# Adaptive Filtering in Biomedical Signal Processing

## Enhancing Signal Quality for Accurate Healthcare

Exploring Principles, Algorithms, Applications, and Future Directions

Fardin Afdideh

Karolinska Institutet

# Overview

## Foundations

- Historical Context and Evolution
- What are Adaptive Filters?
- Why Biomedical Signals Need Them?

## Theory

- How Adaptive Filters Work?
- Key Concepts: Convergence, Cost Function & SSE
- Pros & Cons
- Linear/Nonlinear Adaptive Filters

## Algorithms

- General Adaptive Filter Algorithm
- LMS, NLMS, RLS, AP, and more
- Selection Criteria for an Adaptive Filter

## Applications

- System Identification, Inverse System Identification, Noise Cancellation, Prediction
- ECG/EEG/EMG Signal Enhancement
- Blind Adaptation
- Trade-offs and Considerations
- Future Directions

## Implementation

- MATLAB & Python tools

## Conclusion

- Conclusion & Future Directions
- Dive Deeper
- Q&A

# Foundations

# Historical Context and Evolution

- Evolved from **Communication Theory** to improve signal quality in noisy environments.

- Adapted for processing physiological signals such as ECG and EEG.

- Significant advancements in medical technology, enabling **precise monitoring and interpretation of complex biological data**.

# What are Adaptive Filters?

- **Real-Time Adaptation/Optimization:** Unlike conventional **Digital Filters** with **fixed coefficients** designed based on **prior knowledge** of the **signal and noise**, adaptive filters are specialized digital filters that **dynamically self-adjust**, without the need for intervention by the user or prior knowledge, their parameters in an **automatic** way to optimize performance in **real-time in varying filter environments**.
- **Key Distinction:** Crucial in environments where **signal characteristics or system conditions are unknown or change over time (non-stationary)**.
- **Core Principle:** By **continuously learning from the incoming data (no prior knowledge needed) and adjusting in real-time their parameters**, they can maintain **optimal or near-optimal** performance even when the **characteristics of the signal and noise evolve**.
- Essential tools in various applications, including telecommunications, audio processing, noise/echo cancellation, system identification, channel equalization, and biomedical signal processing.

# Why Biomedical Signals Need Them?

- In numerous biomedical signals, the statistical properties of the signals of interest and the corrupting noise are not constant over time (**non-stationary**).
- Biomedical signals (ECG, EEG, …) are often contaminated by **unknown** and/or **time-varying** noise and interferences.
  - → Noise sources: muscle tremors, baseline wander, electrical artifacts.
- In this non-stationary context, the traditional fixed filters, which designed based on **static assumptions** and have static filter coefficients, **may remove important physiological information** along with the noise (unwanted information).
- Biomedical signal and noise may have a **spectral overlap**.

# Theory

# How Adaptive Filters Work?

- **Filtering** process:
  - → Applies the current set of filter coefficients to the incoming input signal to generate an output signal.

This **iterative** refinement of the filter coefficients allows the adaptive filter to **progressively** improve its performance **over time**.

- **Adjustment/Adaptation** process:
  - → In order to adapt to new "filter environments", filter coefficients change, based on the **filter's output and a desired signal discrepancy**, to converge to an optimal state.
  - → Adaptation is governed by an **optimization algorithm** with Mean Square Error (**MSE**) as a common cost function.
  - → The error signal is the difference between the filter's output and a desired or target signal.

# Key Concepts: Convergence, Cost Function & Steady-state Error

- Convergence
  - → The process by which the filter coefficients approach their (near-)optimal values, leading to a **minimum in the cost function**
  - → Depends on:
    - **statistical properties of the input** signal,
    - the **specific adaptive algorithm** being used, and
    - the algorithm's **learning rate or step size**.
- Cost Function
  - → **Mathematically formalizes** the filter's performance by providing a scalar measure of **how well the filter is operating**, which is typically chosen to be the MSE
- Steady-state Error
  - → **Residual error after the adaptive filter's convergence**, which arises due to factors such as inherent noise in the system, limitations of the chosen algorithm, or the complexity of the relationship between the input and desired signals.

# Pros & Cons

- Pros
  - → **Adaptability**
    - Unlike fixed filters designed based on static assumptions, adaptive filters can track and cancel interference whose characteristics change over time
  - → **Self-adaptability (autonomous)**
  - → Improved performance (**accuracy & efficiency**)
    - Improved signal extraction, noise reduction, and system identification in unknown or time-varying situations where fixed filters would be inherently limited.
  - → **Real-time** performance
  - → **No requirement for prior information** (prior knowledge of the signal and noise statistics)
- Cons
  - → **Convergence speed**
    - In certain complex or highly non-linear scenarios, they might experience slow convergence or converge to a suboptimal solution
  - → **Computational complexity** (e.g., RLS *vs.* fixed filters)
  - → **Stability**

# Linear Adaptive Filters

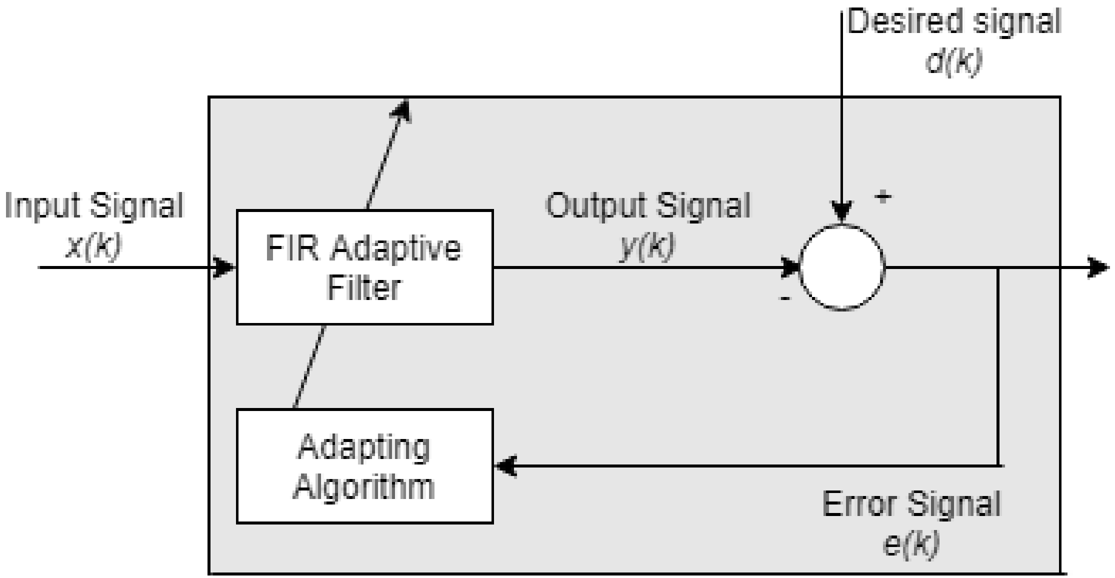
- Linear: A **linear input-output relationship**
  - → **Finite Impulse Response (FIR):** The filter output depends (weighted sum) only on a finite number of **past input** samples (the most common type due to their inherent stability, linear phase characteristics, and ease of implementation).
  - → **Infinite Impulse Response (IIR):** The filter depends on **both the current and past input samples as well as the past output** samples.
  - → **Kalman:** Uses a **recursive Bayesian estimator** for optimal state estimation.
  - → Lattice: Uses a lattice structure composed of a cascade of elementary building blocks called stages.
  - → **Subband**: Decomposes the input signal into multiple frequency subbands using filter banks before performing adaptive filtering and reconstructing the full-band signal.
  - → **Adaptive Linear Combiner (ALC):** The output is obtained by linearly combining a set of input signals using adaptive weights.
  - → Fast Transversal, …

# Nonlinear Adaptive Filters

- Nonlinear: A **nonlinear input-output relationship**
  - → **Volterra:** It includes higher-order terms based on the Volterra series, which is a power series expansion similar to the Taylor series.
  - → **Spline:** Uses B-splines (basis splines) to model and adapt to nonlinearity.
  - → **Neural Network-Based:** Uses artificial neural networks (ANNs) to model nonlinearity.
  - → **Kernel:** To handle nonlinearity in input-output, it uses kernel methods to **map the input signal into a higher-dimensional space** where the relationships between the input and output become linear before performing adaptive filtering.
  - → **Bilinear:** It uses a bilinear (two input signals each affecting the output in a linear fashion) transformation to model nonlinearity.
  - → …

# Algorithms

# General Adaptive Filter Algorithm




https://se.mathworks.com/help/dsp/ug/overview-of-adaptive-filters-and-applications.html

# Least Mean Squares (LMS)

- Simple implementation & low computational complexity: Suitable for real-time implementations and situations with limited resources.
- Effective in noise cancellation and echo reduction.
- Iteratively minimizes the instantaneous error through the gradient descent algorithm, and updating the weights in the direction opposite to the gradient of the MSE.
- Limitations:
  - → Can be slow in convergence, particularly when the input signal is highly correlated (as the optimal range for step-size is theoretically inversely proportional to the largest eigenvalue of the input signal's autocorrelation matrix).
  - → Can have stability issues (oscillate around the optimal solution without converging).
  - → Sensitivity to step-size (learning rate), which controls both the convergence speed and the stability of the algorithm.
  - → Sensitivity to the scaling of the input signal, making the careful selection of the step size parameter crucial.

# Normalized Least Mean Squares (NLMS)

- A variant of LMS
- Improved convergence speed and robustness compared to LMS.
- Effective in non-stationary environments with varying noise levels.
- Normalizes the step size, which often leads to faster convergence and reduced sensitivity to the input signal's power level.
- Increased computational complexity compared to LMS.

# Recursive Least Squares (RLS)

- Uses matrix operations to update filter coefficients.
- RLS (a variant of LMS) *vs.* LMS
  - → Faster convergence.
    - Faster adaptation in rapidly changing systems, and maintaining high accuracy in tracking signals.
  - → Smaller steady state error with respect to the unknown system.
  - → Higher computational complexity.
  - → Convergence to the optimum weights (LMS) *vs.* convergence to the weights minimizing a weighted linear least squares cost function (RLS).
  - → In performance, RLS approaches the Kalman filter in adaptive filtering applications.
  - → More robust to the scaling of the input signal.
  - → Adaptation based on the error at the current time (LMS) *vs.* adaptation based on the total error computed from the beginning (RLS).
    - No memory involved as the older error values are discarded (LMS) *vs.* memory-intensive as all (with a forgetting factor giving exponentially less weight to older error samples) error data is considered in the total error (RLS).

# Affine Projection (AP)

- Generalization of the LMS algorithm by incorporating multiple past input vectors (rather than only the most recent input sample in LMS) in the weight update process.
- Faster and more robust especially in correlated environments compared to the LMS.
- Higher computational complexity and memory-intensive compared to the LMS.
- A balance between convergence speed and computational complexity compared to the RLS.
- Other LMS's variants:
  - → Variable Step Size LMS (VSS-LMS): Instead of using a fixed step size, VSS-LMS updates the step size in order to significantly reduce the output error.
  - → Block LMS: An adaptive LMS filter, where the adaptation of filter weights occurs once for every block of samples.
  - → Filtered-X LMS (FxLMS): Takes the secondary path, which is the path from the output of the adaptive filter to the error signal, into consideration, to implement real-world ANC.

# Additional Algorithms

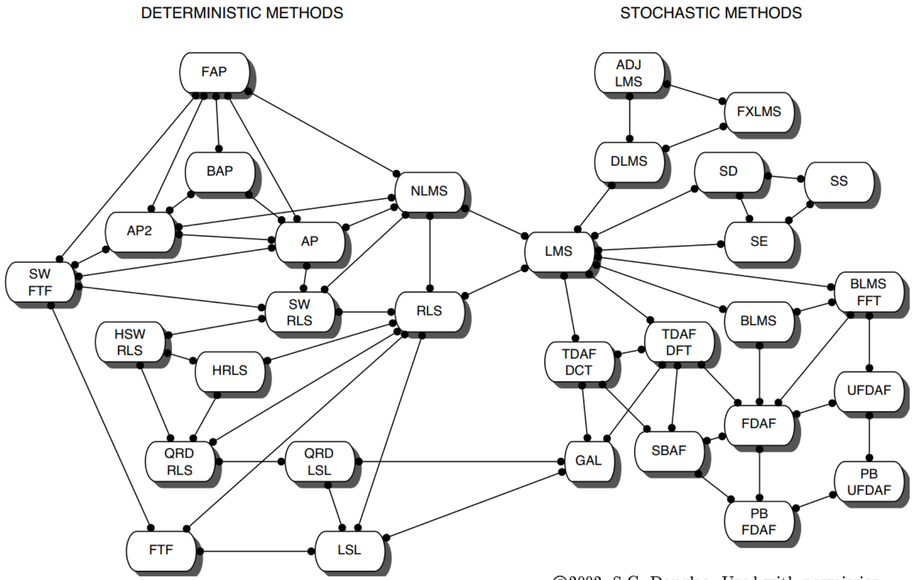
- Gradient Adaptive Lattice (GAL): Efficient parameter estimation and numerical stability.
- Fractional Tap-Length (FTL): Dynamically adjusts tap-lengths for enhanced performance.
- Frequency-Domain Adaptive Filter
- Hybrid Techniques: Combining multiple algorithms (e.g., LMS with RLS) can leverage strengths and mitigate limitations.

# Additional Algorithms

Table 1: Adaptive algorithms within the adaptive filters toolset.

| *Acronym* | *Algorithm Name* | *Ref.* | *Acronym* | *Algorithm Name* | *Ref.* |
|---|---|---|---|---|---|
| | *Simple Updates* | | | *Delayed Updates* | |
| LMS | least-mean-square | [1] | DLMS | delayed LMS | [13] |
| SE | sign-error | [6] | FXLMS | filtered-X LMS | [10] |
| SD | sign-data | [4] | ADJLMS | adjoint LMS | [26] |
| SS | sign-sign | [2] | | *Projection Methods* | |
| | *Block and Frequency Domain Methods* | | NLMS | normalized LMS | [3] |
| BLMS | block LMS | [18] | AP | affine projection $\mathcal{O}(N^3)$ | [8] |
| BLMSFFT | block LMS (FFT) | [18] | AP2 | affine projection $\mathcal{O}(N^2)$ | [15] |
| FDAF | frequency-domain adaptive filter | [18] | FAP | fast affine projection | [23, 24] |
| UFDAF | unconstrained FDAF | [18] | BAP | block affine projection | [21] |
| PBFDAF | partitioned-block FDAF | [14] | | *Conventional Least-Squares Methods* | |
| PBUFDAF | partitioned-block unconstrained FDAF | [14] | RLS | recursive least-squares | [25] |
| | *Transform Domain and Subband Methods* | | SWRLS | sliding-window RLS | [25] |
| TDAFDFT | transform-domain adaptive filter (DFT) | [18] | | *Square-Root Least-Squares Methods* | |
| TDAFDCT | transform-domain adaptive filter (DCT) | [18] | QRDRLS | QR-decomposition RLS | [25] |
| SBAF | subband adaptive filter | [29] | HRLS | Householder RLS | [27, 28] |
| | *Lattice Structures* | | HSWRLS | Householder sliding-window RLS | [28] |
| GAL | gradient adaptive lattice | [5] | | *Fast Transversal Filters* | |
| LSL | least-squares lattice | [17] | FTF | fast transversal filter | [16] |
| QRDLSL | QR-decomposition least-squares lattice | [17] | SWFTF | sliding-window fast transversal filter | [19] |

FIG. 1: TAXONOMY OF ADAPTIVE FILTERS



# Selection Criteria for an Adaptive Filter

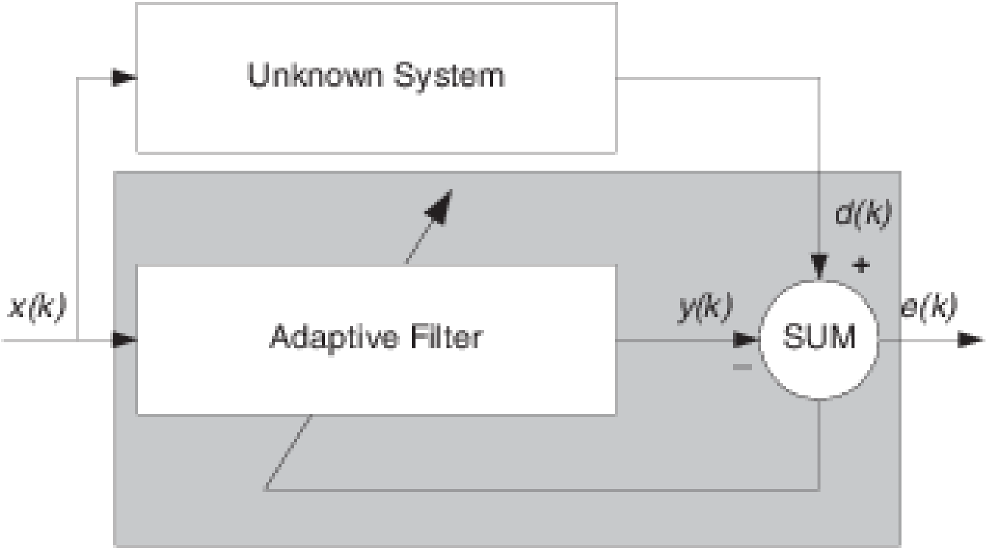
- Convergence speed
- Computational complexity
- Stability
- Performance
- Application constraints
  - → computational resources,
  - → desired convergence speed,
  - → nature of the signals being processed.

# Applications

# System Identification

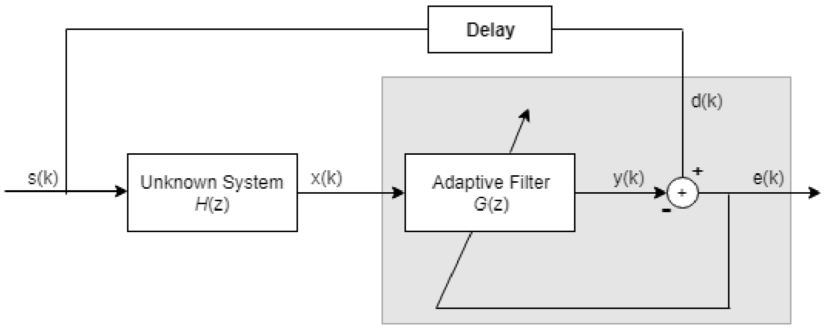
- Identifying an unknown system by placing it in parallel with the adaptive filter

- Applications include identifying communication channels or frequency response of a room



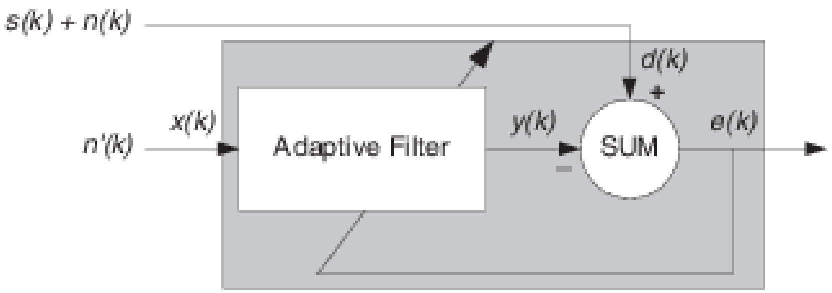
# Inverse System Identification (Equalization)

- Adaptive filter adapts to become the inverse of the unknown system
- Unknown system in series with the adaptive filter
- Requires a delay in the desired signal path to maintain causality
- Used in telephone systems to compensate for the distortion introduced by a communication channel



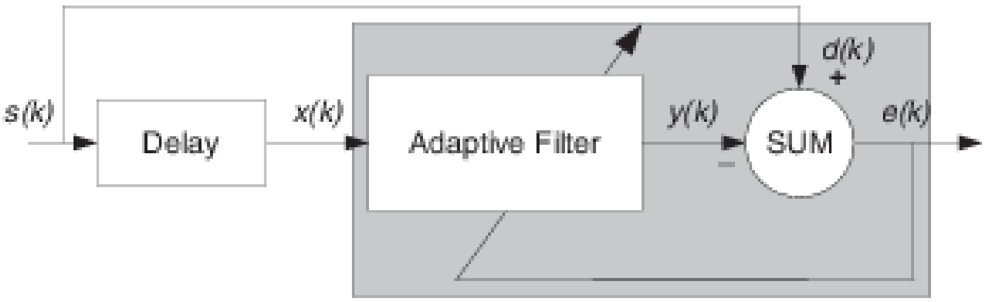
# Noise Cancellation

- Removes noise from a signal in real-time
- Requires a reference signal correlated with the noise
  → Misadaptation: if the reference signal is not sufficiently correlated with the noise
- Error signal converges to the input data signal



# Prediction

- Predicts future values of a periodic signal
- Assumes the signal is steady or slowly varying and periodic
- Uses past values to predict future values



# ECG Signal Enhancement

- Improving the SNR and accuracy of heart disease diagnoses (arrhythmia detection)
- Non-invasive fetal ECG (fECG) extraction:
  - → Maternal ECG (mECG) from the maternal chest as a reference, with the abdominal recording (strong mECG + weak fECG) as the input signal
- Noise/Artifact cancellation in (portable) ECG systems
  - → Power line interference reduction:
    - Adaptive filters can effectively remove this narrowband interference without significantly distorting the underlying ECG signal, especially when the exact frequency of the interference may fluctuate slightly
    - Adaptive line enhancement (ALE): A specialized type of filter not requiring a separate reference signal.
  - → Muscle artifacts reduction: A filter with a reference signal that is correlated with the muscle activity, which is a high-frequency noise, can remove muscle artifacts.
  - → Motion artifacts reduction: The data from motion sensors, e.g., accelerometers, can form the reference signal.
  - → Baseline wander (a low-frequency drift in the ECG baseline often caused by respiration or patient movement) reduction.

# EEG Signal Enhancement

- Noise/Artifact cancellation in EEG systems
  - → Eye blinks artifacts reduction: simultaneously recorded EOG signals as a reference signal
  - → Cardiac artifacts reduction: concurrently recorded ECG signal as a reference signal
  - → Power line interference reduction
  - → Baseline drift reduction
  - → Muscle/Head movements interference reduction, …
- Brain wave patterns recognition enhancement in EEG decoding applications such as BCI
- Real-time EEG monitoring during surgeries or sleep studies

# EMG Signal Enhancement

- Enhancement: Removing artifacts such as ECG interference, unwanted/ involuntary motion artifacts, …
  - → EMG signal analysis in patients with movement disorders, e.g., cervical dystonia (involuntary muscle contractions in the neck)

# Other Biomedical Applications

- Feedback cancellation in hearing aids
- Motion artifacts cancellation in PPG processing

# Blind Adaptation: A Workaround for Real-world Applications

- There are real-world situations in which the desired signal is never available.
- Adaptation without a reference (desired) signal, relying instead on only the **statistical properties of a "hypothetical" desired signal**

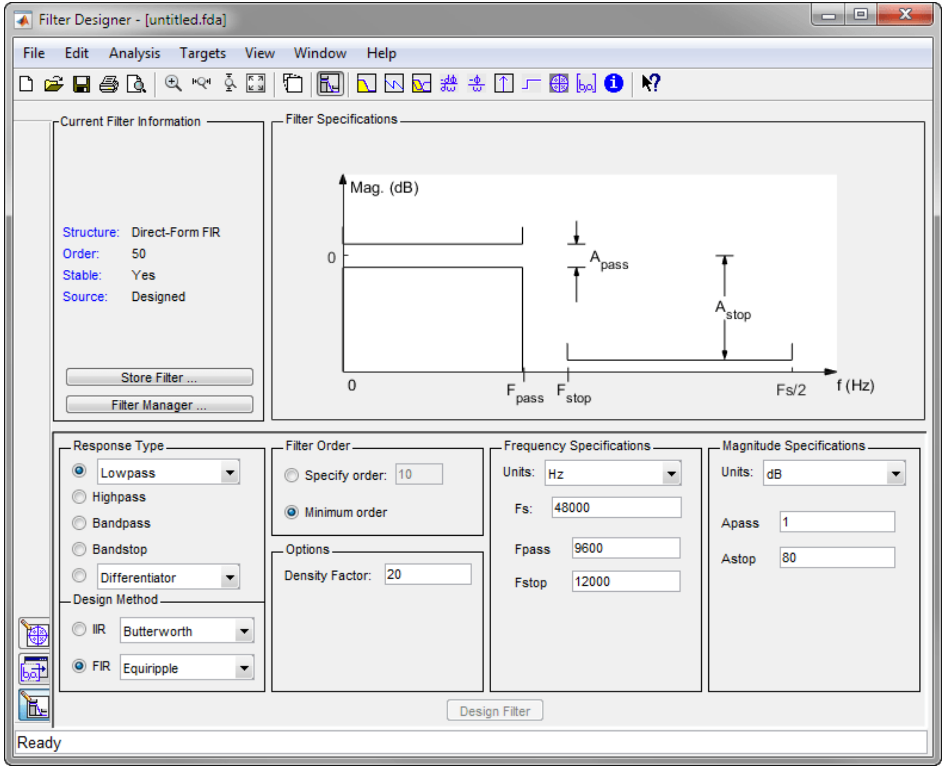
# Trade-offs and Considerations

- Over-filtering: Can inadvertently remove important physiological information.
- Managing Trade-offs: Adaptive filters allow for effective management of the trade-off, providing a realistic representation of physiological signals.
- Power Consumption: Efficient power usage prolongs battery life and minimizes heat generation, critical for portable devices.
- Selection of step size: Its improper selection might make the convergence speed unnecessarily slow or introduce excess MSE.

# Implementation

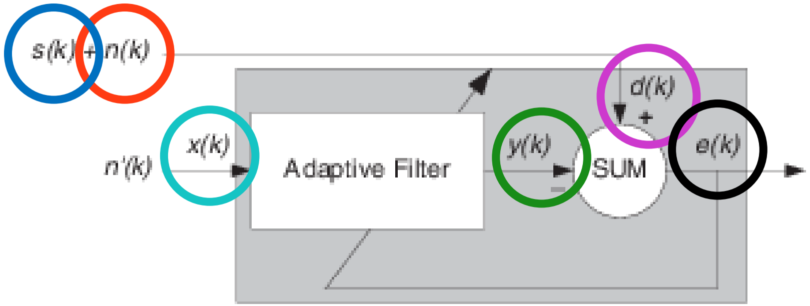
# MATLAB Tools

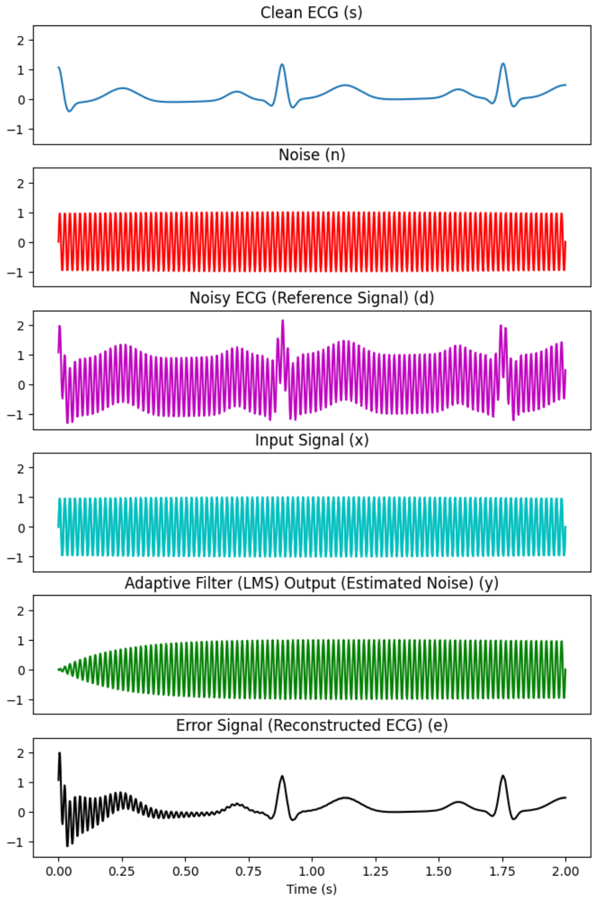
- ”Filter Designer” GUI  in ”Signal Processing Toolbox™” -> fixed/static-coefficient digital filters (no adaptive filters)
- “adaptfilt.algorithm() in “Filter Design Toolbox” -> adaptive filters, but obsolete
- “dsp.algorithm() in “DSP System Toolbox™” -> adaptive filters



# Python Tools

- Power Line Interference Cancellation of an ECG Signal using Adaptive Filters
  - → NeuroKit2: Python Toolbox for Neurophysiological Signal Processing
  - → Padasip: Python Adaptive Signal Processing
  - → [Notebook]



# Conclusion

# Conclusion

- Adaptive filters are indispensable for enhancing the **accuracy** and **reliability** of physiological data analysis.
- Driving innovations in **patient monitoring** and **diagnostic systems**.

# Future Directions

- Integration with:
  - → Modern signal processing techniques, e.g., Machine Learning
  - → Traditional signal processing techniques, e.g., wavelet transform, empirical mode decomposition (EMD), …
- Improved efficiency in real-time applications
  - → **Sparse** adaptive filtering is a more efficient adaptive filter with reduced computational and memory requirements.
- Improved **robustness** and **stability**
- Energy-efficient Implementations in technologies and hardware, e.g., IoT, autonomous systems, …
  - → Implementation on resource-limited wearable and implantable biomedical devices without compromising performance
- Enhanced noise cancellation algorithms

# Dive Deeper


- Widrow, B., Stearns, S. D. (1985). Adaptive Signal Processing. India: Prentice-Hall.
- Henderson (2021): Adaptive techniques in audio analysis and processing.
- Puglisi (2023): Integration of deep learning methods into adaptive filtering for enhanced audio signal processing.
- Prabhakaran and Varughese (2023): Outlines the fundamental concepts of adaptive filters.
- Rangayyan, R.M., Krishnan, S. (2024). Biomedical Signal Analysis.
- Baraniak, J., Hauer, J., Schuhmann, N., and Leugering, G (2007). Implementation of Adaptive Filters for Biomedical Applications.

# Q&A

- Thank you!
- Questions?

Karolinska
Institutet