaussian.
  - Most robotics systems are nonlinear.

Non-linear Gaussian Systems

Let's change our assumptions to allow non-linearity:

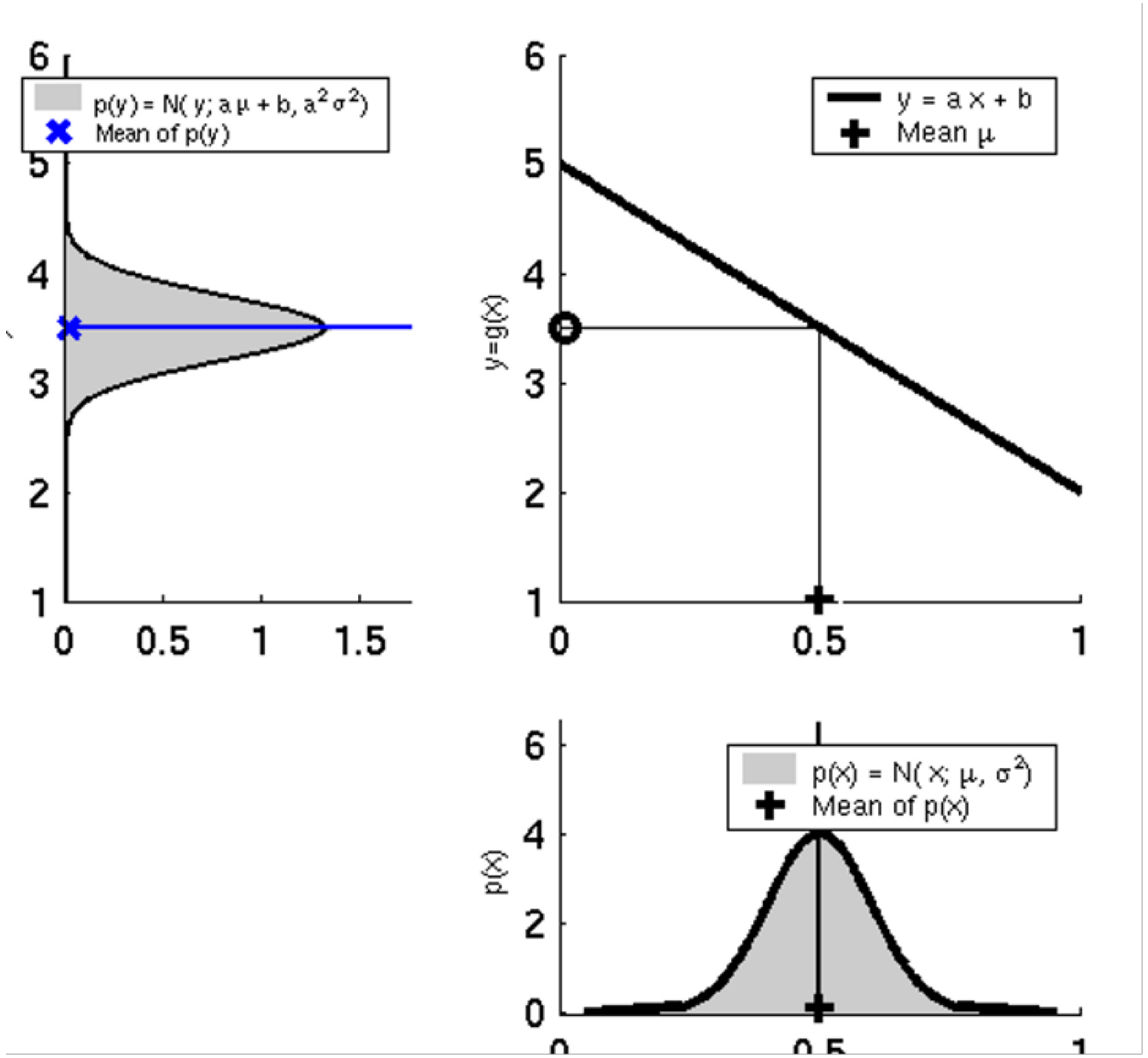
- States, controls, and observations are vectors: $x \in \mathbf{R}^d$ and $u \in \mathbf{R}^k$ and $z \in \mathbf{R}^m$.
- State transition and observation function are non-linear Gaussians:
  - $x_t = f(x_{t-1}, u_t) + w_t$ where $w_t \sim \mathcal{N}(0,Q)$, $Q \in \mathbf{R}^{d\times d}$, and f is a non-linear function.

    $$\implies p(x_t \mid x_{t-1}, u_t) = \mathcal{N}(x; g(x_{t-1}, u_t), Q)$$

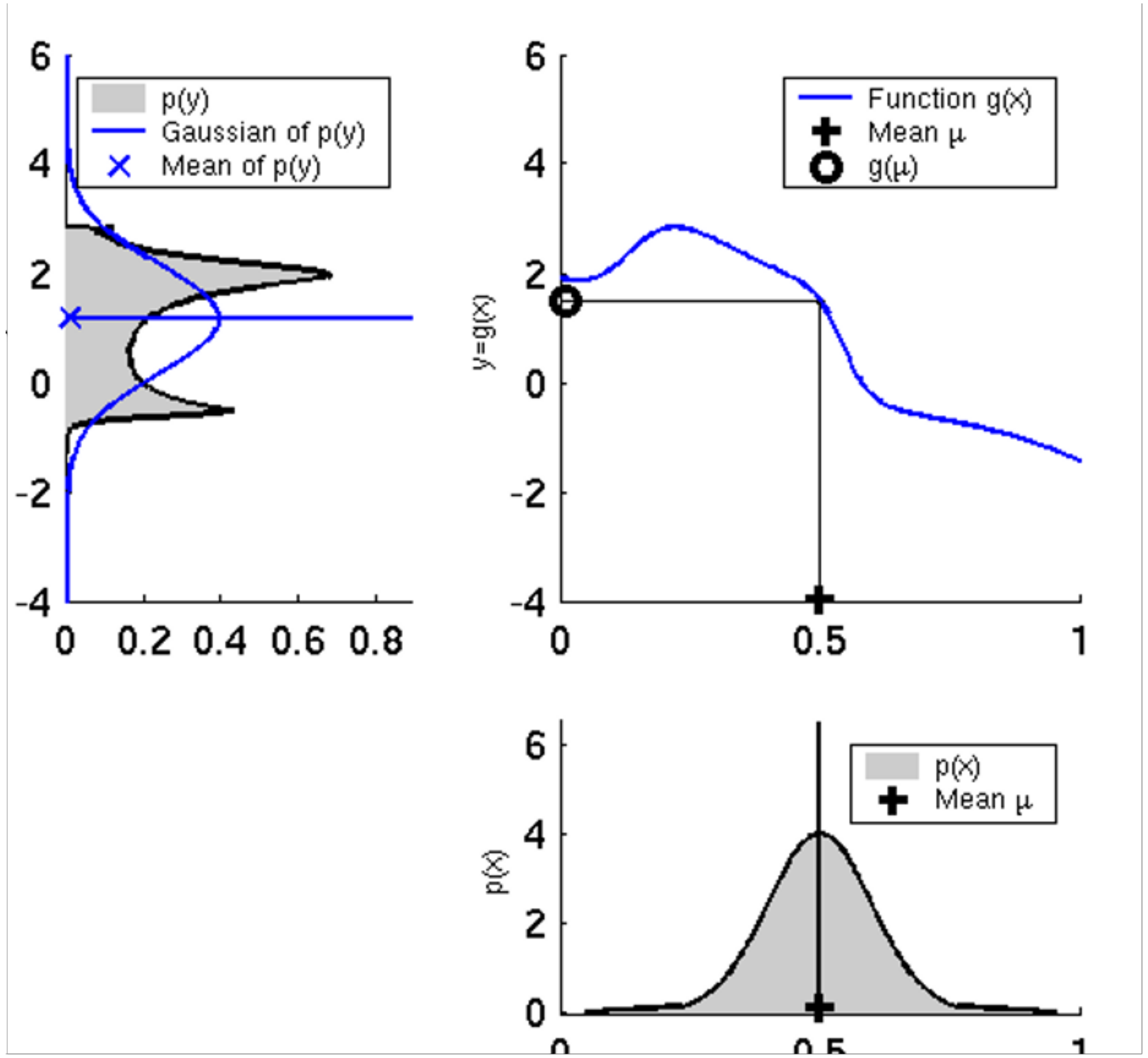
  - $z_t = h(x_t) + v_t$ where $v_t \sim \mathcal{N}(0,R)$, $R \in \mathbf{R}^{m\times m}$, and h is a non-linear function.

    $$\implies g(z_t \mid x_t) = \mathcal{N}(z; h(x_t), R)$$

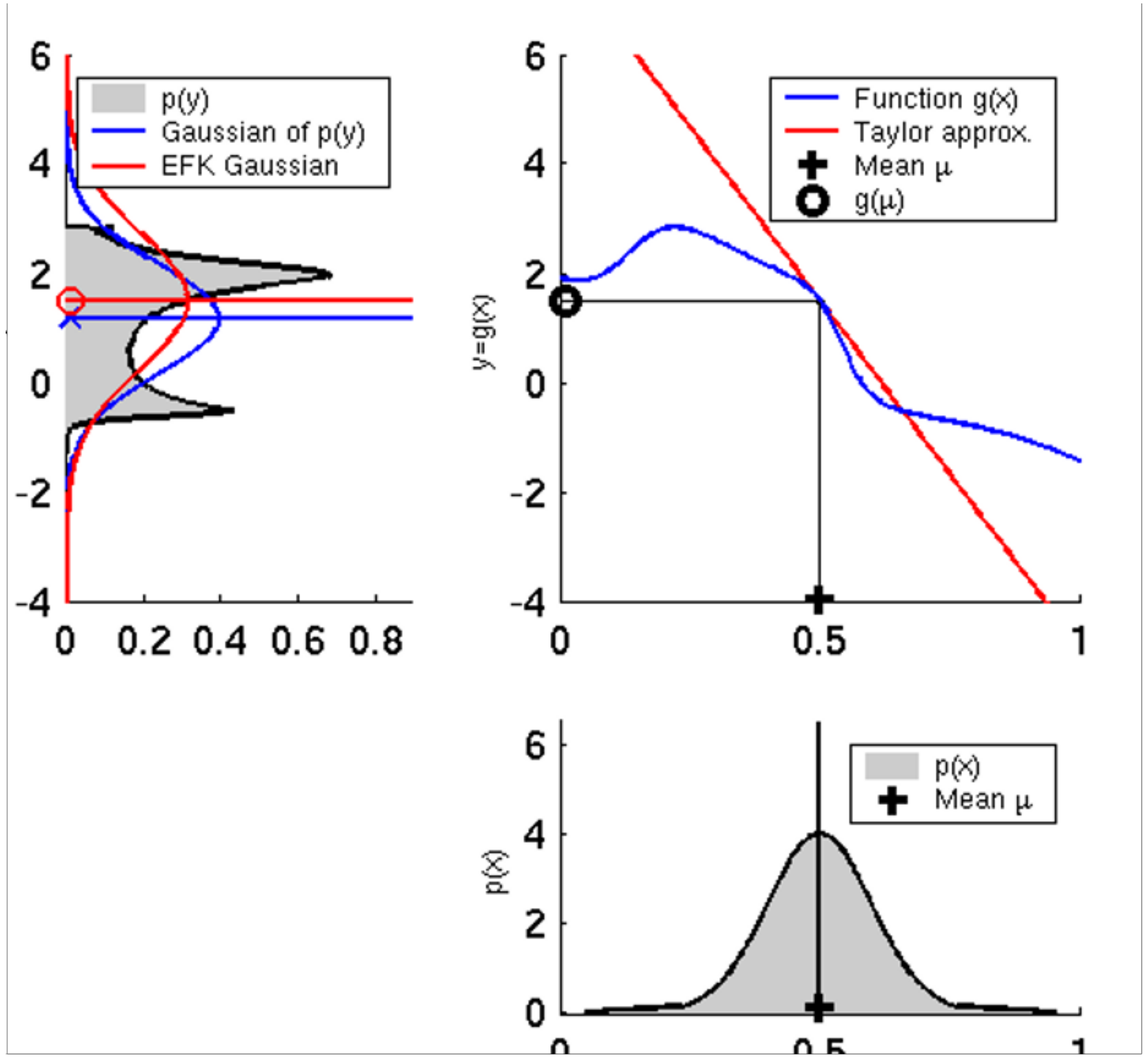
Why do we need linearity?



Why do we need linearity?

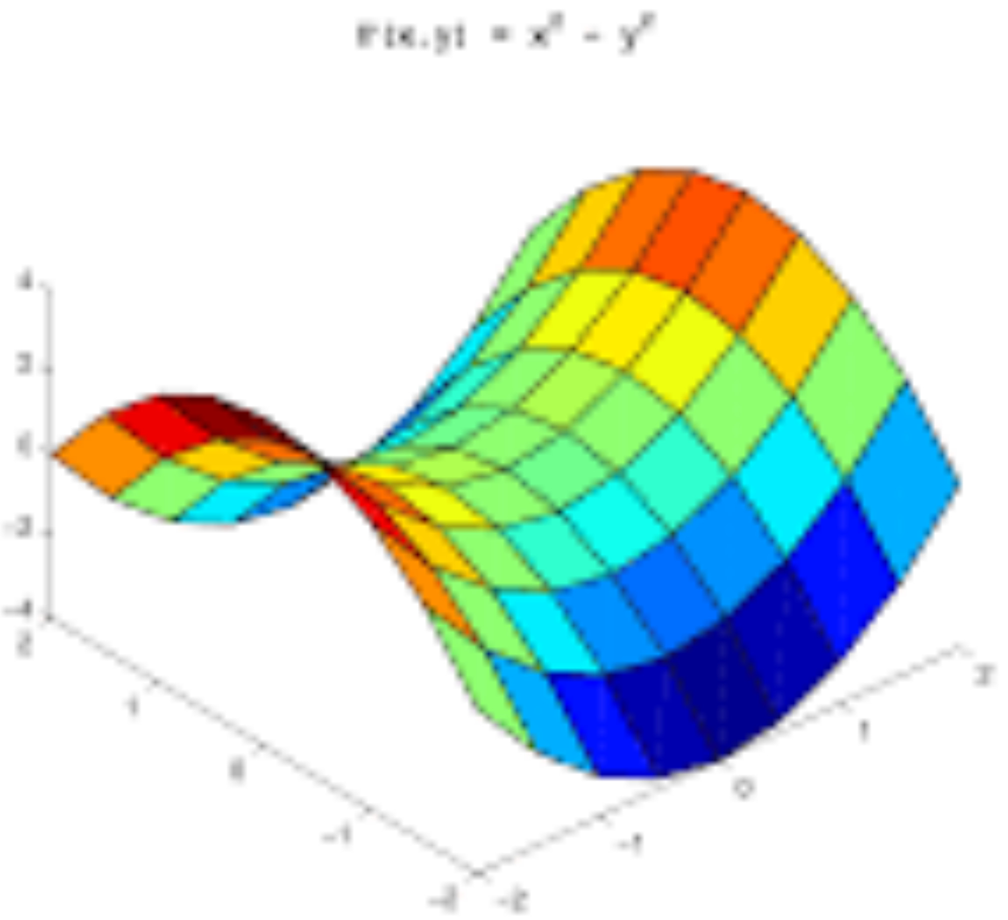


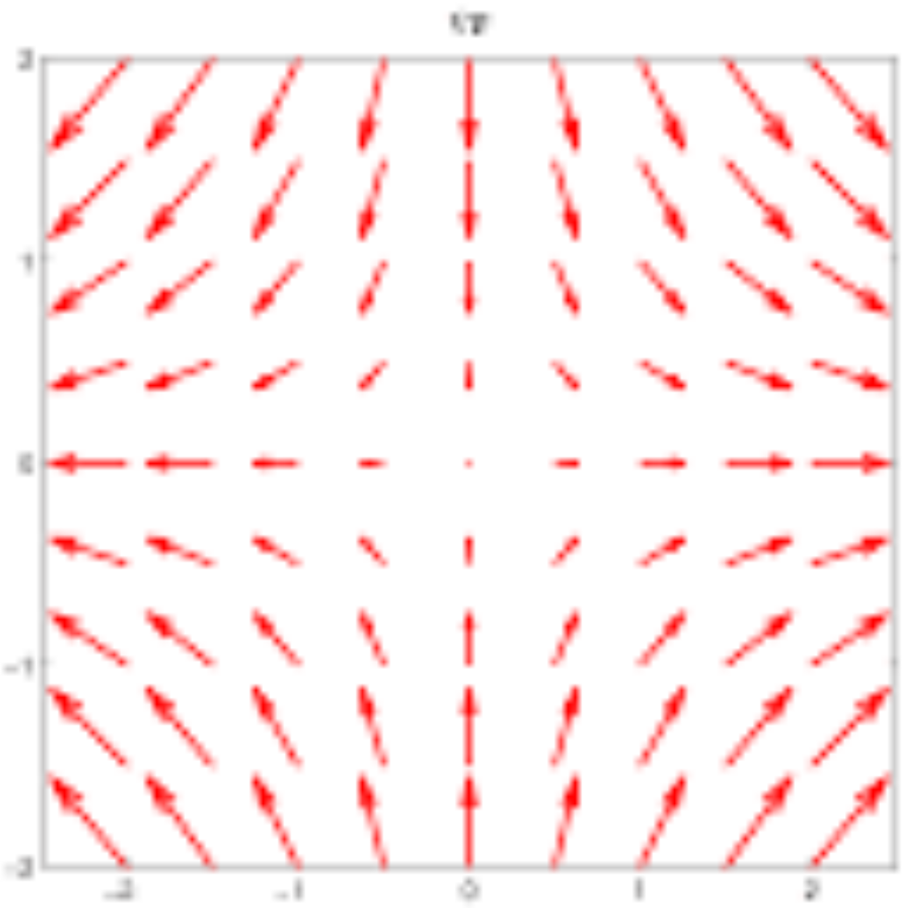
Why do we need linearity?



Calculus Review: Partial Derivatives

- Given a function, $f(x_1, \ldots, x_n)$.
- The partial derivative $\frac{\partial f}{\partial x_i}$ captures the rate of change of f as one of the x_i increases.
- The gradient is the vector of partial derivatives: $\nabla_x f = [\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}]$





Calculus Review: Jacobian Matrix

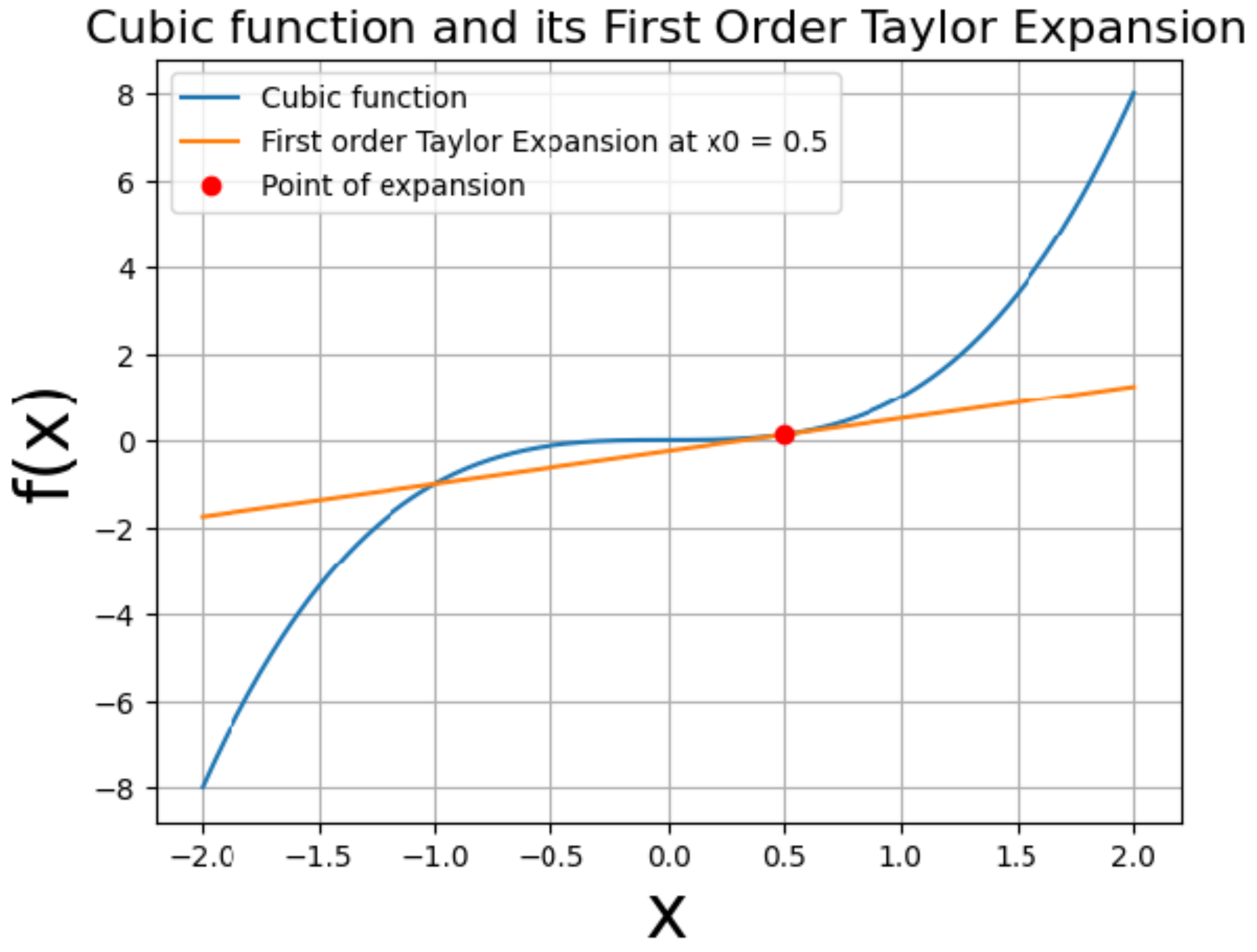
- Given a function, $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$.
  - Equivalently, $x = (x_1, ..., x_n)$ and $f(x) = (f_1(x), ..., f_m(x))$.
- The Jacobian, J, is the matrix of partial derivatives of f.
  - Entry (i, j) captures how fast $f_i(x)$ is changing as x_j increases.

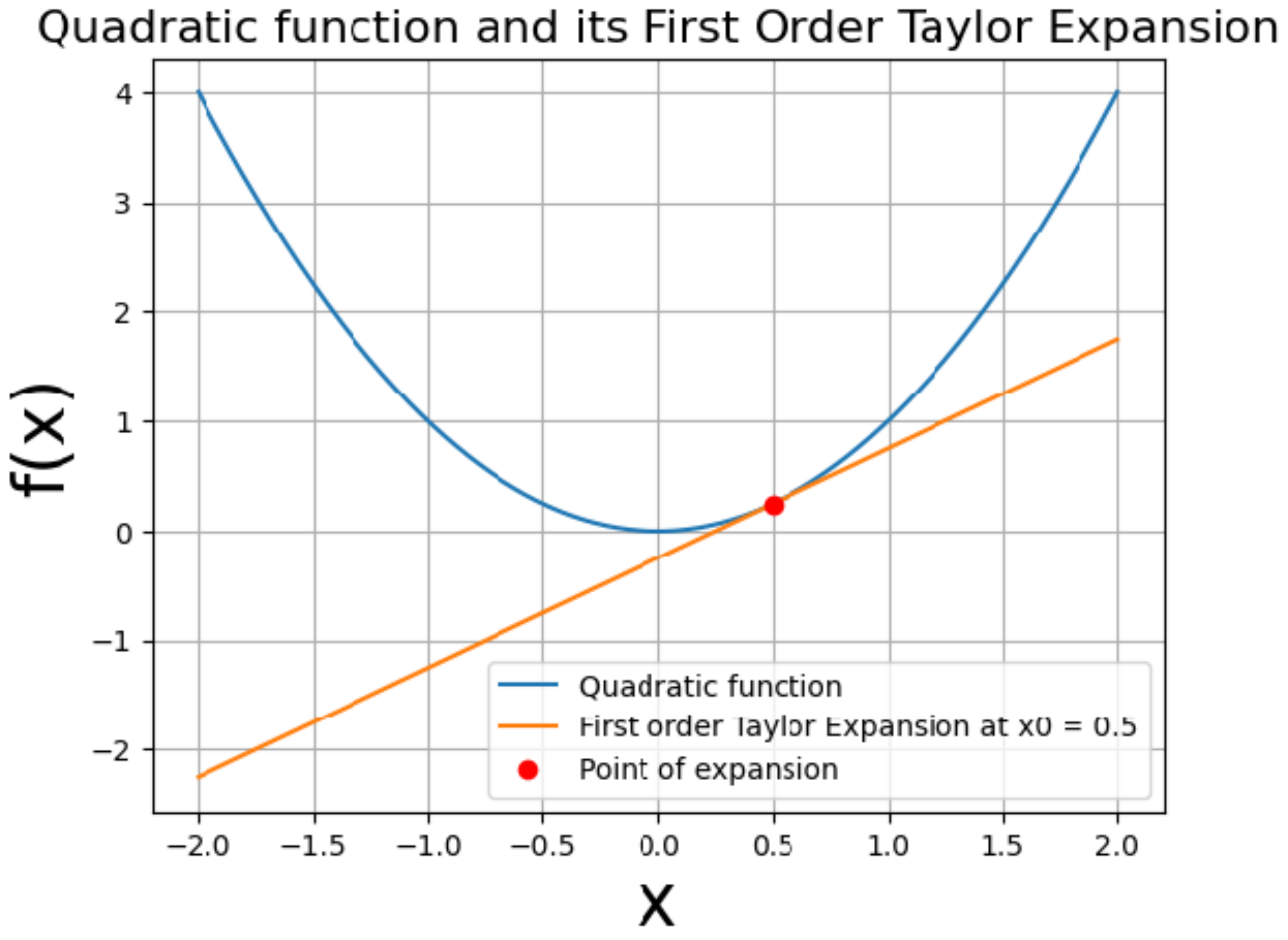
$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

https://discourse.julialang.org/t/visualizing-jacobians-gradients-hows-this/36799

Calculus Review: Taylor Series Expansion

- Goal is to approximate a given function f (possibly non-linear) with a linear function.





f has one input and one output.

$$f(x) \approx f(x_0) + (x - x_0)\frac{\partial f}{\partial x}\bigg|_{x=x_0}$$

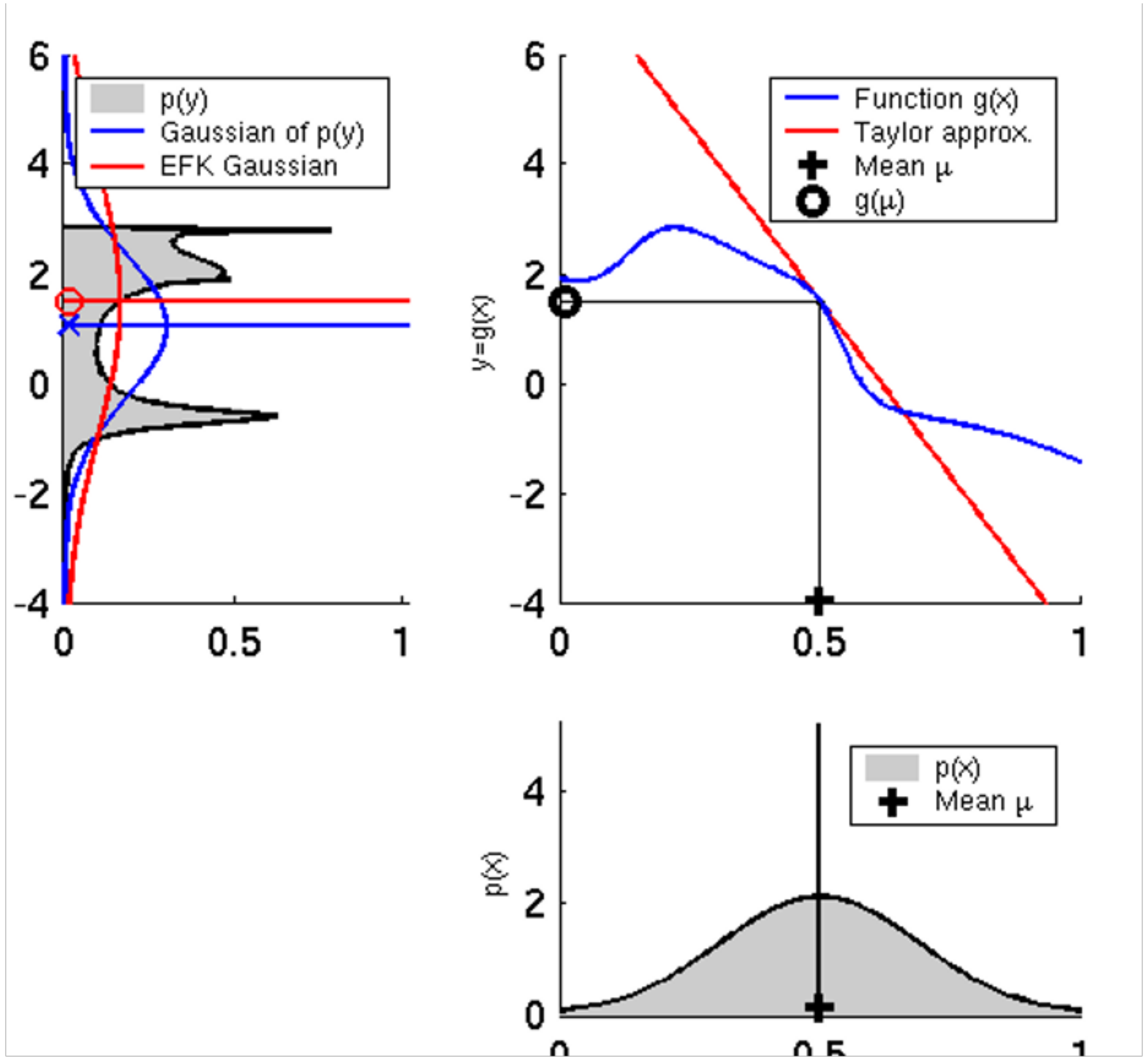
f has multiple inputs and outputs.

$$f(x) \approx f(x_0) + J^\top (x - x_0)$$

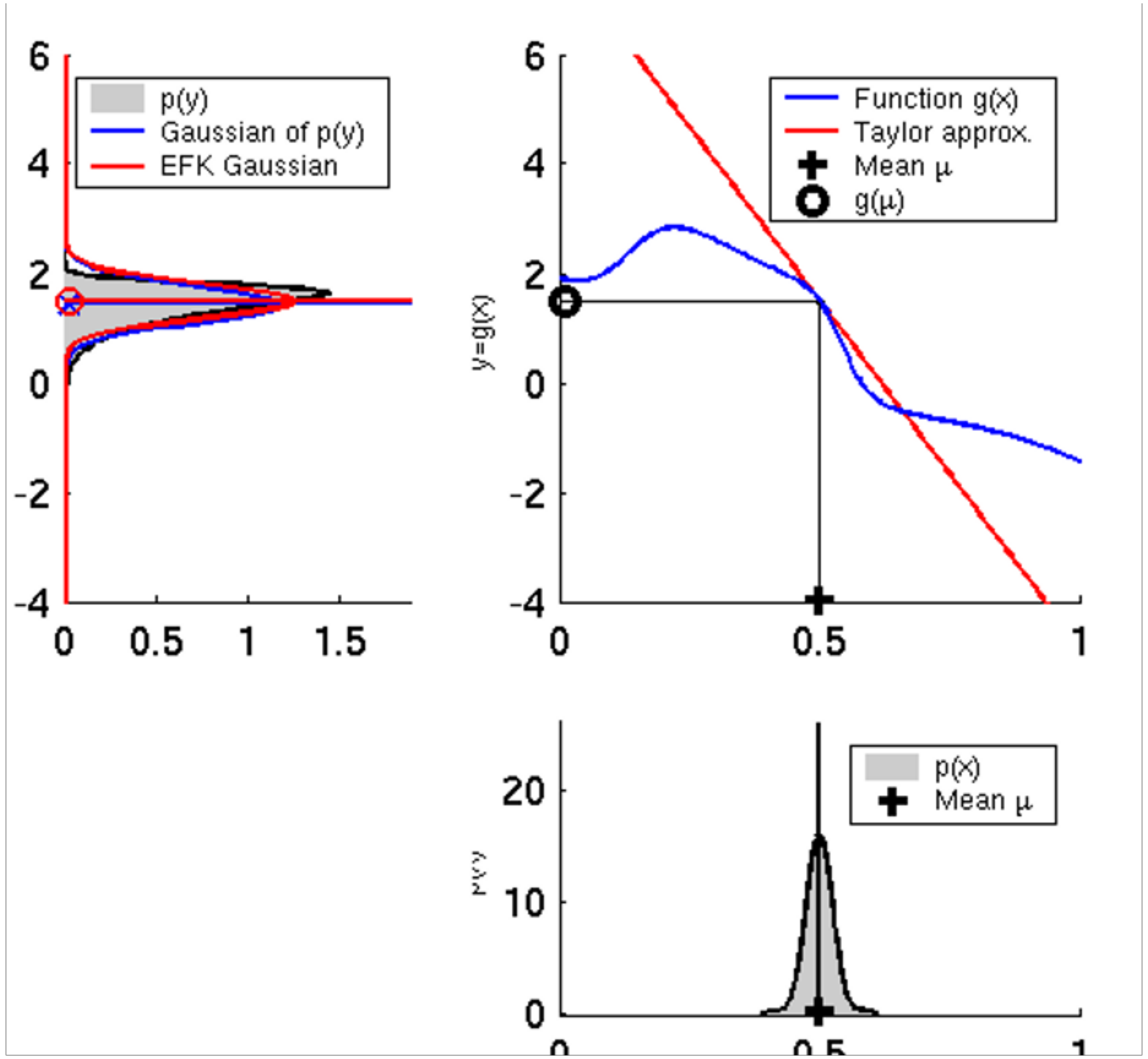
Extended Kalman Filter

- Intuition for EKF: linearize the non-linear system with a Taylor expansion and then apply Kalman filtering to the linearized system.
- $f(x_{t-1}, u_t) \approx f(\mu_{t-1}, u_t) + G_t^\top (x_{t-1} - \mu_{t-1})$ where G_t is the Jacobian of f at μ_{t-1}.
- $h(x_t) \approx h(\mu_t) + H_t^\top (x_t - \mu_t)$ where H_t is the Jacobian of h at μ_t.
- **Note:** the expansion point is set to be the mean of the current belief.
- Can also view the basic Kalman filter as a special case of EKF when f and g are linear functions.

Expanding at the Mean



Expanding at the Mean



Extended Kalman Filter

- Initialize belief: $\mathcal{N}(x_0, \mu_0, \Sigma_0)$
- Prediction:
- Correction:

Kalman Filter	**Extended Kalman Filter**
$\bar{\mu}_t = A\mu_{t-1} + Bu_t$	$\bar{\mu}_t = f(\mu_{t-1}, u_t)$
$\overline{\Sigma}_t = A^T \Sigma A + R$	$\overline{\Sigma}_t = G_t^T \Sigma G_t + R$
$\mu_t = \bar{\mu}_t + K_t(z_t - H\bar{\mu}_t)$	$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
$\Sigma_t = (I - K_t H)\overline{\Sigma}_t$	$\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t$

$$K_t = \overline{\Sigma}_t H_t^\top (H_t \overline{\Sigma}_t H_t^\top + R)^{-1}$$

Advantages / Disadvantages

- Extended Kalman filters:
    - Same strengths of Kalman filters (except optimality)
    - Relax the assumption of linear state transitions and observations.
    - Widely used.
- But…
    - Lose optimality guarantees.
    - Taylor expansion can be a bad approximation for highly non-linear systems.
    - If learning models, need accurate estimation of gradients.

Tracking multiple hypotheses

- Gaussians are unimodal distributions — key limitation of KF and EKF.
- One extension of KFs and EKFs is to use a Gaussian mixture model representation.
- Each possible mode is represented by a different Gaussian and updates are similar to KF/EKF updates.
  - But, must include mechanisms for splitting or pruning individual modes when they become very unlikely.

Practice

- A robot is using the following model of its environment:
  - $f(x_{t-1}, u_t) = Ax_{t-1} + Bu_t + w_t$ where w_t is Gaussian noise and A and B are matrices. The Jacobian of f is A.
  - $g(x_t) = Hx_t + v_t$ where v_t is Gaussian noise, H is a matrix, and the Jacobian of g is H.

What are the extended Kalman filter updates under this model?

Practice

- A robot is using the following model of its environment:
  - $f(x_{t-1}, u_t) = Ax_{t-1} + Bu_t + w_t$ where w_t is Gaussian noise and A and B are matrices. The Jacobian of f is A.
  - $g(x_t) = Hx_t + v_t$ where v_t is Gaussian noise, H is a matrix, and the Jacobian of g is H.

What are the extended Kalman filter updates under this model?

Note that f and g are already linear functions. Consequently, the EKF reduces to the KF.

Summary

- Extended the linear Gaussian model to the non-linear Gaussian model.
- Introduced the extended Kalman filter as a generalization of the Kalman filter for non-linear Gaussian assumption.
- Discussed pros and cons of the EKF.

Action Items

- Read on particle filter for next week; send a reading response by 12 pm on Monday.