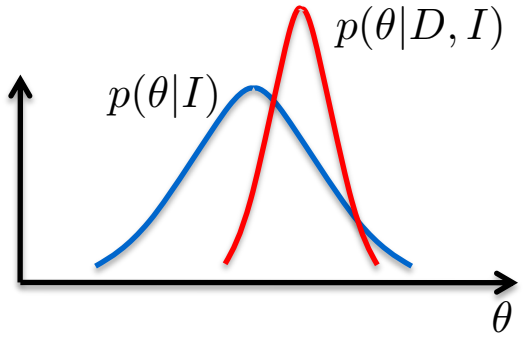
THE UNIVERSITY OF NEWCASTLE AUSTRALIA
FACULTY OF ENGINEERING AND BUILT ENVIRONMENT
www.newcastle.edu.au

Nonlinear Kalman filtering



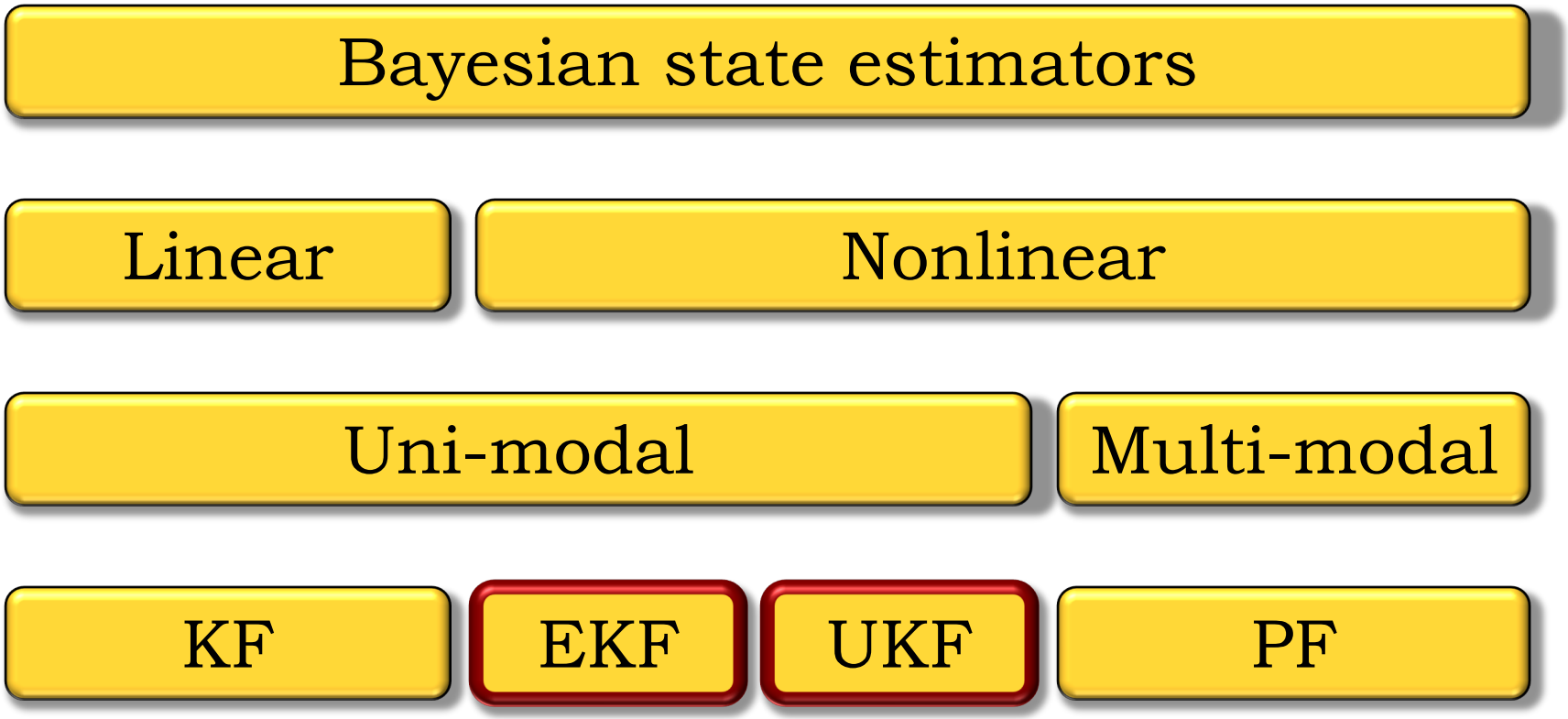
Dr. Chris Renton
School of Engineering

NL transformations

Linearisation and EKF

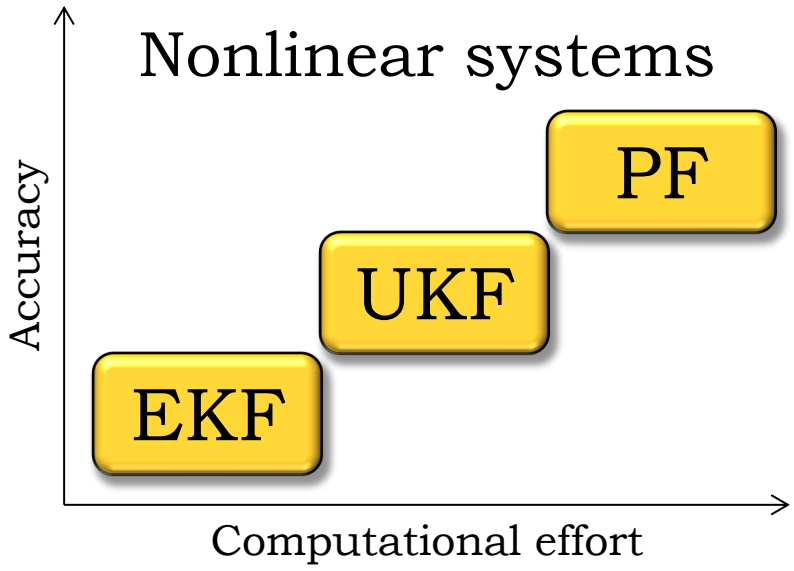
UT and UKF

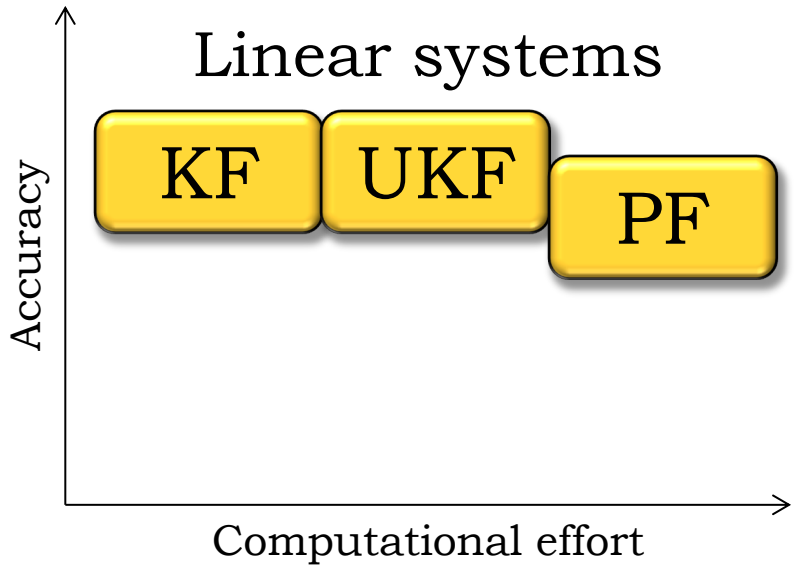
Classification of BSEs



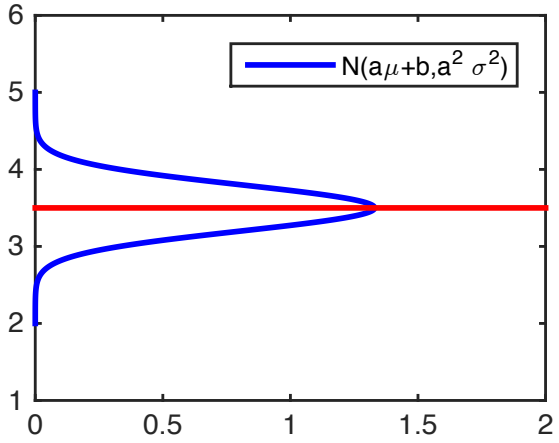
Nonlinear estimation

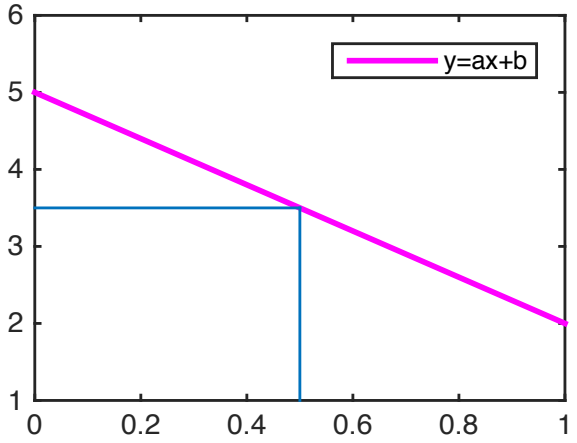
If model non-linearities are significant, we can employ a nonlinear estimator to reduce the estimator error and often improve robustness. If a model is linear, there is little to be gained by using a nonlinear estimator.

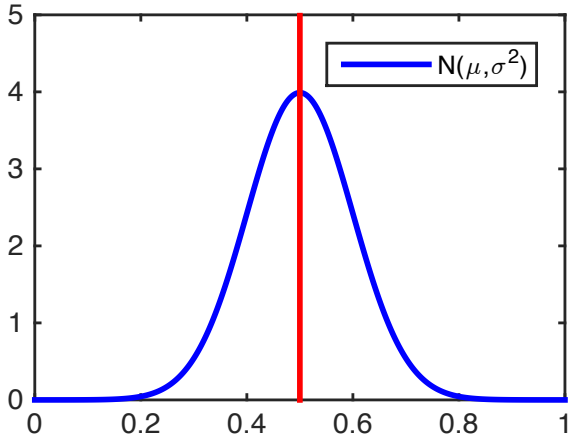




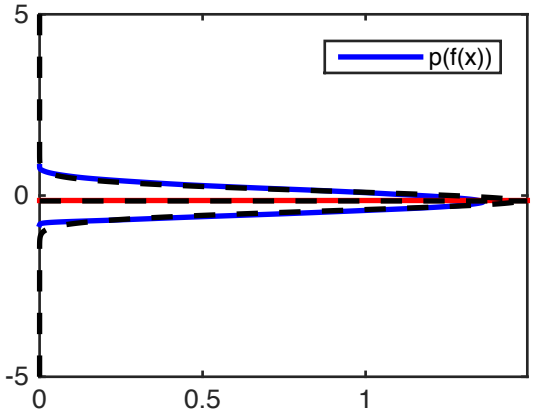
Kalman filter (review)

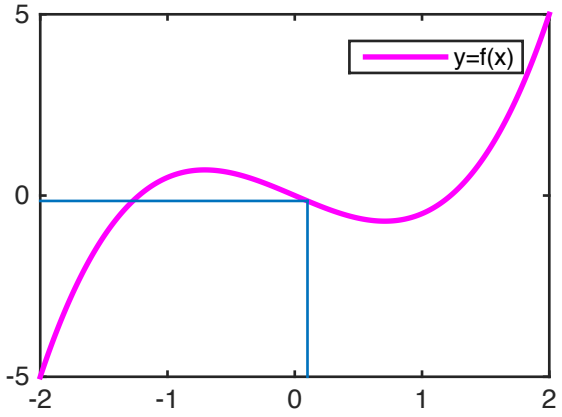






Kalman filter (a problem)

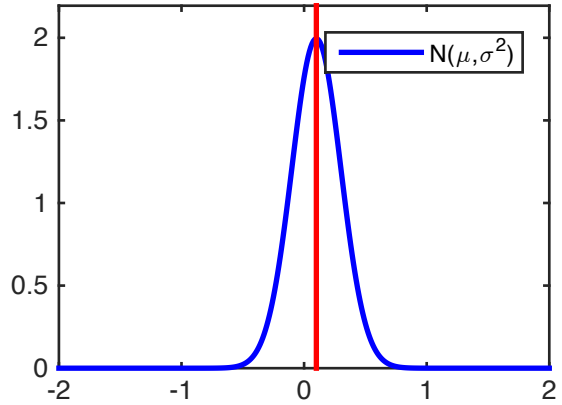




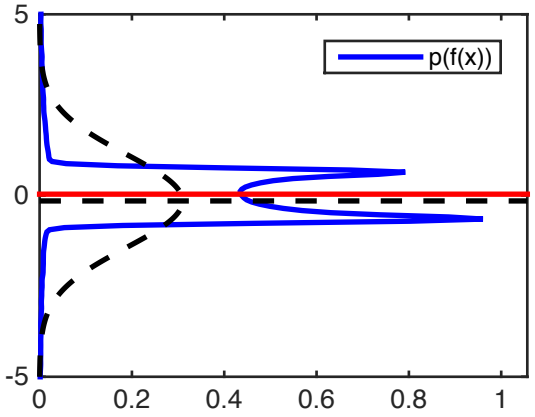
$$y = f(x) = x^3 - \frac{3}{2}x$$

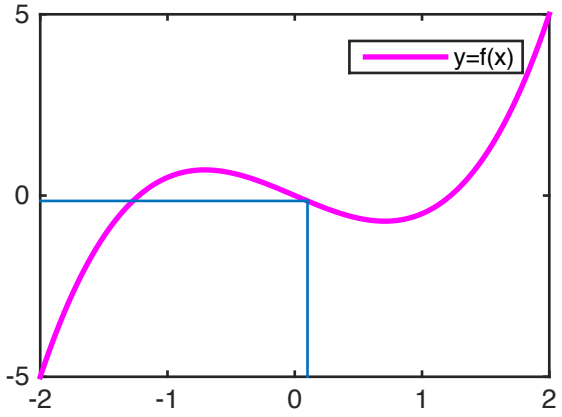
$$p_Y(y) = \sum_{x \in f^{-1}(y)} \left| \frac{\partial f(x)}{\partial x} \right|^{-1} p_X(x)$$

$$f^{-1}(y) = \{x \mid f(x) = y\}$$



Kalman filter (a problem)

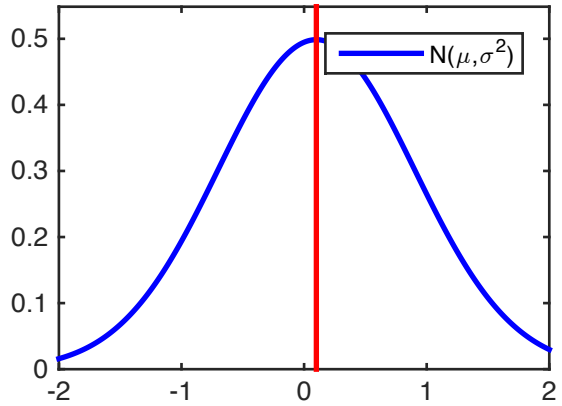




$$y = f(x) = x^3 - \frac{3}{2}x$$

$$p_Y(y) = \sum_{x \in f^{-1}(y)} \left| \frac{\partial f(x)}{\partial x} \right|^{-1} p_X(x)$$

$$f^{-1}(y) = \{x \mid f(x) = y\}$$



Nonlinear Kalman filter

Consider the following nonlinear model with additive Gaussian noise

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \\ \mathbf{w}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \\ \mathbf{v}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{R})\end{aligned}$$

Assume predicted and filtered densities can be approximated as Gaussian pdfs

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{k|k-1}, \mathbf{P}_{k|k-1})$$

$$\boldsymbol{\mu}_{k|k-1} \triangleq \mathrm{E}[\mathbf{x}_k|\mathbf{y}_{1:k-1}]$$

$$\mathbf{P}_{k|k-1} \triangleq \mathrm{cov}(\mathbf{x}_k|\mathbf{y}_{1:k-1})$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{k|k}, \mathbf{P}_{k|k})$$

$$\boldsymbol{\mu}_{k|k} \triangleq \mathrm{E}[\mathbf{x}_k|\mathbf{y}_{1:k}]$$

$$\mathbf{P}_{k|k} \triangleq \mathrm{cov}(\mathbf{x}_k|\mathbf{y}_{1:k})$$

Nonlinear Kalman filter

1. Form the prediction density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ by propagating $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ through the transformation

$$\mathbf{x}_{k-1} \mapsto \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$$

$$(\boldsymbol{\mu}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \mapsto (\boldsymbol{\mu}_{k|k-1}, \mathbf{P}_{k|k-1})$$

2. Form the joint pdf $p(\mathbf{x}_k,\mathbf{y}_k|\mathbf{y}_{1:k-1})$ by propagating $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ through the transformation

$$\mathbf{x}_k \mapsto \begin{bmatrix}\mathbf{x}_k \\ \mathbf{y}_k\end{bmatrix} = \begin{bmatrix}\mathbf{x}_k \\ \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k\end{bmatrix}$$

$$(\boldsymbol{\mu}_{k|k-1}, \mathbf{P}_{k|k-1}) \mapsto \left(\begin{bmatrix}\boldsymbol{\mu}_{k|k-1} \\ \boldsymbol{\mu}_{\mathbf{y}}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{k|k-1} & \mathbf{P}_{\mathbf{xy}} \\ \mathbf{P}_{\mathbf{yx}} & \mathbf{P}_{\mathbf{yy}}\end{bmatrix}\right)$$

3. Compute the conditional pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ by conditioning the joint pdf on $\mathbf{y}_k$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}_{1:k-1}) = \frac{p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}$$

$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_{\mathbf{y}})$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}\mathbf{P}_{\mathbf{yx}}$$

Nonlinear Kalman filter

We have a choice in how to perform the transformations

- Linearise the models about a nominal operating state and then use affine Gaussian transformations.
  This leads to the **Linearised Kalman Filter** (LKF).

- Linearise the process model about the previous filtered mean and linearise the measurement model about the predicted mean and then use affine Gaussian transformations.
  This leads to the **Extended Kalman Filter** (EKF).

- Use the Unscented Transform to evaluate the mean and covariance of the prediction density $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ and the joint density $p(\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})$.
  This leads to the **Unscented Kalman Filter** (UKF).

Extended Kalman filter

Recall the model

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k$$
$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$$

Use a Taylor expansion of the process model about previous filtered mean and a Taylor expansion of the measurement model about the predicted mean

$$\mathbf{x}_{k+1} \approx \mathbf{f}_k(\boldsymbol{\mu}_{k|k}) + \mathbf{A}_k(\mathbf{x}_k - \boldsymbol{\mu}_{k|k}) + \mathbf{w}_k$$
$$\mathbf{y}_k \approx \mathbf{h}_k(\boldsymbol{\mu}_{k|k-1}) + \mathbf{C}_k(\mathbf{x}_k - \boldsymbol{\mu}_{k|k-1}) + \mathbf{v}_k$$

where

$$\mathbf{A}_k \triangleq \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k}} \qquad \mathbf{C}_k \triangleq \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k-1}}$$

Extended Kalman filter

In summary, the extended Kalman filter consists of the steps:

1. Initialise the prior with some sensible values

$$\boldsymbol{\mu}_{0|0} = \mathrm{E}[\mathbf{x}_0]$$
$$\mathbf{P}_{0|0} = \mathrm{E}[(\mathbf{x}_0 - \boldsymbol{\mu}_{0|0})(\mathbf{x}_0 - \boldsymbol{\mu}_{0|0})^\mathsf{T}]$$

2. Repeat the following for $k = 1,2,\ldots$
   a. Prediction

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{f}_{k-1}(\boldsymbol{\mu}_{k-1|k-1})$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^\mathsf{T} + \mathbf{Q}$$

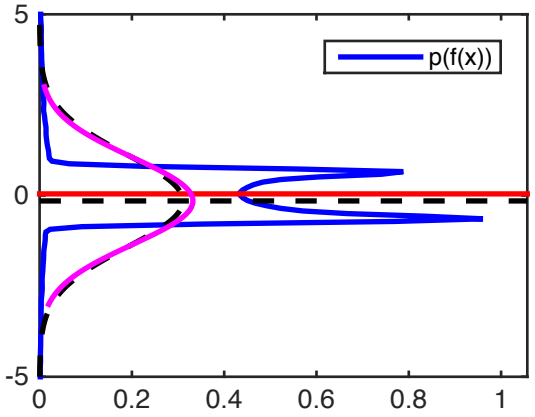
$$\mathbf{A}_k \triangleq \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k}}$$

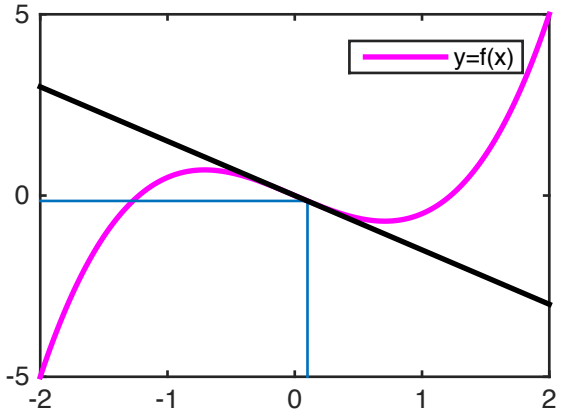
   b. Correction

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}_k^\mathsf{T}(\mathbf{C}_k\mathbf{P}_{k|k-1}\mathbf{C}_k^\mathsf{T} + \mathbf{R})^{-1}$$
$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\boldsymbol{\mu}_{k|k-1}))$$
$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k|k-1}$$

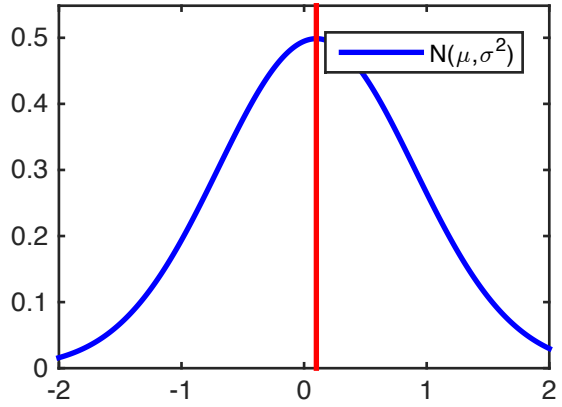
$$\mathbf{C}_k \triangleq \left.\frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k-1}}$$

Extended Kalman Filter





$$y = x^3 - \frac{3}{2}x$$



Parameter estimation

By including parameters in the state vector, we can jointly estimate both states and parameters.

We can use a random walk model for the parameters and then tune the process noise to inform the filter how quickly to adjust the parameters to explain the measurements.

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{w}_k$$

The resulting model for the joint state and parameter estimator is usually nonlinear, as parameters and states are often combined multiplicatively.

The state estimator is still bound by observability; now of both states and parameters.

The estimator will try to explain the data using whatever states and parameters are provided, without careful tuning of the parameter process noise.

Example – Mass damper

Consider estimating the position, x_1, and velocity, x_2, of a mass-damper system while jointly estimating the linear damping coefficient, x_3.

This leads to the following state-space model:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{1}{m} x_2 x_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} u + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

The product between the damping coefficient and the velocity leads to a nonlinear model.

Example – 1DOF Ballistic coefficient estimation

Consider the problem of tracking an object falling vertically through the troposphere while simultaneously estimating its ballistic coefficient.

Let x_1 = height, x_2 = velocity, x_3 = bal. coeff.

Acceleration due to aerodynamic drag:

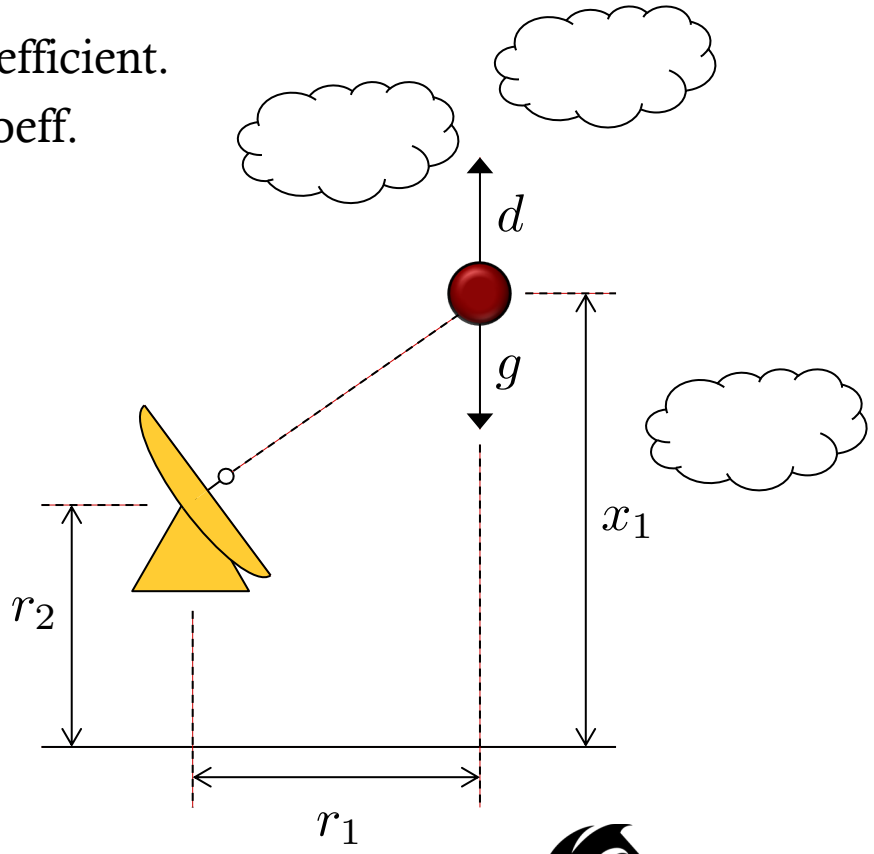
$$d(\mathbf{x}) = \frac{1}{2}\rho(x_1)x_2^2 x_3$$

where

$$\rho(x_1) = \frac{Mp(x_1)}{RT(x_1)}$$

$$T(x_1) = T_0 - Lx_1$$

$$p(x_1) = p_0\left(1 - \frac{Lx_1}{T_0}\right)^{\frac{gM}{RL}}$$



Example – 1DOF Ballistic coefficient estimation

The process model in continuous time is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ d(\mathbf{x}) - g \\ 0 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

We can discretise this using the Euler approximation for the sampling time T,

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} + T x_{2,k} \\ x_{2,k} + T(d(\mathbf{x}_k) - g) \\ x_{3,k} \end{bmatrix} + \begin{bmatrix} w_{1,k} \\ w_{2,k} \\ w_{3,k} \end{bmatrix}$$

Example – 1DOF Ballistic coefficient estimation

The EKF needs the linearisation about the current state estimate

$$\mathbf{A}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k}} = \begin{bmatrix} 1 & T & 0 \\ \cdot & \cdot & \cdot \\ 0 & 0 & 1 \end{bmatrix}$$ (see example code)

The measurement model is $y_k = \sqrt{r_1^2 + (x_{1,k} - r_2)^2}$ so therefore

$$\mathbf{C}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \boldsymbol{\mu}_{k|k-1}} = \begin{bmatrix} \frac{x_{1,k} - r_2}{\sqrt{r_1^2 + (x_{1,k} - r_2)^2}} & 0 & 0 \end{bmatrix}$$

The system has no input, so **B = 0** and **D = 0**, but these matrices aren't used by the EKF anyway.

Constrained parameters

When performing parameter estimation, we must be careful to avoid escaping the domain of feasibility for the model.

Sometimes, we can reparameterise the system to avoid violating constraints.

For example, if we want to enforce some parameter $a > 0$, we can define a new parameter, $\tilde{a} = \ln(a)$, so that

$$\exp(\tilde{a}) \triangleq a > 0, \qquad -\infty < \tilde{a} < \infty$$

and then express the model in terms of the new parameter $\tilde{a}$.

There are more sophisticated techniques for dealing with nontrivial inequality constraints. These project infeasible points back onto the boundary of the feasible region. This complicates the KF and EKF algorithms, but are relatively straightforward to implement in the UKF and PF.

Sensor/data fusion

With multiple sensors, we can combine the data gathered in a way that provides a better estimate than any one particular sensor. Examples:

- Accelerometer and rate gyro
- GPS and odometry
- Multiple position encoders

Implementation is usually straightforward: write a measurement equation for each sensor and vertically concatenate these models.

Even when vastly different quality sensors are being used, we should not discard information from poor sensors, but instead inform the estimator (via the measurement noise covariance) how much trust to place in each sensor.

Sensor/data fusion

"Robustness in the usual sense of the word can always be achieved merely by throwing away cogent information! It is hard to believe that anyone could really want this if he were aware of it; but those with only orthodox training do not think in terms of information content and so do not realize when they are wasting information."

Probability Theory: The Logic of Science

E.T. Jaynes

Sensor/data fusion

"Euler failed to solve it, but not because of the magnitude of this computation; he failed even to comprehend the principle needed to solve it. Instead of seeing that by combining many observations their errors tend to cancel, he thought that this would only 'multiply the errors' and make things worse. In other words, Euler concentrated his attention entirely on the worst possible thing that could happen, as if it were certain to happen - which makes him perhaps the first really devout believer in Murphy's Law."

Probability Theory: The Logic of Science

E.T. Jaynes

Sensor/data fusion (example)

Consider a Wiener process acceleration model

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}_t + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{w}, \qquad \mathbf{x}_{k+1} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} w_k$$

And assume that we measure position and acceleration

$$y_{1,k} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + v_{1,k}, \qquad y_{2,k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + v_{2,k}$$

How do we estimate the velocity?

Sampling estimators

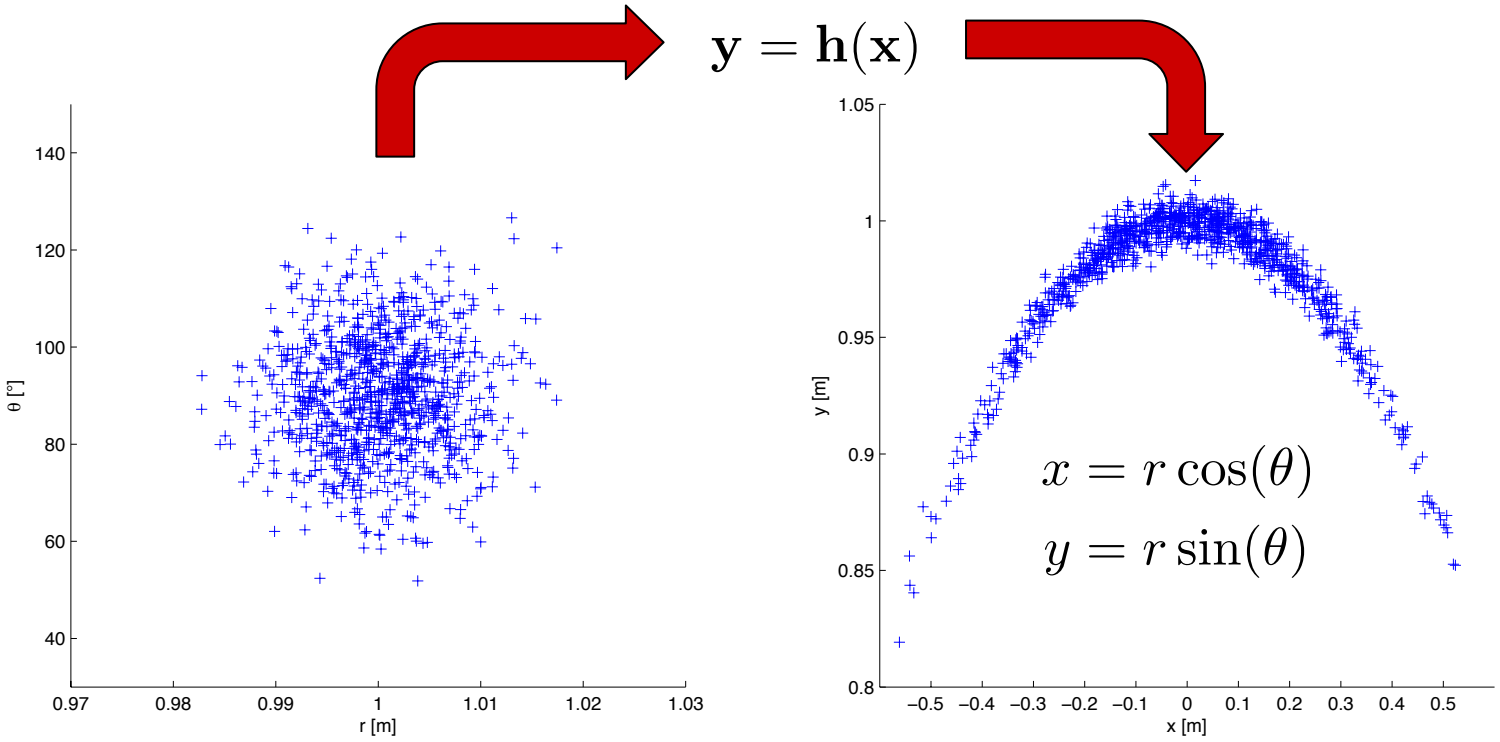
UKF and PF operate by transforming points through the process and measurement models, and reconstructing approximations of the pdfs involved.

Unlike the EKF, this approach does not require derivative information about the model. This means we have a tool to solve estimation problems where either the model derivatives are unavailable, or the models are nonsmooth (e.g., localisation on grid maps). Such models may also arise as the result of spatial discretisation of a set of PDEs (e.g., FEA, CFD), which must be solved numerically, and like the model itself, closed-form solution for the derivatives are not available.

The sampling estimators also permit a straightforward way of dealing with constraints. We will see this later.

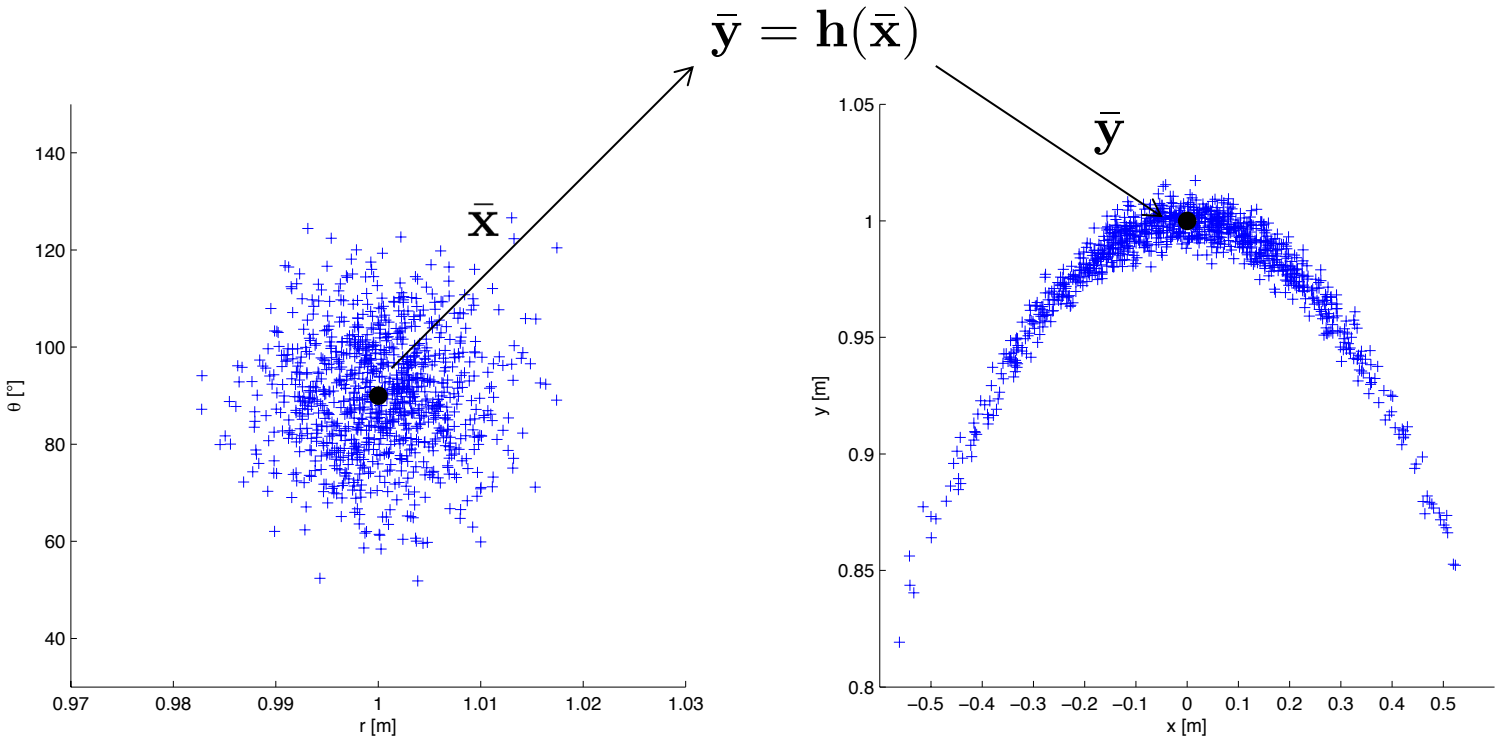
Propagation through a nonlinear transformation

Consider a polar to Cartesian transformation.



Propagation through a nonlinear transformation

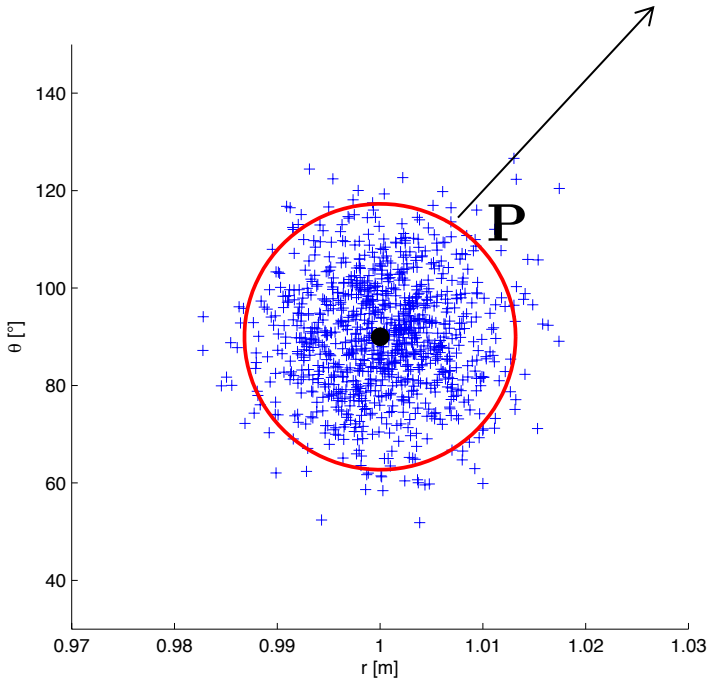
Propagate the mean through the linearised transformation.

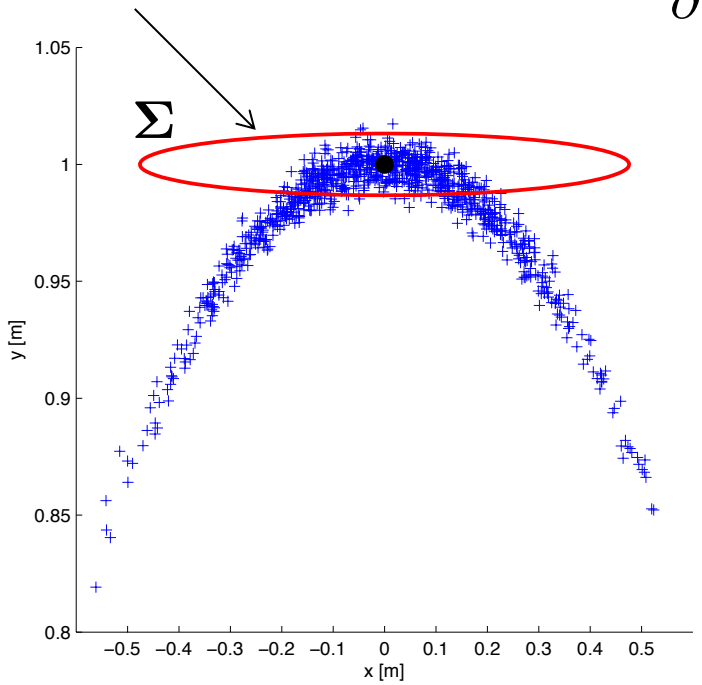


Propagation through a nonlinear transformation

Propagate the covariance through the linearised transformation.

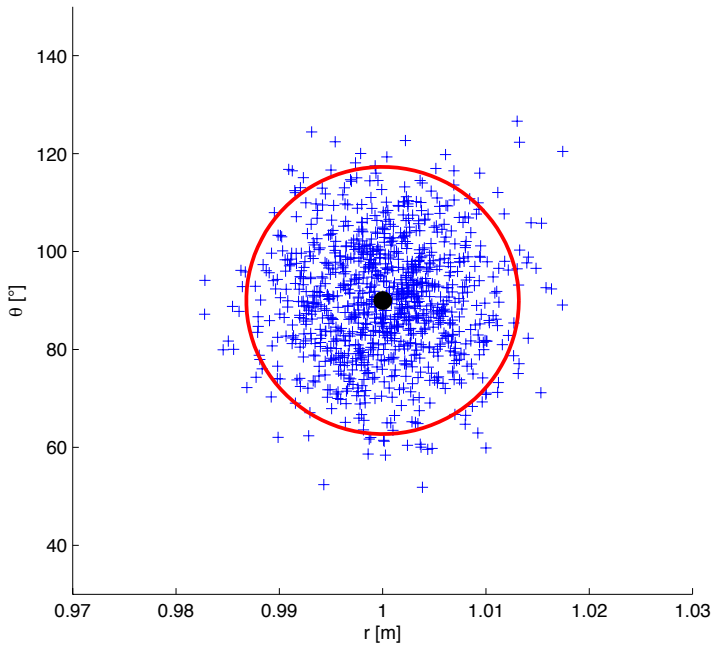
$$\boldsymbol{\Sigma} = \mathbf{C}\mathbf{P}\mathbf{C}^{\mathsf{T}} \qquad \mathbf{C} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\bar{\mathbf{x}}}$$

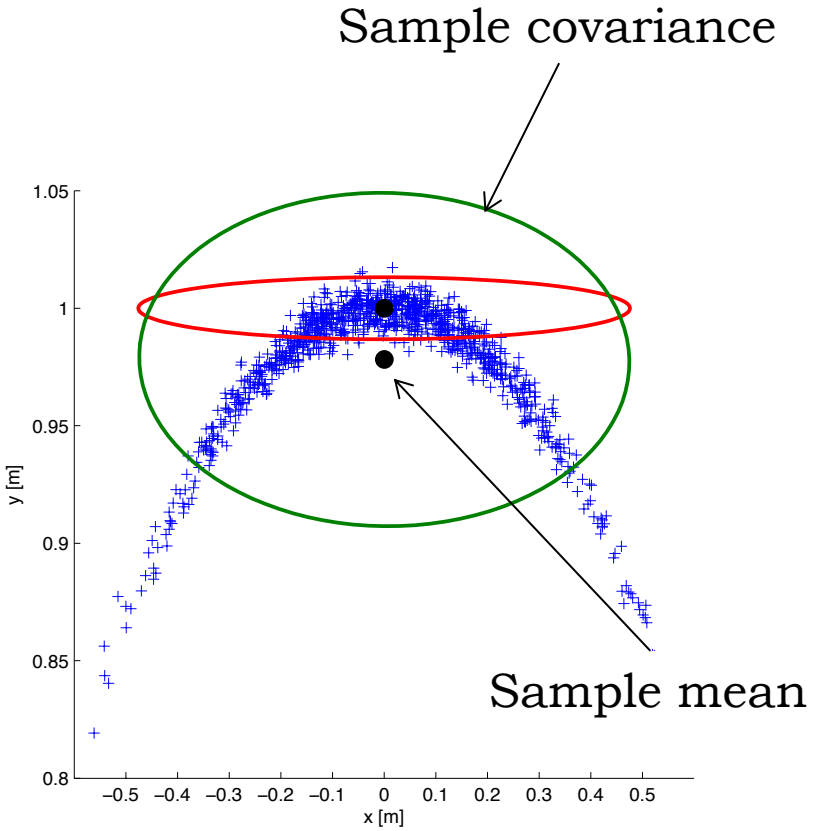




Propagation through a nonlinear transformation

The sample mean and sample covariance of the transformed data disagree.





Propagation through a nonlinear transformation

Clearly, in some cases, using the linearisation to propagate the mean and covariance does not provide a good approximation of the true mean and covariance of the transformed RV.

How can we more accurately calculate the mean and covariance of a transformed RV?

One answer is the Unscented Transform (UT), which leads to the Unscented Kalman Filter (UKF).

Another answer is just to push those thousands of points through the models and just see what happens. This approach leads to the Particle Filter (PF).

Unscented Transform

The basic idea is to pick a set of $2n$ **sigma points**, according to

$$\mathbf{x}^{(i)} = \boldsymbol{\mu}_{\mathbf{x}} + \left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

$$\mathbf{x}^{(i+n)} = \boldsymbol{\mu}_{\mathbf{x}} - \left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

where $(\bullet)_i$ is the i^{th} row of $(\bullet)$, and

$$\left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right)^{\mathsf{T}} \left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right) = n\mathbf{P}_{\mathbf{x}}$$

is the upper Cholesky factorisation of $n\mathbf{P}_{\mathbf{x}}$.

Then, we propagate the sigma points through the function of interest, and compute the sample mean and sample covariance of the transformed points.

Unscented Transform

The Unscented Transform $(\boldsymbol{\mu}_{\mathbf{x}}, \mathbf{P}_{\mathbf{x}}) \mapsto (\boldsymbol{\mu}_{\mathbf{y}}, \mathbf{P}_{\mathbf{y}})$ consists of the steps:

1. Form $2n$ sigma point vectors, $\mathbf{x}^{(i)}$,

$$\mathbf{x}^{(i)} = \boldsymbol{\mu}_{\mathbf{x}} + \left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

$$\mathbf{x}^{(i+n)} = \boldsymbol{\mu}_{\mathbf{x}} - \left(\sqrt{n\mathbf{P}_{\mathbf{x}}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

2. Transform the sigma points through the function of interest

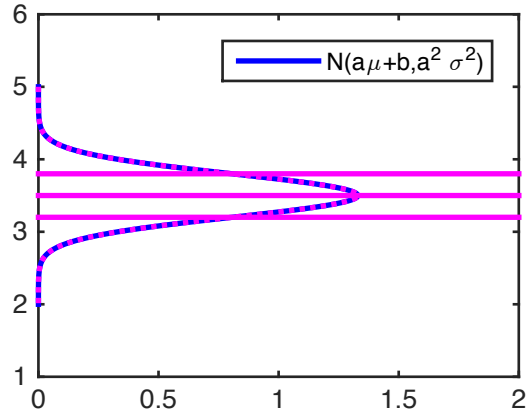
$$\mathbf{y}^{(i)} = \mathbf{h}(\mathbf{x}^{(i)}), \quad i = 1, \ldots, 2n$$

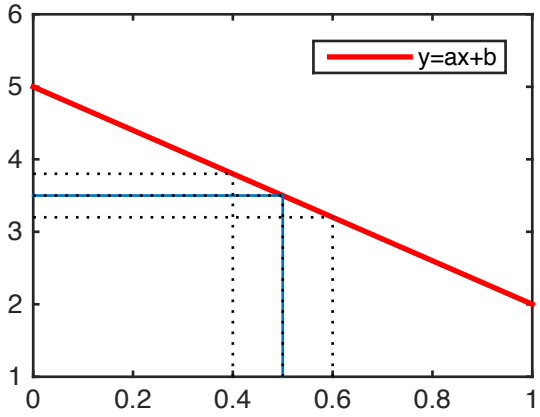
3. Approximate the mean and covariance as

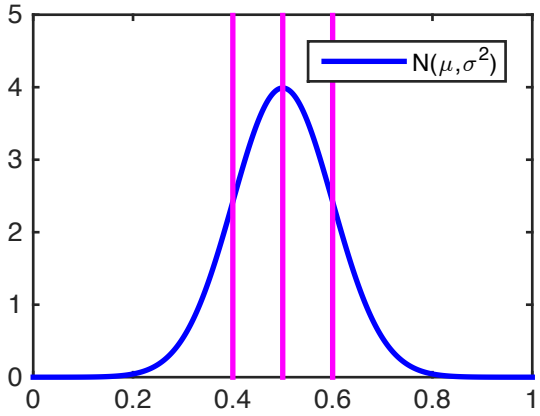
$$\boldsymbol{\mu}_{\mathbf{y}} = \frac{1}{2n} \sum_{i=1}^{2n} \mathbf{y}^{(i)}$$

$$\mathbf{P}_{\mathbf{y}} = \frac{1}{2n} \sum_{i=1}^{2n} \left(\mathbf{y}^{(i)} - \boldsymbol{\mu}_{\mathbf{y}}\right) \left(\mathbf{y}^{(i)} - \boldsymbol{\mu}_{\mathbf{y}}\right)^{\mathsf{T}}$$

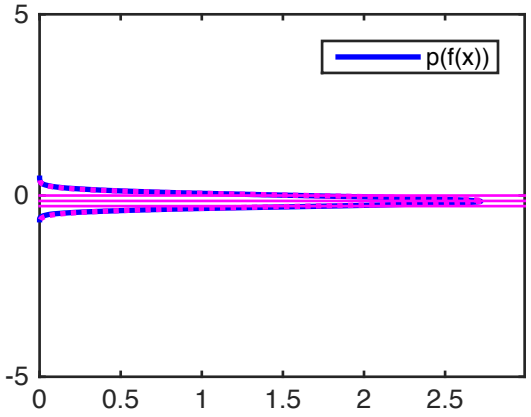
Unscented Transform

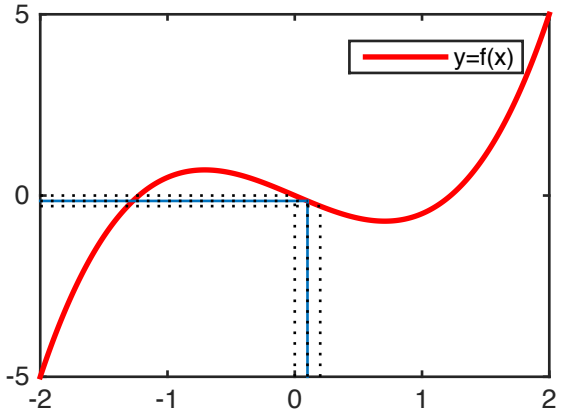




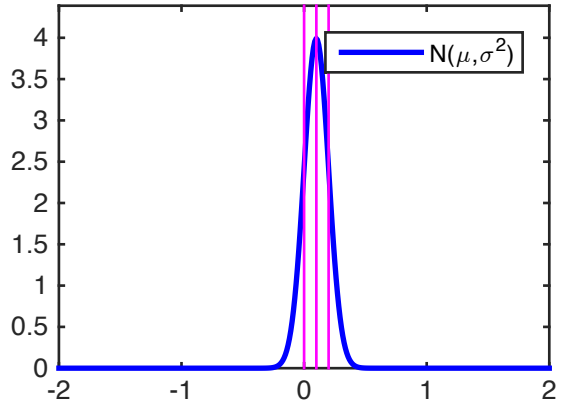


Unscented Transform

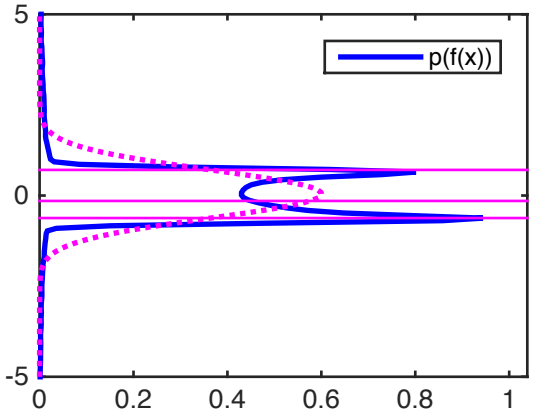


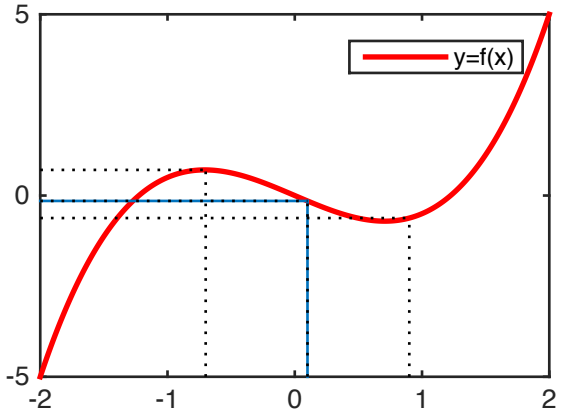


$$y = x^3 - \frac{3}{2}x$$

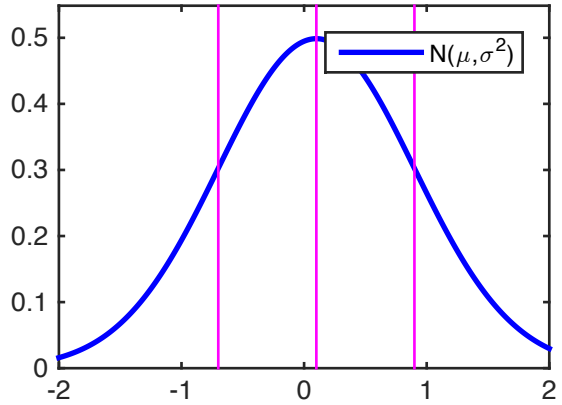


Unscented Transform



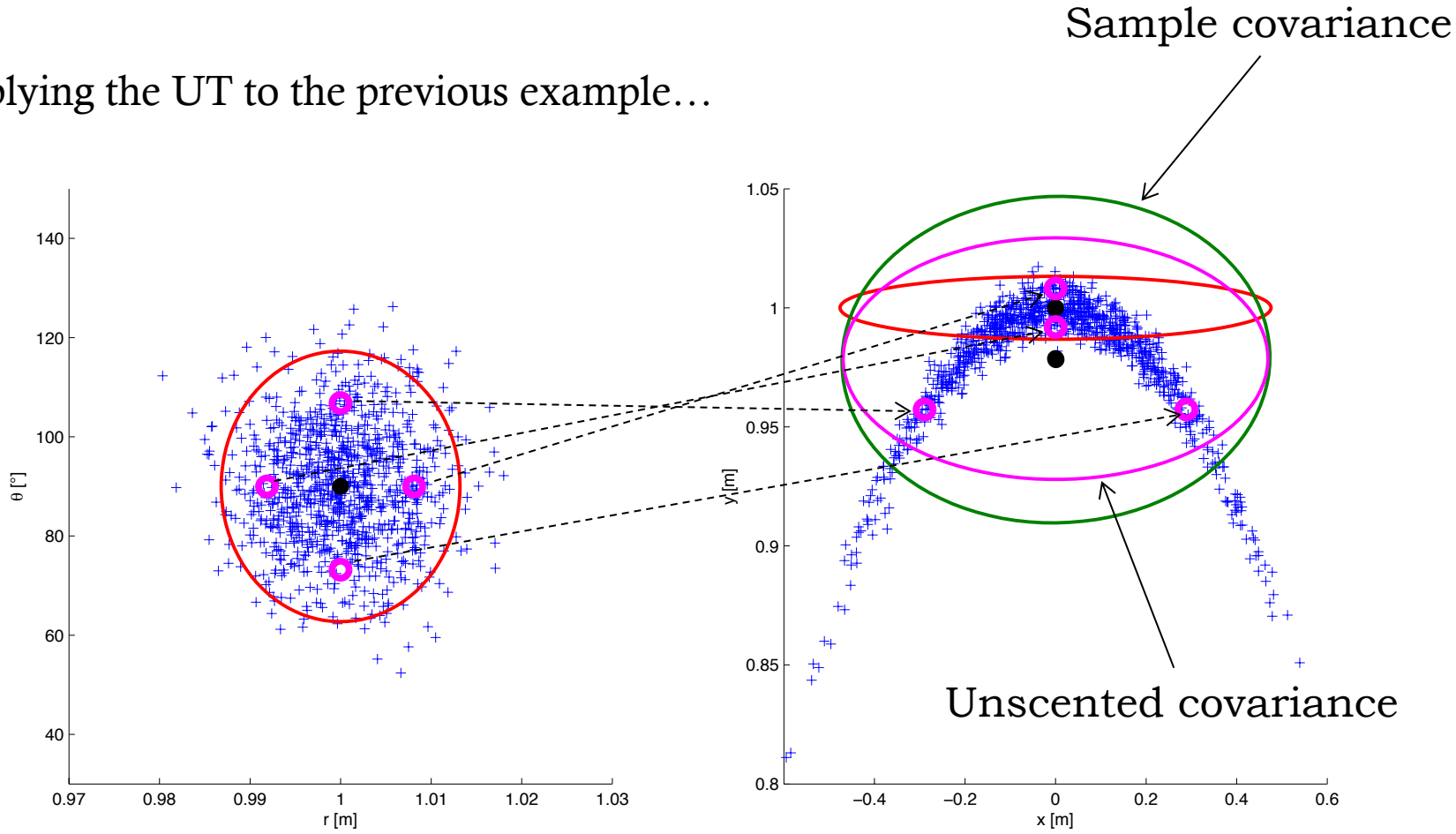


$$y = x^3 - \frac{3}{2}x$$



Unscented Transform

Applying the UT to the previous example…



Unscented Kalman filter

The Unscented Transform computes the mean and covariance correctly up to at least second order for *any* nonlinearity, whereas the linearisation is only correct to first order.

Use the UT for each transformation step and we get the UKF.

Unscented Kalman filter

1. Form the prediction density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ by propagating $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ through the transformation

$$\mathbf{x}_{k-1} \mapsto \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}$$

   a. Use UT to map through noiseless process model

$$\boldsymbol{\Phi}_{\mathbf{f}} : \mathbf{x}_{k-1} \mapsto \mathbf{f}_{k-1}(\mathbf{x}_{k-1})$$
$$\mathcal{UT}\{\boldsymbol{\Phi}_{\mathbf{f}}\} : (\boldsymbol{\mu}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \mapsto (\boldsymbol{\mu}_{k|k-1}, \mathbf{P}_{\mathbf{ff}})$$

   b. Add process noise

$$\mathbf{P}_{k|k-1} = \mathbf{P}_{\mathbf{ff}} + \mathbf{Q}$$

Unscented Kalman filter

2. Form the joint pdf $p(\mathbf{x}_k, \mathbf{y}_k | \mathbf{y}_{1:k-1})$ by propagating $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ through the transformation

$$\boldsymbol{\Phi}_{\mathbf{xy}} : \mathbf{x}_k \mapsto \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \end{bmatrix}$$

   a. Use UT to map through noiseless measurement model

$$\boldsymbol{\Phi}_{\mathbf{xh}} : \mathbf{x}_k \mapsto \begin{bmatrix} \mathbf{x}_k \\ \mathbf{h}_k(\mathbf{x}_k) \end{bmatrix}$$

$$\mathcal{UT}\{\boldsymbol{\Phi}_{\mathbf{xh}}\} : (\boldsymbol{\mu}_{k|k-1}, \mathbf{P}_{k|k-1}) \mapsto \left(\begin{bmatrix} \boldsymbol{\mu}_{k|k-1} \\ \boldsymbol{\mu}_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{k|k-1} & \mathbf{P}_{\mathbf{xy}} \\ \mathbf{P}_{\mathbf{yx}} & \mathbf{P}_{\mathbf{hh}} \end{bmatrix} \right)$$

   b. Add measurement noise

$$\mathbf{P}_{\mathbf{yy}} = \mathbf{P}_{\mathbf{hh}} + \mathbf{R}$$

Unscented Kalman filter

3. Compute the conditional pdf $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ by conditioning the joint pdf on $\mathbf{y}_k$

$$p(\mathbf{x},\mathbf{y}) = p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_\mathbf{x}\\ \boldsymbol{\mu}_\mathbf{y}\end{bmatrix}, \begin{bmatrix}\mathbf{P}_{\mathbf{xx}} & \mathbf{P}_{\mathbf{xy}}\\ \mathbf{P}_{\mathbf{yx}} & \mathbf{P}_{\mathbf{yy}}\end{bmatrix}\right)$$

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}, \mathbf{P}_{\mathbf{x}|\mathbf{y}}) \\ &= \mathcal{N}(\mathbf{x}; \underbrace{\boldsymbol{\mu}_\mathbf{x} + \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}(\mathbf{y}-\boldsymbol{\mu}_\mathbf{y})}_{\boldsymbol{\mu}_{\mathbf{x}|\mathbf{y}}=\mathrm{E}[\mathbf{x}|\mathbf{y}]}, \underbrace{\mathbf{P}_{\mathbf{xx}} - \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}\mathbf{P}_{\mathbf{yx}}}_{\mathbf{P}_{\mathbf{x}|\mathbf{y}}=\mathrm{cov}(\mathbf{x}|\mathbf{y})}) \end{aligned}$$

$$\begin{aligned} \boldsymbol{\mu}_{k|k} &= \boldsymbol{\mu}_{k|k-1} + \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_\mathbf{y}) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1}\mathbf{P}_{\mathbf{yx}} \end{aligned}$$

Unscented Kalman filter

In summary, the Unscented Kalman filter consists of the steps:

1. Initialise the prior with some sensible values

$$\boldsymbol{\mu}_{0|0} = \mathrm{E}[\mathbf{x}_0]$$
$$\mathbf{P}_{0|0} = \mathrm{E}[(\mathbf{x}_0 - \boldsymbol{\mu}_{0|0})(\mathbf{x}_0 - \boldsymbol{\mu}_{0|0})^{\mathsf{T}}]$$

2. Repeat the following for $k = 1,2,\ldots$
   a. Prediction
      i. Generate sigma points

$$\mathbf{x}^{(i)}_{k-1|k-1} = \boldsymbol{\mu}_{k-1|k-1} + \left(\sqrt{n\mathbf{P}_{k-1|k-1}}\right)^{\mathsf{T}}_i, \qquad i = 1,\ldots,n$$
$$\mathbf{x}^{(i+n)}_{k-1|k-1} = \boldsymbol{\mu}_{k-1|k-1} - \left(\sqrt{n\mathbf{P}_{k-1|k-1}}\right)^{\mathsf{T}}_i, \qquad i = 1,\ldots,n$$

Unscented Kalman filter

2. Repeat the following for $k = 1,2,\ldots$ *(continued)*
   a. Prediction *(continued)*
      ii. Transform sigma points through process model

$$\mathbf{x}_{k|k-1}^{(i)} = \mathbf{f}_{k-1}(\mathbf{x}_{k-1|k-1}^{(i)}), \qquad i = 1, \ldots, 2n$$

      iii. Obtain the prediction mean

$$\boldsymbol{\mu}_{k|k-1} = \frac{1}{2n} \sum_{i=1}^{2n} \mathbf{x}_{k|k-1}^{(i)}$$

      iv. Obtain the prediction covariance

$$\mathbf{P}_{k|k-1} = \frac{1}{2n} \sum_{i=1}^{2n} \left(\mathbf{x}_{k|k-1}^{(i)} - \boldsymbol{\mu}_{k|k-1} \right) \left(\mathbf{x}_{k|k-1}^{(i)} - \boldsymbol{\mu}_{k|k-1} \right)^{\mathsf{T}} + \mathbf{Q}$$

Unscented Kalman filter

2. Repeat the following for $k = 1,2,\ldots$ *(continued)*
   b. Correction
      i. Re-generate sigma points

$$\mathbf{x}_{k|k-1}^{(i)} = \boldsymbol{\mu}_{k|k-1} + \left(\sqrt{n\mathbf{P}_{k|k-1}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

$$\mathbf{x}_{k|k-1}^{(i+n)} = \boldsymbol{\mu}_{k|k-1} - \left(\sqrt{n\mathbf{P}_{k|k-1}}\right)_i^{\mathsf{T}}, \qquad i = 1, \ldots, n$$

      ii. Transform sigma points through measurement model

$$\mathbf{y}_k^{(i)} = \mathbf{h}_k(\mathbf{x}_{k|k-1}^{(i)})$$

      iii. Obtain predicted measurement estimate

$$\boldsymbol{\mu}_{\mathbf{y}} = \frac{1}{2n}\sum_{i=1}^{2n} \mathbf{y}_k^{(i)}$$

Unscented Kalman filter

2. Repeat the following for $k = 1,2,\ldots$ *(continued)*
   b. Correction *(continued)*
      iv. Obtain predicted measurement covariance

$$\mathbf{P_{yy}} = \frac{1}{2n}\sum_{i=1}^{2n}\left(\mathbf{y}_k^{(i)} - \boldsymbol{\mu}_\mathbf{y}\right)\left(\mathbf{y}_k^{(i)} - \boldsymbol{\mu}_\mathbf{y}\right)^\mathsf{T} + \mathbf{R}$$

      v. Obtain cross-covariance between state and measurement

$$\mathbf{P_{xy}} = \frac{1}{2n}\sum_{i=1}^{2n}\left(\mathbf{x}_{k|k-1}^{(i)} - \boldsymbol{\mu}_{k|k-1}\right)\left(\mathbf{y}_k^{(i)} - \boldsymbol{\mu}_\mathbf{y}\right)^\mathsf{T}$$

      vi. Update the filtered mean and covariance estimate

$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \mathbf{P_{xy}}\mathbf{P_{yy}}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_\mathbf{y})$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P_{xy}}\mathbf{P_{yy}}^{-1}\mathbf{P_{xy}}^\mathsf{T}$$

UKF variants

More general choices of sigma points exist with different (non-uniform) weighting, e.g., general ($2n$+1), simplex (n+1), spherical (n+1). These can achieve better accuracy (3rd higher order accuracy or higher) or require less computational effort.

Other modifications such as square-root filtering improve numerical precision, by working directly with the square root of the covariance instead of the covariance itself,

$$\mathbf{S}^{\mathsf{T}}\mathbf{S} = \mathbf{P}$$

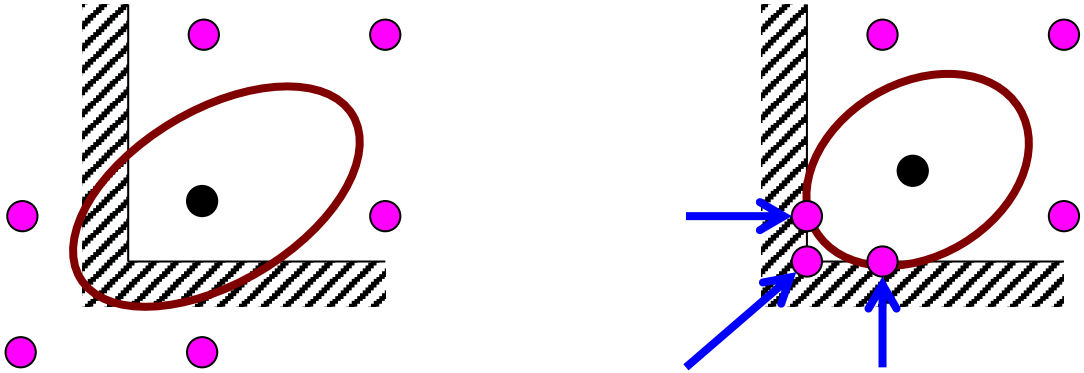
The matrix $\mathbf{S}$ is the upper Cholesky decomposition of $\mathbf{P}$.

We will revisit square root implementations of Kalman filters.

Dealing with constraints

Both the UKF and PF employ types of particles to sample around likely states and parameters. If the estimation error covariance is large, the filter may try to evaluate the process model in a region which is not valid (e.g., negative damping coefficients) leading to problems.

A straightforward way to deal with this is to "clip" or project infeasible points back onto the boundary of the feasible region.



For more details, see Kandepu *et al.* (2008).

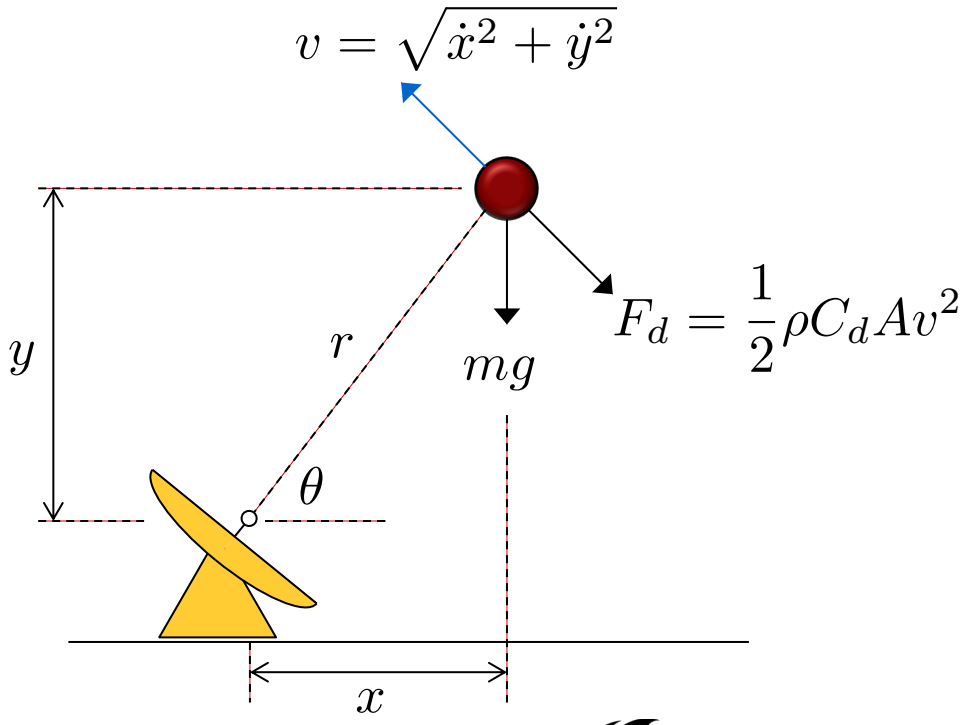
Example – 2D projectile tracking

Consider the problem of tracking a ballistic projectile in 2D using range and angle information, while estimating the coefficient of drag online.

$$m\ddot{x} = -\frac{1}{2}\rho C_d A \dot{x}\sqrt{\dot{x}^2 + \dot{y}^2}$$

$$m\ddot{y} = -\frac{1}{2}\rho C_d A \dot{y}\sqrt{\dot{x}^2 + \dot{y}^2} - mg$$

$$\begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \text{atan2}(y, x) \end{bmatrix}$$



Example – 2D projectile tracking

Choose the state vector $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & C_d \end{bmatrix}^\mathsf{T}$

The process model in continuous time is given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{1}{2m}\rho x_5 A x_2 \sqrt{x_2^2 + x_4^2} \\ x_4 \\ -\frac{1}{2m}\rho x_5 A x_4 \sqrt{x_2^2 + x_4^2} - g \\ 0 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix}$$

The measurement model is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \sqrt{x_1^2 + x_3^2} \\ \mathrm{atan2}(x_3, x_1) \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Example – Blackbox model fitting

When significant residuals remain after fitting data gathered from physical processes to models derived from first principles (whitebox models), we can employ blackbox models to explain the remaining error. We use a training data set to fit the blackbox model, and then use a validation data set to evaluate the performance and to avoid overfitting.

$$\mathbf{y} = \mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2$$

An approach to fit this type of model is to configure a SRUKF as a parameter estimator. The unscented transform can cope with the significant nonlinearity in the model, and, unlike methods based on gradient descent, this algorithm is resistant to becoming trapped in local minima due to the injection of artificial process noise.

References

- Simon, D. (2006) "*Optimal State Estimation*" John Wiley & Sons
  - Chapters 13 & 14

- S.J. Julier and J.K. Uhlmann (1997) "*A new extension of the Kalman filter to nonlinear systems*" Int. Symp. Aerospace Defense Sensing, Simul. and Controls

- R. Kandepu, L. Imsland, and B.A. Foss (2008) "*Constrained state estimation using the unscented Kalman filter*" 16th Mediterranean Conference on Control and Automation