CEU
Universidad
San Pablo

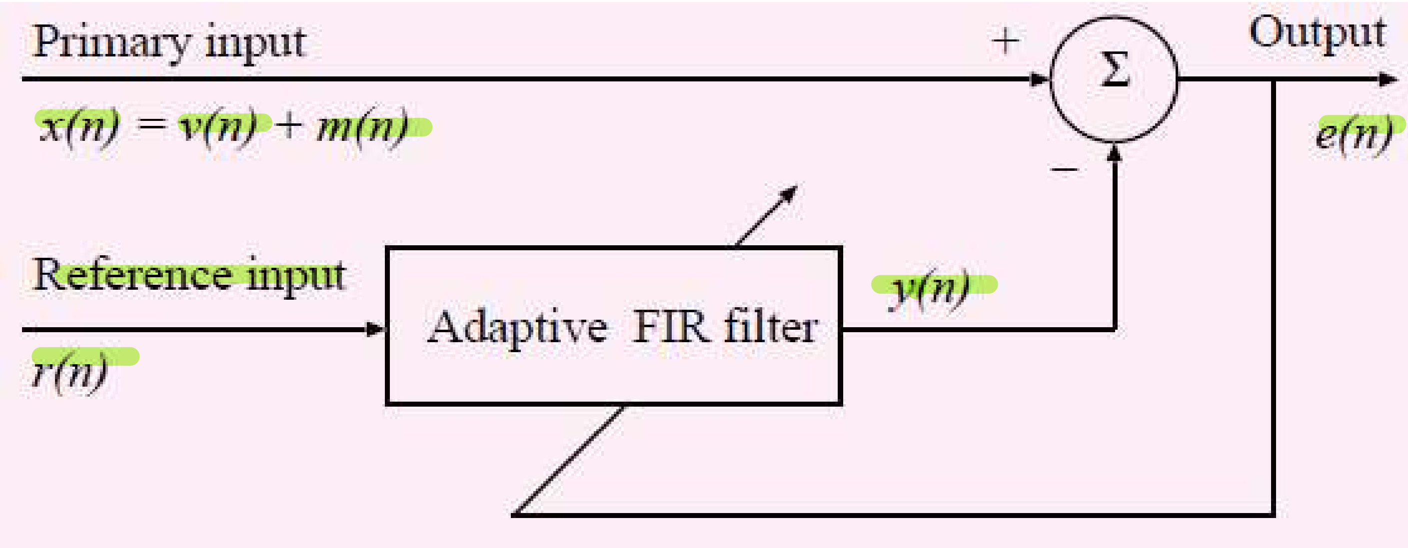
UNIT 5: Adaptive filtering

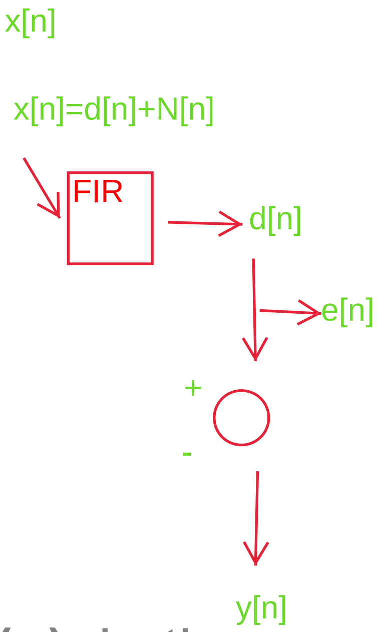
Gabriel Caffarena Fernández
3rd Year Biomedical Engineering Degree
EPS – Univ. San Pablo – CEU

Adaptive filtering

- Wiener filters provides a means to design optimal filters given that the statistics of the desired signals and the noise are known
- The Wiener filtering theory can be extended to the design of adaptive filtering, that is filters that modify their coefficient in order to optimize a quality criterium
- Adaptive filters enable working with unknown and not necessarily stationary signals
- In this unit we focus on the **adaptive noise canceller (ANC)**

Adaptive noise cancelling





- The *primary input* is composed of the desired signal **v(n)** a noise **m(n)**, both uncorrelated and with zero mean
- The *reference* **r(n)** *input* has zero mean and it is correlated with **m(n)** and uncorrelated with **v(n)**
- The filter produces **y(n)** which is an estimation of noise **m(n)**
- Sustracting the estimation of the noise to **x(n)** leads to **e(n)**, an estimation of the desired signal **v(n)**

Adaptive noise cancelling

- We are going to follow an approach very similar to that of Wiener filtering

  Minimized the MSE, considering that **e(n)**, which is the estimate of **v(n)**, is the error

- The primary input is

DESIRED SIGNAL

NOISE

r[n]--- FIR --- y[n]

CORRELATED

$$x(n) = v(n) + m(n)$$

- The output signal is

$$e(n) = x(n) - y(n) = v(n) + m(n) - y(n)$$

= V[N]

a ----------------b---------------

- The MSE is

v(n) has zero mean and it is uncorrelated with m(n) and y(n)

$$MSE = E[e^2(n)]$$

$$= E[v^2(n)] + E[\{m(n) - y(n)\}^2] + 2E[v(n)\{m(n) - y(n)\}]$$

FIR uncorrelated 0

$$= E[v^2(n)] + E[\{m(n) - y(n)\}^2]$$

Adaptive noise cancelling

-----FIXED------ --------RELATED TO THE FILTER'S COEFFICIENT

$$\text{MSE}= E[v^2(n)] + E[\{m(n) - y(n)\}^2]$$

- So, to minize MSE we have to minimize

y[n]= m[n]

$$E[\{m(n) - y(n)\}^2]$$

- Thus, once the system is minized the estimation **y(n)** is the minimum MSE estimate of m(t)
- As consequence, **e(n)** is also the minimum MSE estimate of **v(n)**
- Minimizing the estimation error of the noise, maximizes the SNR at the output

Adaptive noise cancelling

- Let us now introduce vectors to obtain more compact expressions

$$\vec{w} = [w_0, w_1, w_2, \dots, w_{M-1}]^T$$ M coefficents

$$\vec{r}(n) = [r(n), r(n-1), \dots, r(n-M+1)]^T$$

current value

I--------------------(m-1) past values --------------------------I

- The output of the system is

Y[N]

$$e(n) = x(n) - \vec{w}^T \vec{r}(n)$$

- There are several methods to minimize **e(n)** in real-time
- Note that the only constraint that we are imposing is that of the correlation level between **m(n)**, **r(n)** and **v(n)**, and that the means are al zero
- We will focus on the **Least Mean Squares** method **(LMS)**

The LMS adaptive filter

- The aim of the LMS filter is to minimize the squared error, assuming that the current squared error is a good approximation of the MSE

a^T * b = b^T * a*

$$MSE = E[e^2(n)] \approx e^2(n)$$

- Therefore, the filter is continuously minimizing $e^2(n)$

x[n] - y[n] x[n] -y[n]

$$e^2(n) = (x(n) - \vec{w}^T \vec{r}(n))(x(n) - \vec{r}^T(n)\vec{w}(n))$$

$$= x^2(n) - 2x(n)\vec{r}^T \vec{w}(n) + \vec{w}^T(n)\vec{r}(n)\ \vec{r}^T(n)\vec{w}(n)$$

- The squared error is a hyper-paraboloidal (bowl-like) that is never negative
- The aim of the LMS algorithm is to reach the bottom of the “bowl”
- Since we are not optimizing the MSE but the squared error, there is no need to solve the Winer-Hopf equations
- We resort to the **steepest descent algorithm** to find the minimum

The LMS adaptive filter

$$e^2(n) = x^2(n) - 2x(n)\vec{r}^T\vec{w}(n) + \vec{w}^T(n)\vec{r}(n)\,\vec{r}^T(n)\vec{w}(n)$$

- Given the current filter coefficients **w(n)**, the next set of coefficients **w(n+1)** is corrected considering the negative of the gradient of **e^2(n)**

$$\vec{w}(n+1) = \vec{w}(n) - \mu\nabla e^2(n)$$

- The parameter μ controls the stability and the rate of convergence

small μ → high stability → slow convergence

high μ → low stability → fast convergence

- Moreover, the gradient is estimated through the first order derivative

$$\nabla e^2(n) \approx -2x(n)\vec{r}(n) + 2\{\vec{w}^T(n)\vec{r}(n)\}\vec{r}(n) = -2e(n)\vec{r}(n)$$

- Thus, the weights are updated as follows

$$\vec{w}(n+1) = \vec{w}(n) + 2\mu e(n)\vec{r}(n)$$

Example: ECG and EMG

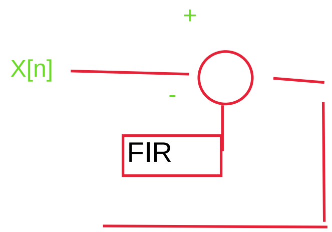
- We consider an ECG **v(n)** signal that is corrupted due to muscular activity (**m1(n)**) and noise (**n1(n)**)

ECG EMG'

$$x(n) = v(n) + m(n)$$

$$m(n) = m_1(n) + n_1(n)$$

EMG noise



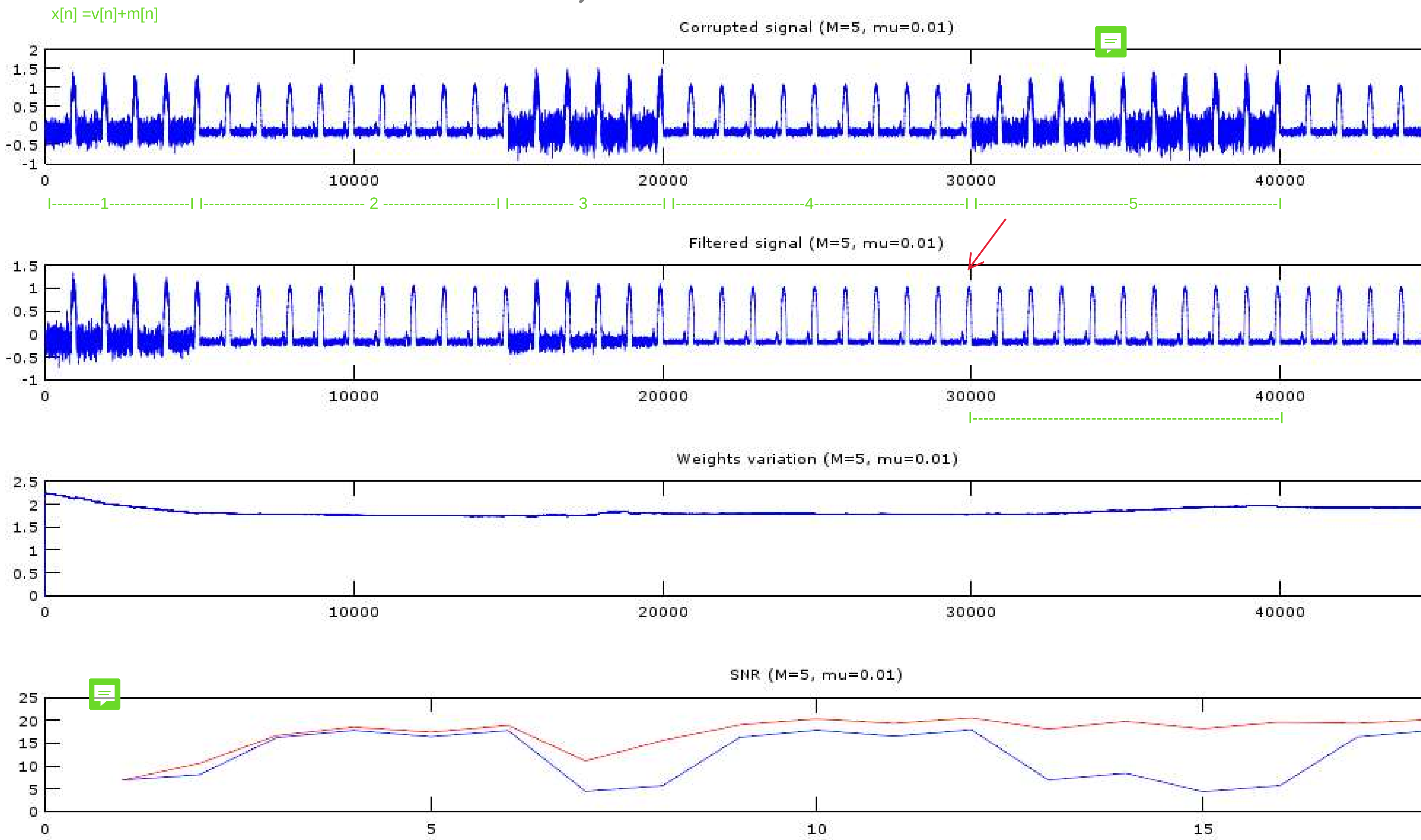
- The secondary input **r(n)** is the muscular activity obtained through an EMG sensor. This signal is corrupted with noise

$$r(n) = k \cdot m_1(n) + n_2(n), 0 < k < 1$$

EMG' EMG noise

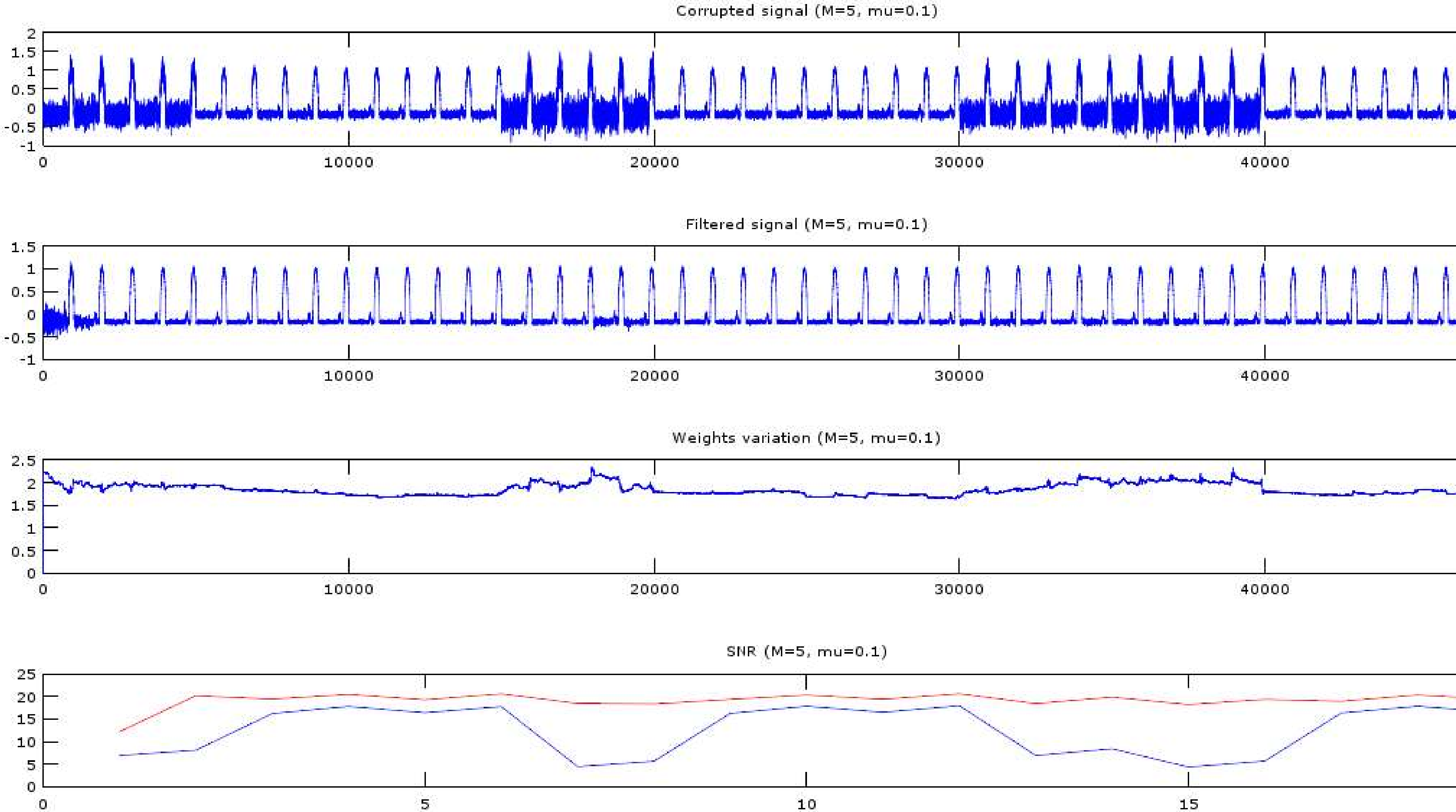
Example: ECG and EMG

M=5, mu=0.01



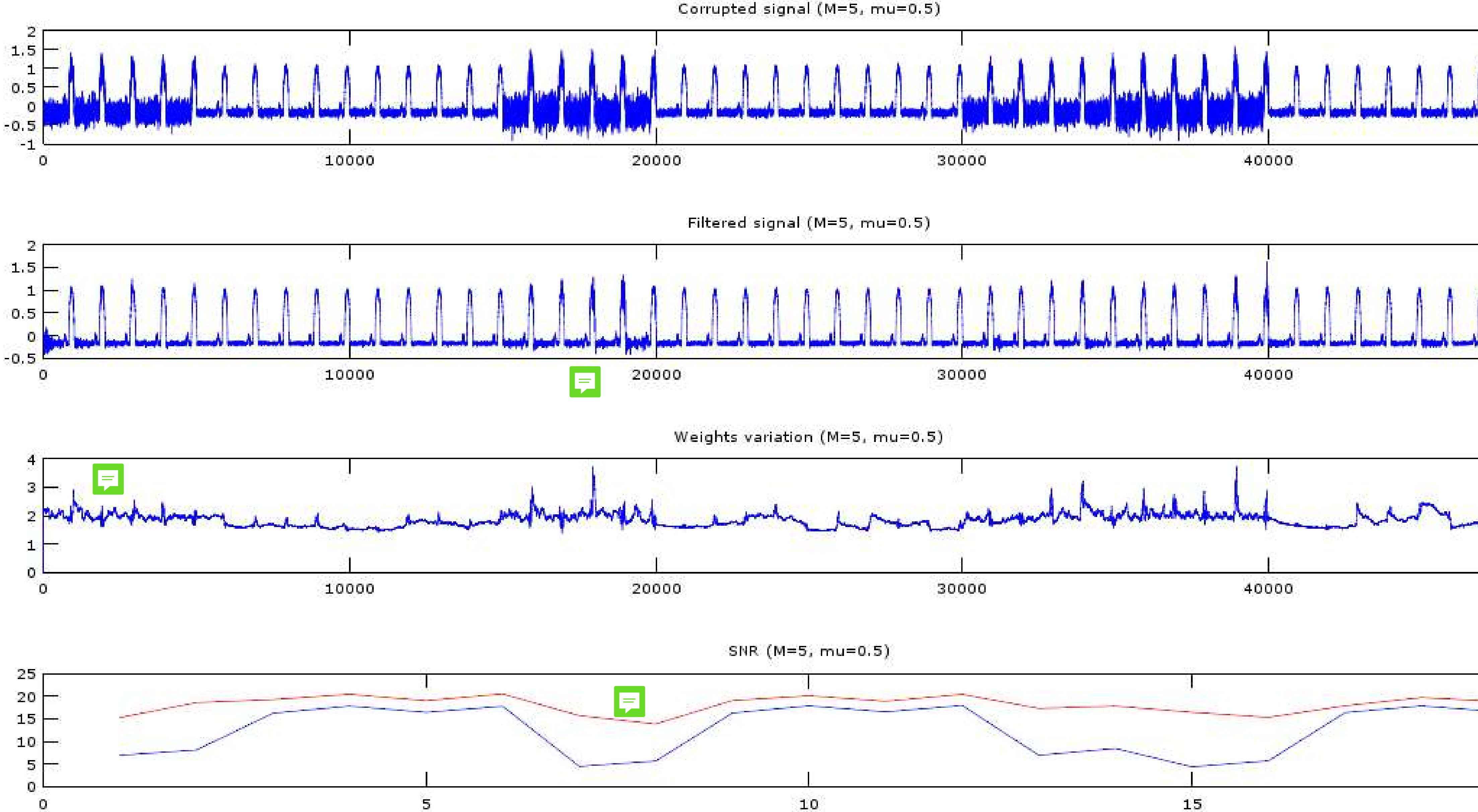
Example: ECG and EMG

M=5, mu=0.1



Example: ECG and EMG

M=5, mu=0.5



SUMMARY

- Adaptive filtering
- LMS filter
- ECG example