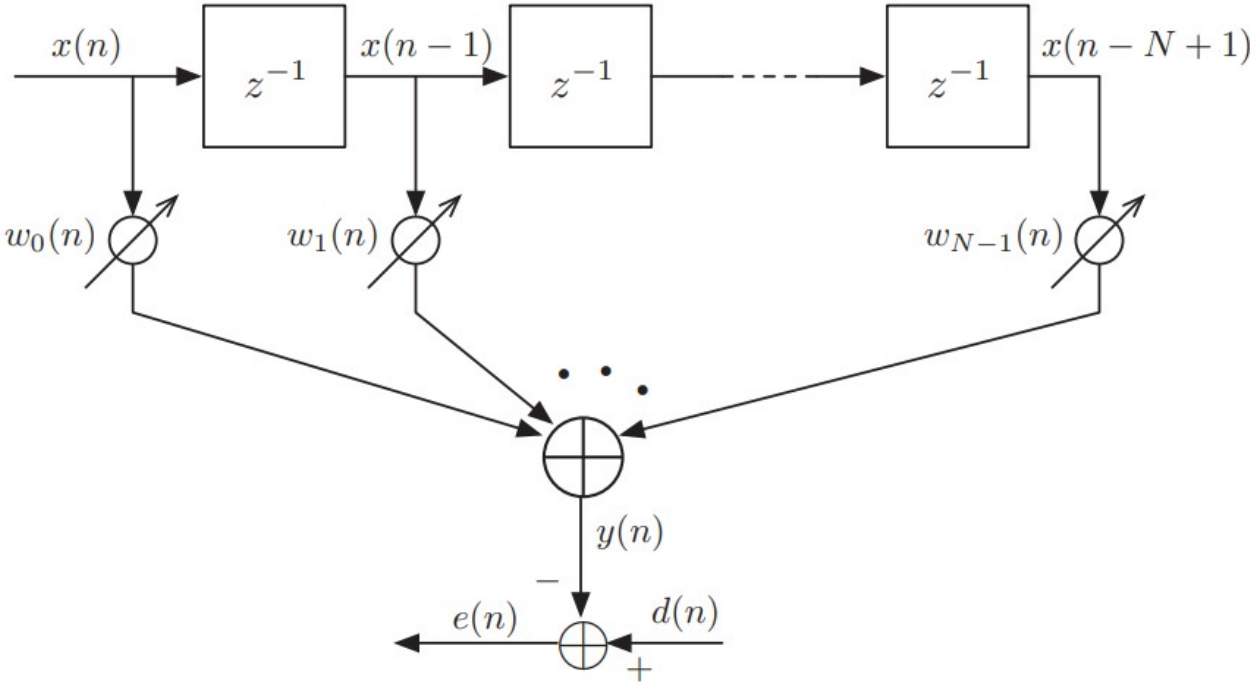
ASP Midterm Project

潘俊宏N96111037 郭承宗Q36134027 陳致融E14109022

Agenda

- LMS Algorithm
- NLMS Algorithm
- APA Algorithm
- Performance Analysis & Conclusion
  - LMS versus NLMS
    - Fixed frequency & Dynamic SNR
    - Fixed SNR & Dynamic frequency
  - APA versus NLMS
    - Convergence Test
    - Fixed SNR & Dynamic frequency
    - Fixed frequency & Dynamic SNR

LMS algorithm



$$y(n) = \sum_{i=0}^{N-1} w_i(n)x(n-i)$$

$$e(n) = d(n) - y(n)$$

Replace x(n), w(n) representation

$$x(n) = \left[x(n)x(n-1)\cdots x(n-N+1)\right]^{\mathrm{T}}$$

$$w(n) = \left[w(n)w(n-1)\cdots w(n-N+1)\right]^{\mathrm{T}}$$

LMS algorithm

Input: Tap-weight vector, $\mathbf{w}(n)$,
Input vector, $\mathbf{x}(n)$,
and desired output, $d(n)$

Output: Filter output, $y(n)$,
Tap-weight vector update, $\mathbf{w}(n+1)$

---

1. Filtering:

$$y(n) = \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)$$

2. Error estimation:

$$e(n) = d(n) - y(n)$$

3. Tap-weight vector adaptation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n)$$

LMS algorithm

Derivation of LMS algorithm

Cost function $\xi = E\left[e^2(n)\right]$ Replace estimate $\hat{\xi} = e^2(n)$

By steepest - descent method

Substituting ξ in the steepest-descent recursion $w(n+1) = w(n) - \mu\nabla e^2(n)$, μ is the step-size

∇ is the gradient operator $\nabla = \left[\frac{\partial}{\partial w_0}\frac{\partial}{\partial w_1}\cdots\frac{\partial}{\partial w_{N-1}}\right]^T$ $e(n) = d(n) - y(n)$ $\nabla\frac{\partial^2 e(n)}{\partial w_i} = -2e(n)\frac{\partial y(n)}{\partial w_i}$

$w(n+1) = w(n) - \mu\nabla e^2(n)$ $w(n+1) = w(n) + 2\mu e(n)x(n)$

The eminent feature of the LMS algorithm, is its simplicity.

Its implementation requires, **2N + 1 multiplications and 2N additions**.

LMS algorithm

LMS algorithm convergence

optimum tap-weight vector w_o ⟹ definition **weight-error vector** $v(n) = \mathrm{w}(n) - w_o$

$$e(n) = d(n) - \mathrm{w}^{\mathrm{T}}(n)x(n) = d(n) - x^{\mathrm{T}}(n)w(n)$$

$$= d(n) - x^{\mathrm{T}}(n)w_o - x^{\mathrm{T}}(n)\left(w(n) - w_o\right) = \mathrm{e}_o(n) - x^{\mathrm{T}}(n)v(n) \quad \text{where} \quad \mathrm{e}_o(n) = d(n) - x^{\mathrm{T}}(n)w_o$$

optimum recursion equation

$$v(n+1) = \left[I - 2\mu x(n)x^{\mathrm{T}}(n)\right]v(n) + 2\mu e_o(n)x(n)$$

orthogonality

$$E\left[v(n+1)\right] = E\left[\left(I - 2\mu x(n)x^{\mathrm{T}}(n)\right)v(n)\right] + 2\mu E\left[e_o(n)x(n)\right] = E\left[\left(I - 2\mu x(n)x^{\mathrm{T}}(n)\right)v(n)\right]$$

independent ⟹ $E\left[x(n)x^{\mathrm{T}}(n)v(n)\right] = E\left[x(n)x^{\mathrm{T}}(n)\right]E\left[v(n)\right]$

$E\left[v(n+1)\right] = (\mathrm{I} - 2\mu R)E[v(n)]$ where $R = \mathrm{E}[\mathrm{x}(\mathrm{n})\mathrm{x}^{\mathrm{T}}(\mathrm{n})]$ is correlation matrix of the input vector x(n)

Convergence to zero $0 < \mu < \dfrac{1}{\lambda_{\max}}$ ⟹ $\lambda_{\max}$ is the maximum eigenvalue of R

NLMS algorithm

Normalized LMS algorithm

LMS algorithm problem : When the adjustment amount of the tap weight (N) is proportional to the input vector x(n)

⇩

gradient noise amplification problem (梯度躁聲放大)

LMS recursion equation $w(n+1) = w(n) + 2\mu e(n)x(n)$, μ is the step-size

⇩ Idea : Modify the step size μ in the LMS algorithm, make it have time-varying μ(n)

$$w(n+1) = w(n) + \mu(n)e(n)x(n)$$

Considered a constrained optimization problem in LMS

$$\min_{w(n+1)} \left\| w(n+1) - w(n) \right\|^2$$

$$\text{subject to } w^{T}(n+1)x(n) = d(n)$$

⇒ Use Lagrange multiplier method solve it

NLMS algorithm

Lagrange multiplier method

Cost function $\left[w^{\mathrm{T}}(n+1)-w^{\mathrm{T}}(n)\right]\left[w(n+1)-w(n)\right]+\lambda\left[d(n)-w^{\mathrm{T}}(n+1)x(n)\right]$

⇩ derivative by $w^{\mathrm{T}}(n+1)$ to find extremum value

$$w(n+1)-w(n)+\lambda x(n)=0 \quad \text{substitute into} \quad w^{\mathrm{T}}(n+1)x(n)=d(n)$$

$$\left[w^{\mathrm{T}}(n)+\lambda x^{\mathrm{T}}(n)\right]x(n)=d(n)$$

$$\lambda=\frac{d(n)-w^{\mathrm{T}}(n)x(n)}{\|x(n)\|^2}=\frac{e(n)}{\|x(n)\|^2} \Longrightarrow w(n+1)=w(n)-\frac{\tilde{\mu}}{\|x(n)\|^2}x(n)e(n)$$

time-varying μ(n)

In developing the normalize LMS algorithm, when $\|x(n)\|^2$ is small, numerical difficulties may arise.

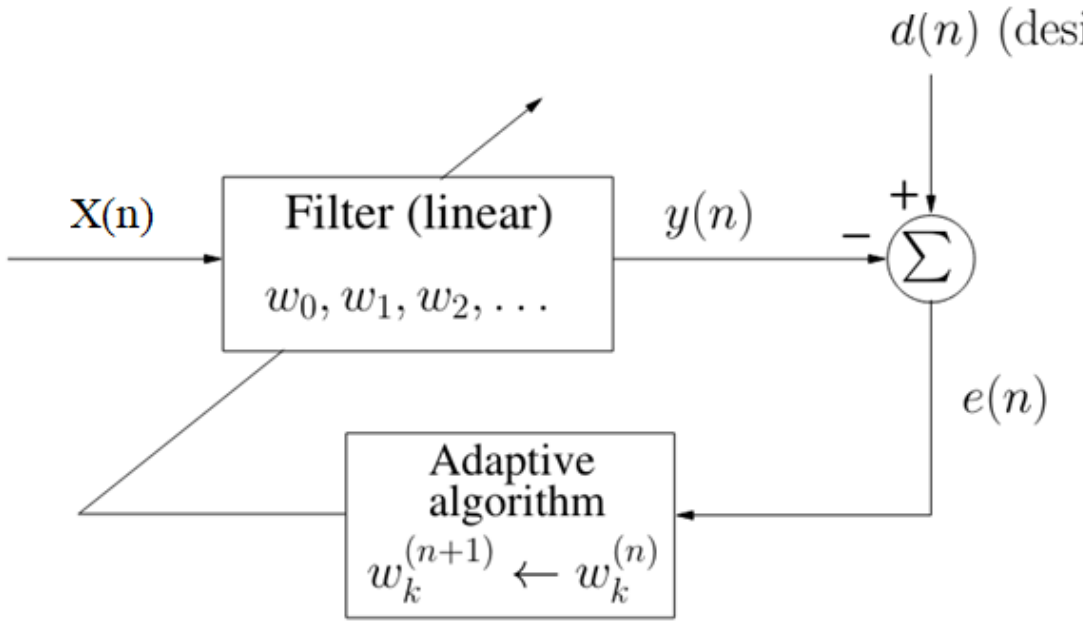
To overcome this problem, we add a relax value (ε) in denominator.

Recursive equation $w(n+1)=w(n)-\frac{\tilde{\mu}}{\|x(n)\|^2+\varepsilon}x(n)e(n)$

Note : The eminent feature of the NLMS algorithm, it is more complexity than LMS algorithm. Its implementation requires

3N + 1 multiplications

Affine projection algorithm



$$\mathbf{X}(n) = [\mathbf{x}(n)\mathbf{x}(n-1)\ldots\mathbf{x}(n-M+1)]$$
$$\mathbf{d}(n) = [d(n)d(n-1)\cdots d(n-M+1)]^{\mathrm{T}}$$
$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^{\mathrm{T}}(n)\mathbf{w}(n)$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \tilde{\mu}\mathbf{X}(n)\left(\mathbf{X}^{\mathrm{T}}(n)\mathbf{X}(n) + \psi\mathbf{I}\right)^{-1}\mathbf{e}(n)$$

Affine projection algorithm

$w^T(n+1)x(n-k) = d(n-k), k = 0,1, \dots M-1 \quad M$ constraints

can be written as $\rightarrow X^T(n)w(n+1) = d(n)$

where $X(n) = [x(n)\, x(n-1) \dots x(n-M+1)]$ and $d(n) = [d(n)\, d(n-1) \dots d(n-M+1)]^T$

by method of Lagrange multipliers

$\rightarrow \xi^C = \|w - w(n)\|^2 + (X^T(n)w - d(n))^T\lambda$ and $\nabla_w \xi^C = 0 \quad \nabla_\Lambda \xi^C = 0$

w is replaced by $w(n+1)$ then we have

$\rightarrow 2\big(w(n+1) - w(n)\big) + X(n)\lambda = 0$ and $X^T(n)w(n+1) - d(n) = 0$

define $e(n) = d(n) - X^T(n)w(n)$

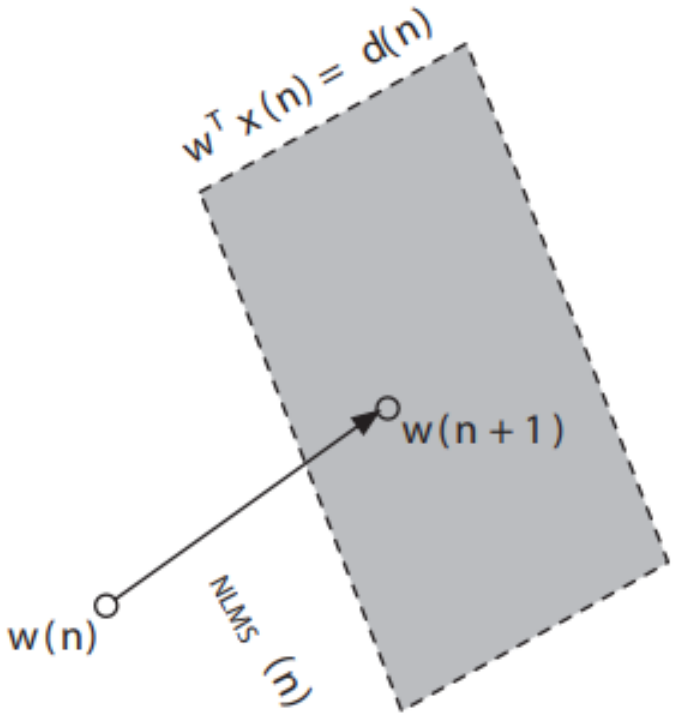
we can obtain $\rightarrow \quad \lambda = -2\big(X^T(n)X(n)\big)^{-1}e(n)$

then we substitute into $2\big(w(n+1) - w(n)\big) + X(n)\lambda = 0$ and add $\tilde{\mu}$ and ψ we can get

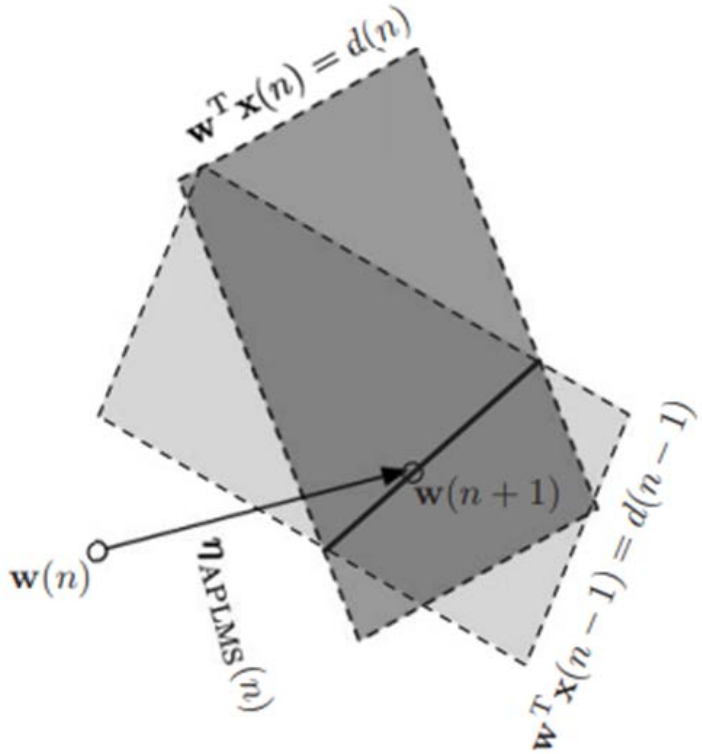
$\rightarrow w(n+1) = w(n) + \tilde{\mu}X(n)(X^T(n)X(n) + \psi I)^{-1}e(n)$

$\tilde{\mu}$ is step size parameter and ψ is numerical stability parameter

Affine projection algorithm



Geometrical interpretation of the NLMS



2 constraints of the Affine projection algorithm

Affine projection algorithm

Input: Tap-weight vector, $\mathbf{w}(n)$,
Input matrix, $\mathbf{X}(n) = [\mathbf{x}(n)\mathbf{x}(n-1) \quad \cdots \mathbf{x}(n-M+1)]$,
and Desired output vector, $\mathbf{d}(n) = [d(n) \quad d(n-1) \cdots d(n-M+1)]^{\mathrm{T}}$

Output: Filter output, $y(n)$,
Tap-weight vector update, $\mathbf{w}(n+1)$

---

1. Filtering:

$$\mathbf{y}(n) = \mathbf{X}^{\mathrm{T}}(n)\mathbf{w}(n), \; y(n) = \text{the first element of } \mathbf{y}(n)$$

2. Error estimation:

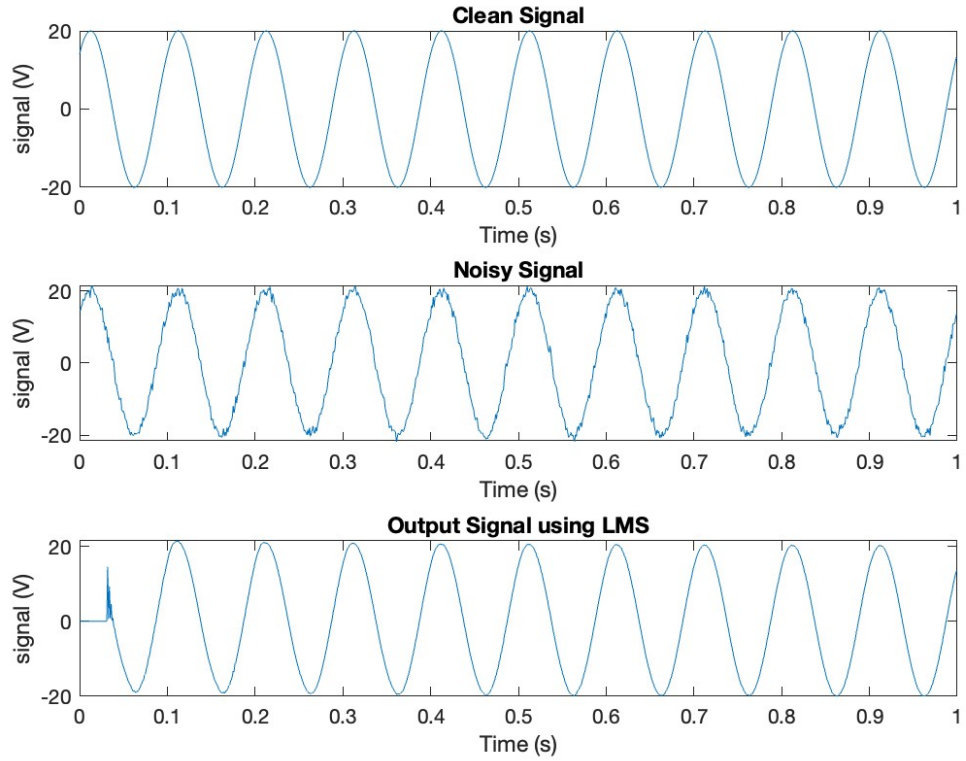
$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$$

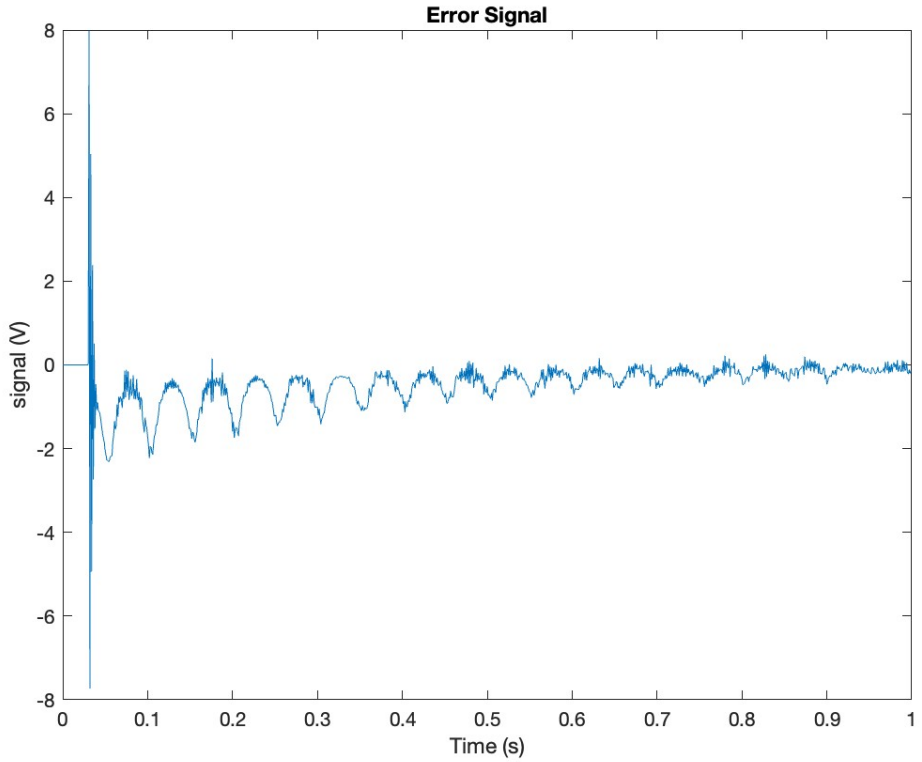
3. Tap-weight vector adaptation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \tilde{\mu}\mathbf{X}(n)(\mathbf{X}^{\mathrm{T}}(n)\mathbf{X}(n) + \psi\mathbf{I})^{-1}\mathbf{e}(n)$$

Performance Analysis

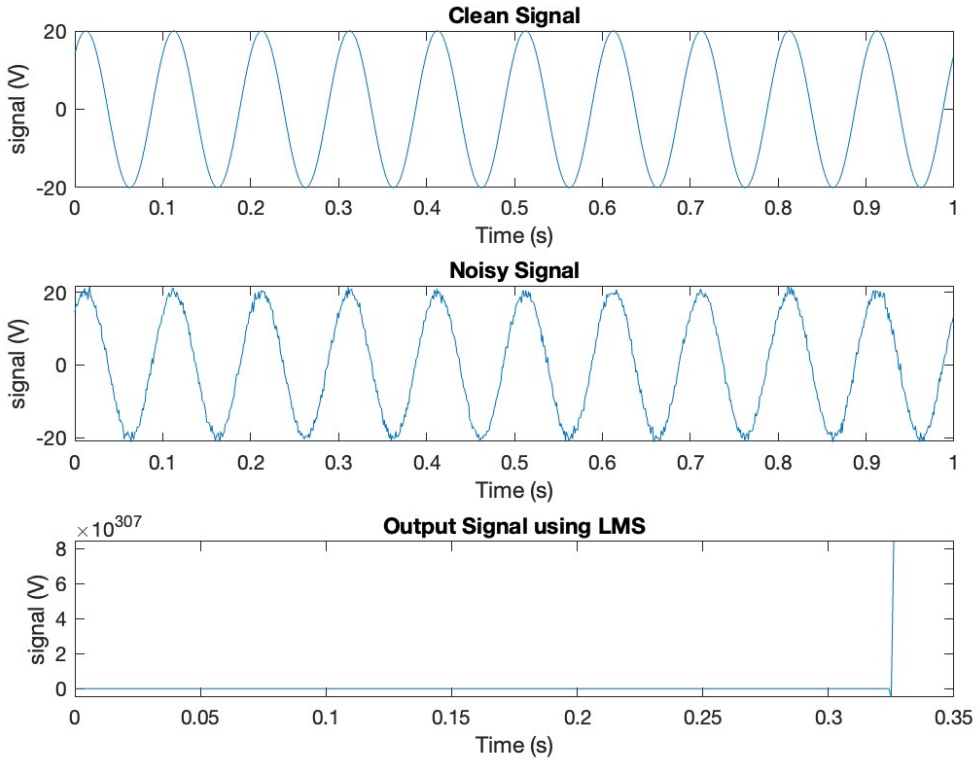
- LMS
  - $\mu = 0.0001, SNR = 25, f = 10$
  - Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
  - Noice $\sim N(0,0.64)$

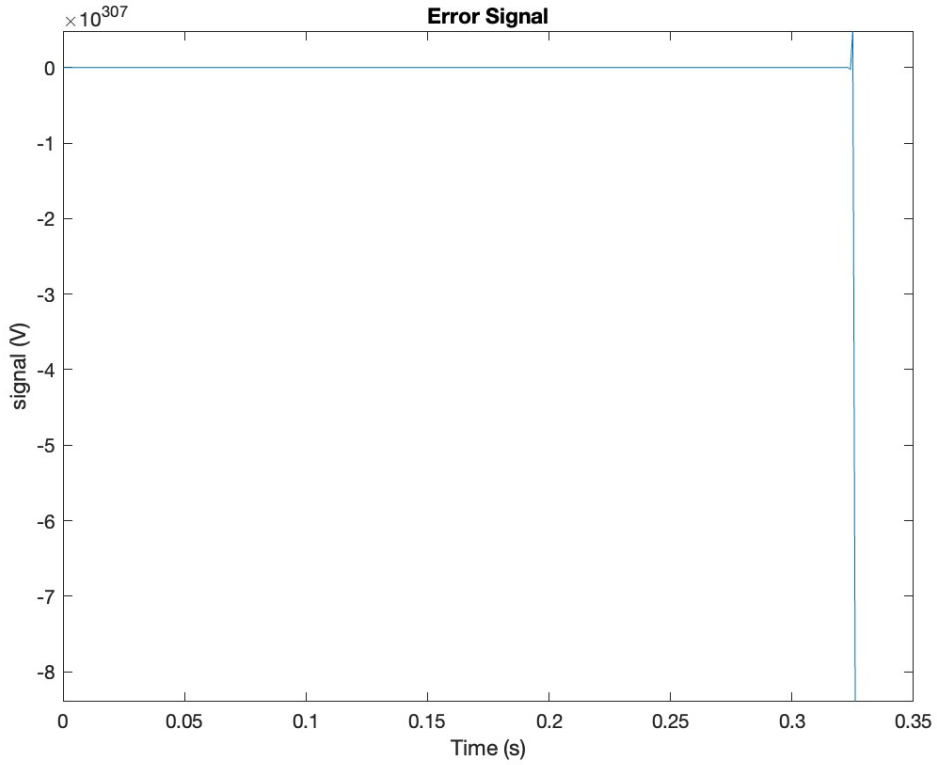




Performance Analysis

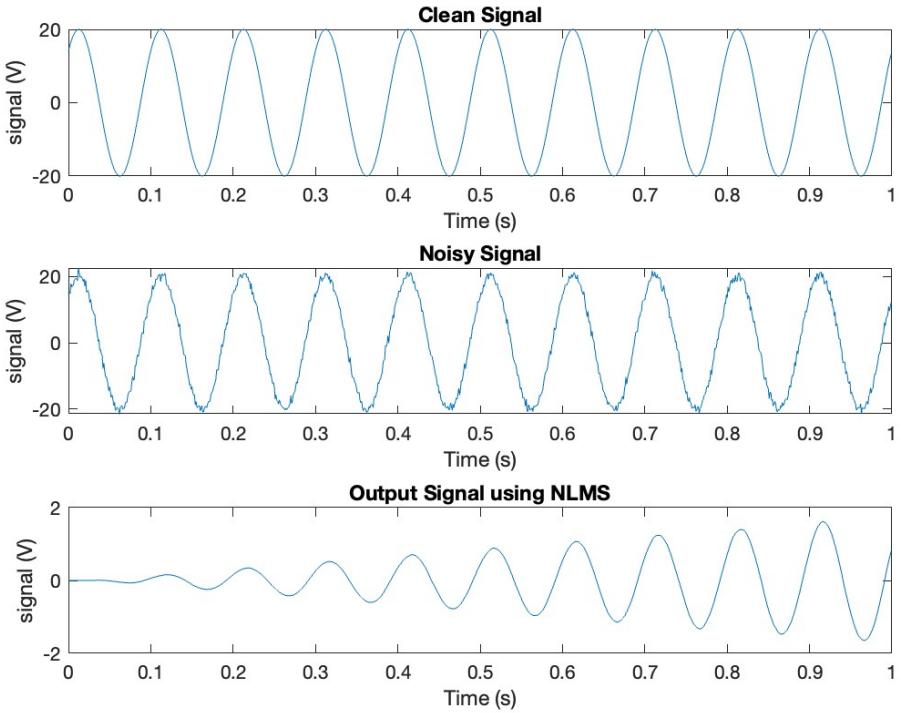
- LMS
  - $\mu = 0.001, SNR = 25, f = 10$
  - Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
  - Noice $\sim N(0,0.64)$

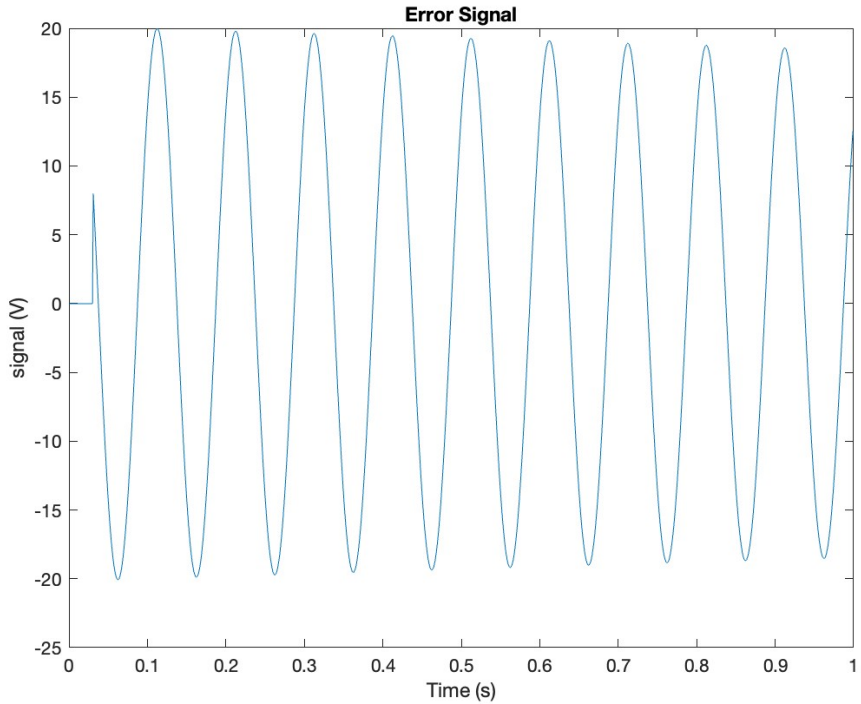




Performance Analysis

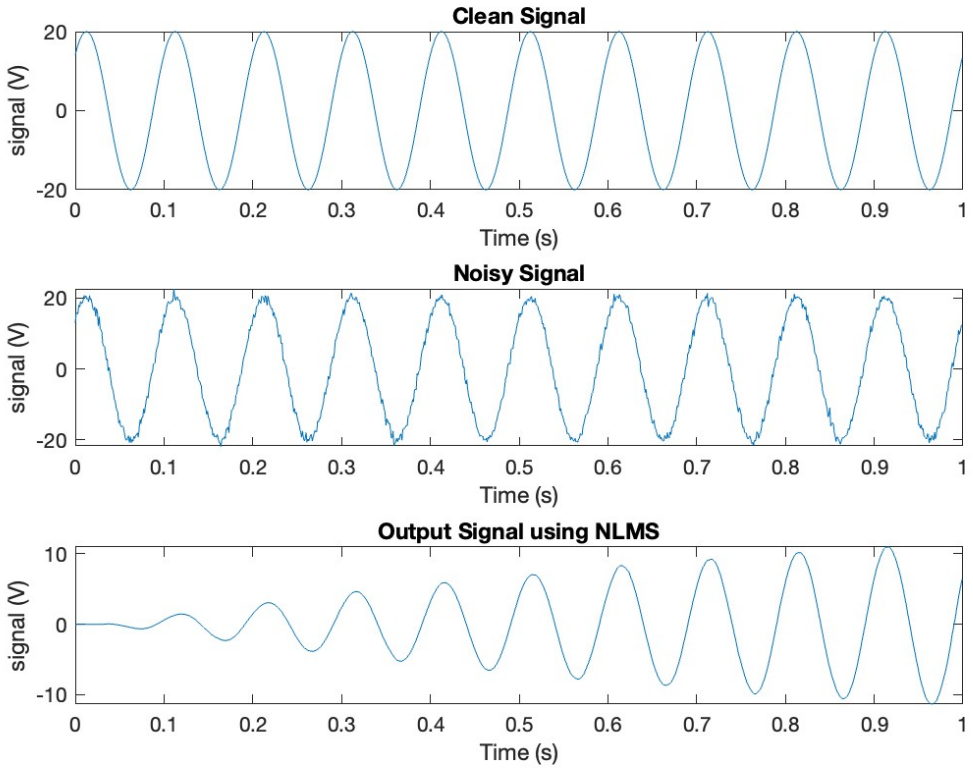
- NLMS
  - $\mu = 0.0001, SNR = 25, f = 10$
  - Clean Signal $= A\cos(2\pi f t) + A\sin(2\pi f t)$
  - Noice $\sim N(0,0.64)$

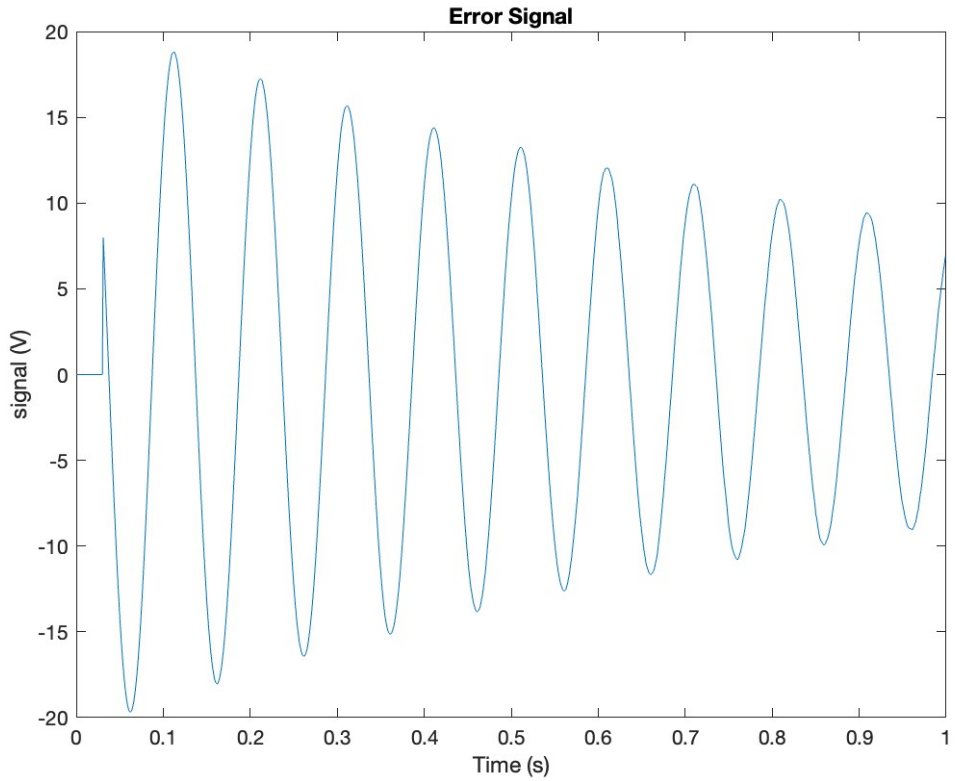




Performance Analysis

- NLMS
  - $\mu = 0.001, SNR = 25, f = 10$
  - Clean Signal $= A\cos(2\pi f t) + A\sin(2\pi f t)$
  - Noice $\sim N(0,0.64)$



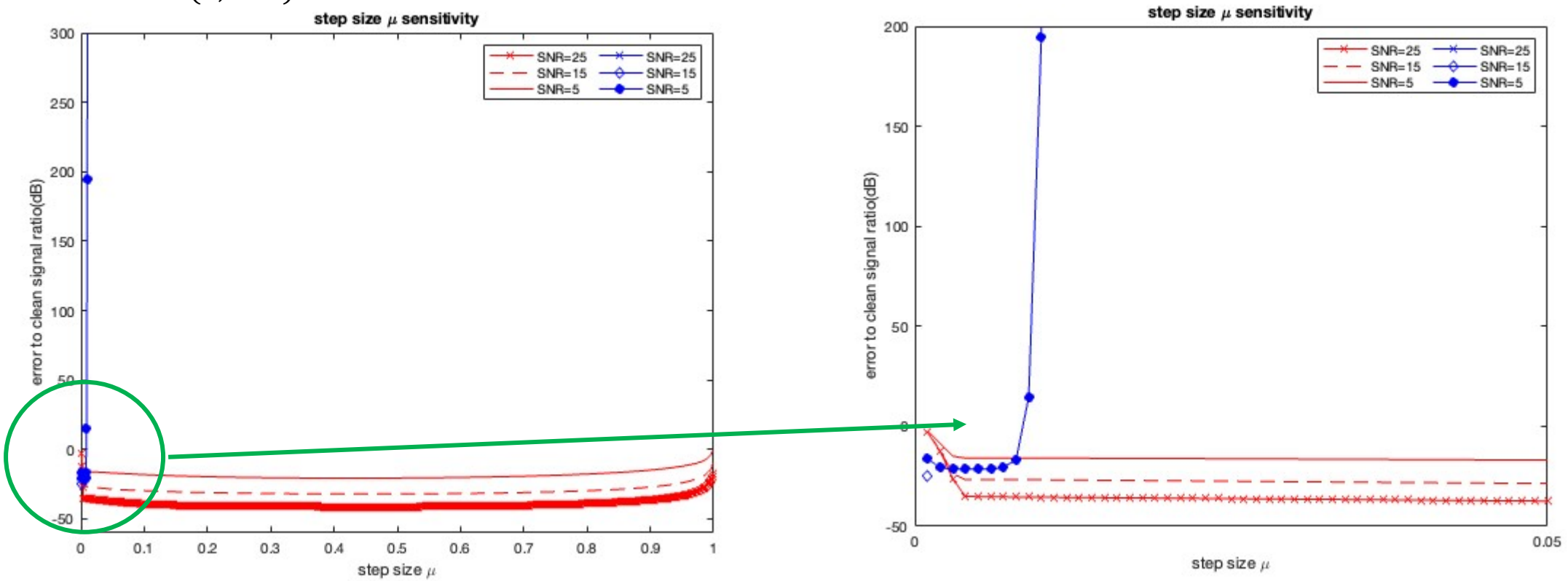


Performance Analysis

LMS versus NLMS

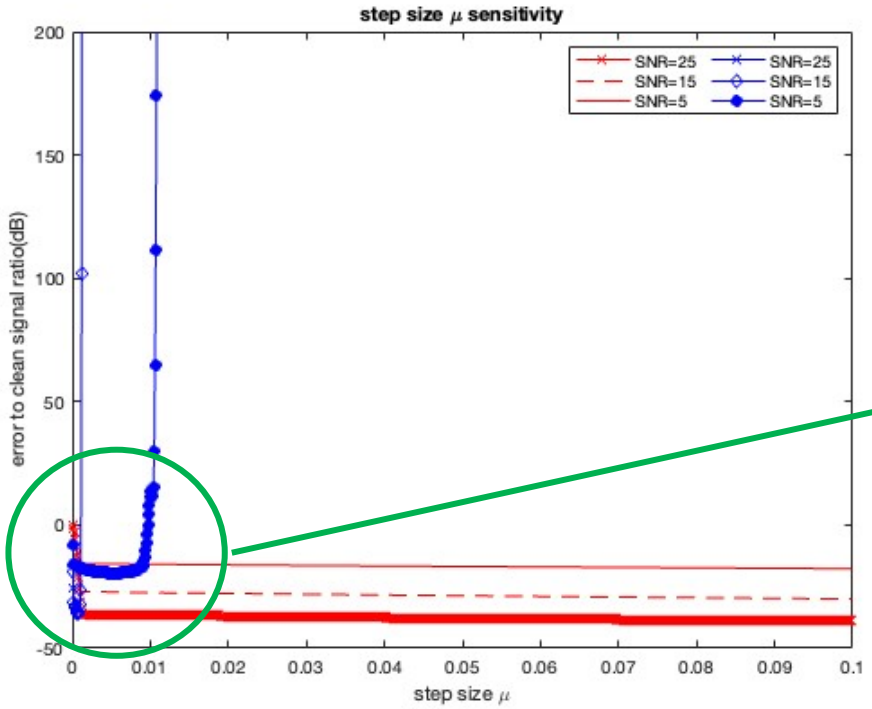
Performance Analysis

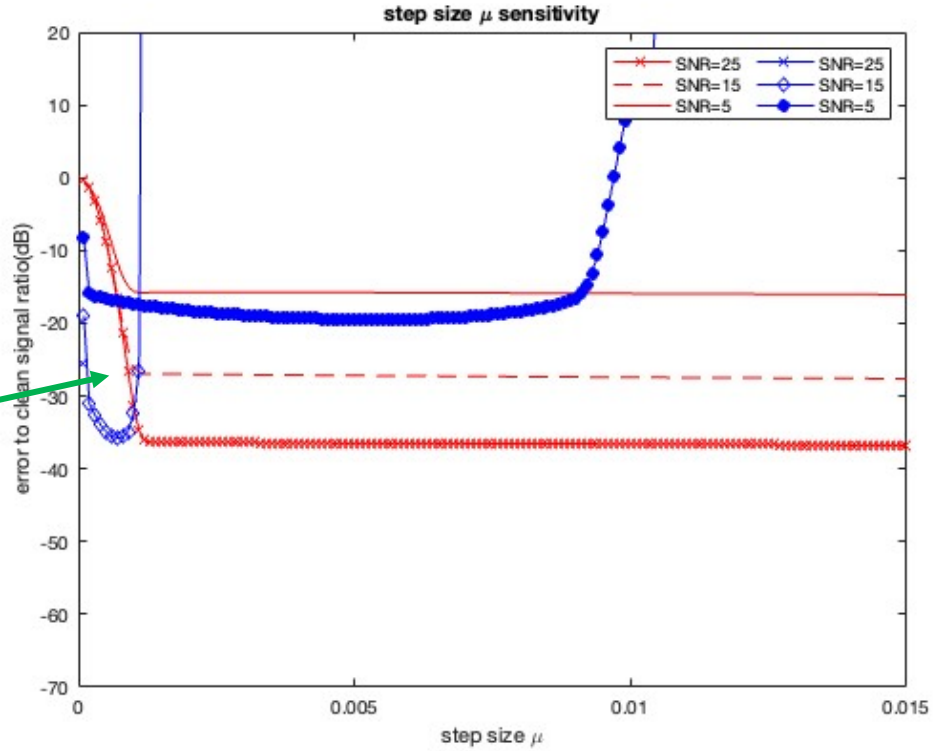
- LMS versus NLMS
  - Fixed frequency f & Dynamic SNR
- $SNR = \{25,15,5\},\ f = 10$
- $0.001 \leq \mu \leq 0.999$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$



Performance Analysis

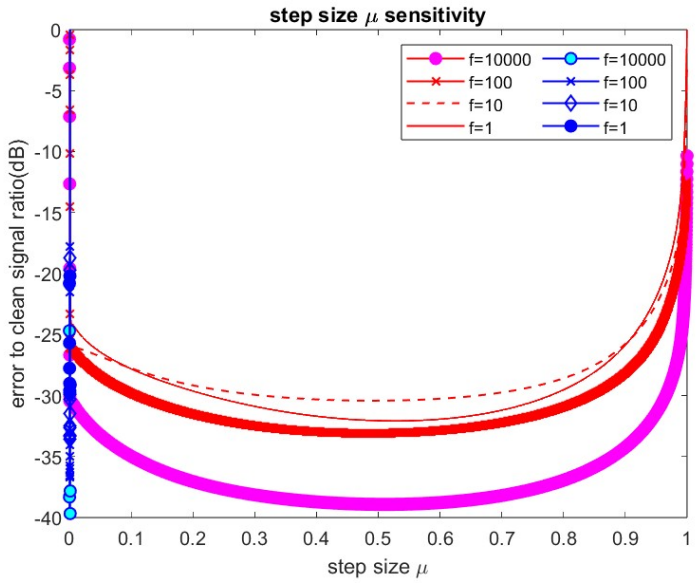
- LMS versus NLMS
  - Fixed frequency f & Dynamic SNR
- $SNR = \{25,15,5\},\ f = 10,$
- $0.0001 \leq \mu \leq 0.9999$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$

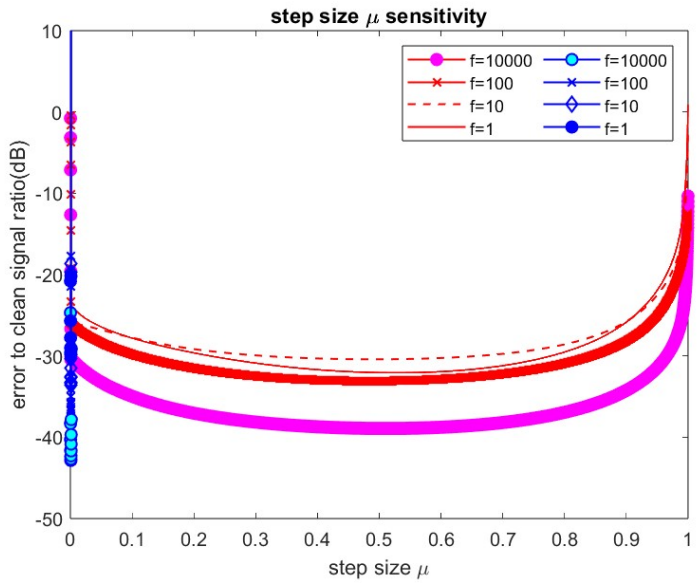


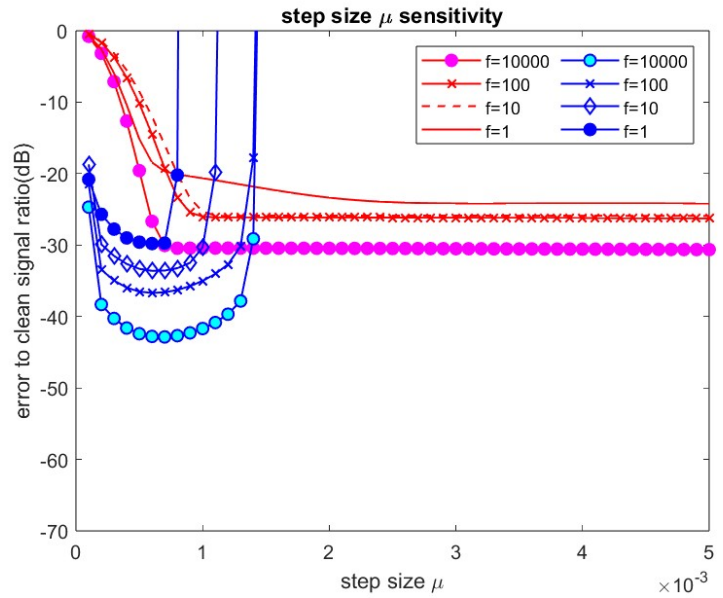


Performance Analysis

- LMS versus NLMS
  - Fixed SNR & Dynamic frequency f
- $SNR = 25, f = \{1,10,100,10000\}$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$





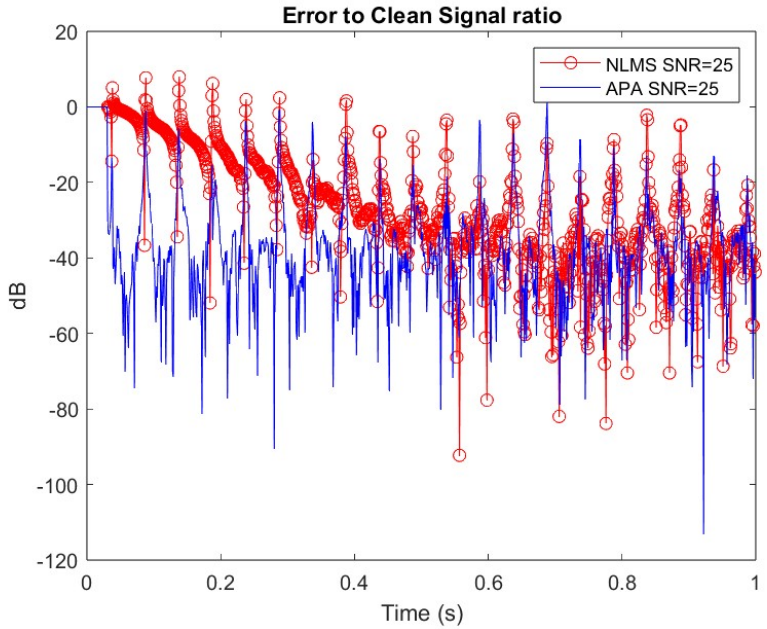


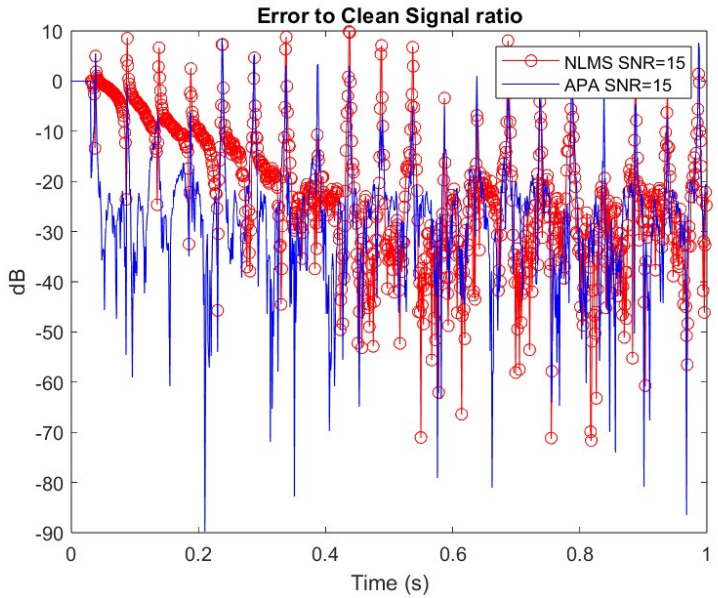
Performance Analysis

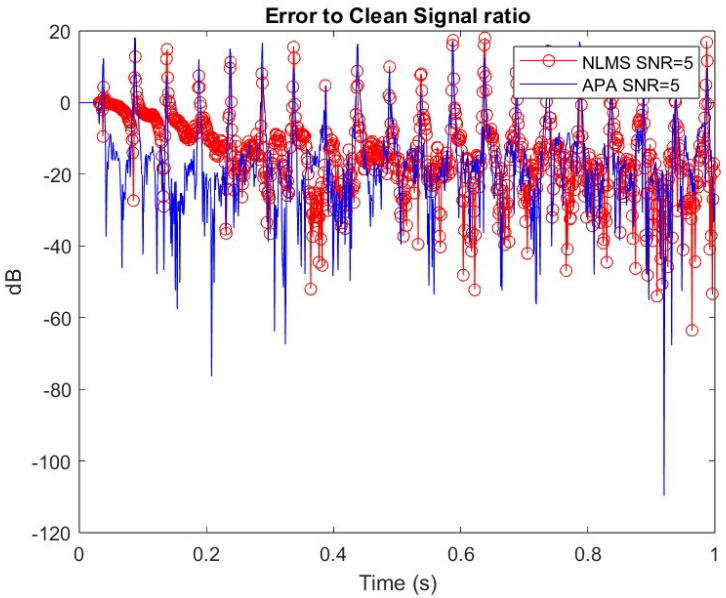
APA versus NLMS

Performance Analysis

- APA versus NLMS
  - Convergence Test
- $SNR = \{25,15,5\},\ f = 10, \mu = 0.01$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$

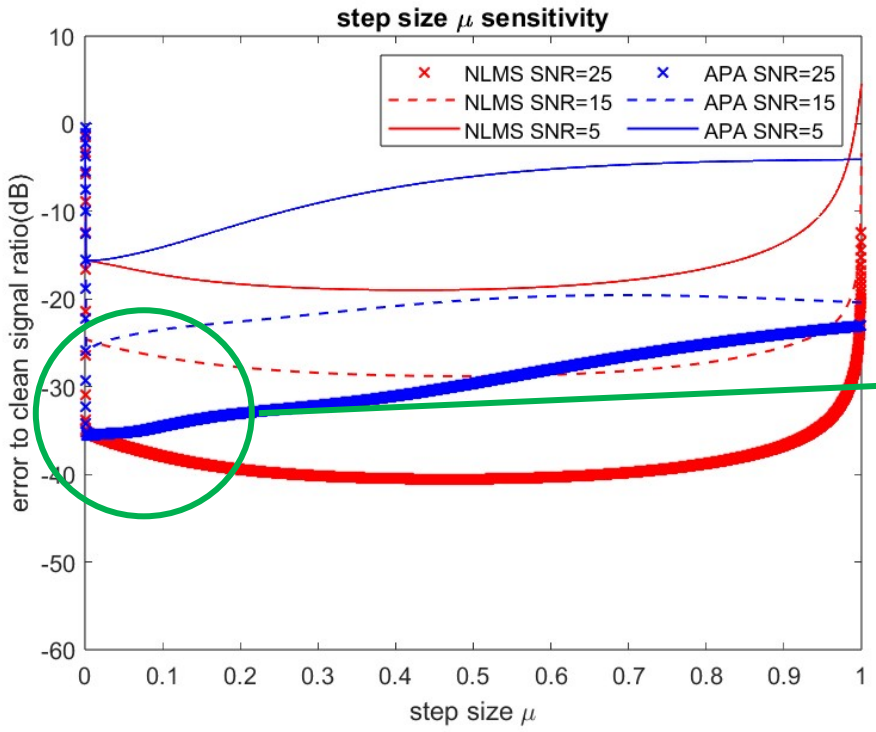


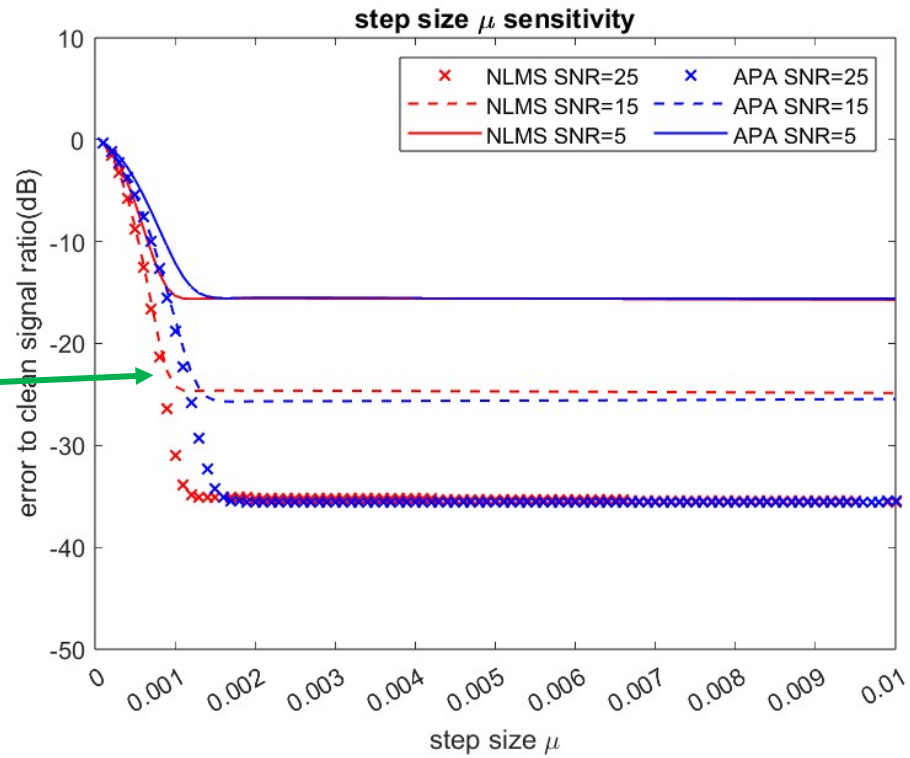




Performance Analysis

- APA versus NLMS
  - Fixed frequency f & Dynamic SNR
- $SNR = \{25,15,5\},\ f = 10$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$





Performance Analysis

- APA versus NLMS
  - Fixed SNR & Dynamic frequency f
- $SNR = 25, f = \{1,10,100,10000\}$
- Clean Signal $= A\cos(2\pi ft) + A\sin(2\pi ft)$
- Noice $\sim N(0,0.64)$