Cyber-Physical Systems

Dr. Jonathan Jaramillo

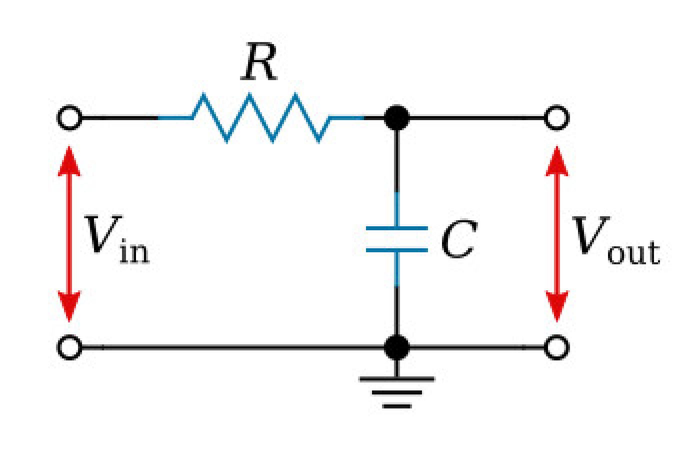
Signal Processing

Signal Processing

- Sensor interpretation
- Control and stability
- Communication
- Diagnostics and fault detection
- Adaption and learning

Types of Signal

- Analog
  - Signal processing is performed with circuit components
- Digital
  - Algorithmic/computational processing requires real time computational systems

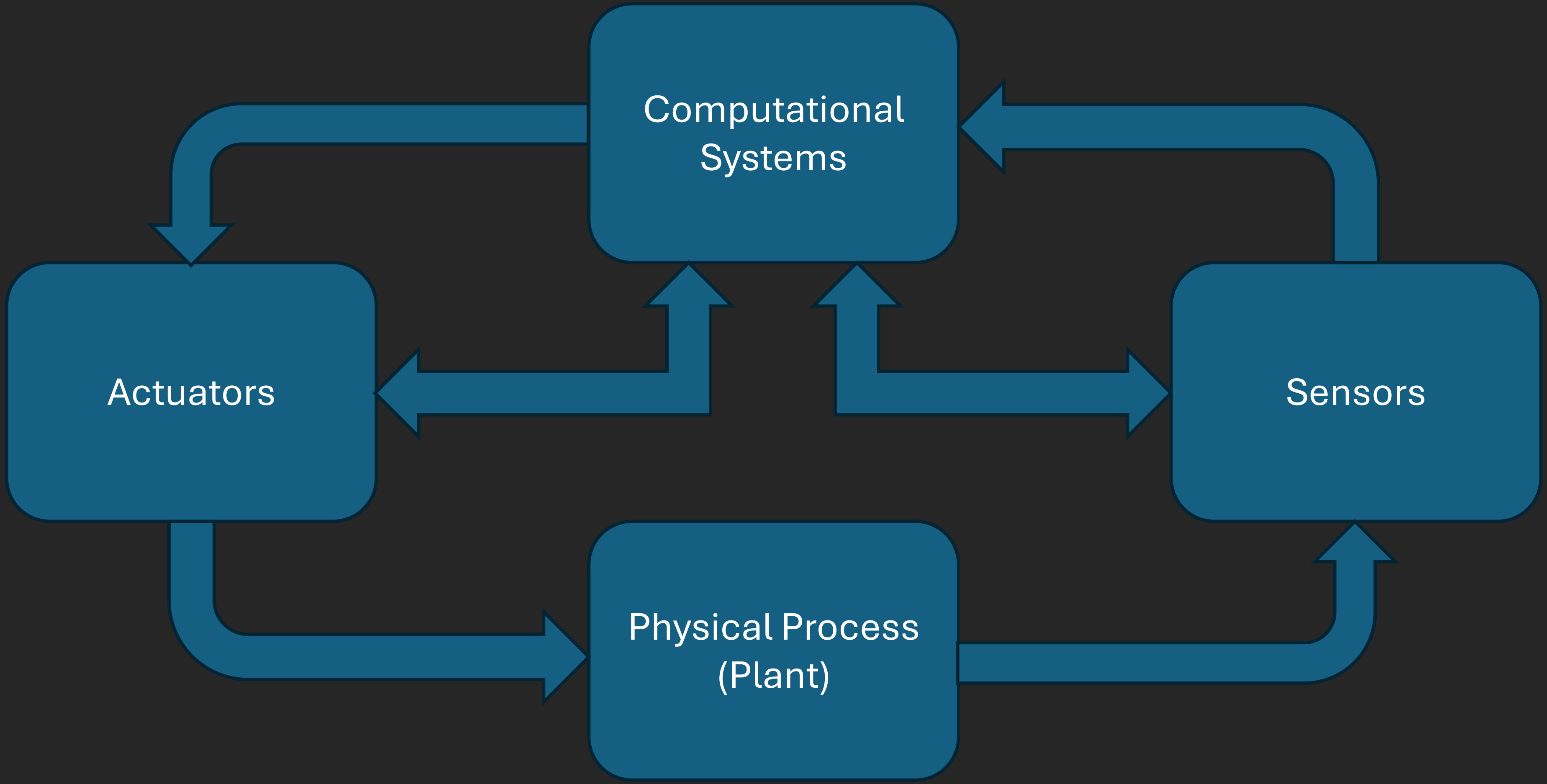


$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k]$$

Digital Signals Considerations

- Sampling frequency – Nyquist theorem
- Quantization and data types
- Computational systems
  - Microcontrollers
  - Digital Signal Processors (DSPs)
  - FPGAs
  - CPUs
  - GPUs
  - ASICS

What are Cyber-Physical Systems?



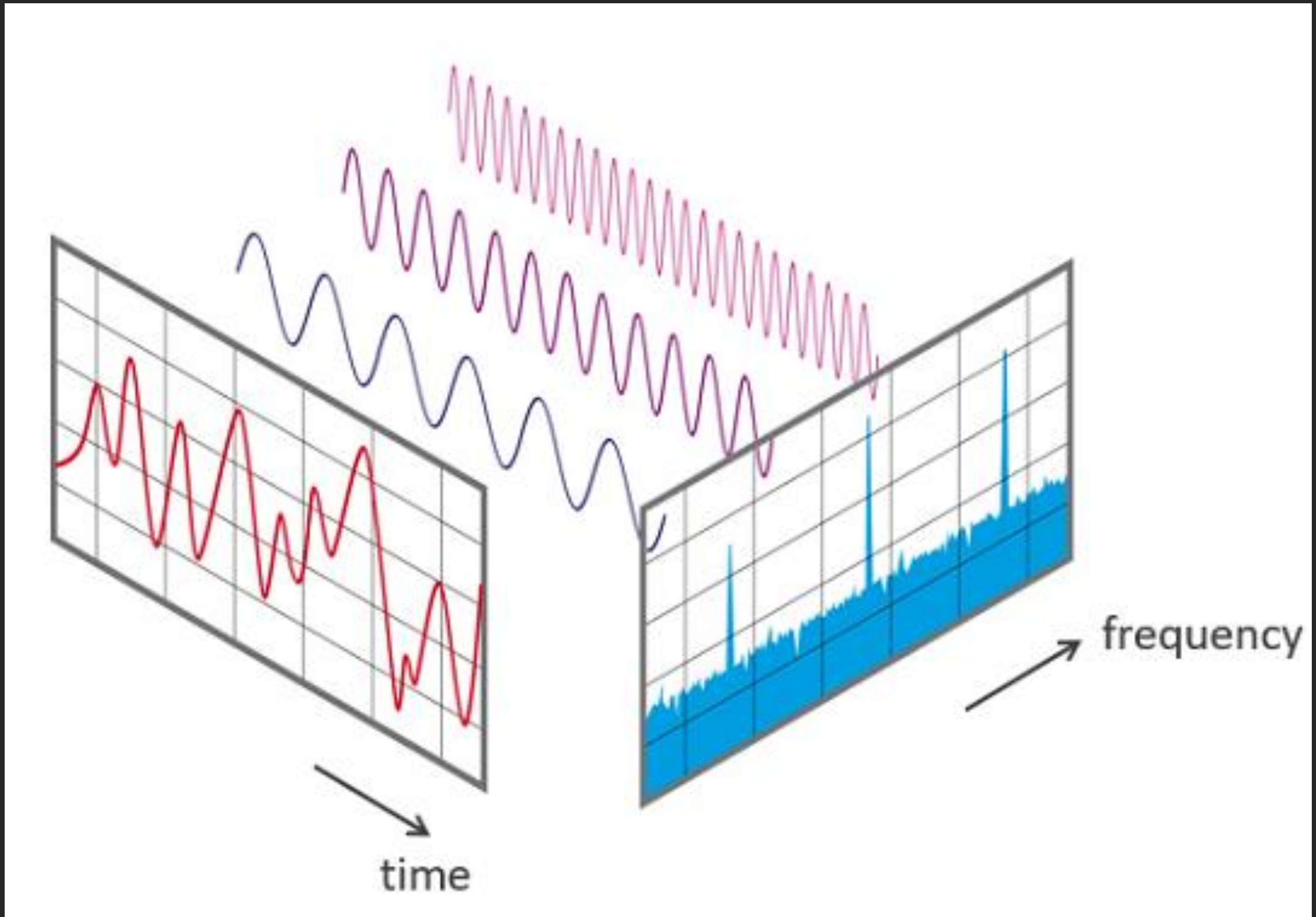
Concepts in Signal Processing

- Fourier Analysis
- Filtering
- Noise Reduction
- Correlation
- Autoregressive Models
- Decimation and Interpolation

Fourier Analysis

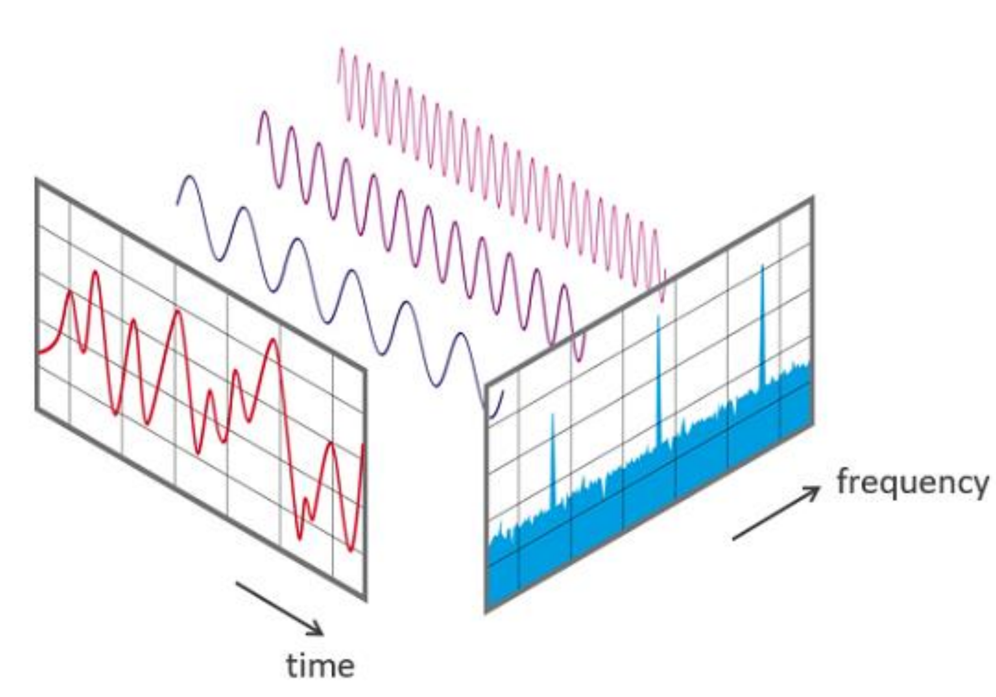
Fourier Analysis

- Provides a way to analyze the frequency content of a signal
- Converts between time domain and frequency domain



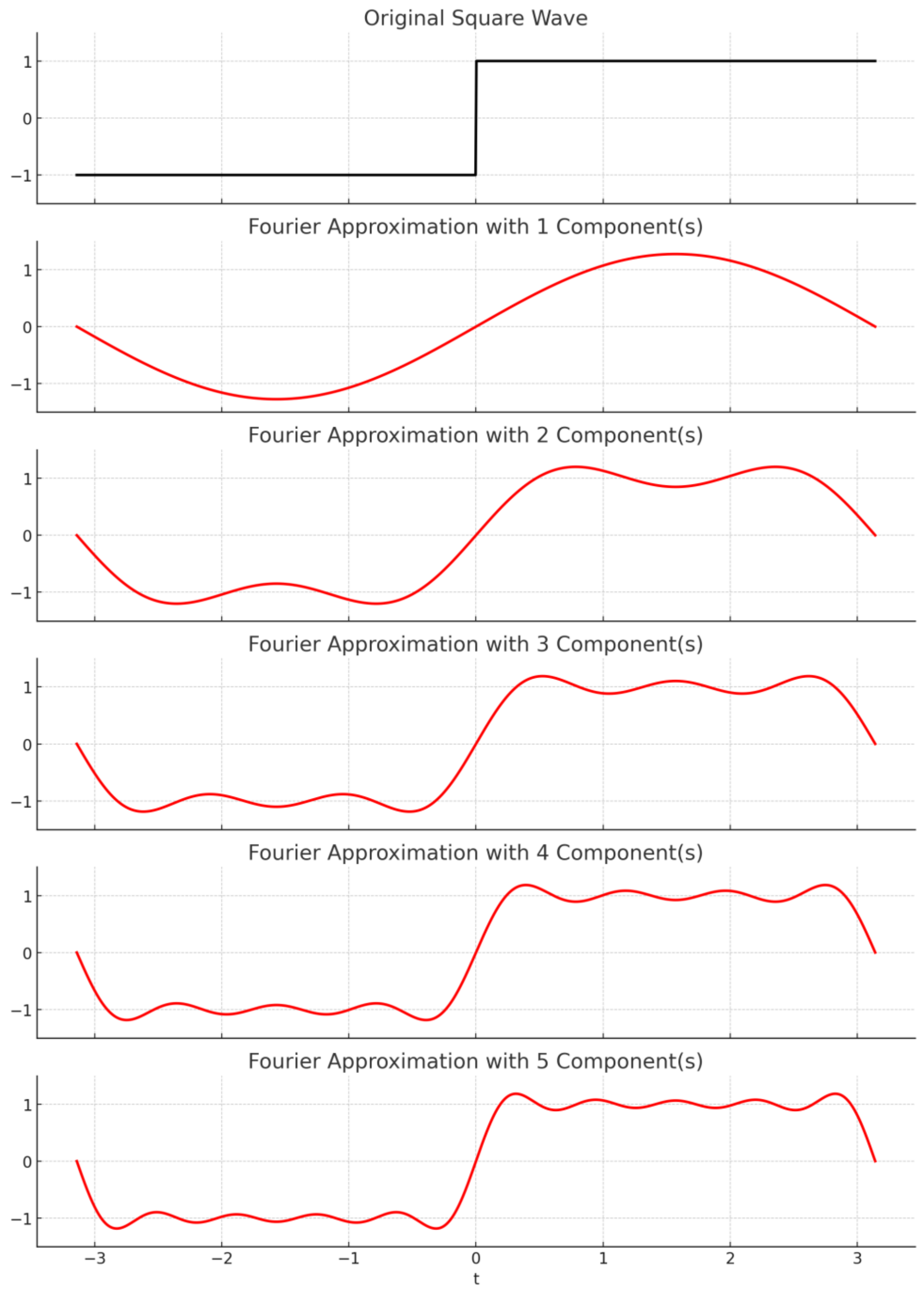
Fourier Series

- Express a periodic signal as a superposition of sinewaves
- $f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi x}{T} + b_n \sin \frac{n\pi x}{T} \right)$
- $f(x) = \sum_{n=1}^{\infty} c_n \, e^{i2\frac{n}{T}x}$



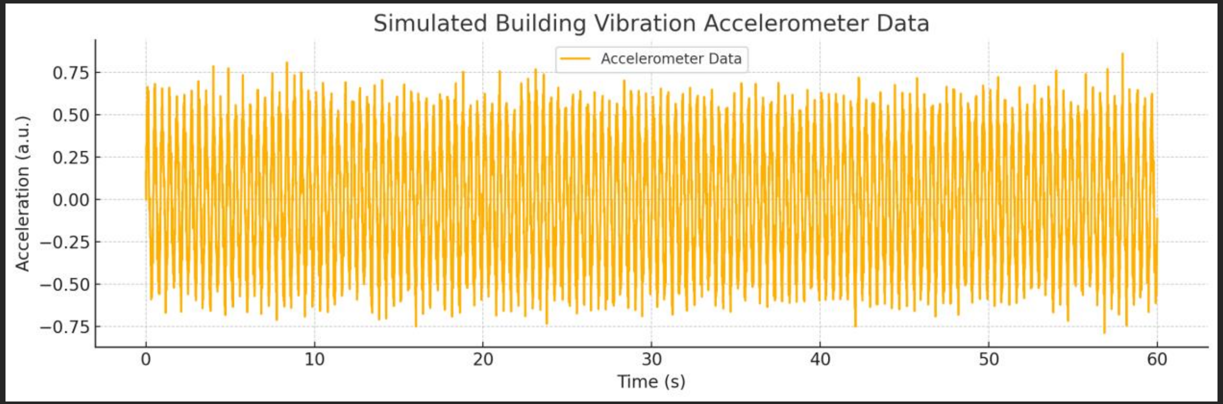
Example – Square Wave

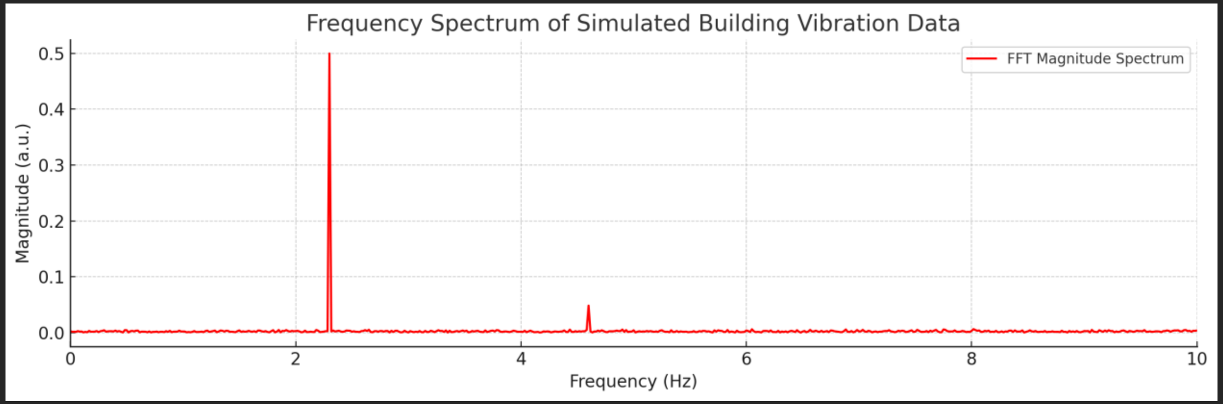
- $f(t) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{1}{2n+1} \sin\big((2n+1)t\big)$
- $f_1(t) = \frac{4}{\pi}(\sin t)$
- $f_2(t) = \frac{4}{\pi}\left(\sin t + \frac{1}{3}\sin 3t\right)$
- $f_3(t) = \frac{4}{\pi}\left(\sin t + \frac{1}{3}\sin 3t + \frac{1}{5}\sin 5t\right)$



Discrete Fourier Analysis

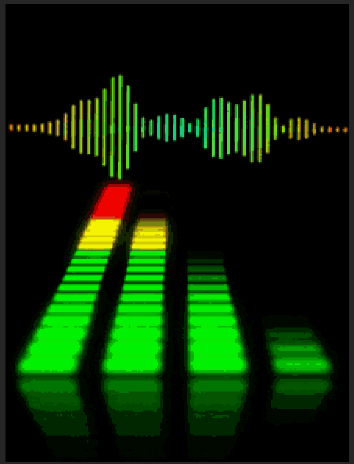
- $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{i\frac{2\pi}{N}kn}$
- $X[k] = \sum_{n=0}^{N-1} x[n]e^{-i\frac{2\pi}{N}kn}$
- Magnitude of $X[k]$ indicates the frequency amplitude
- $f[k] = \frac{kf_s}{N}$

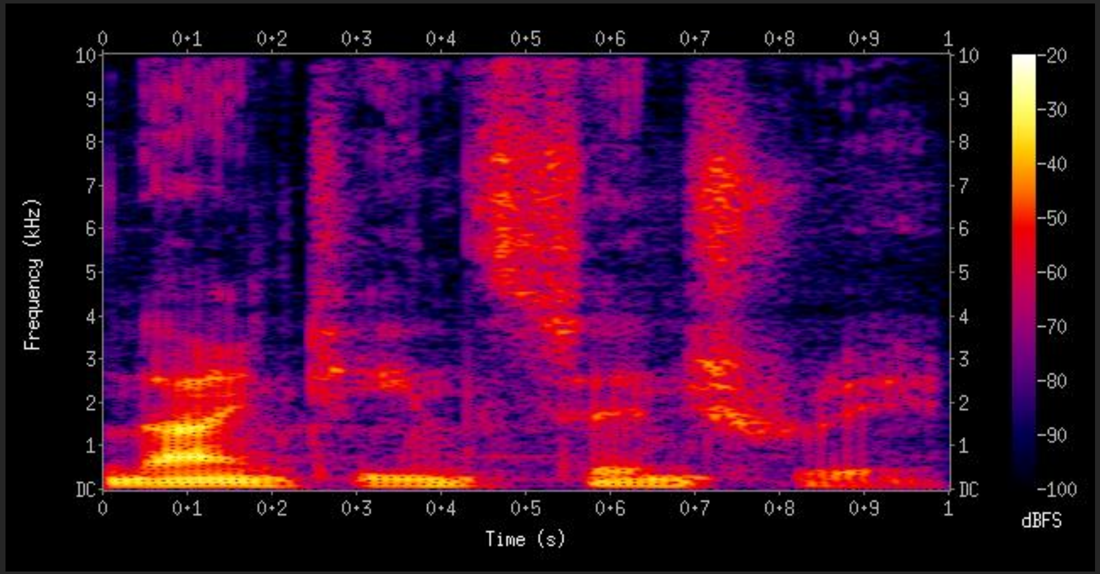




Discrete Frequency Analysis in Practice

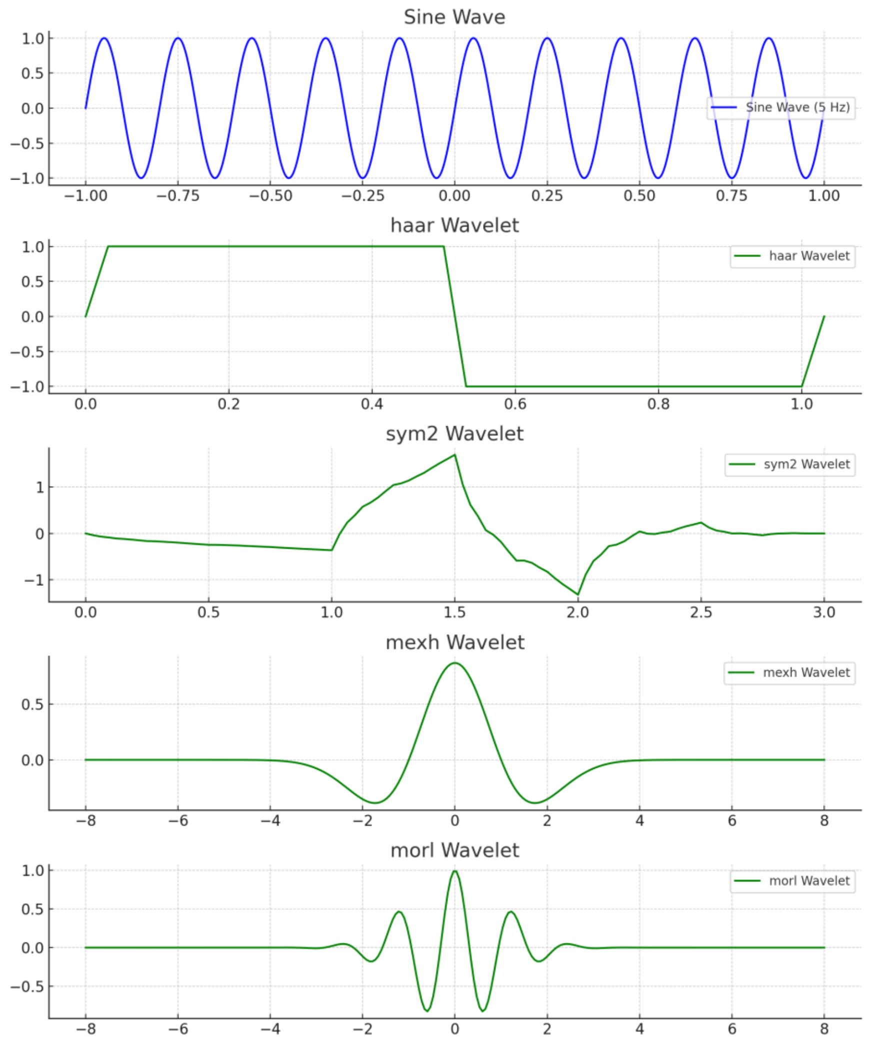
- Short-Time Frequency Transform (STFT)
  - Break the signal into small windows and apply DFT
- Window Sise
  - Short Window -> Good time resolution, poor frequency resolution
  - Long Window -> Poor time resolution, good frequency resolution





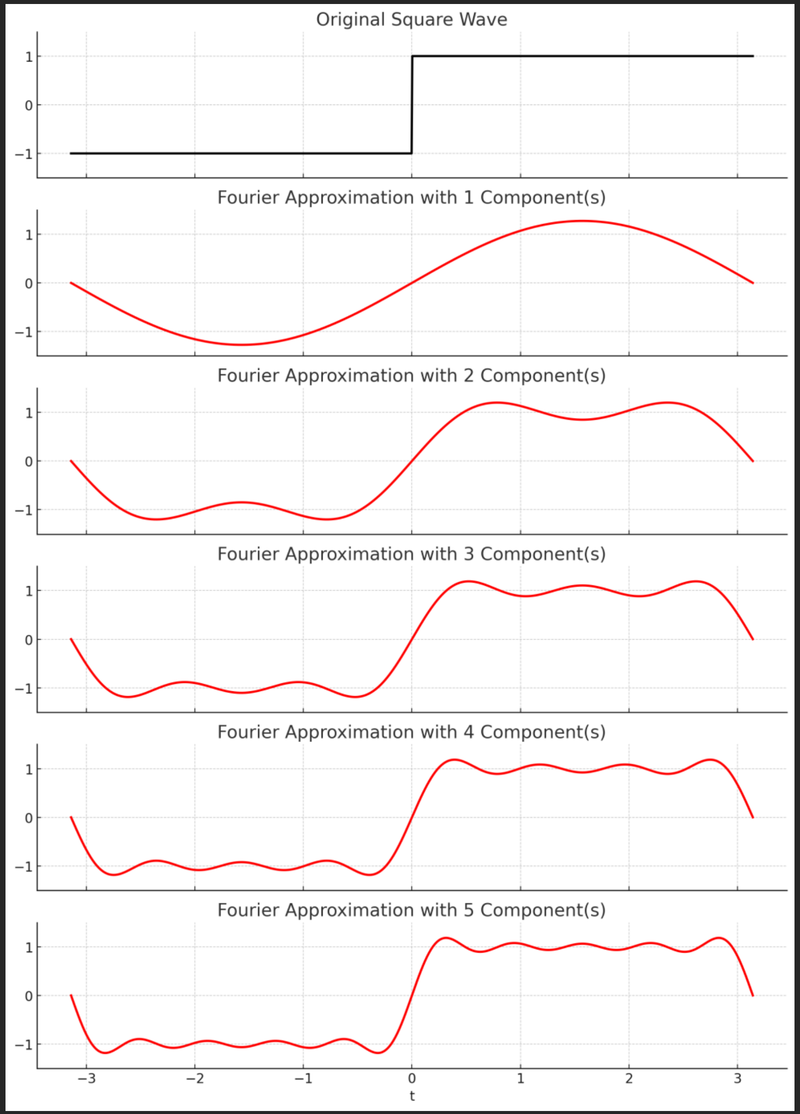
Discrete Frequency Analysis in Practice

- Wavelet Transform
  - Dynamically shifts the window of transform
  - Doesn't decompose signal into sine waves
  - Uses a "mother" wavelet
- Better a localizing frequency content



Signal Compression

- DFT and Wavelet transforms deconstruct a signal into “larger” and “smaller” features
- $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{i\frac{2\pi}{N}kn}$
- $X[k] = \sum_{n=0}^{N-1} x[n]e^{-i\frac{2\pi}{N}kn}$
- Only saving some $X[k]$ will result in a loss of detail.



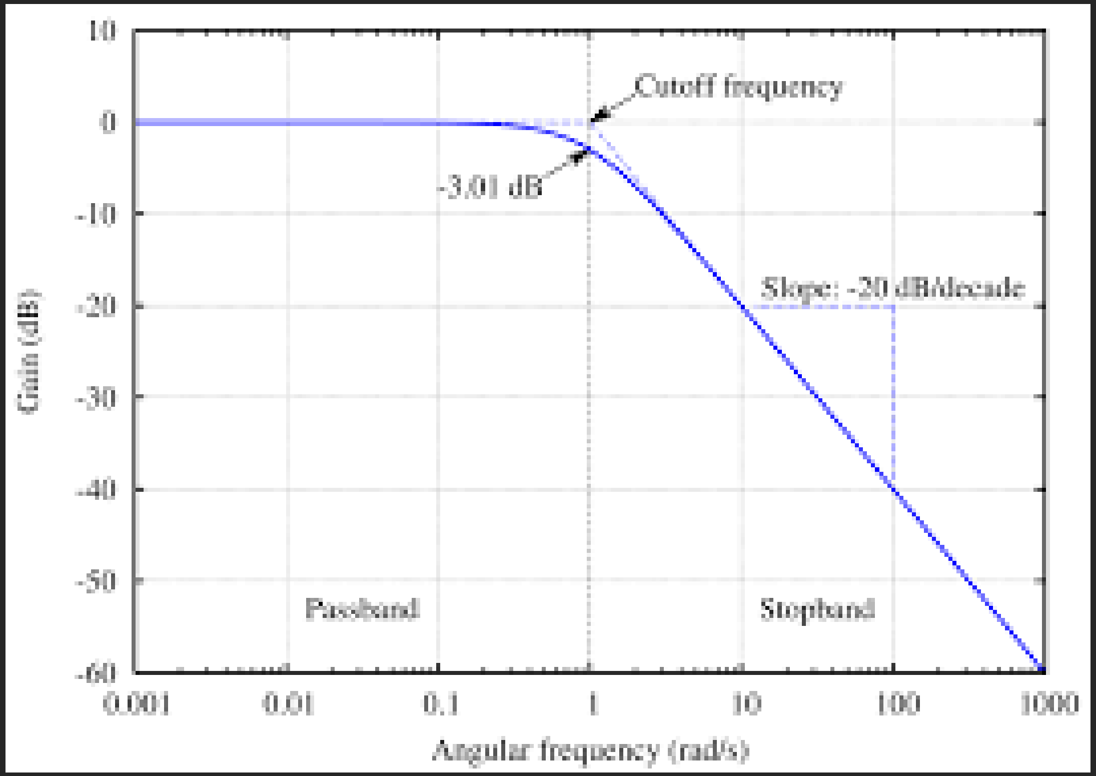
Filtering & Noise Reduction

Filtering

- Define – The process of manipulating the frequency content of a signal
- Types of filtering
  - Causal (on-line) Filtering – process data points as they come in
  - Non-causal (off-line) Filtering – process entire dataset at once
  - Finite Impulse Response – only requires inputs
  - Infinite Impose Response – requires input and all past outputs
- Applications
  - Noise Reduction, Signal Extraction, System Stability

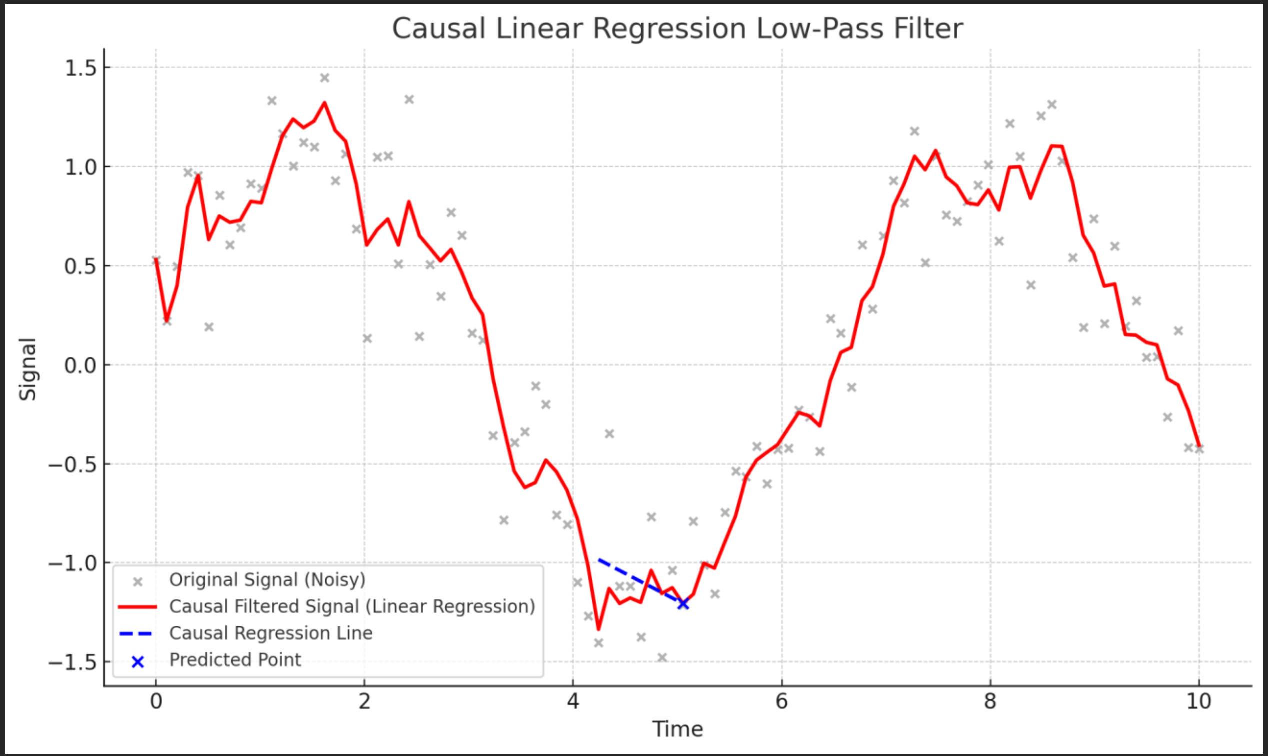
Low-pass Filters

- Remove high-frequency content from a signal
- Used to remove high-frequency noise
  - Electrical
  - Vibration
  - Thermal

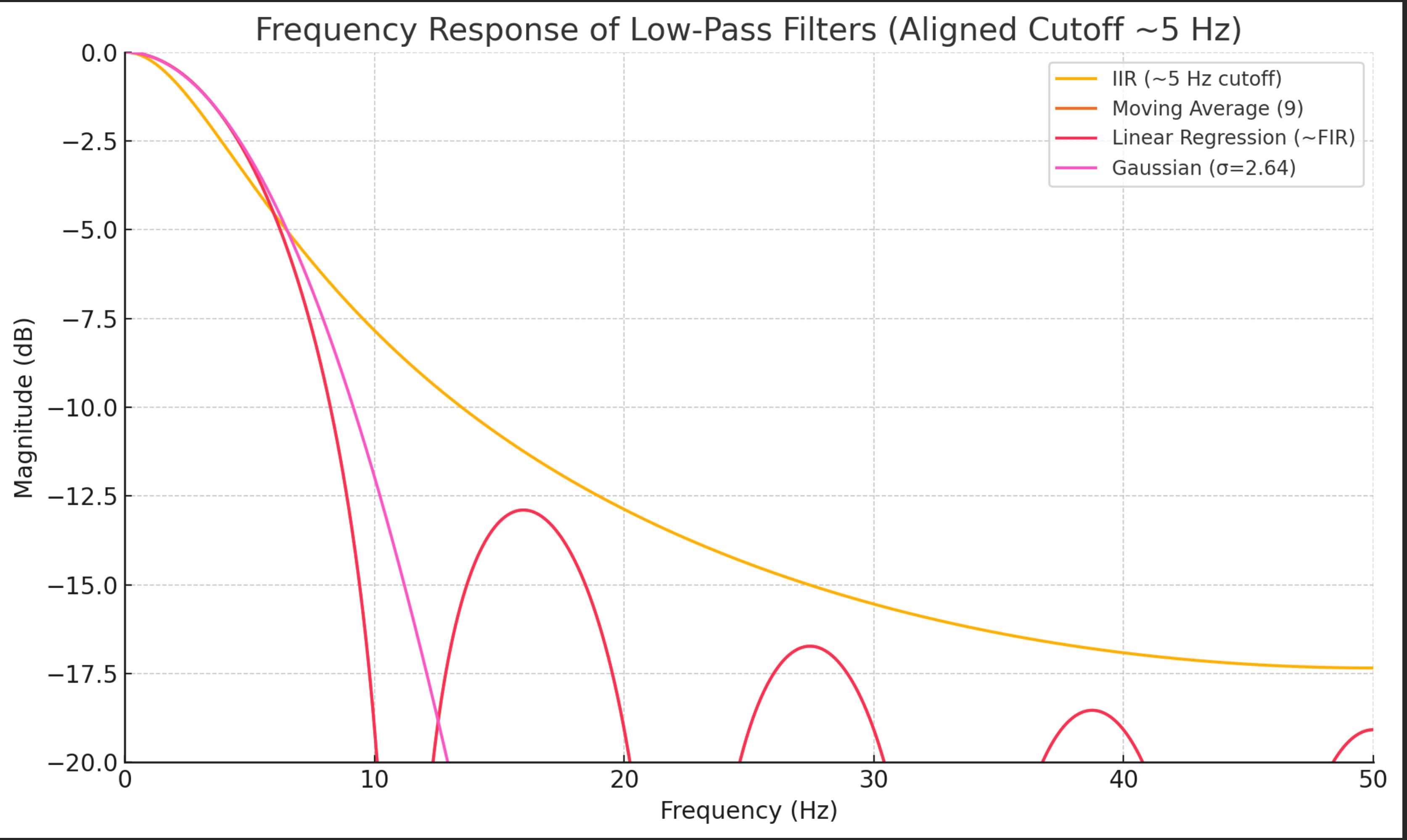


Types of Low-Pass Filters

- Example
  - $y[n] = \alpha x[n] + (1 - \alpha)y[n-1]$
- Linear regression
  - Fits the last N points to a line and projects
- Moving Average
  - Averages the last N values
- Gaussian Filter
  - Like moving average but discounts with distance

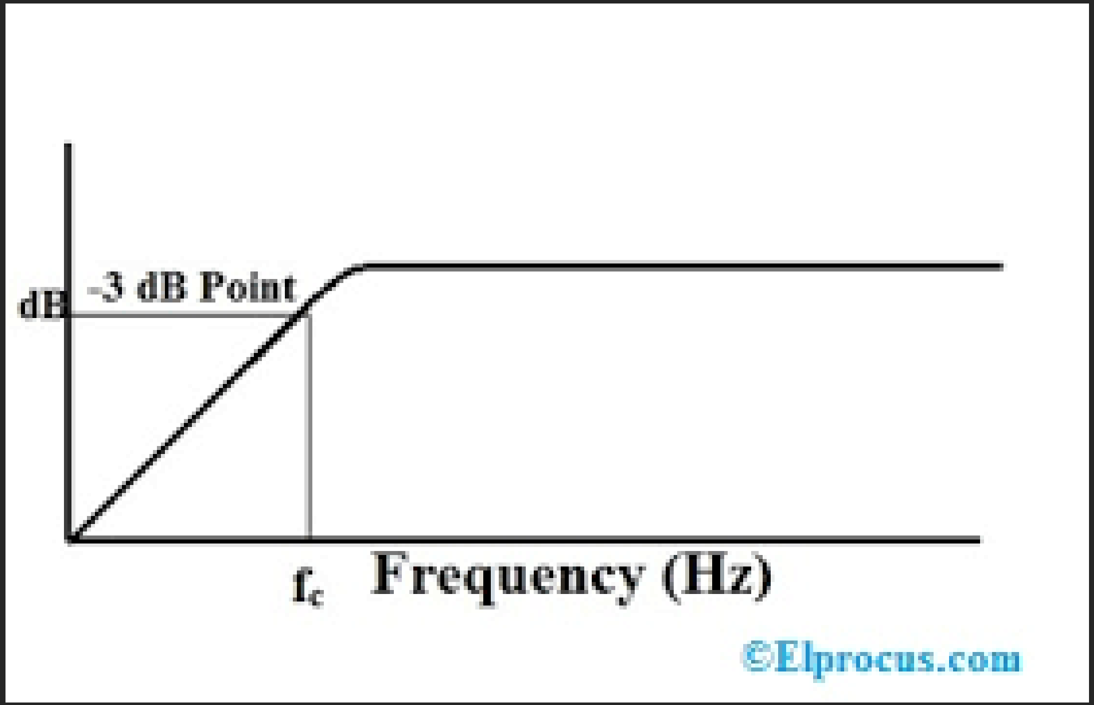


Frequency Response



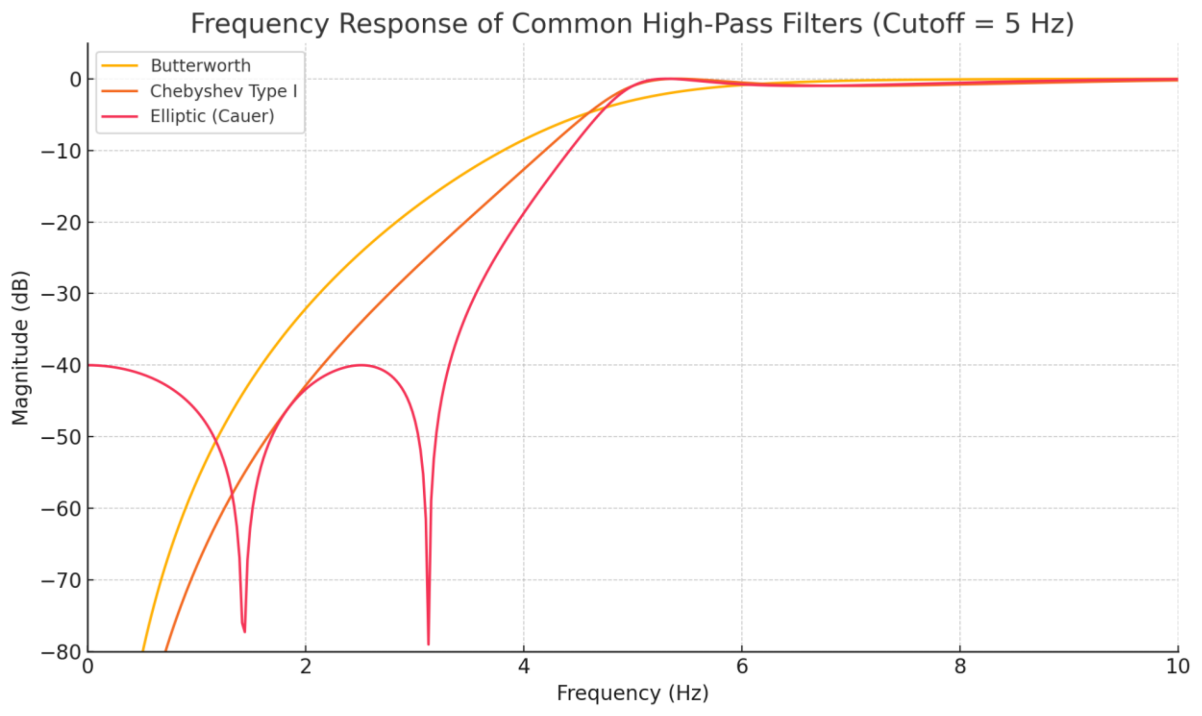
High-Pass Filters

- Removes low frequency content
- Used to remove drift
  - Gyroscope
  - Audio Signals
  - Spike detections



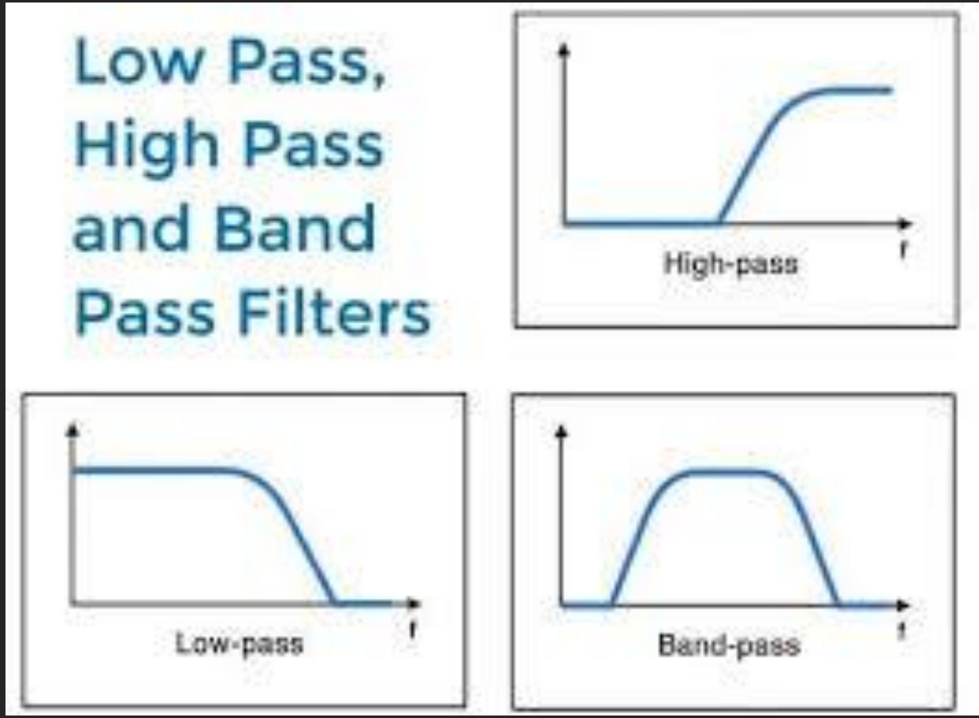
Types of High-pass filters

- First Order IIR
  - $y[n] = \alpha(y[n-1] + x[n] - x[n-1])$
- Moving Difference
  - Subtract the mean of a moving window
- Low pass filters
  - $y[n] = x[n] - y_{LP}[n]$



Band-pass Filter

- Combines a low-pass and high-pass filter to only let a narrow band of frequencies through



Noise Reduction

- Characterize Noise using Fourier Analysis
  - High frequency
  - Low frequency
  - Particular frequencies
- Intensify latency requirements
  - Causal/non-causal
- Identify Computational resources

Correlation and Auto Regressive Modeling

Correlation

- Correlation – measurement of the similarity of two signals
- Use cases
  - Pattern Matching
  - Event detection
  - Signal alignment for sensor fusion

Cross Correlation

- Measures the similarity between two different signals
- Direct – performs correlation on the signal
  - $R_{xy}[k] = \sum_n x[n]y[n+k]$
- Advantages
  - Simple, effective for short signals,
- Disadvantages
  - Computationally expensive for long signals

Cross Correlation

- FFT-Based
  - $R_{xy}[k] = F^{-1}\{X[f]Y^*[f]\}$
- Advantages
  - Much faster for longer signals, ideal for real-time applications
  - $O(NlogN)$
- Disadvantages
  - Require zero padding
  - More complex to implement

Auto Correlation

- Measures the similarity between a signal and itself
- Applications
  - Periodic signal detection, fundamental frequency
  - Fault detection, predictive maintenance
  - Noise reduction

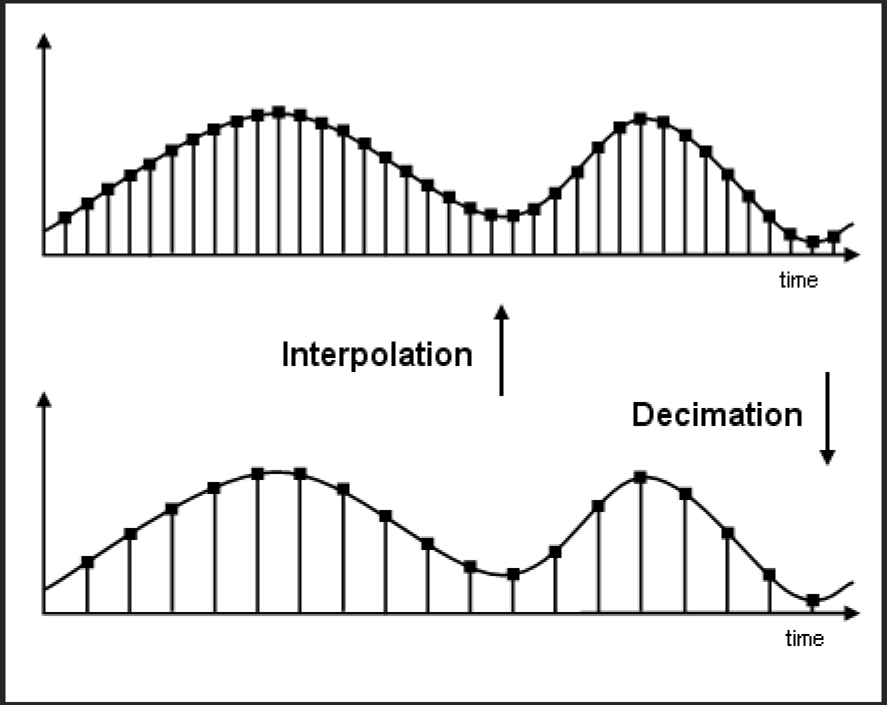
Auto Regressive Modeling

- A method for predicting future values from past values
- Linear model
  - $x[n] = \sum_{i=1}^{p} a_i x[n-1] - e[n]$
- Applications
  - System identification
  - Prediction and forcasting
  - Anomaly detection

Decimation and Interpolation

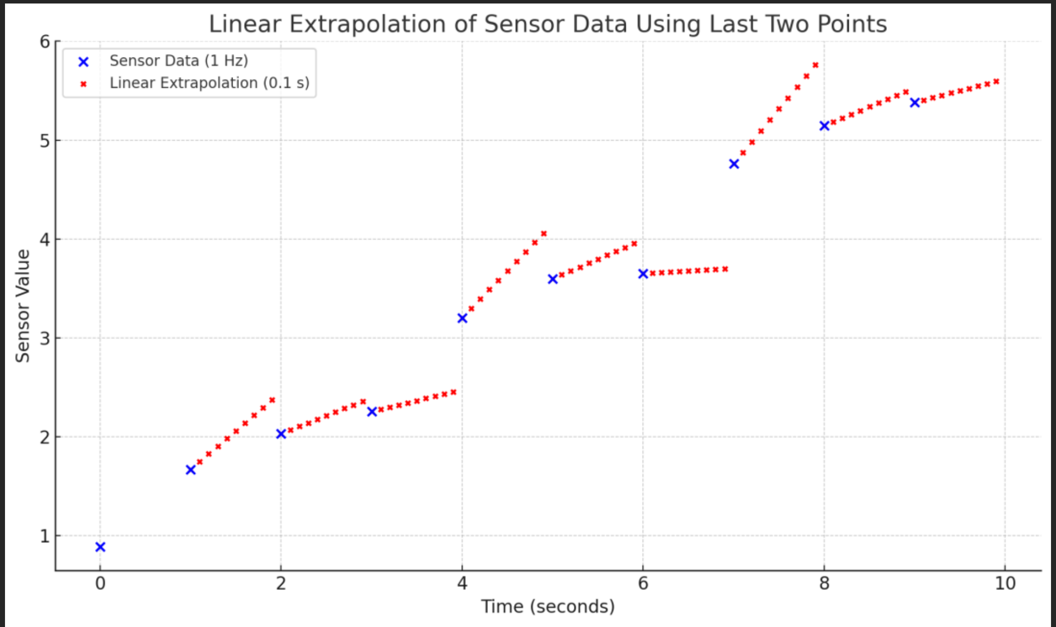
Decimation

- Down sampling a signal by taking every mth value.
- Low-pass filter to remove noise above the new Nyquist frequency
- Applications
  - Data reduction
  - Energy efficiency
  - Signal alignment



Interpolation

- The process of increasing the sample rate of a signal by placing values between existing values
- Causal Interpolation
  - Extrapolate previous data to new time using curve fit
- Non-causal Interpolation
  - Fill in data using low-pass filter



Concepts in Signal Processing

- Fourier Analysis
- Filtering
- Noise Reduction
- Correlation
- Autoregressive Models
- Decimation and Interpolation

State Estimation

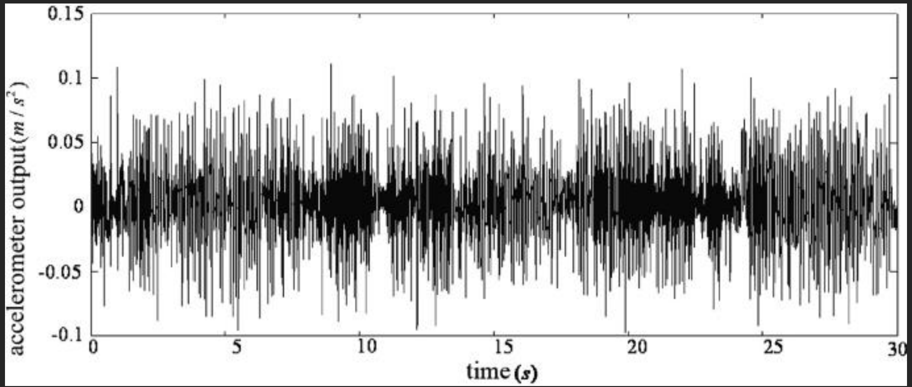
State Estimation

- Process used to infer the internal state of a system that cannot be directly measured due to incomplete and/or noisy measurements
- Components:
  - Sensors Models – Leverages sensor noise characteristics to optimally filter and fuse sensor readings
  - System Dynamic Models – Predicts the behavior of a system based on equations of motion and actuator inputs

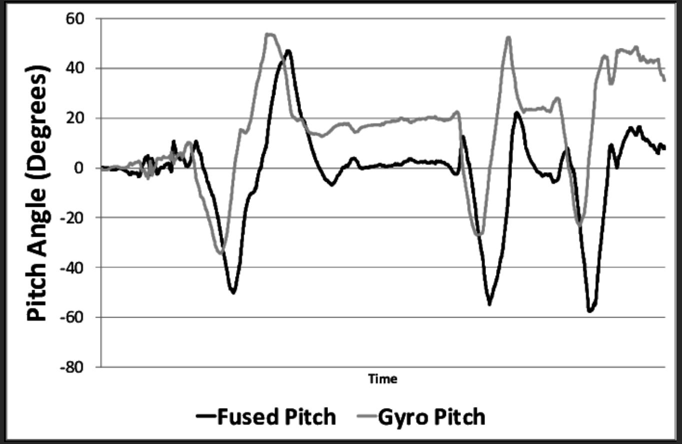
Sensor Models

- Sensor models define the mapping from the true state of a system to the measurements obtained by sensors
- $y = h(x) + v$
- $y =$ sensor reading
- $h(x) =$ function that maps state of the system to sensor reading
- $v =$ sensor noise

IMU Example

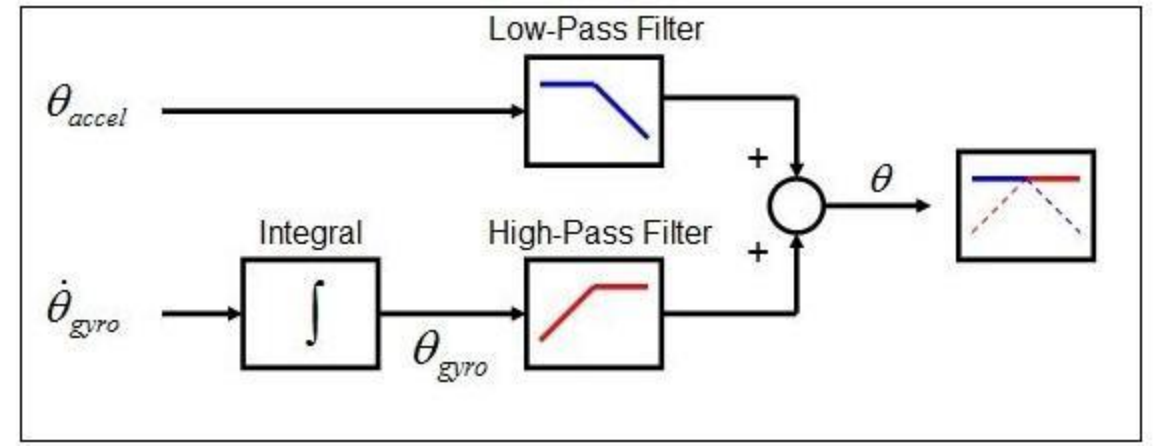


- Accelerometer Noise
  - Low Pass Filter
  - $\theta_{a\_lp}[n] = \alpha\theta_{accel}[n] + (1-\alpha)\theta_{a\_lp}[n-1]$
- Gyroscope Drift
  - High Pass Filter
  - $\theta_{g\_hp}[n] = \alpha\left(\theta_{g\_hp}[n-1] + \theta_{gyro}[n] - \theta_{gyro}[n-1]\right)$



Commentary Filter

- Combine the accelerometer low frequency component with the gyroscope high frequency component



- $\theta_{est}[k] = \alpha(\theta_{est} + \dot{\theta}_{gyro}[k]\Delta t) + (1-\alpha)\theta_{accel}$

Commentary Filter

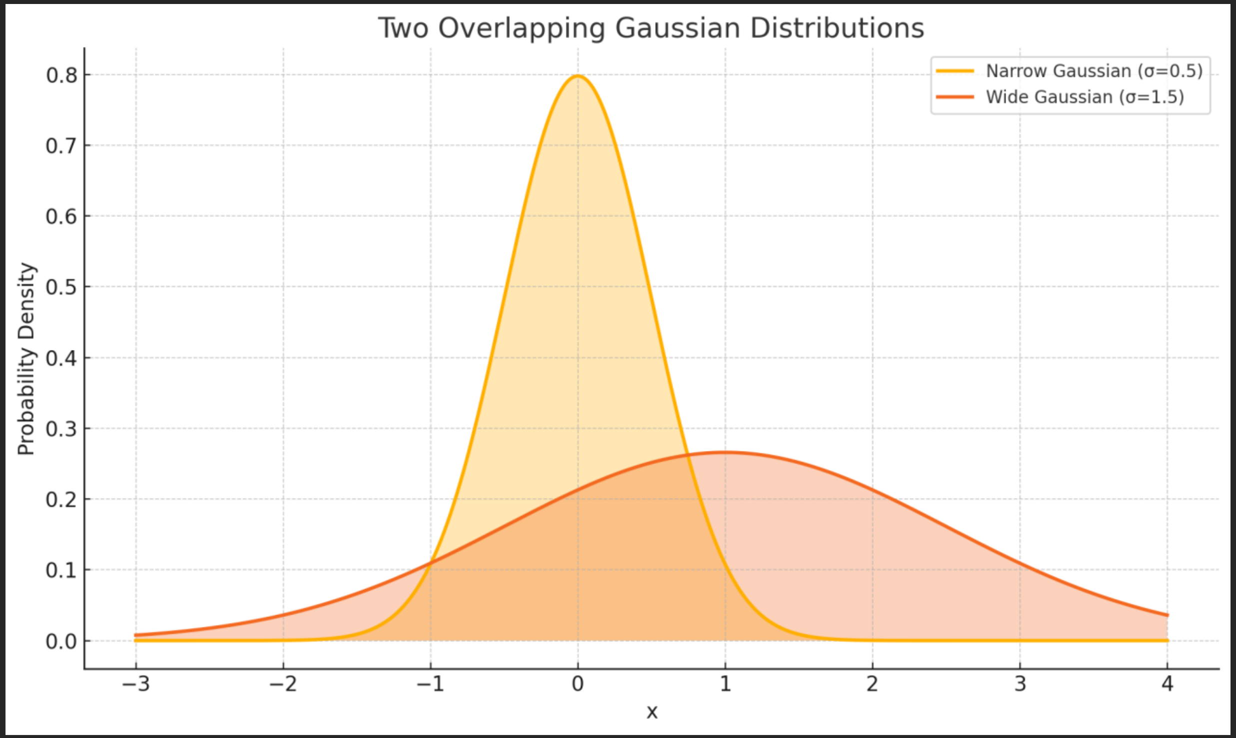
- $\theta_{est}[k] = \alpha(\theta_{est} + \dot{\theta}_{gyro}[k]\Delta t) + (1 - \alpha)\theta_{accel}$

- When $\alpha = 1$
  - Defaults to only gyroscope reading
- When $\alpha = 0$
  - Defaults to only accelerometer reading

Probabilistic State Estimation

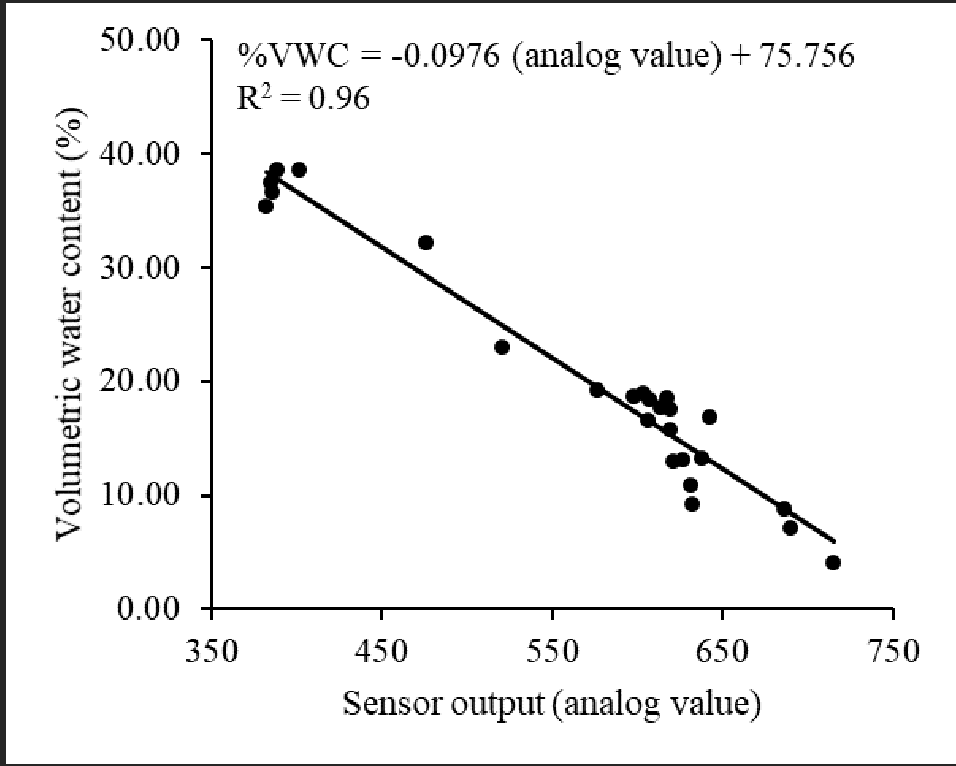
- Classical State Estimate
  - Represents the state as a single value
- Probabilistic State Estimation
  - Represents the state as a probability distribution

- Allows for combining data based on uncertainty



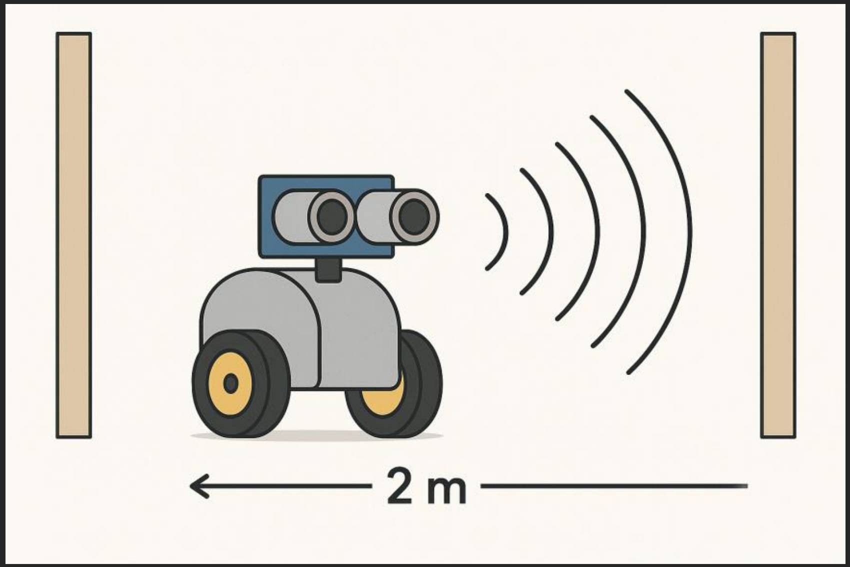
Linear Gaussian Sensor Model

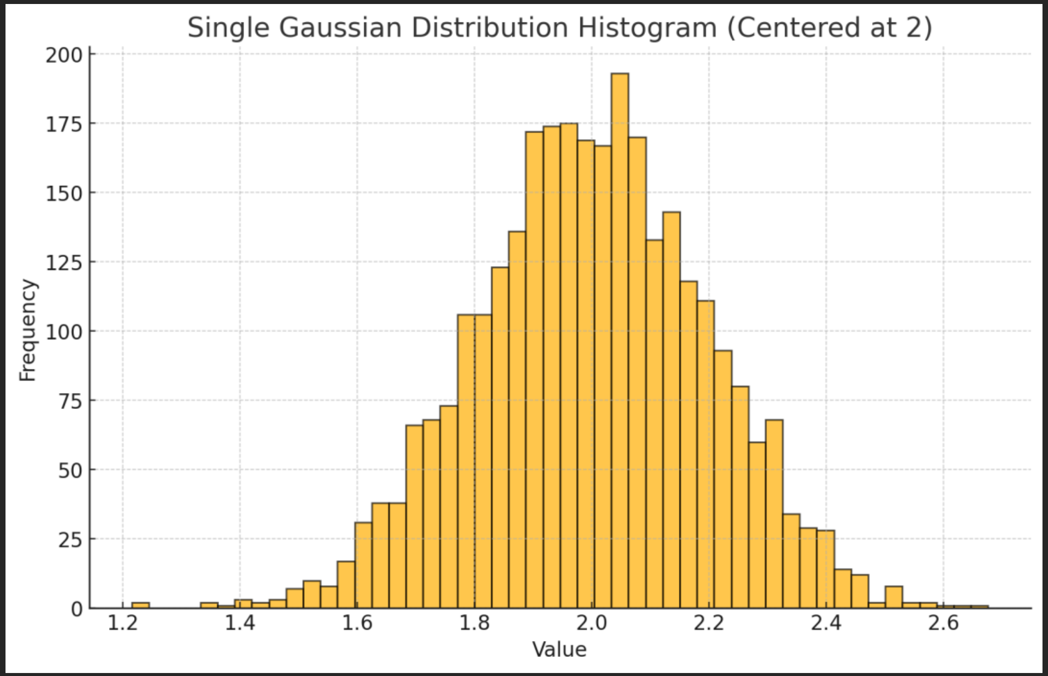
- Sensor readings is a linear function of the state
- $y = Hx + v$
- $v \sim N(0, \sigma)$



Linear Gaussian Sensor Model

- State Estimators
  - Mean
  - Median
  - Max
  - Min

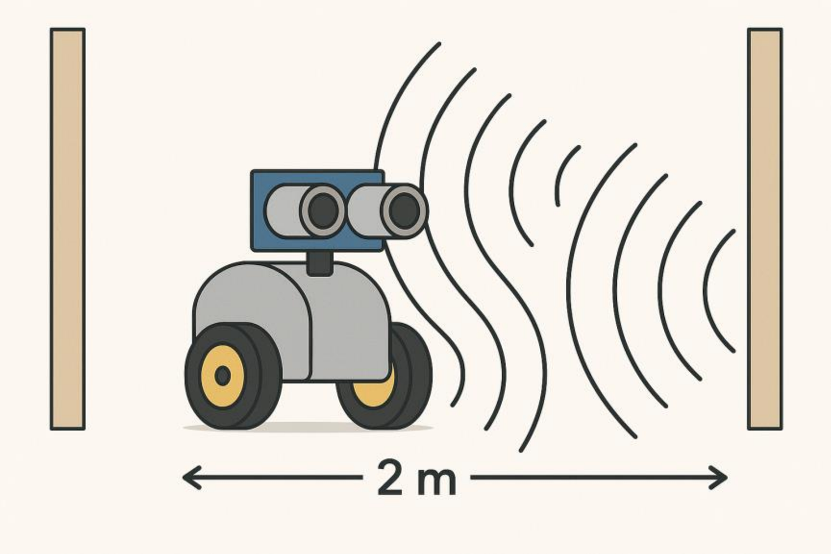


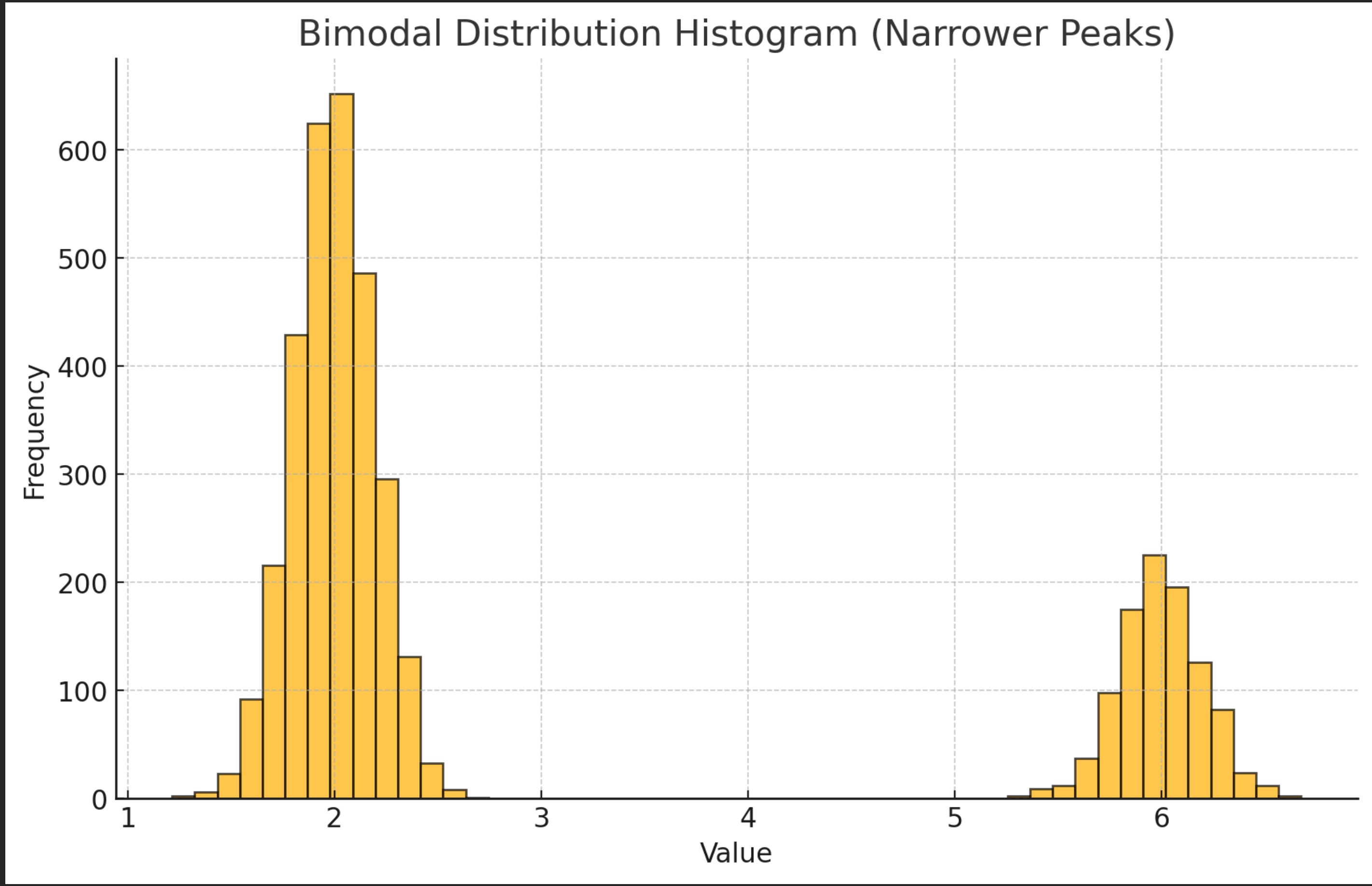


Maximum Likelihood Estimator

- The method that minimizes the sum of squared errors
- $SSE = \sum_{i=0}^{N}(y_i - \widehat{y_i})^2$
- y_i = true value
- $\widehat{y_i}$ = Estimate (e.g. mean, median)

Linear Bimodal Gaussian Sensor Model



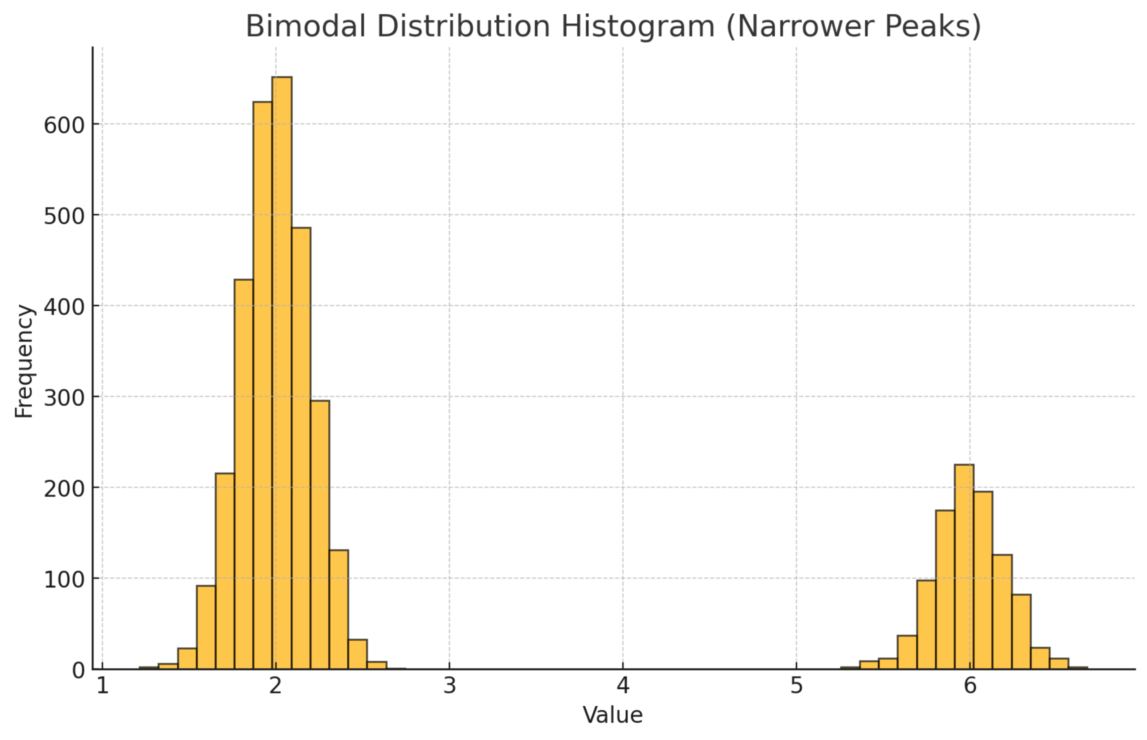


Random Sampling Consensus (RANSAC)

- Select a subsample of the data
- Run the model on the subsample
- Measure the distance off data from model
- Select inliers and outliers
- Repeat and select model with most inliers
- Run model on inliers

Bimodal Gaussian Sensor Model

- Repeat a bunch of times
  - Select 10 out of 100 points
  - Measure distance from mean
  - Select inliers by threshold
- Select inliers of max run
- Take mean of inliers



Motion Models

System Dynamics

- A set of equations that govern the motion of a system
  - Kinematic equations
  - Newton's laws
  - Simple update
- Example – Simple update
  - $x_k = x_{k-1} + u_k + w$ where $w_k \sim N(0, Q)$

Example – Spring Mass System

- Newton's law
  - $F = ma$
  - Forces: External ($u(t)$), Spring ($-kx$), Friction ($-c\dot{x}$)
- Equation of Motion
  - $m\ddot{x} = u(t) - c\dot{x} - kx$
  - $u(t) =$ External Force
  - $x =$ Position of Mass
  - $m =$ Mass
  - $k =$ Spring Constant

Example – Spring Mass System

- $m\ddot{x} = u(t) - c\dot{x} - kx$
- Rewrite
  - $x_1(t) = x(t)$
  - $x_2(t) = \dot{x}(\mathrm{t})$
- Take Derivatives
  - $\dot{x}_1(\mathrm{t}) = \mathrm{x}_2(\mathrm{t})$
  - $\dot{x}_2(\mathrm{t}) = \ddot{x}_1(t) = -\frac{k}{m}x_1(t) - \frac{-c}{m}x_2(t) + \frac{1}{m}u(t)$

Example – Spring Mass System

- $\dot{x}_1(\mathrm{t}) = \mathrm{x}_2(\mathrm{t})$
- $\dot{x}_2(\mathrm{t}) = -\frac{k}{m}x_1(t) - \frac{c}{m}x_2(t) + \frac{1}{m}u(t)$

- $\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$

- $\dot{x}(\mathrm{t}) = Ax(\mathrm{t}) + \mathrm{Bu}(\mathrm{t})$

Continuous Vs Discrete Time Motion Model

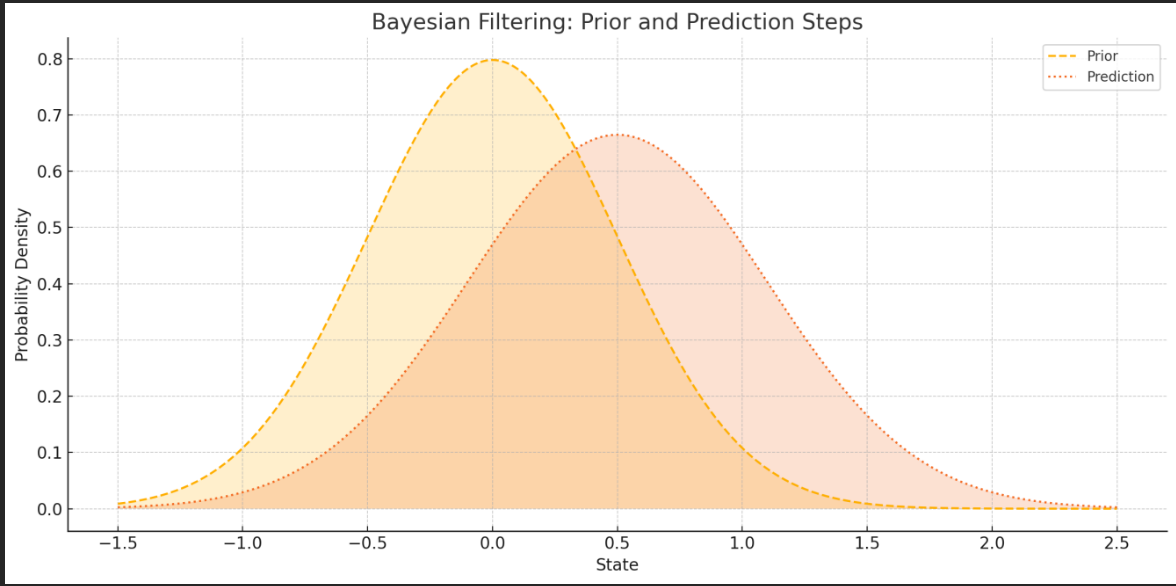
- Continuous Time
  - $\dot{x}(\mathrm{t}) = A_c x(\mathrm{t}) + \mathrm{B}_c \mathrm{u}(\mathrm{t})$
- Discrete Time
  - $x_k = A_d x_{k-1} + B_d u_k$
- Conversion between continuous and discrete time domain
  - $A_d = e^{A_c \Delta t}$
  - $B_d = \int_0^{\Delta t} e^{A_c \tau} d\tau B$

Simplified Motion Model

- Systems Dynamics
  - Treat $u(t)$ as a force and modeling system dynamics
- Simplified Model
  - Treat $u(t)$ as a Δx with no system dynamics
  - $x_k = x_{k-1} + u_k$

Process Noise

- Uncertainty in the system dynamic model
- $x_k = A_d x_{k-1} + B_d u_k + w$
- $w \sim N(0, Q)$
- Every time the system propagates forward in time, you become less certain about its location



Bayesian Estimators & Kalman Filter

Bayes Law

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Example

- $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
- Prior – Suppose 1% of the population has a certain type of cancer
  - $P(Cancer) = .01$
- Sensitivity – Test identifies cancer 90% of the time when present
  - $P(Positive|Cancer) = .90$
- Specificity – False positives occur 5% of the time
  - $P(Positive|No\ Cancer) = .05$

Example

- $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
- $P(Cancer|Positive) = \frac{P(Positive|Cancer)P(Cancer)}{P(Positive)}$
- $\mathrm{P}(Positive) = \mathrm{P(Positive|Cancer)P(Cancer) + P(Positive|No\ Cancer)P(No\ Cancer)}$
- $\mathrm{P}(Positive) = (.90 \times .01) + (.05 \times .99) = 0.0585$
- $P(Cancer|Positive) = \frac{.90 \times .1}{0.0585} = 0.1538$

- What is the probability that someone has cancer?

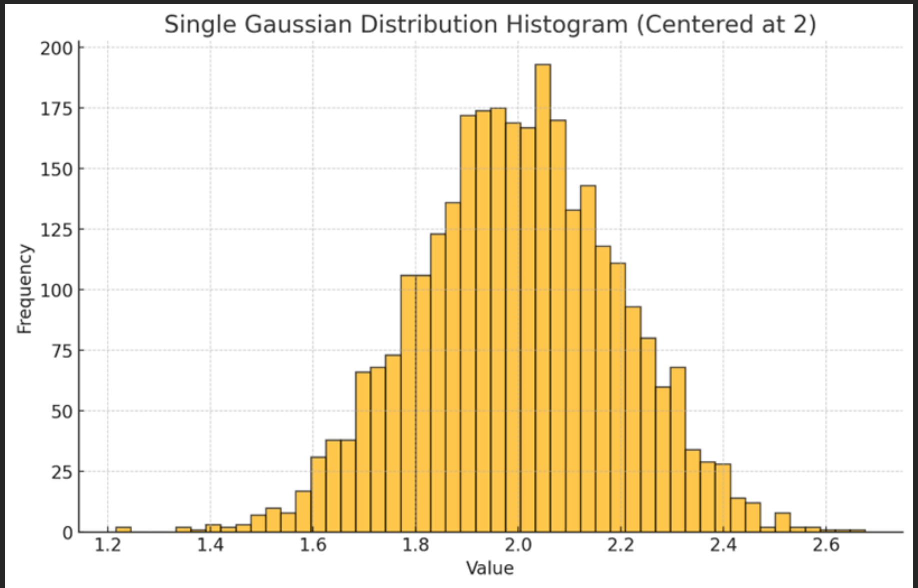
Sensor Models

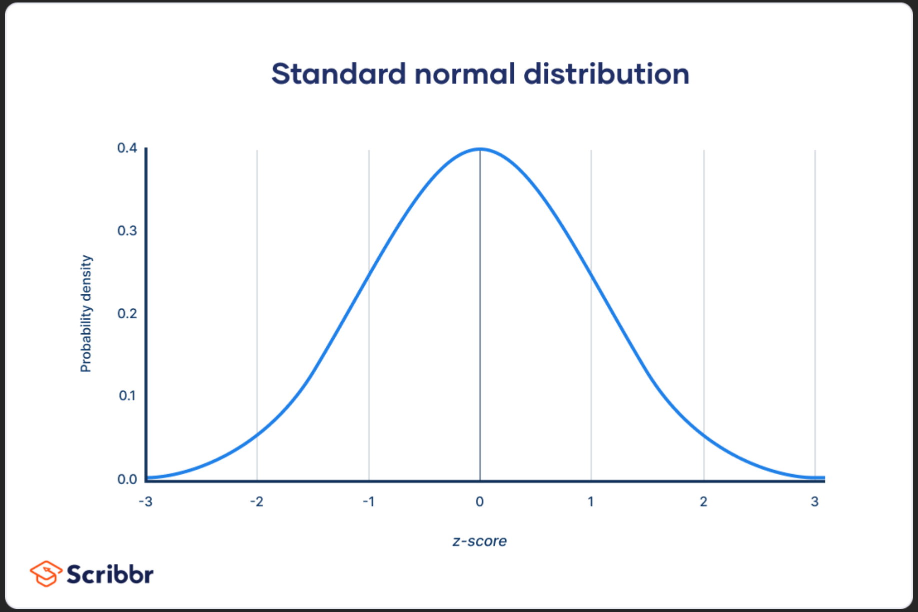
- Bayesian Estimators maximize conditional probability
- $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
- x = State
- y = Sensor Reading
- $p(x = 2m|y = 2.1m) = \frac{p(y=2.1m|x=2m)p(x=2m)}{p(y=2.1m)}$
- $p(y = 2.1m|x = 2m)$ = Sensor Model

Sensor Models

- $p(y = 2.1m \mid x = 2m)$
- $p(y|x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}$

- Why is this important?
- $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$
- $p(x)$ from motion model



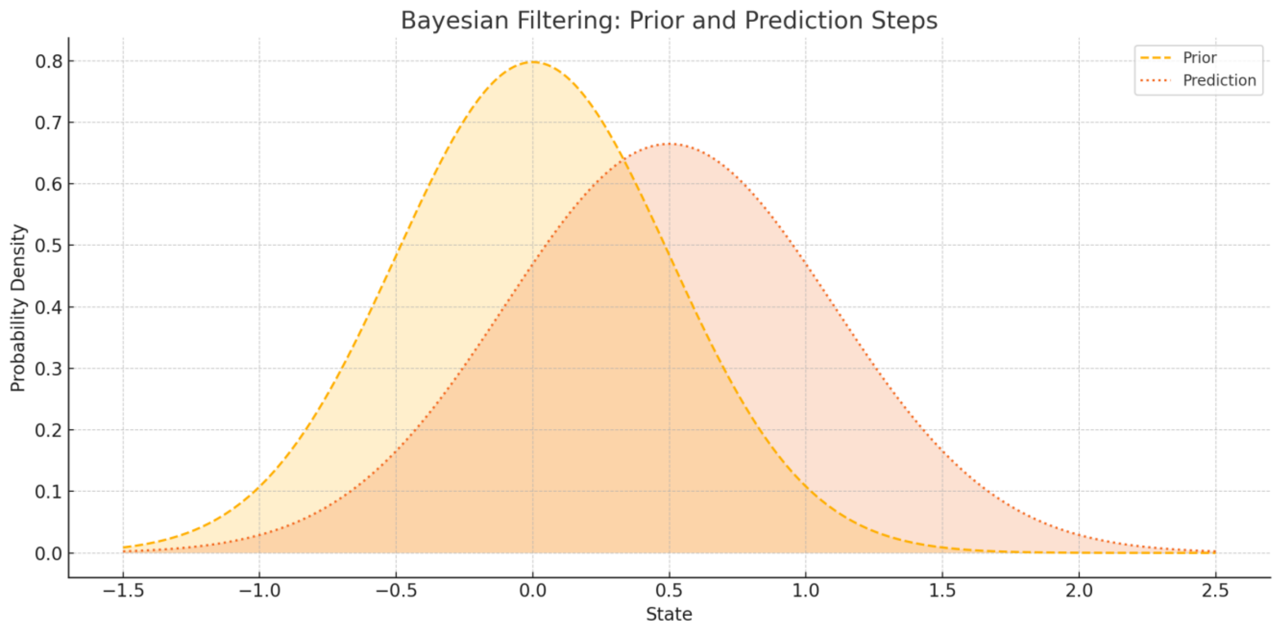


Combining Motion and Sensor Models

- System Equations
  - $x_k = x_{k-1} + u_k + w$ where $w_k \sim N(0, Q)$
  - $y_k = x_k + v$ where $v \sim N(0, R)$
- Bayesian Filter
  - Predict the state
  - Update variance of prediction
  - Calculate gain
  - Update state estimate
  - Update variance of state estimation

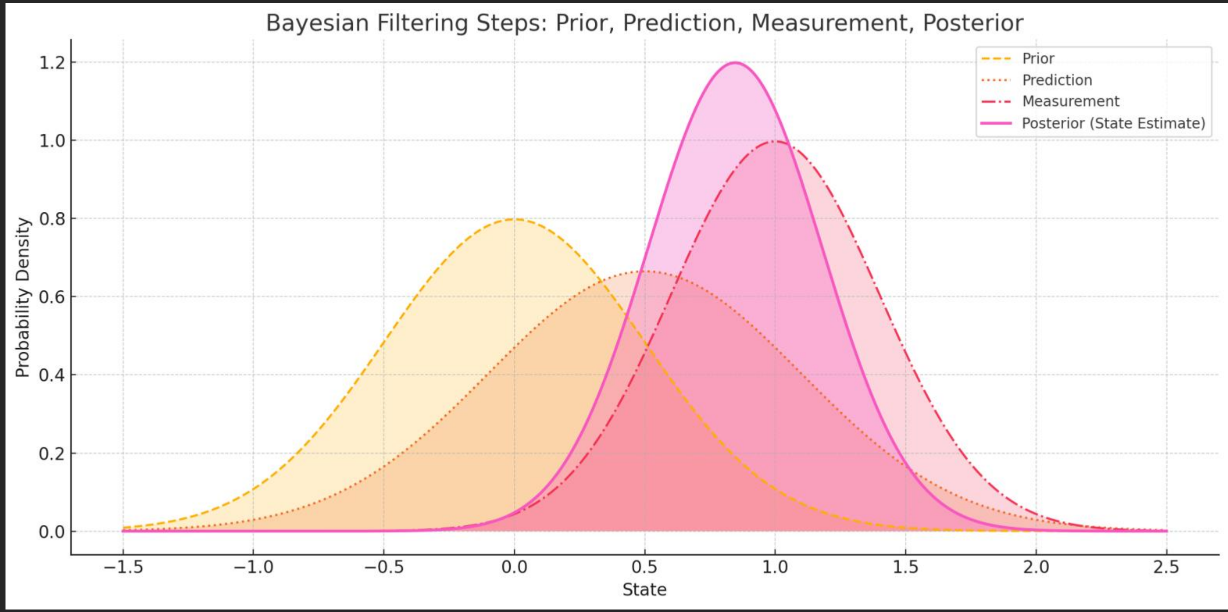
Prediction Step

- System Equations
  - $x_k = x_{k-1} + u_k + w$ where $w_k \sim N(0, Q)$
- Update Prediction
  - $\hat{x}_k = \hat{x}_{k-1} + u_k$
- Update Prediction Variance
  - $P_{k|k-1} = P_{k-1} + Q$



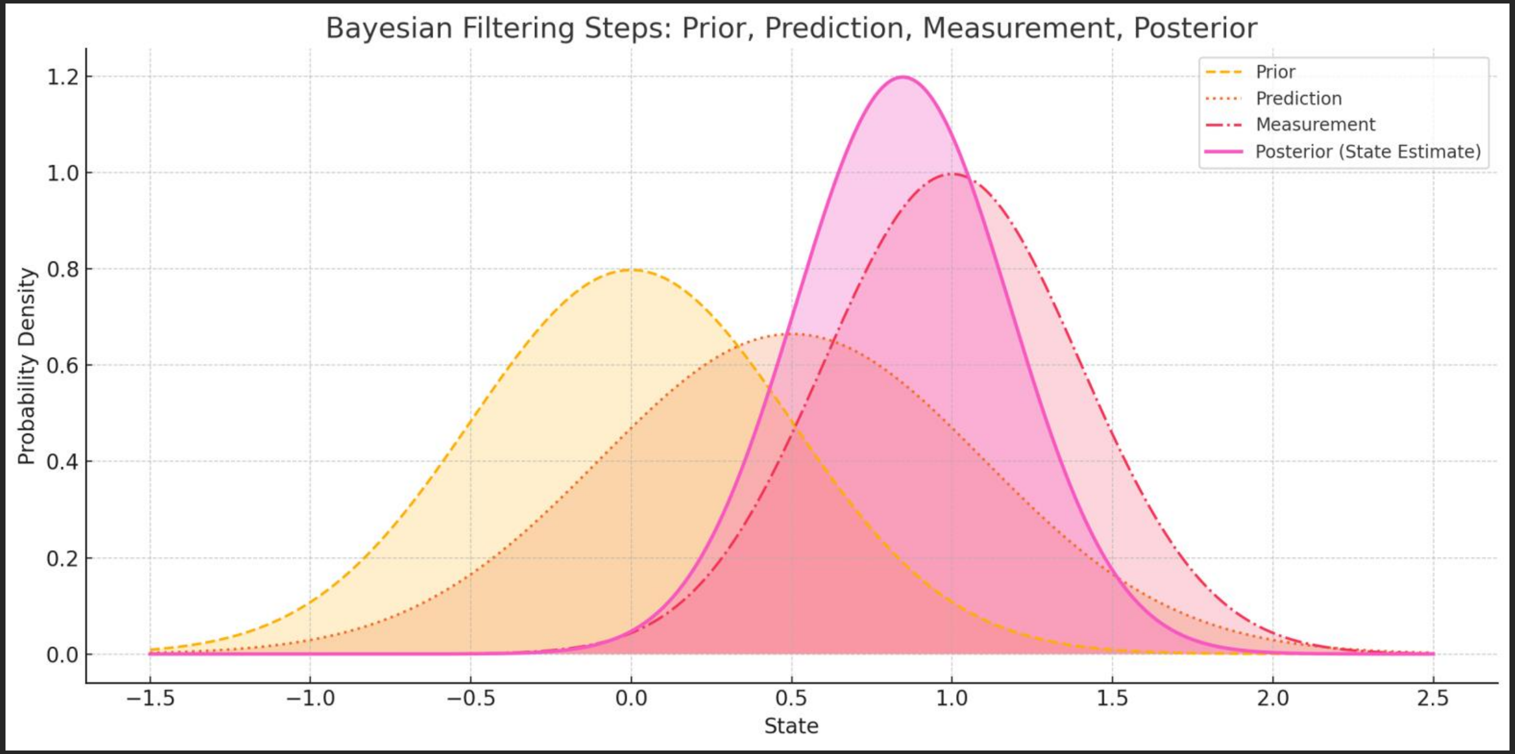
Update Step

- Calculate Gain
  - $K_k = \frac{P_{k|k-1}}{P_{k|k-1}+R}$
- Update State Estimate
  - $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - \hat{x}_{k|k-1}\right)$
- Update State Estimate Variance
  - $P_{k|k} = (1 - K_k)P_{k|k-1}$
- What if $K_k = 1$?



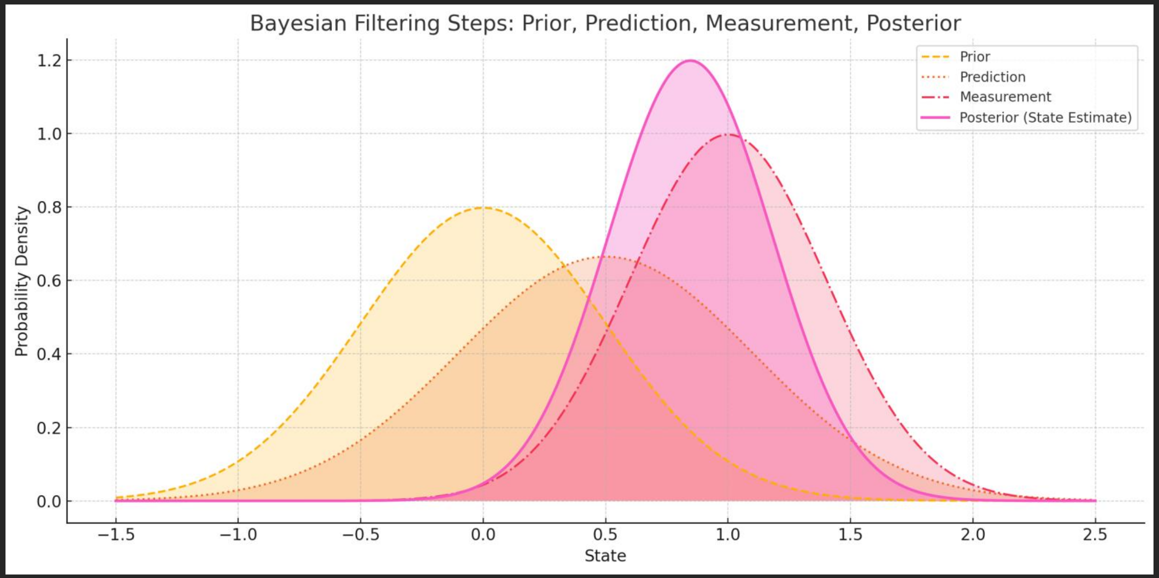
Bayesian Filter

- Predict the state
- Update variance of prediction
- Calculate gain
- Update state estimate
- Update variance of state estimation



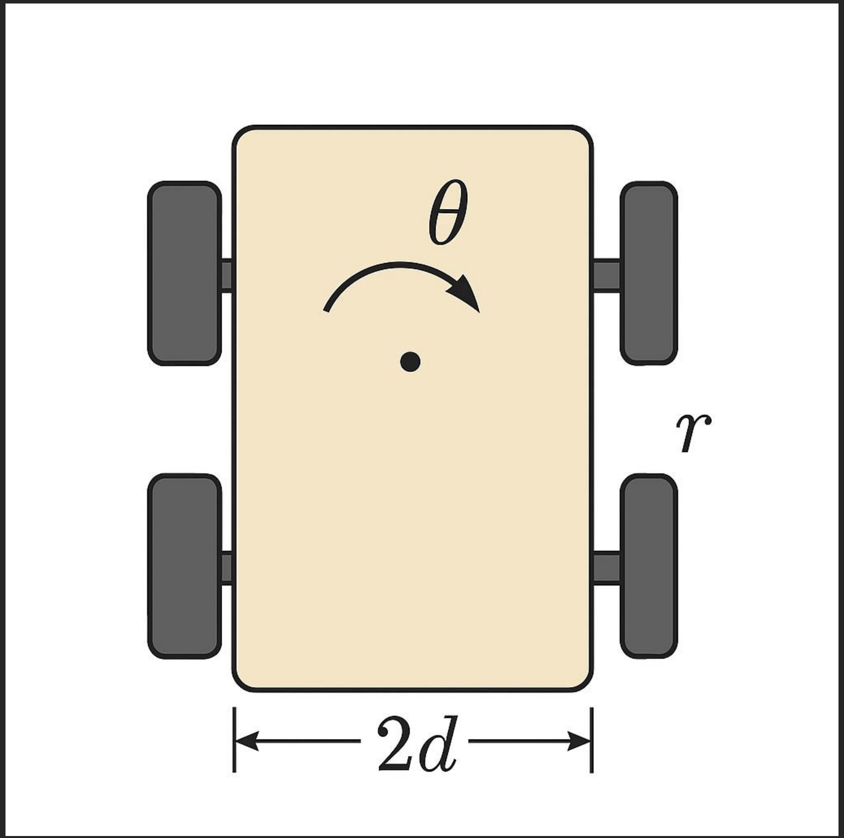
Kalman Filter

- System Equations
  - $x_k = Ax_{k-1} + Bu_k + w_k$
  - $y_k = Hx_k + v_k$
- Filter
  - $\hat{x}_{k|k-1} = A\hat{x}_{k-1} + Bu_k$
  - $P_{k|k-1} = AP_{k-1}A^T + Q$
  - $K_k = P_{k|k-1}A^T(AP_{kk-1}A^T + R)^{-1}$
  - $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - H\hat{x}_{k|k-1}\right)$
  - $P_{k|k} = (I - K_kH)P_{k|k-1}$



Example – Robot Orientation

- Motion Model
  - $I\ddot{\theta} = \tau$
- Friction
  - $I\ddot{\theta} = \mathrm{F} \cdot \mathrm{d} - \mathrm{b}\dot{\theta}$
- Force exerted by wheel
  - $F = \frac{K_r}{rR} \mathrm{V}_{\mathrm{supply}} \cdot \mathrm{u}$
- Equation of motion
  - $\ddot{\theta} = -\frac{b}{I}\dot{\theta} + \frac{K_r}{rRI} \mathrm{V}_{\mathrm{supply}} \cdot \mathrm{u}$

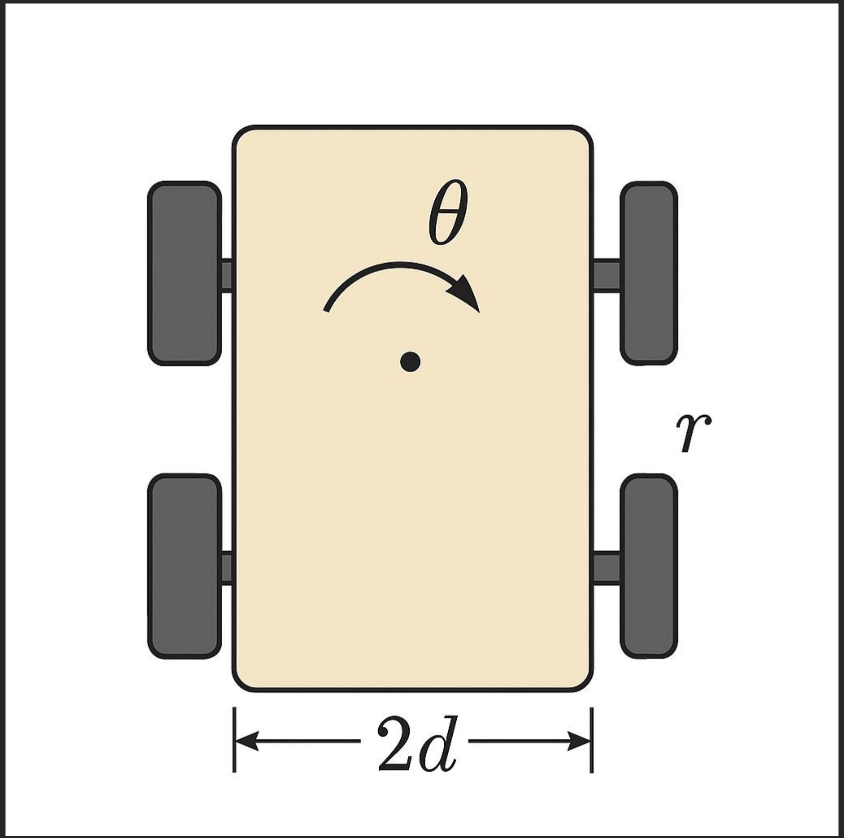


Example – Robot Orientation

- $\ddot{\theta} = -\frac{b}{I}\dot{\theta} + \frac{K_r}{rRI}\mathrm{V_{supply}} \cdot \mathrm{u}$

- $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$

- $\dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{I} \end{bmatrix}}_{\mathrm{A}} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{K_r}{rRI}\mathrm{V_{supply}} \end{bmatrix}}_{\mathrm{B}} u$

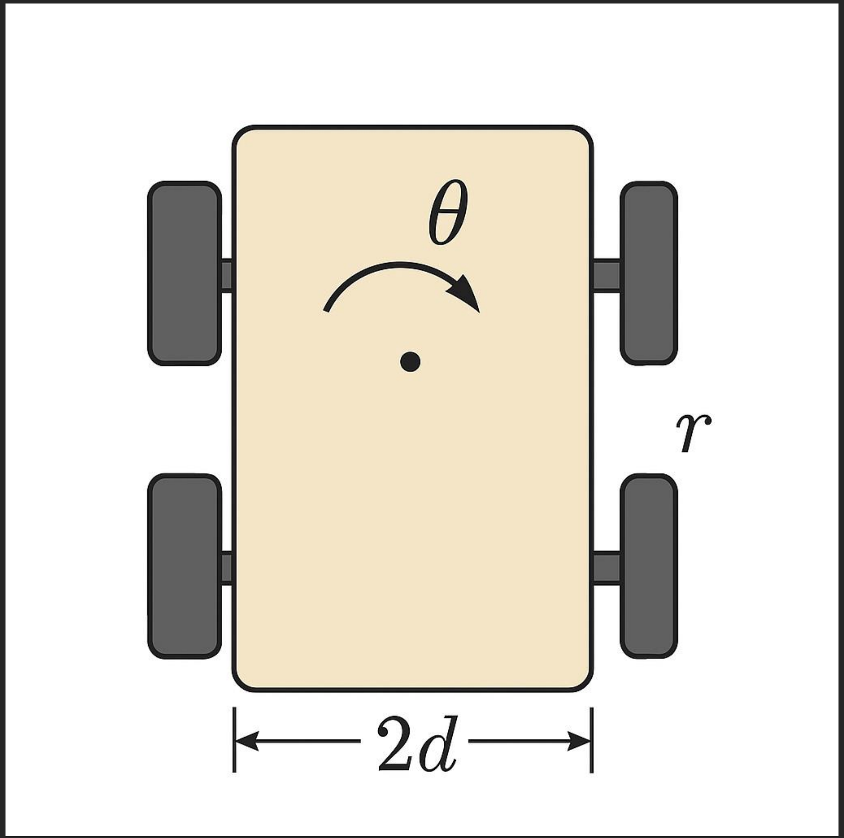


Example – Robot Orientation

- $\dot{x} = A_c x + B_c u$

- $A_c = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{I} \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ \frac{K_r}{rRI} \mathrm{V}_{\mathrm{supply}} \end{bmatrix}$

- $A_d = e^{A_c \Delta t}$
- $B_d = \int_0^{\Delta t} e^{A_c \tau} d\tau B$



Example – Robot Orientation

- $x[k+1] = A_d x[k] + B_d u$

- How do we get A_d and B_d?
  - Analytically or Computationally
- What if we don't know K_r, r, or R?
  - Characterize the system – input a given u and measure the resulting $\ddot{\theta}$
  - Ignore equations of motion and use encoders

- $\theta[k+1] = \theta[k] + \Delta\theta_{encoder}$

Example – Robot Orientation

- System state
  - Orientation
  - Gyroscope bias

- $\theta[k+1] = \theta[k] + \Delta\theta_{encoder} + w_\theta$
- $b[k+1] = b[k] + w_b$

- $x[k+1] = \begin{bmatrix} \theta[k+1] \\ b[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta[k] \\ b[k] \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Delta\theta_{encoder} + \mathrm{Q}$

Example – Robot Orientation

- Sensor – Gyroscope readings

- $y[k] = \theta[k] + b[k] + w_g$

- $\mathrm{y[k]} = [1 \quad 1] \begin{bmatrix} \theta[k+1] \\ b[k+1] \end{bmatrix} + \mathrm{R}$

Example – Robot Orientation

- $x[k+1] = Ax[k] + \mathrm{B}\Delta\theta_{encoder} + \mathrm{Q}$
- $\mathrm{y[k]} = H\mathrm{x[k]} + \mathrm{R}$
- $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H = [1 \;\; 1]$
- Filter
  - $\hat{x}_{k|k-1} = A\hat{x}_{k-1} + Bu_k$
  - $P_{k|k-1} = AP_{k-1}A^T + Q$
  - $K_k = P_{k|k-1}A^T(AP_{kk-1}A^T + R)^{-1}$
  - $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - H\hat{x}_{k|k-1}\right)$
  - $P_{k|k} = (I - K_kH)P_{k|k-1}$