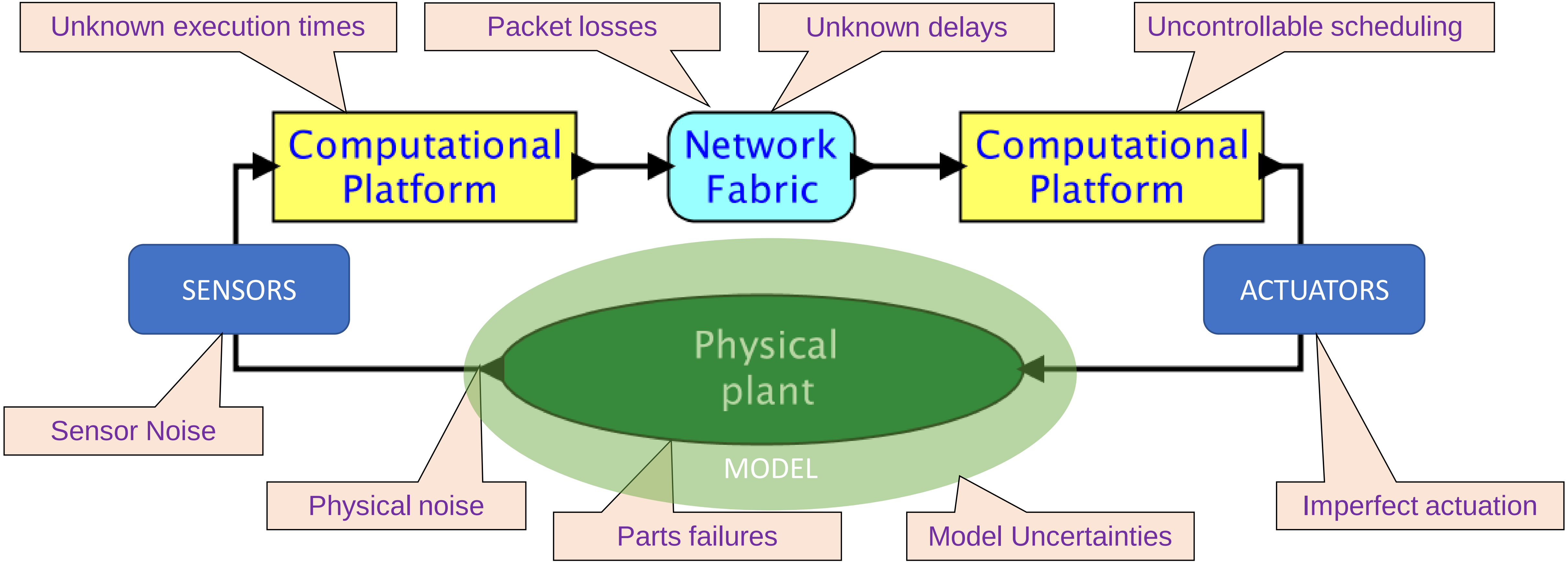
Kalman Filters

Maqsood

BIG PICTURE: CPS



“Essentially, all models are wrong, but some are useful.”

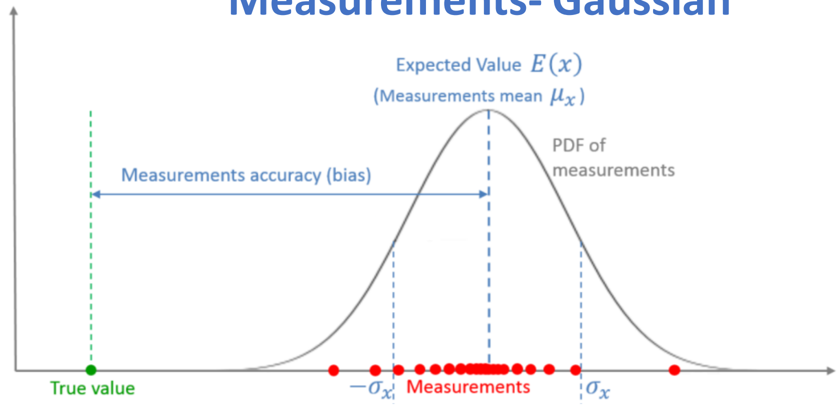
“A CPS system is only as good as the Sensors”

“Everything is an approximation”

Background Knowledge

- Measurement is a **random variable**, described by the **Probability Density Function (PDF)**.
- Measurements mean is the **Expected Value** of the random variable.
- Offset between the measurements mean and the true value is the **measurements accuracy** (or **bias** or **measurement error)**.
- The dispersion of the distribution is known as **precision** or (**measurement noise** or **measurement uncertainty)**.

Measurements- Gaussian



Mean

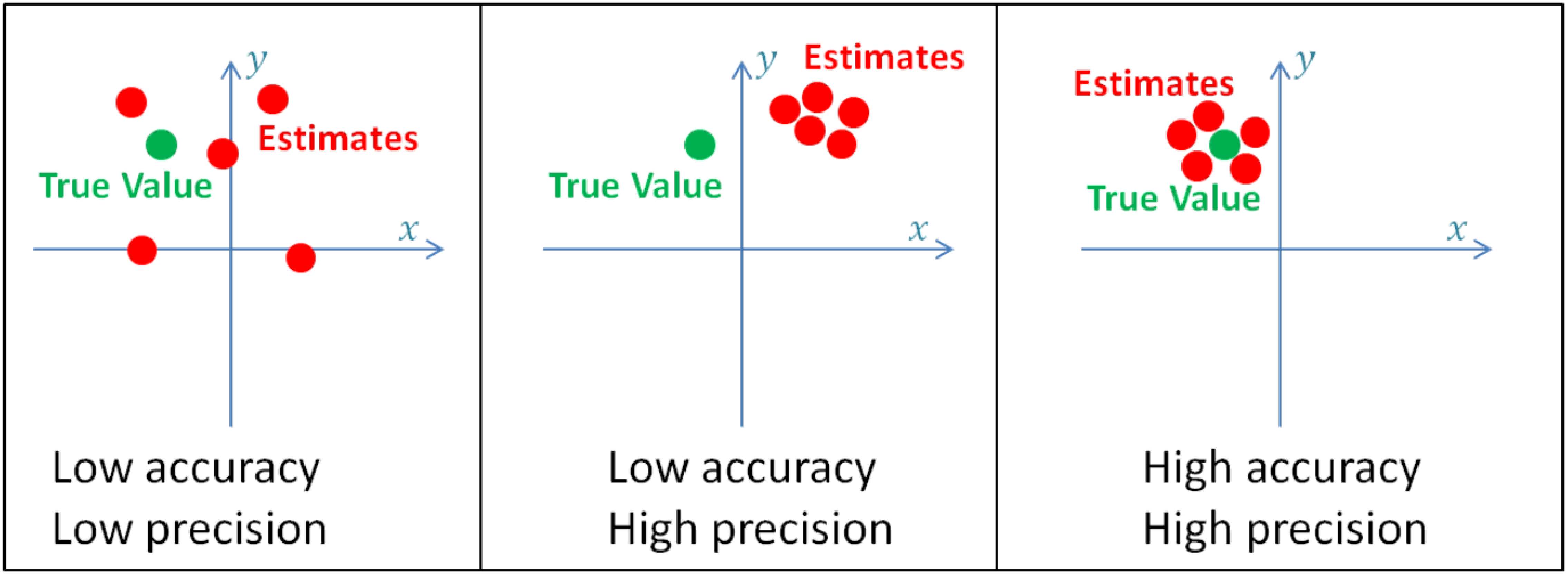
$$E(X) = \mu_X = \frac{1}{N}\sum_{n=1}^{N}(x_n$$

Variance

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

Gaussian PDF $f\left(x;\mu,\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$

Accuracy & Precision



Kalman Filters

What is a Kalman Filter:

- A Kalman filter is an ***optimal estimator*** – i.e. infers parameters of interest from *indirect, inaccurate and uncertain observations*. It is recursive so that *new measurements can be processed as they arrive*.

Optimal in what sense:

- **If Noise is Gaussian:** the Kalman filter minimizes the mean square error of the estimated parameters.
- **If Noise is NOT Gaussian:** Kalman filter is still the best *linear* estimator. Non-linear estimators may be better.
  - **Gauss-Markov Theorem** – Optimal among all Linear, Unbiased Estimators
  - **Rao–Blackwell theorem** – Optimal among Non-linear Estimators with Gaussian Noise

Kalman Filters…

An Estimator: Optimal under Linear or Gaussian and is **On-Line.**

Why is Kalman Filtering so popular:

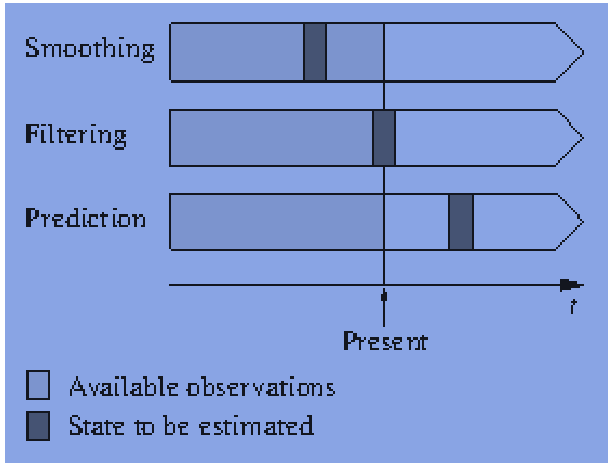
- Good results in practice due to optimality and structure.
- Convenient form for online real time processing.
- Easy to formulate and implement given a basic understanding.
- Measurement equations need not be inverted.

Why use the word "Filter"

- The process of finding the "best estimate" from noisy data amounts to "filtering out" the noise.
- Kalman filter doesn't just clean up the data measurements, but also projects them onto the state estimate.

Kalman Filter: Smoothing, Filtering, Prediction

- Additional Reading and Acknowledgements:
  - https://www.kalmanfilter.net/
  - https://www.mathworks.com/videos/series/understanding-kalman-filters.html
  - http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf
- Real-time optimal estimation is desired when new data Arrives
  - Smoothing (Take advantage of noise reduction)
  - Filtering
  - Prediction (extrapolate based on model)
    - Applications: controllers, tracking, etc.

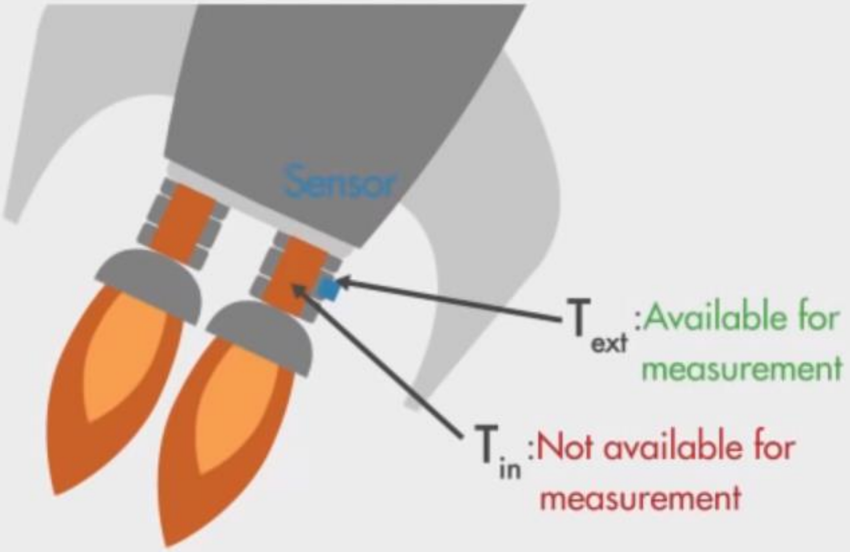


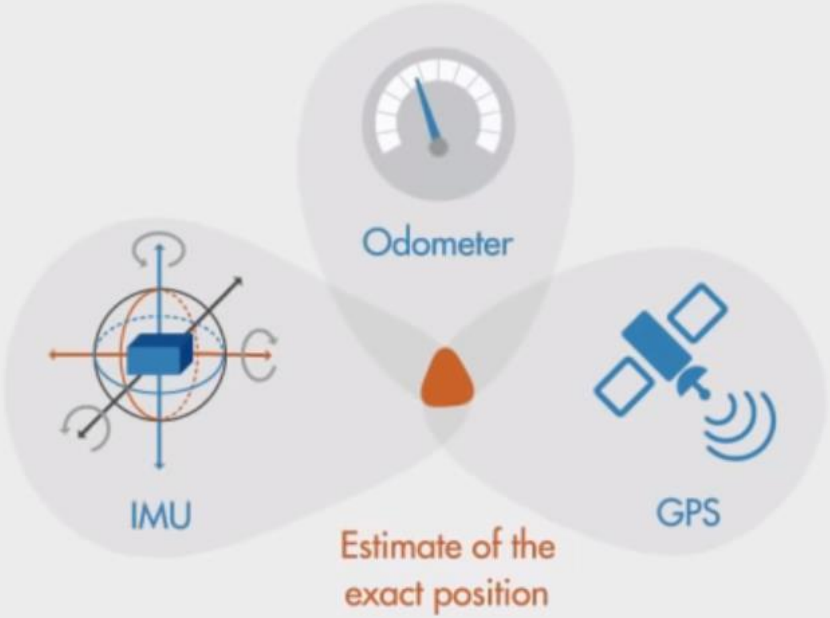
Kalman Filter: Mechanism

- **Required:** 1. System Model and 2. Observations.
- Model may be uncertain, Measurements may be Noisy
- **Prediction-correction framework:** Optimal combination of system model and observations

Kalman filter is used when:

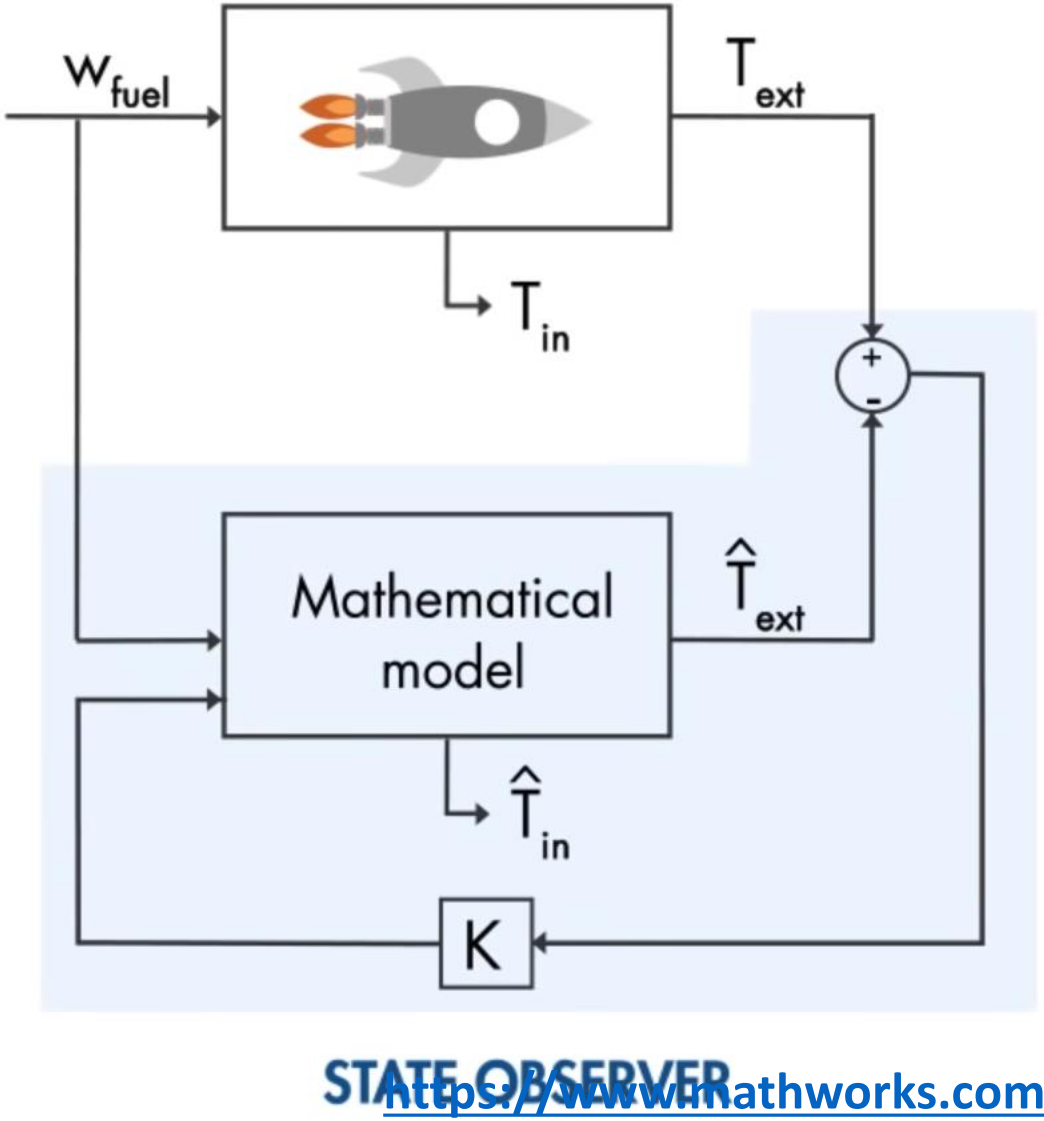
- The variables of interest can only be measured indirectly.

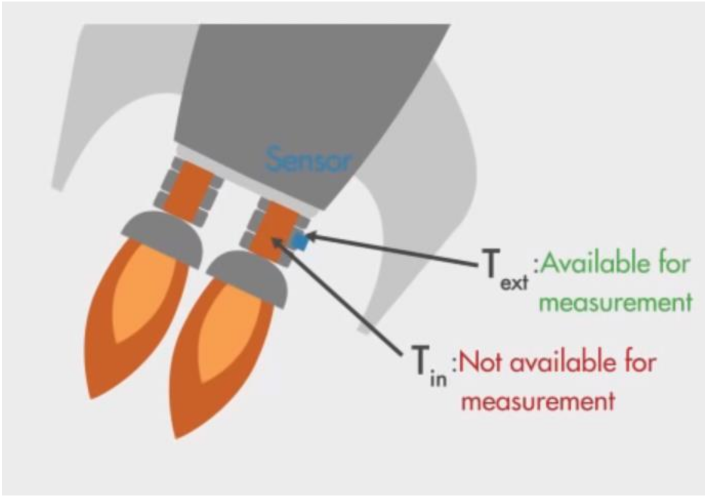


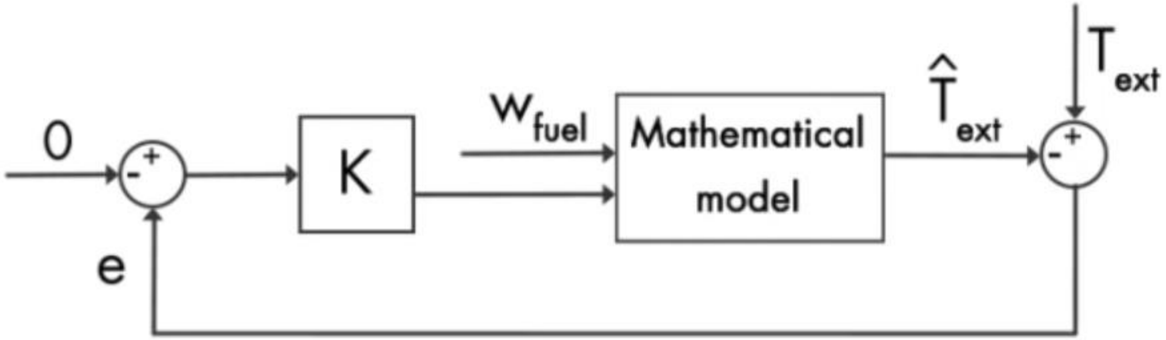


- Measurements are available from various sensors but might be subject to noise.

Intuition: State Observer: Estimating state of a Rocket



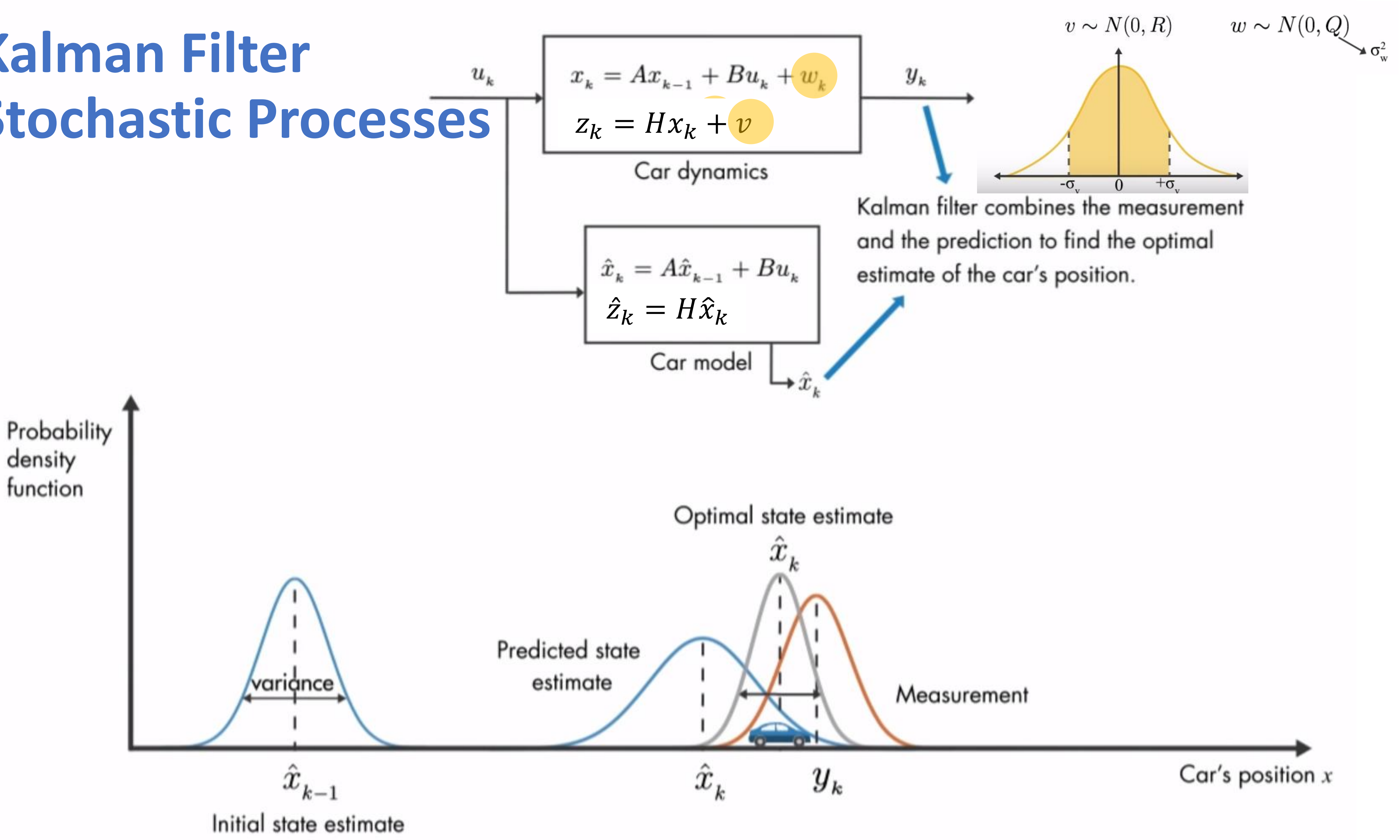




$$e = \text{error} = T_{ext} - \hat{T}_{ext}$$

https://www.mathworks.com/videos/series/understanding-kalman-filters.html

Kalman Filter
Stochastic Processes



$$x_k = Ax_{k-1} + Bu_k + w_k$$

$$z_k = Hx_k + v$$

Car dynamics

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

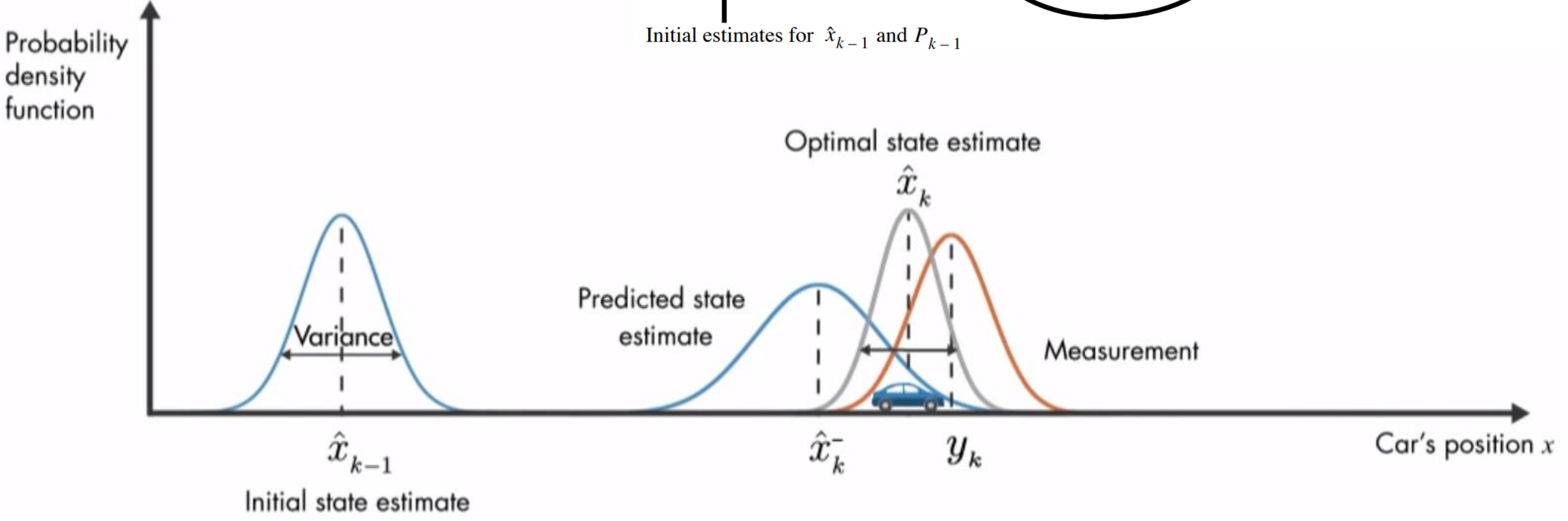
(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}



Simple Example: Data Acquisition Intuition

- Measurement of a single point z_1
- Variance σ_1^2 (uncertainty σ_1)
- Best estimate of true position $\hat{x}_1 = z_1$
- Uncertainty in best estimate $\hat{\sigma}_1^2 = \sigma_1^2$

- Second measurement z_2, variance σ_2^2
- Best estimate of true position?

Minimum Variance Estimator

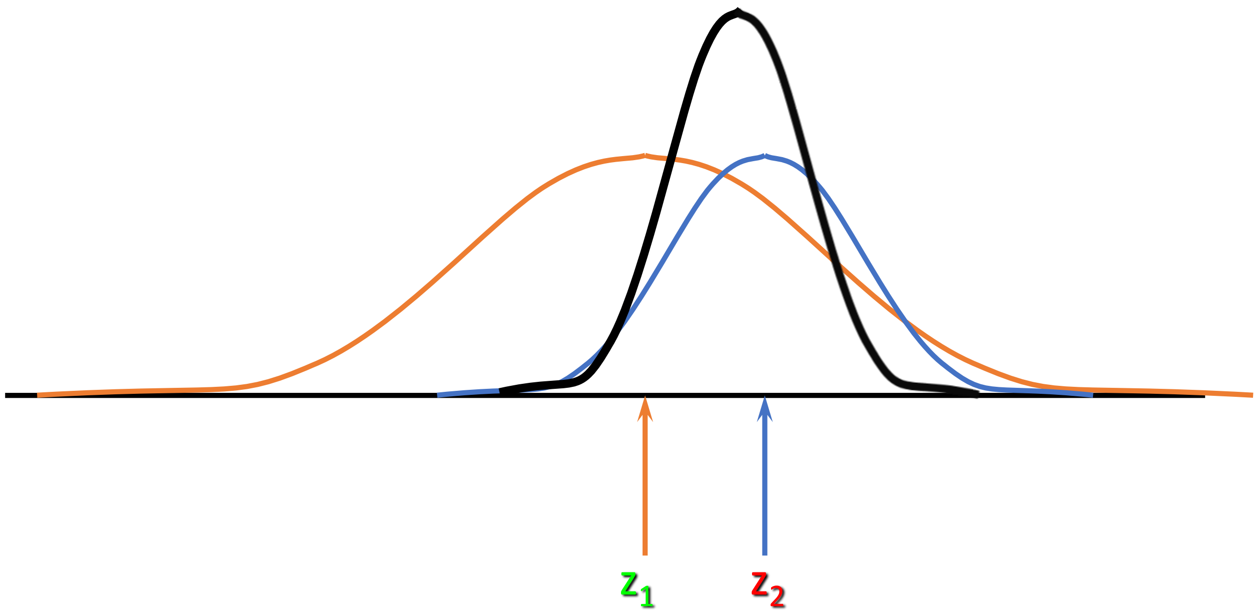
- Second measurement z_2, variance σ_2^2
- Best estimate of true position: weighted average

$$\hat{x}_2 = \frac{\frac{1}{\sigma_1^2} z_1 + \frac{1}{\sigma_2^2} z_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}}$$

$$= \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(z_2 - \hat{x}_1)$$

- Uncertainty in best estimate

$$\hat{\sigma}_2^2 = \frac{1}{\frac{1}{\hat{\sigma}_1^2} + \frac{1}{\sigma_2^2}}$$



State Space Representation

- For "standard" Kalman filtering, everything must be linear

System model: $x_k = Ax_{k-1} + Bu + w$

  - The matrix A is *state transition matrix*
  - The matrix B is *input matrix*
  - The vector *w* represents *additive noise*, assumed to have covariance *Q*

Measurement model: $z_k = Hx_k + v$

  - Matrix *C* is *measurement matrix*
  - The vector v is *measurement noise*, assumed to have covariance *R*

- Best estimate of state $\hat{x}$ with covariance *P*

Further Reading: http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

Prediction/Correction

prediction of new state based on passed state x'_k
predicted observation z'_k
new observation z_k
new estimate of state $\hat{x}_k$

- **Prediction:** of new state (Ignoring input *u*)

$$
\begin{aligned}
x'_k &= A\hat{x}_{k-1} \\
P'_k &= AP_{k-1}A^{\mathrm{T}} + Q \\
z'_k &= Ax'_k
\end{aligned}
$$

Pk is the error covariance matrix at time k

$$P_k = E\left[e_k e_k^T\right] = E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right]$$

- **Correction:** To Account for new measurements

Kalman Gain: Weighting of process model vs. measurements

$$
\begin{aligned}
K_k &= P'_k H^{\mathrm{T}}\left(HP'_k H^{\mathrm{T}} + R\right)^{-1} \\
\hat{x}_k &= x'_k + K_k(z_k - H x'_k) \\
P_k &= (I - K_k H)P'_k
\end{aligned}
$$

Kalman Filter Definition: For 1-D Case

$$K_n = \frac{Uncertainty \quad in \quad Estimate}{Uncertainty \quad in \quad Estimate \quad + \quad Uncertainty \quad in \quad Measurement}$$

Where:
$p_{n,n-1}$ is the extrapolated estimate uncertainty
r_n is the measurement uncertainty

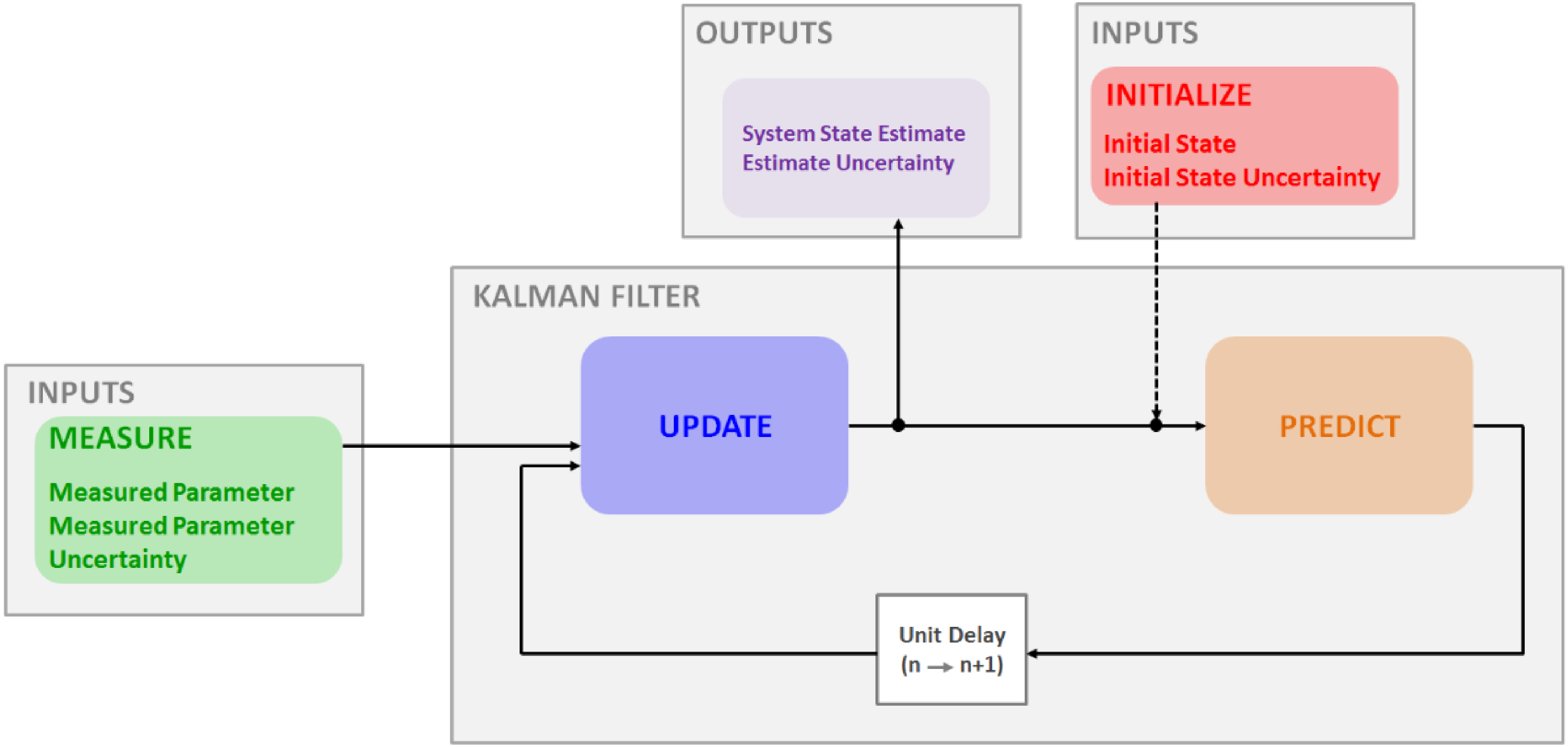
$$p_{n,n} = \ (1 - K_n)\, p_{n,n-1}$$

Where:
K_n is the Kalman Gain
$p_{n,n-1}$ is the estimate uncertainty that was calculated during the previous filter estimation
$p_{n,n}$ is the estimate uncertainty of the current sate

Equation	Equation Name	Alternative names used in the literature
$\hat{x}_{n,n} = \ \hat{x}_{n,n-1} + K_n \left(z_n - \hat{x}_{n,n-1} \right)$	State Update	Filtering Equation
$\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1} + \Delta t \hat{\dot{x}}_{n-1,n-1}$ $\hat{\dot{x}}_{n,n-1} = \hat{\dot{x}}_{n-1,n-1}$ (For constant velocity dynamics)	State Extrapolation	Predictor Equation Transition Equation Prediction Equation Dynamic Model State Space Model
$K_n = \frac{p_{n,n-1}}{p_{n,n-1}+r_n}$	Kalman Gain	Weight Equation
$p_{n,n} = \ (1 - K_n)\, p_{n,n-1}$	Covariance Update	Corrector Equation
$p_{n,n-1} = p_{n-1,n-1}$ (For constant dynamics)	Covariance Extrapolation	Predictor Covariance Equation

Further Reading: https://www.kalmanfilter.net/kalman1d.html

Kalman Filter: Systematic View



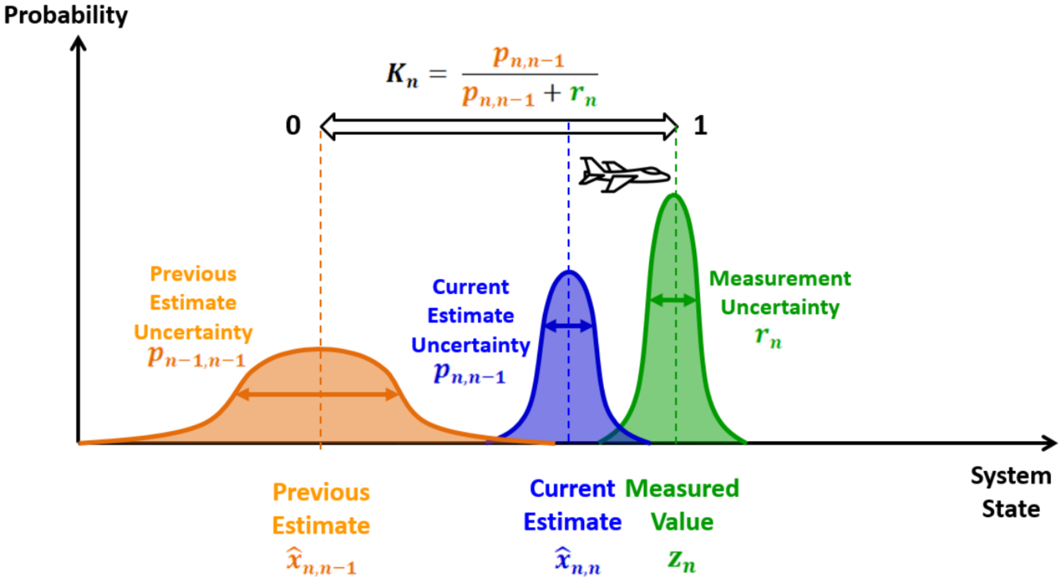
The Kalman Gain Intuition: For 1D Case

The estimate of the current state = Predicted value of the current state + Factor x (Measurement − Predicted value of the current state)

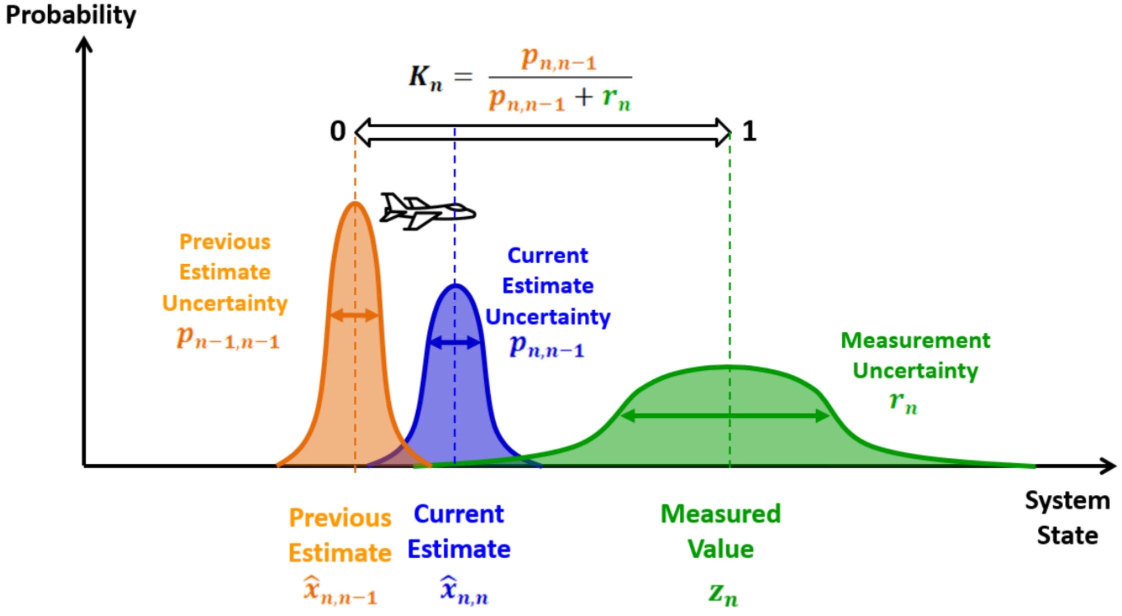
$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n\left(z_n - \hat{x}_{n,n-1}\right)$$

$$p_{n,n} = (1 - K_n)p_{n,n-1}$$

HIGH KALMAN GAIN



LOW KALMAN GAIN



Example 1: Estimating Temperature of Liquid in Tank Numerical Example

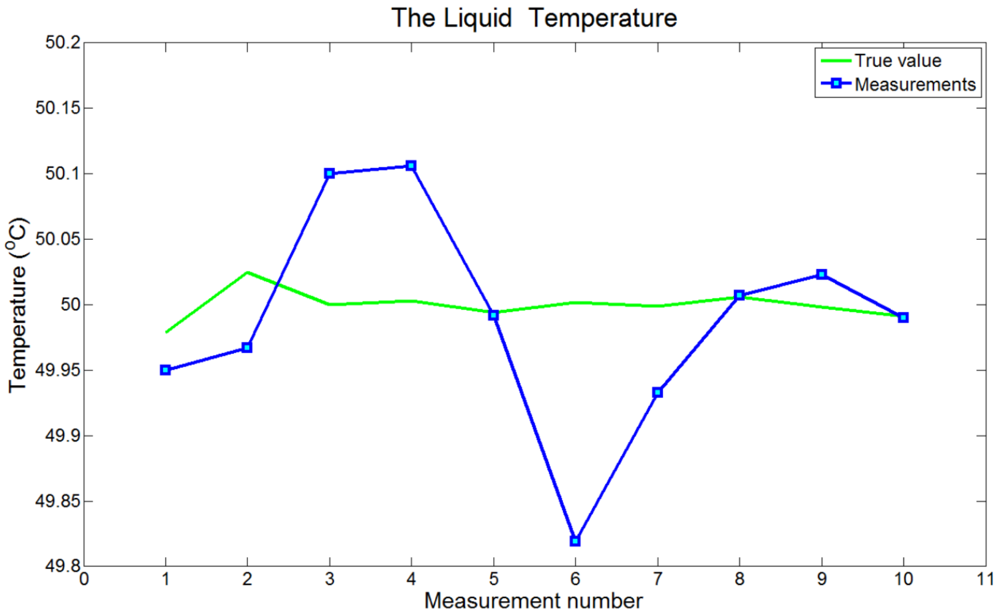
$$x_n = T + w_n$$

where:

T is the constant temperature

w_n is a random process noise with variance q

- Let us assume the true temperature of 50 degrees Celsius.
- We think that we have an accurate model, thus we set the process noise variance (q) to 0.0001.
- The measurement error (standard deviation) is 0.1 degrees Celsius.
- The measurements are taken every 5 seconds.
- The true liquid temperature at the measurement points is: 49.979^oC, 50.025^oC, 50^oC, 50.003^oC, ^oC, and 49.991^oC.
- The set of measurements is: 49.95^oC, 49.967^oC, 50.1^oC, 50.106^oC, 49.992^oC, 49.819^oC, 49.933^o



For Further Details: https://www.kalmanfilter.net/kalman1d.html

$$x_n = T + w_n$$

where:

T is the constant temperature

w_n is a random process noise with variance q

ITERATION ZERO

- **INITIALIZATION**

$$\hat{x}_{0,0} = 10^oC$$

$$p_{0,0} = 100^2 = 10,000$$

- **PREDICTION**

$$\hat{x}_{1,0} = 10^oC$$

$$p_{1,0} = p_{0,0} + q = 10000 + 0.0001 = 10000.0001$$

FIRST ITERATION

- **STEP 1 - MEASURE**

$$z_1 = 49.95^oC$$

$$r_1 = 0.01$$

- **STEP 2 - UPDATE**

$$K_1 = \frac{p_{1,0}}{p_{1,0} + r_1} = \frac{10000.0001}{10000.0001 + 0.01} = 0.999999$$

$$\hat{x}_{1,1} = \hat{x}_{1,0} + K_1 (z_1 - \hat{x}_{1,0}) = 10 + 0.999999\,(49.95 - 10) = 49.95^oC$$

$$p_{1,1} = (1 - K_1)\, p_{1,0} = (1 - 0.999999)\, 10000.0001 = 0.01$$

- **STEP 3 - PREDICT**

$$\hat{x}_{2,1} = \hat{x}_{1,1} = 49.95^oC$$

$$p_{2,1} = p_{1,1} + q = 0.01 + 0.0001 = 0.0101$$

SECOND ITERATION

- **STEP 1 - MEASURE**

$$z_2 = 49.967^oC$$

$$r_2 = 0.01$$

- **STEP 2 - UPDATE**

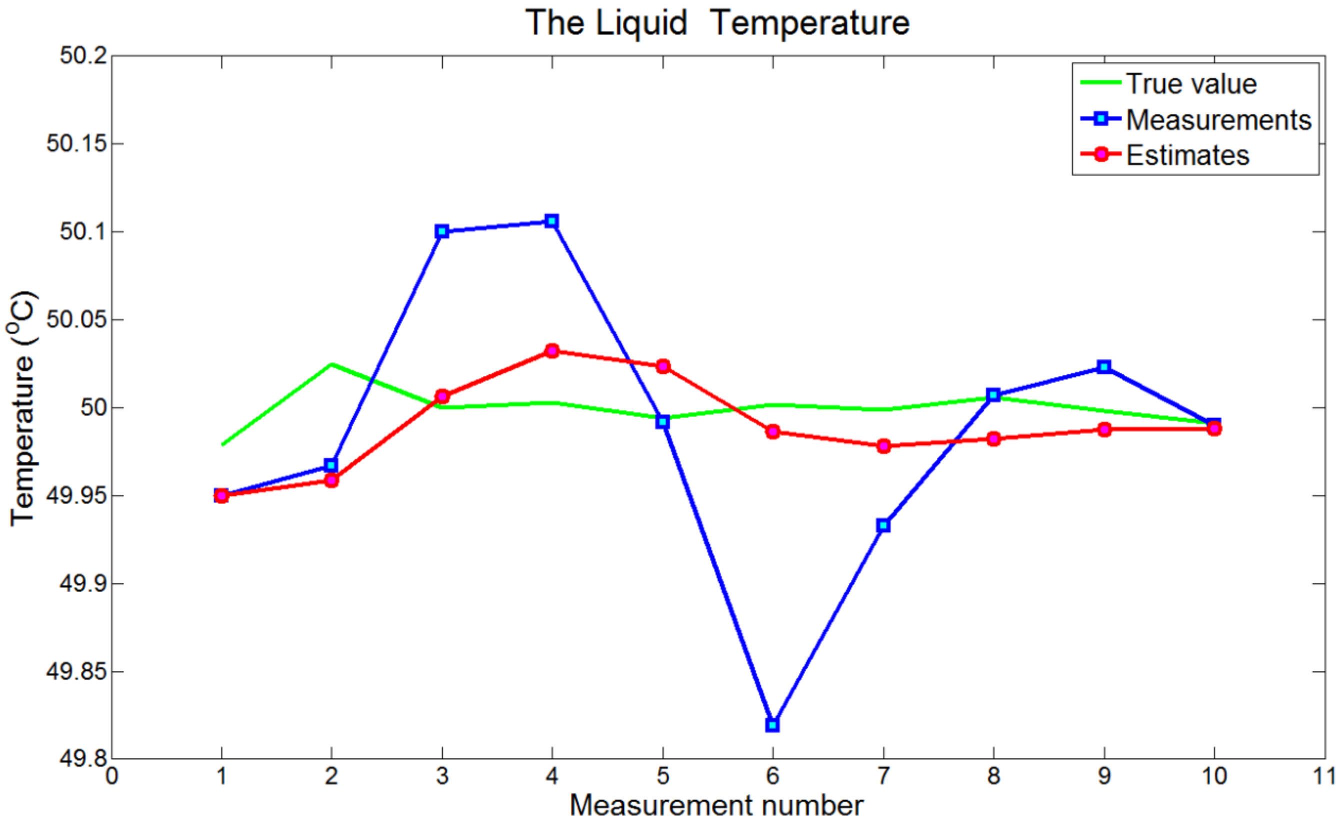
$$K_2 = \frac{p_{2,1}}{p_{2,1} + r_2}$$

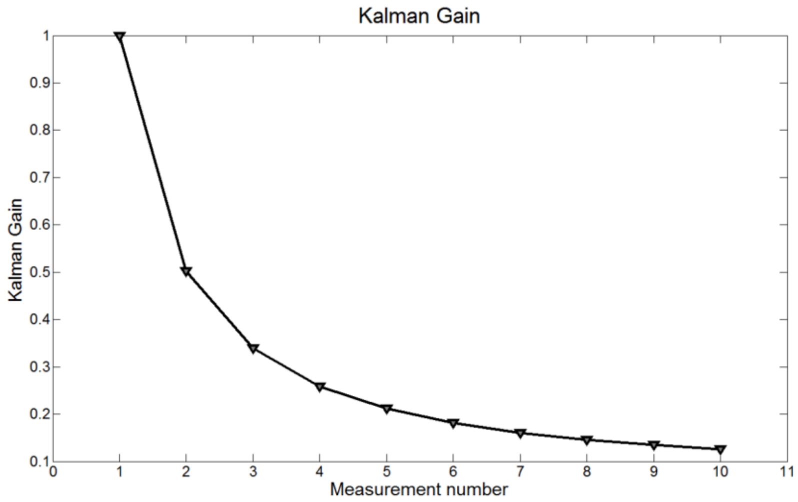
$$\hat{x}_{2,2} = \hat{x}_{2,1} + K_2 (z_2 - \hat{x}_{2,1})$$

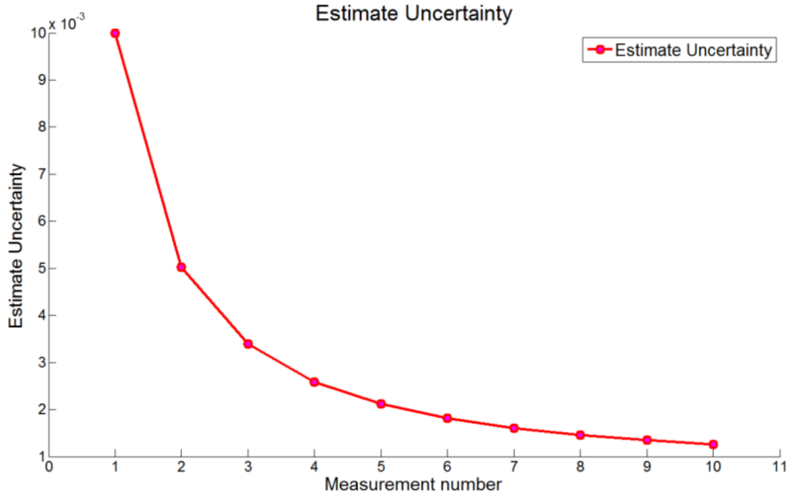
$$p_{2,2} = (1 - K_2)\, p_{2,1}$$

- **STEP 3 - PREDICT**

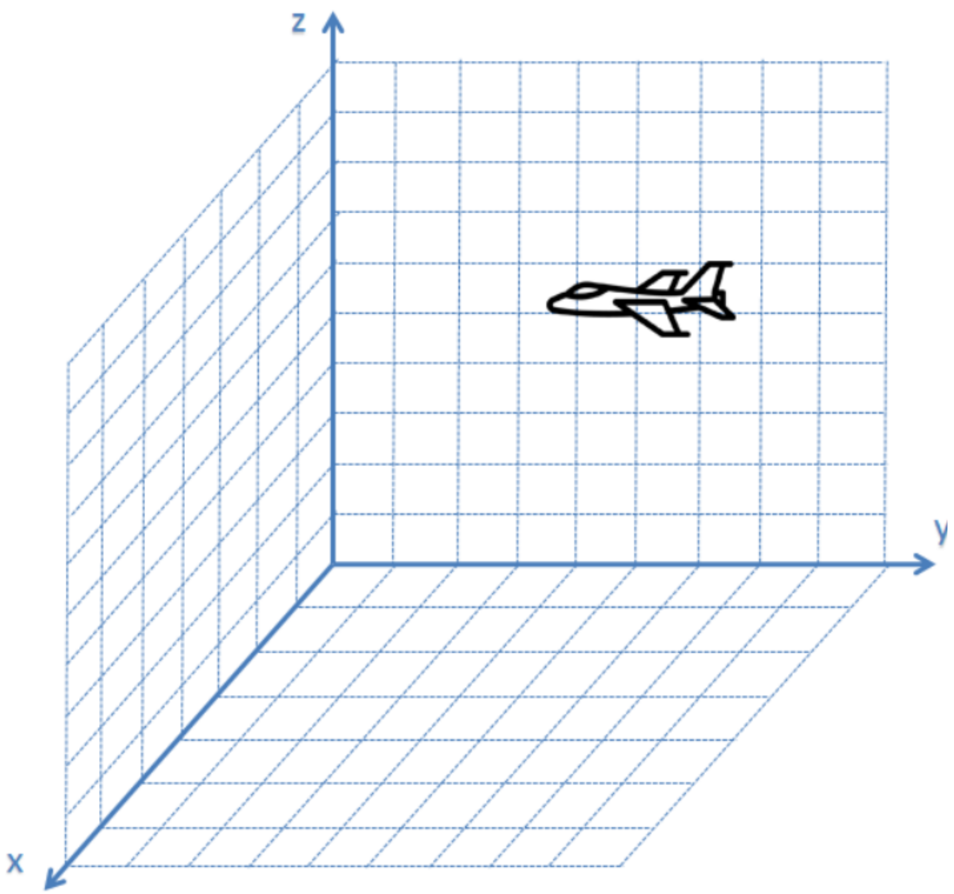
Estimating Temperature of Liquid in Tank







EXAMPLE 2: AIRPLANE CONSTANT ACCELERATION MODEL
Determining The State Space Mode



Estimated State Vector

$$\hat{\boldsymbol{x}}_n = \begin{bmatrix} \hat{x}_n \\ \hat{y}_n \\ \hat{z}_n \\ \hat{\dot{x}}_n \\ \hat{\dot{y}}_n \\ \hat{\dot{z}}_n \end{bmatrix}$$

Control vector

$$\hat{\boldsymbol{u}}_n = \begin{bmatrix} \hat{\ddot{x}}_n \\ \hat{\ddot{y}}_n \\ \hat{\ddot{z}}_n \end{bmatrix}$$

$$x = x_0 + v_0\Delta t + \frac{1}{2}a\Delta t^2$$

Where:

x is the target's position

x_0 is the target's initial position

v_0 is the target's initial velocity

a is the target's acceleration

Δt is the time interval (5 seconds in our example)

$$\begin{cases} x = x_0 + v_{x0}\Delta t + \frac{1}{2}a_x\Delta t^2 \\ y = y_0 + v_{y0}\Delta t + \frac{1}{2}a_y\Delta t^2 \\ z = z_0 + v_{z0}\Delta t + \frac{1}{2}a_z\Delta t^2 \end{cases}$$

State transition matrix

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Control Matrix

$$\boldsymbol{G} = \begin{bmatrix} 0.5\Delta t^2 & 0 & 0 \\ 0 & 0.5\Delta t^2 & 0 \\ 0 & 0 & 0.5\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix}$$

The state extrapolation equation is:

$$\boldsymbol{\hat{x}}_{\boldsymbol{n+1,n}} = \boldsymbol{F}\boldsymbol{\hat{x}}_{\boldsymbol{n,n}} + \boldsymbol{G}\boldsymbol{\hat{u}}_{\boldsymbol{n,n}}$$

$$\begin{bmatrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{z}_{n+1,n} \\ \hat{\dot{x}}_{n+1,n} \\ \hat{\dot{y}}_{n+1,n} \\ \hat{\dot{z}}_{n+1,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{z}_{n,n} \\ \hat{\dot{x}}_{n,n} \\ \hat{\dot{y}}_{n,n} \\ \hat{\dot{z}}_{n,n} \end{bmatrix} + \begin{bmatrix} 0.5\Delta t^2 & 0 & 0 \\ 0 & 0.5\Delta t^2 & 0 \\ 0 & 0 & 0.5\Delta t^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \hat{\ddot{x}}_{n,n} \\ \hat{\ddot{y}}_{n,n} \\ \hat{\ddot{z}}_{n,n} \end{bmatrix}$$

The matrix multiplication results:

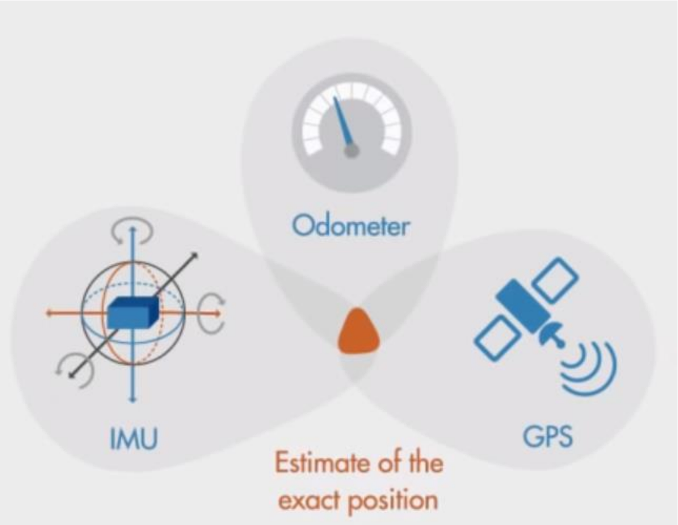
$$\begin{cases} \hat{x}_{n+1,n} = \hat{x}_{n,n} + \hat{\dot{x}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{x}}_{n,n}\Delta t^2 \\ \hat{y}_{n+1,n} = \hat{y}_{n,n} + \hat{\dot{y}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{y}}_{n,n}\Delta t^2 \\ \hat{z}_{n+1,n} = \hat{z}_{n,n} + \hat{\dot{z}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{z}}_{n,n}\Delta t^2 \\ \hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n}\Delta t \\ \hat{\dot{y}}_{n+1,n} = \hat{\dot{y}}_{n,n} + \hat{\ddot{y}}_{n,n}\Delta t \\ \hat{\dot{z}}_{n+1,n} = \hat{\dot{z}}_{n,n} + \hat{\ddot{z}}_{n,n}\Delta t \end{cases}$$

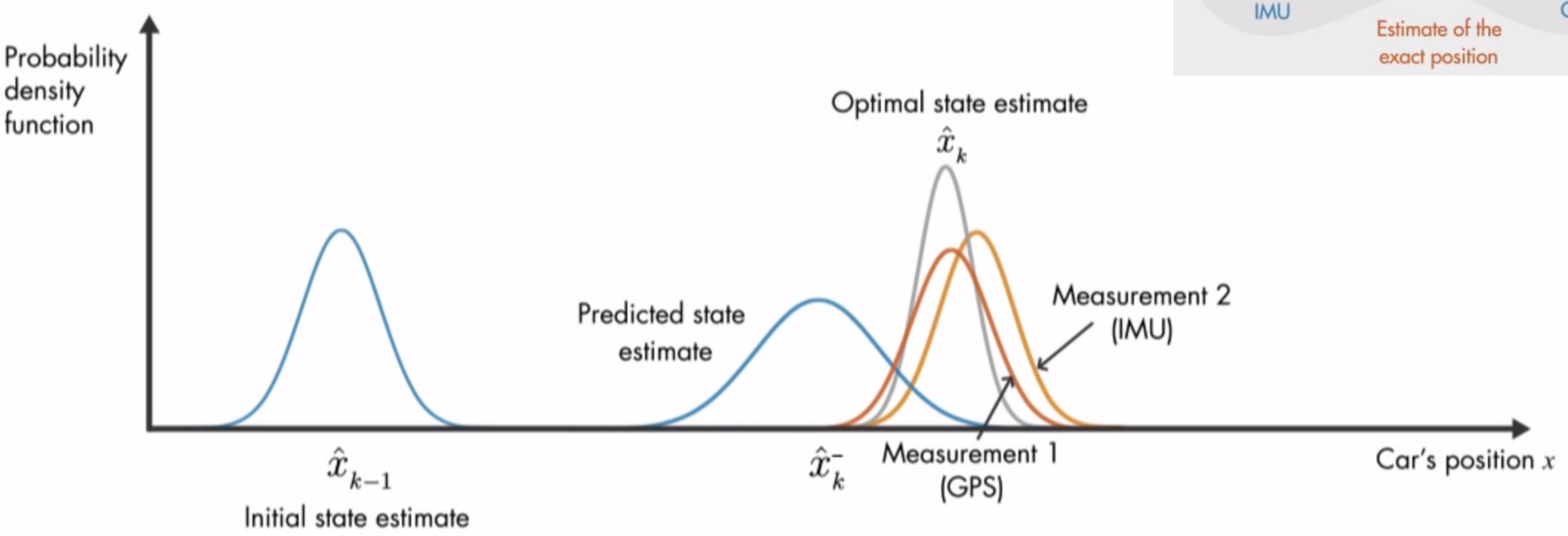
Sensor Fusion

Sensor fusion

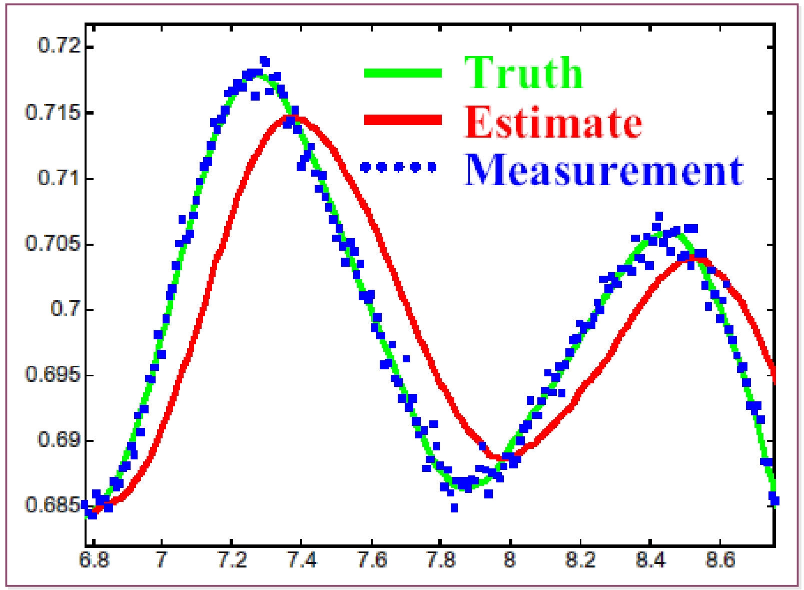
$$\hat{x}_{k\,[1x1]} = \hat{x}^-_{k\,[1x1]} + K_{k\,[1x2]}\left(y_{k\,[2x1]} - C_{[2x1]}\,\hat{x}^-_{k\,[1x1]}\right)$$

Vector of multiple measurements



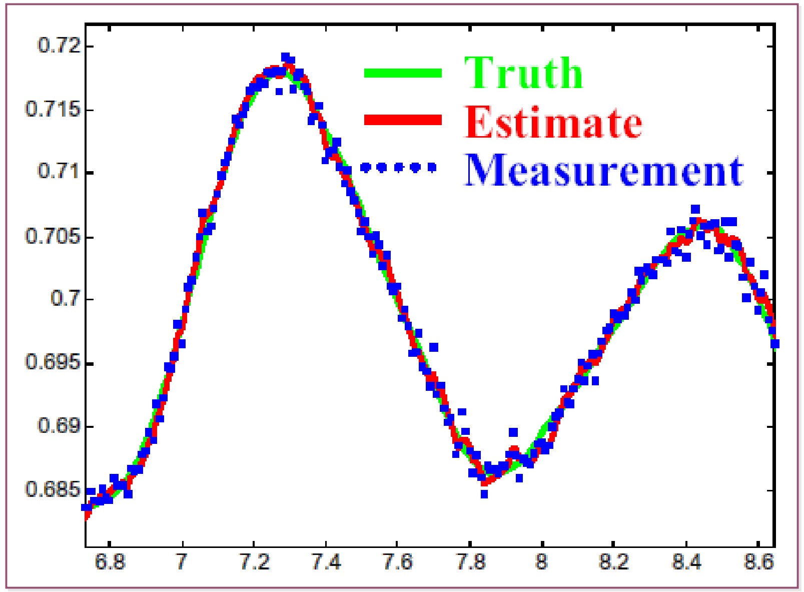


Comparison: Position-Only vs Position-Velocity Model



Position-Only Model

E.g., GPS position measurements

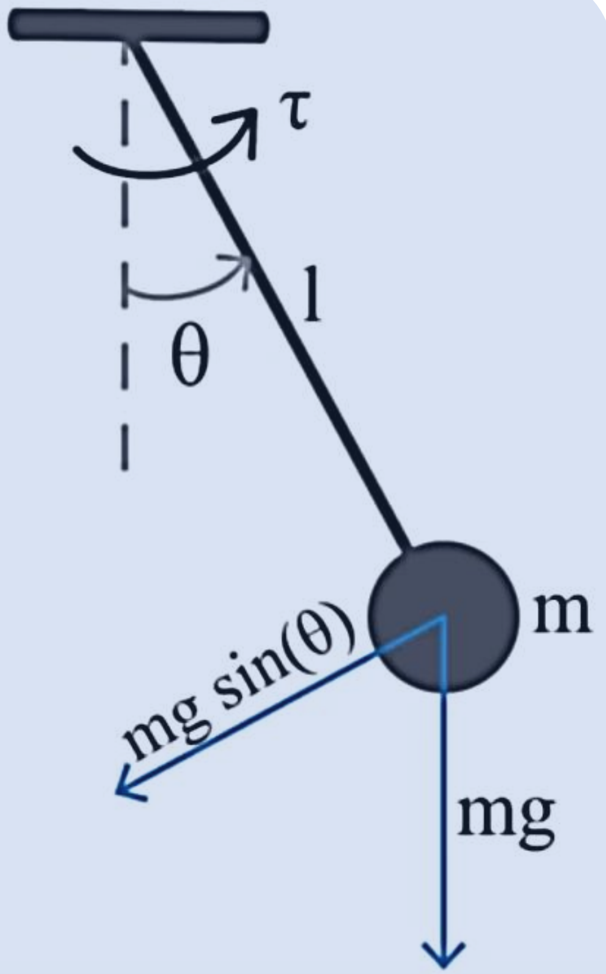


Position-Velocity Model

E.g., GPS position + Odometer speed

[Welch & Bishop]

Example 3: Pendulum Equation of Motion
Determining a Linear State Space Representation



The simple pendulum system with no friction can be represented by:

Dynamic Model

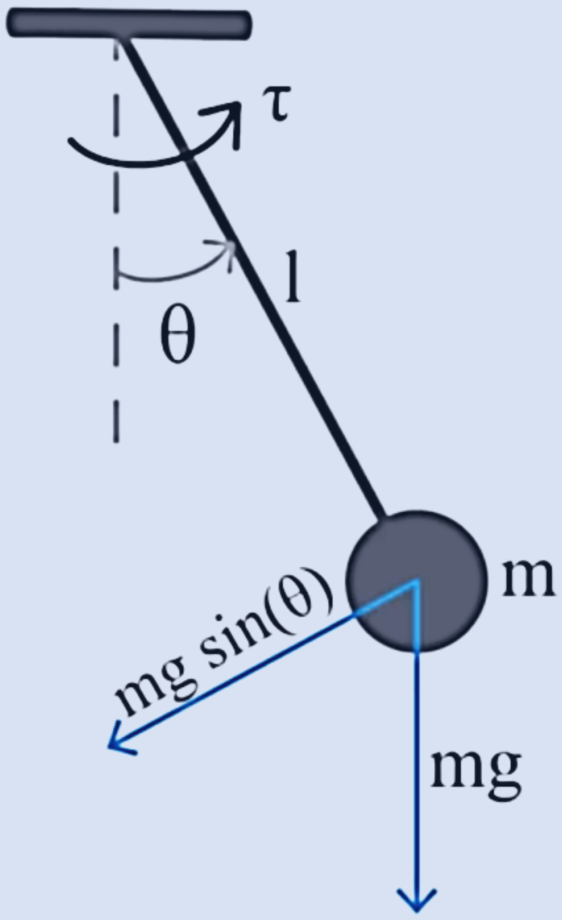
$$I\frac{d^2\theta}{dt^2} + mgl\sin(\theta) = \tau \quad \text{where} \quad I = ml^2$$

and rearranged as:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\sin(\theta) = \frac{1}{ml^2}\tau \quad \Rightarrow \quad \frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = \frac{1}{ml^2}\tau$$

Non-Linear **For Small Angles** **Linear**

Pendulum Equation of Motion



Substituting τ by the input vector u, the linearized system can be represented in state space form as follows:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = \frac{1}{ml^2}u \qquad \text{where} \quad u = \tau$$

Defining States

$x_1 = \theta$

$x_2 = \dot{\theta}$

Dynamic Model

$\dot{x}_1 = \dot{\theta} = x_2$

$\dot{x}_2 = \ddot{\theta} = -\frac{g}{l}x_1 + \frac{1}{ml^2}u$

$y = \theta$

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = Ax + Bu$$

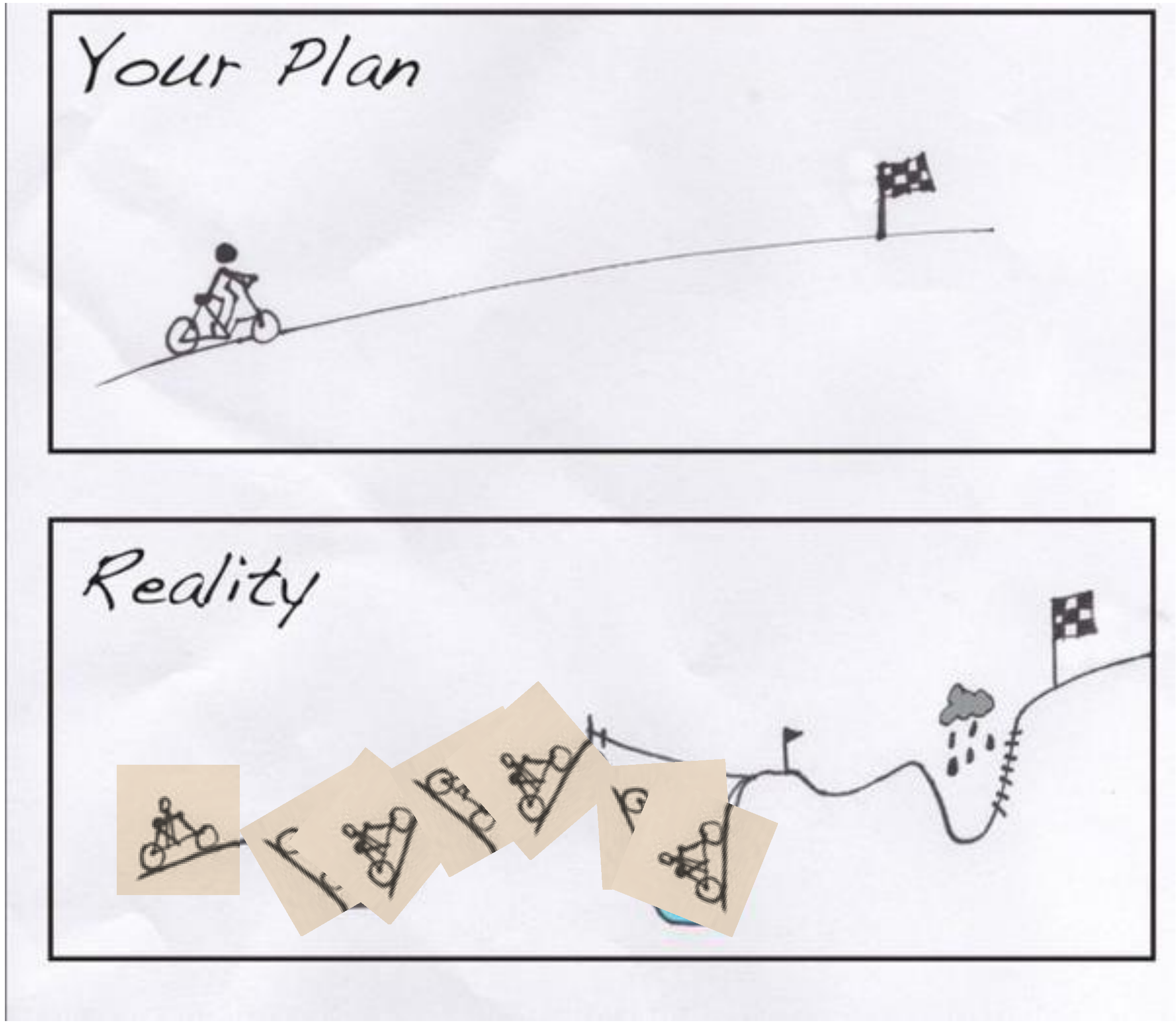
$$y = Cx + Du$$

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$$

System Matrices

$$C = [1 \;\; 0] \qquad D = 0$$

Who said Life is Linear?



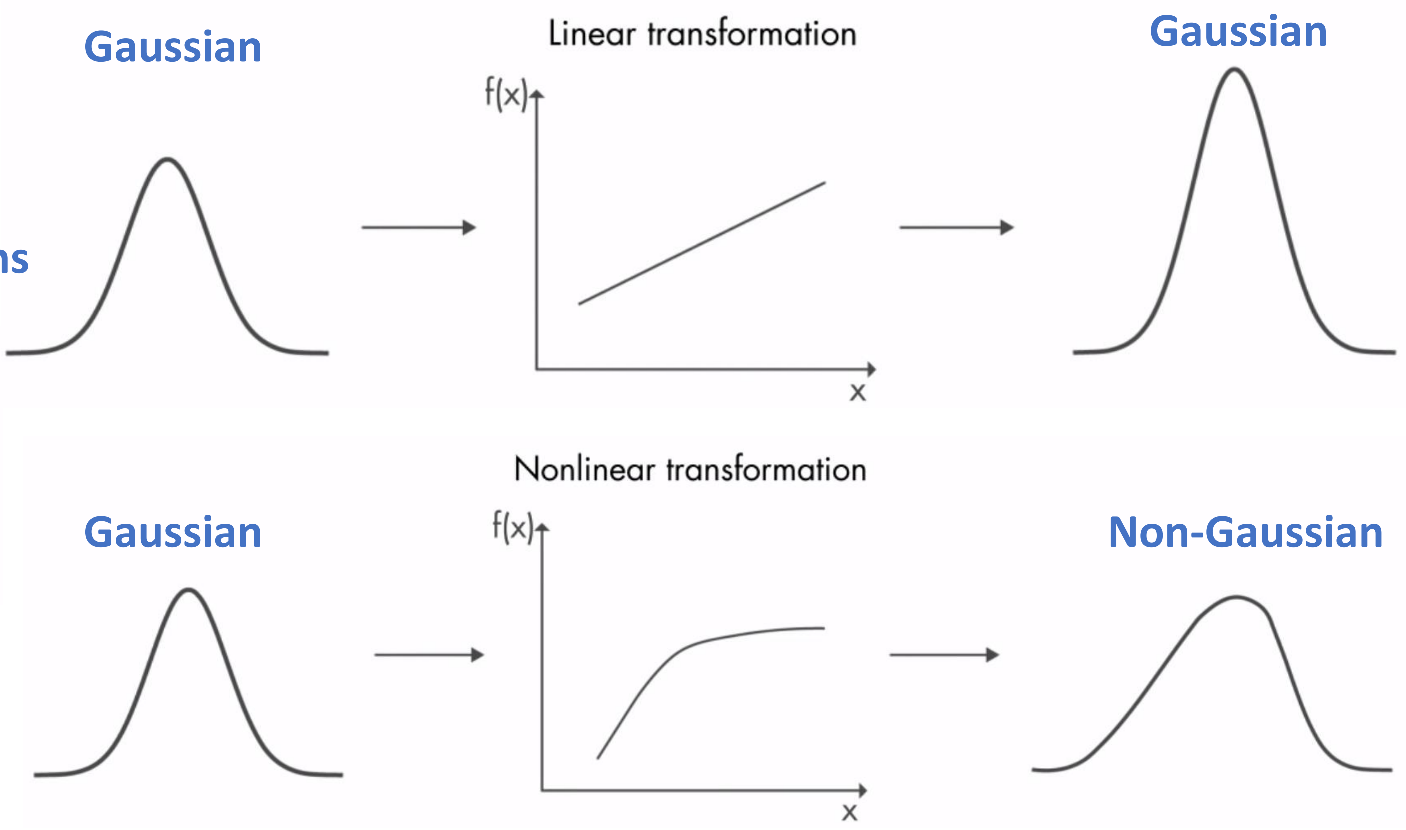
Non-linear Estimation

Kalman Filters are optimal for Linear, Gaussian Systems

$$x_k = f(x_{k-1}, u_k) + w_k$$

$$y_k = g(x_k) + v_k$$

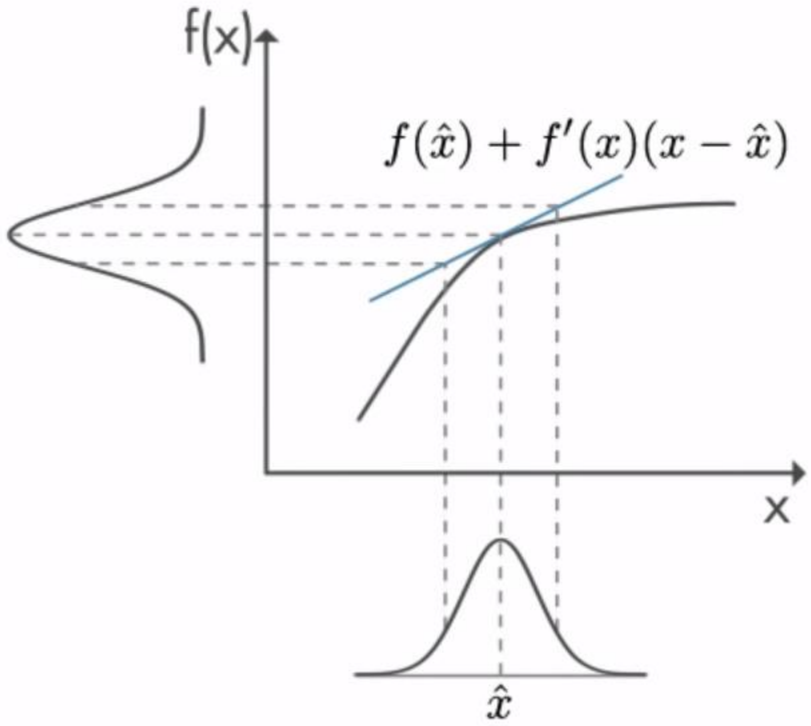
Nonlinear functions



Extended Kalman Filter
Unscented Kalman Filter

Extended Kalman Filters

Nonlinear transformation



System:

$$x_k = f(x_{k-1}, u_k) + w_k$$

$$y_k = g(x_k) + v_k$$

Jacobians:

$$F = \frac{\partial f}{\partial x}\bigg|_{\hat{x}_{k-1}, u_k}$$

$$G = \frac{\partial g}{\partial x}\bigg|_{\hat{x}_k}$$

Linearized system:

$$\Delta x_k \approx F \Delta x_{k-1} + w_k$$

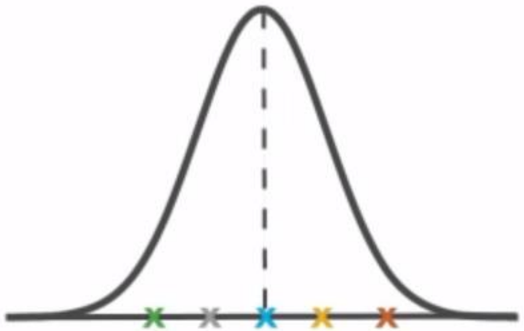
$$\Delta y_k \approx G \Delta x_k + v_k$$

Drawbacks to Using Extended Kalman Filters (EKFs):

- It is difficult to calculate the Jacobians (if they need to be found analytically)
- There is a high computational cost (if the Jacobians can be found numerically)
- EKF only works on systems that have a differentiable model
- EKF is not optimal if the system is highly nonlinear

Unscented Kalman Filters

Gaussian



Sigma points

are selected such that their mean μ and covariance P are the same as the probability distribution.

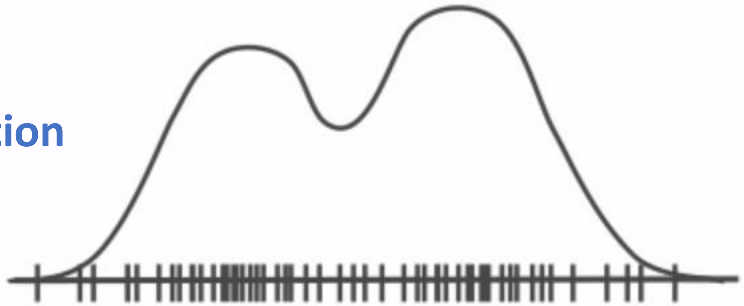
Nonlinear function

Gaussian

The mean μ and covariance P of the transformed sigma points are used to calculate the new state estimate.

Particle Filters

Non-Gaussian Estimation



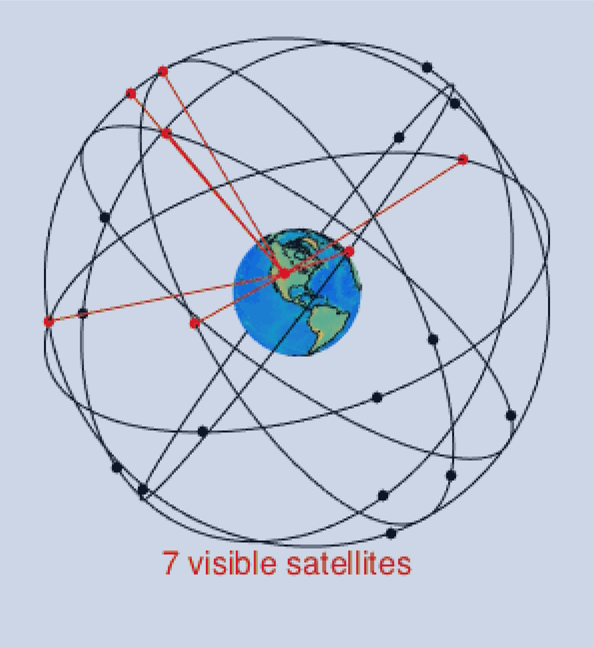
Particles

Comparison

State Estimator	Model	Assumed distribution	Computational cost
Kalman filter (KF)	Linear	Gaussian	Low
Extended Kalman filter (EKF)	Locally linear	Gaussian	Low (if the Jacobians need to be computed analytically) Medium (if the Jacobians can be computed numerically)
Unscented Kalman filter (UKF)	Nonlinear	Gaussian	Medium
Particle filter (PF)	Nonlinear	Non-Gaussian	High

Applications

GPS Tracking

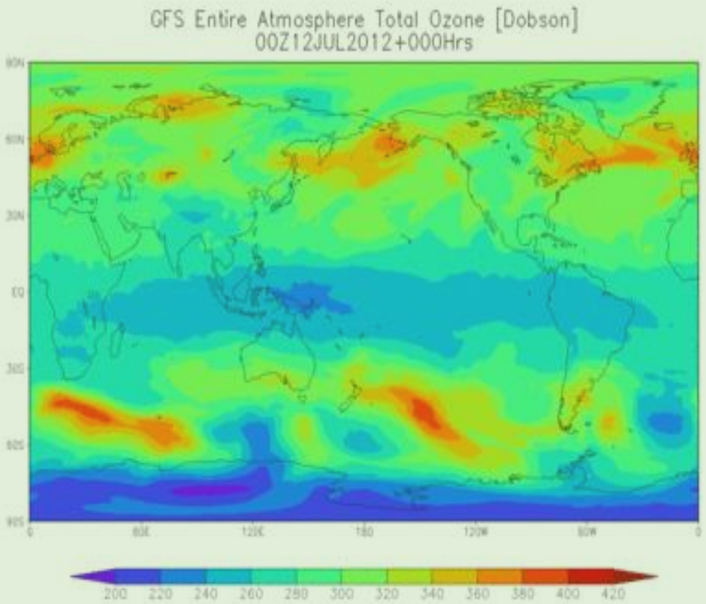


Large Kalman filter system: Including trajectories of 24+ satellites, drift rates and phases of all system clocks, and parameters related to atmospheric propagation delays with time and location

Wind-Mill Tracking

For prolonging life of wind turbines by detecting wind anomalies (wind shear, extreme gusts) utilizing an EKF for regression analysis.

Weather forecasting



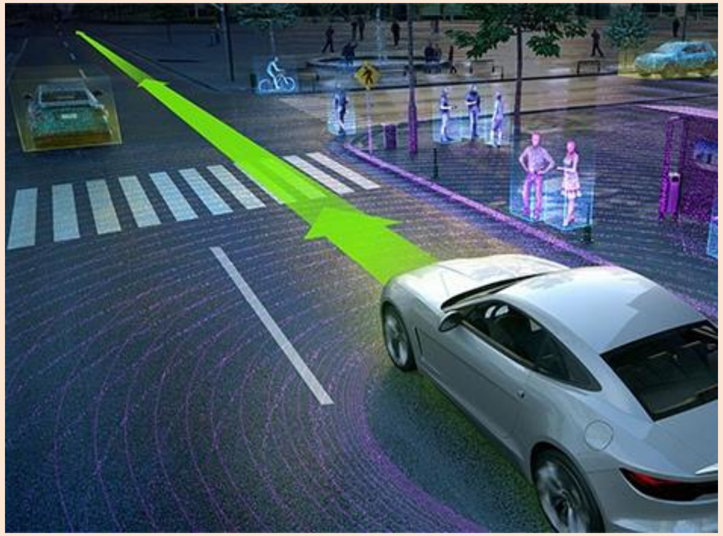
Forecast model. Uses an Ensemble Kalman filter which throws out bad data that would result in a poor forecast.”

Applications

GPS Tracking

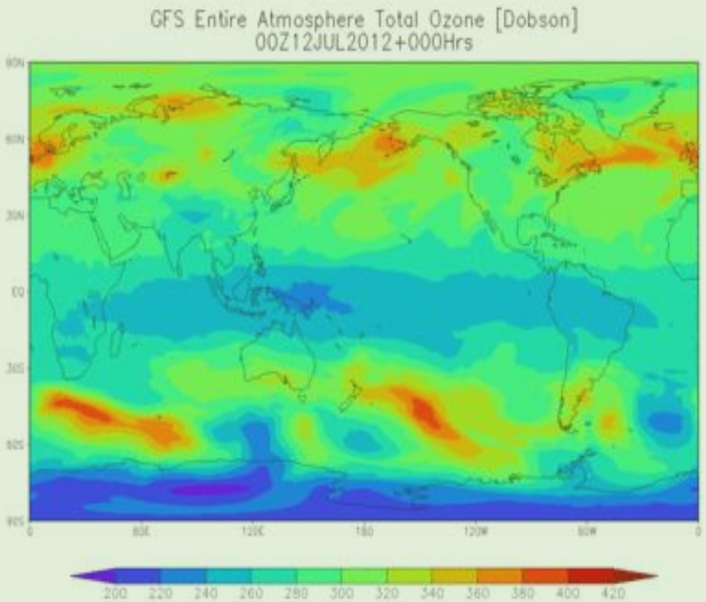
In VR, predictive tracking is used to forecast the position of an object and its trajectory.

Advanced Driver Assistance Systems (ADAS)



Improves efficiency of ADAS and makes vehicle control operations like blind spot detection, stability and traction control, lane departure detection and automatic braking in emergency situations a lot safer and more effective

Weather forecasting



Forecast model. Uses an Ensemble Kalman filter which throws out bad data that would result in a poor forecast.”