Adaptive Filtering

Lecture Notes

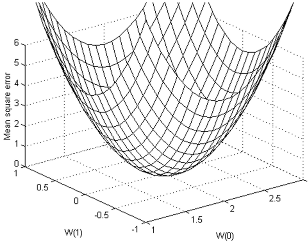
Ahmet Ademoglu, *PhD*,
Bogazici University,
Institute of Biomedical Engineering

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

$$\hat{d}(n) = \sum_{k=0}^{p} w(k)x(n-k)$$

For stationary $d(n)$ and $x(n)$
$\mathbf{R}_x\mathbf{w} = \mathbf{r}_{dx}$
For nonstationary signals, the mean square error $E\{|e(n)^2|\}$ depends on n

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)x(n-k)$$



$$\mathbf{w}_n = [w_n(0), w_n(1), \ldots, w_n(p)]^T,$$
$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-p)]^T$$

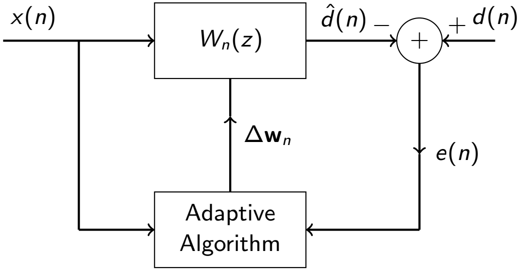
$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}_n$$

Adaptive filtering is about determining $\Delta\mathbf{w}_n$.

In nonstationary environments

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n-k) \text{ and } \hat{r}_{dx}(k) = \frac{1}{N}\sum_{n=0}^{N-1} d(n)x^*(n-k)$$

are changing by time as well.



1. In stationary environments, $\mathbf{w}_n$ converges to the Wiener-Hopf solution

$$\lim_{n\to\infty} \mathbf{w}_n = \mathbf{R}_x^{-1}\mathbf{r}_{dx}$$

2. Estimation of $r_x(k)$ and $r_{dx}(k)$ should be built into the adaptive filter.
3. For nonstationary signals, the filter should adapively track the solution by time.

FIR Adaptive Filters

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)x(n-k) = \mathbf{w}_n^T \mathbf{x}(n)$$

$$\xi(n) = E\{|e(n)|^2\}$$

$$e(n) = d(n) - \hat{d}(n) = d(n) - \mathbf{w}_n^T \mathbf{x}(n)$$

$$\mathbf{R}_x(n)\mathbf{w}_n = \mathbf{r}_{dx}(n)$$

Steepest Descent Algorithm

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla \xi(n)$$

μ : stepsize that affect the rate of $\mathbf{w}_n$ to move down the quadratic surface.
$\nabla \xi(n) = \frac{\partial \xi(n)}{\partial \mathbf{w}^*}$: gradient of the $E\{|e(n)|^2\}$

$$\nabla \xi(n) = E\{e(n) \nabla e^*(n)\} = -E\{e(n)\mathbf{x}^*(n)\}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\{e(n)\mathbf{x}^*(n)\}$$

If $x(n)$ and $d(n)$ are jointly WS stationary,

$$E\{e(n)\mathbf{x}^*(n)\} = E\{d(n)\mathbf{x}^*(n)\} - E\{\mathbf{w}_n^T \mathbf{x}(n)\mathbf{x}^*(n)\} = \mathbf{r}_{dx} - \mathbf{R}_x \mathbf{w}_n$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu(\mathbf{r}_{dx} - \mathbf{R}_x \mathbf{w}_n)$$

Choice of μ

$$\mathbf{w}_{n+1} = (\mathbf{I} - \mu\mathbf{R}_x)\mathbf{w}_n + \mu\mathbf{r}_{dx}$$

$$\mathbf{w}_{n+1} - \mathbf{w} = (\mathbf{I} - \mu\mathbf{R}_x)\mathbf{w}_n + \mu\mathbf{R}_x\mathbf{w} - \mathbf{w} = [\mathbf{I} - \mu\mathbf{R}_x](\underbrace{\mathbf{w}_n - \mathbf{w}}_{\mathbf{c}_n})$$

$$\mathbf{c}_{n+1} = (\mathbf{I} - \mu\mathbf{R}_x)\mathbf{c}_n$$

$\mathbf{R}_x = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$ and since $\mathbf{R}_x$ is Hermitian and positive definite, $\lambda_k \geq 0$.

$$\mathbf{V}^H\mathbf{c}_{n+1} = (\mathbf{I} - \mu\mathbf{\Lambda}^H)\mathbf{V}^H\mathbf{c}_n$$

$\mathbf{u}_n = \mathbf{V}^H\mathbf{c}_n \longrightarrow \mathbf{u}_{n+1} = (\mathbf{I} - \mu\mathbf{\Lambda}^H)\mathbf{u}_n \rightarrow u_n(k) = (1 - \mu\lambda_k)^n u_0(k)$
$\mathbf{c}_n \rightarrow 0$ if and only if $|1 - \mu\lambda_k| < 1$; $k = 0, 1, \ldots, p$

$$0 < \mu < \frac{2}{\lambda_{max}}$$

$$\mathbf{w}_n = \mathbf{w} + \mathbf{c}_n = \mathbf{w} + \sum_{k=0}^{p} u_n(k)\mathbf{v}_k = \mathbf{w} + \sum_{k=0}^{p}(1-\mu\lambda_k)^n u_0(k)\mathbf{v}_k$$

$\mathbf{v}_k$: *Modes of Filter*

$(1-\mu\lambda_k)^{\tau_k} = 1/e \qquad \tau_k = -1/\ln(1-\mu\lambda_k) \approx 1/\mu\lambda_k$

$\tau = \max\{\tau_k\} \approx 1/\mu\lambda_{min} \qquad \mu = \alpha\frac{2}{\lambda_{max}} \qquad \tau \approx \frac{1}{2\alpha}\frac{\lambda_{max}}{\lambda_{min}} = \frac{1}{2\alpha}\chi$

α: Normalized step size $(0 < \alpha < 1)$

χ: Condition number which is inversely related with the rate of convergence.

The decay of $\xi(n)$: *Learning Curve*

$$\xi_{min} = r_d(0) - \mathbf{r}_{dx}^H\mathbf{w}$$

$$\xi(n) = E\{|e(n)|^2\} = E\{|d(n) - \mathbf{w}_n^T\mathbf{x}(n)|^2\} = r_d(0) - \mathbf{r}_{dx}^H\mathbf{w}_n - \mathbf{w}_n^H\mathbf{r}_{dx} + \mathbf{w}^H\mathbf{R}_x\mathbf{w}_n$$

$$\xi(n) = r_d(0) - \mathbf{r}_{dx}^H\mathbf{w} + \mathbf{c}_n^H\mathbf{R}_x\mathbf{c}_n = \xi_{min} + \mathbf{c}_n^H\mathbf{R}_x\mathbf{c}_n = \xi_{min} + \mathbf{u}_n^H\mathbf{\Lambda}_x\mathbf{u}_n$$

$$\xi(n) = \xi_{min} + \sum_{k=0}^{p}\lambda_k|u_n(k)|^2 = \xi_{min} + \sum_{k=0}^{p}\lambda_k(1-\mu\lambda_k)^{2n}|u_0(k)|^2$$

LMS Algorithm

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\{e(n)\mathbf{x}^*(n)\}$$

$$\hat{E}\{e(n)\mathbf{x}^*(n)\} = \frac{1}{L}\sum_{l=0}^{L-1} e(n-l)\mathbf{x}^*(n-l)$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{L}\sum_{l=0}^{L-1} e(n-l)\mathbf{x}^*(n-l)$$

If $L = 1$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n)\mathbf{x}^*(n)$$

```
Input :          Order of the filter $p$
                 Step Size $\mu$
Initialize :     $\mathbf{w}_0 = \mathbf{0}$
Computation :    For $n = 0, 1, 2, \ldots$
                 $y(n) = \mathbf{w}_n^T\mathbf{x}(n)$
                 $e(n) = d(n) - y(n)$
                 $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n)\mathbf{x}^*(n)$
```

Convergence of the LMS Algorithm

Using one sample estimation of $E\{e(n)\mathbf{x}^*(n)\}$

$$\nabla\xi(n) = -E\left\{e(n)\mathbf{x}^*(n)\right\} \longleftrightarrow \hat{\nabla}\xi(n) = -e(n)\mathbf{x}^*(n)$$

It is an unbiased estimation

$$E\{\hat{\nabla}\xi(n)\} = -E\left\{e(n)\mathbf{x}^*(n)\right\} = \nabla\xi(n)$$

$$\lim_{n\to\infty} E\{\mathbf{w}_n\} = \mathbf{w} = \mathbf{R}_x^{-1}\mathbf{r}_{dx}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu[d(n) - \mathbf{w}_n^T\mathbf{x}(n)]\mathbf{x}^*(n)$$

$$E\left\{\mathbf{w}_{n+1}\right\} = E\{\mathbf{w}(n)\} - \mu E\{d(n)\mathbf{x}^*(n)\} - \mu E\{\mathbf{x}^*(n)\mathbf{x}^T(n)\mathbf{w}_n\}$$

Assume that $x(n)$ and $\mathbf{w}_n$ are independent.

$$\begin{aligned} E\left\{\mathbf{w}_{n+1}\right\} &= E\left\{\mathbf{w}_n\right\} + \mu E\{d(n)\mathbf{x}^*(n)\} - \mu E\{\mathbf{x}^*(n)\mathbf{x}^T(n)\}E\{\mathbf{w}_n\} \\ &= (\mathbf{I} - \mu\mathbf{R}_x)E\left\{\mathbf{w}_n\right\} + \mu\mathbf{r}_{dx} \end{aligned}$$

$$\mathbf{u}_n = \mathbf{V}^H[\mathbf{w}_n - \mathbf{w}] \qquad E\{\mathbf{u}_n\} = (\mathbf{I} - \mu\mathbf{\Lambda})^n\mathbf{u}_0$$

If $0 < \mu < 2/\lambda_{max}$ then $E\{\mathbf{u}_n\} \to 0$ and $\mathbf{w}_n \to \mathbf{w}$

$$\lambda_{max} \le \sum_{k=0}^{p} \lambda_k = Tr[\mathbf{R}_x] = (p+1)r_x(0) \qquad \hat{r}_d(0) = \frac{1}{N}\sum_{k=0}^{N-1}|x(n-k)|^2$$

A more conservative bound $\quad 0 < \mu < 2/((p+1)r_x(0))$

Adaptive Linear Prediction using LMS Algorithm

$$x(n) = 1.2728x(n-1) - 0.81x(n-2) + v(n)$$

$v(n)$: unit variance white noise
Optimum Causal linear filter :

$$\hat{x}(n) = 1.2728x(n-1) - 0.81x(n-2)$$

Adaptive filter : $\hat{x}(n) = w_n(1)x(n-1) - w_n(2)x(n-2)$
LMS Algorithm

$$w_{n+1} = w_n(k) + \mu e(n)x^*(n-k) \qquad e(n) = x(n) - \hat{x}(n)$$

$\mathbf{w}_0 = \mathbf{0}$ and $\mu = 0.02$ and $\mu = 0.004$

Adaptive Linear Prediction using LMS Algorithm

```
N=1000;
a1 = 1.2827; a2=-0.81;
sigma_v2 =1;
v=sqrt(sigma_v2)*randn(N,1);
v=(v-mean(v))/std(v);
x=filter(1,[1 -a1 -a2],v);
mu=[0.02; 0.004];
% Initialize LMS
w1(:,1)=zeros(2,1);
w2(:,1)=zeros(2,1);
for n=3:N,
y = w1(:,n-2)'*x(n-[1:2]);
e= x(n)-y;
w1(:,n-1) = w1(:,n-2) + mu(1)*e*x(n-[1:2],1);
y = w2(:,n-2)'*x(n-[1:2]);
e= x(n)-y;
w2(:,n-1) = w2(:,n-2) + mu(2)*e*x(n-[1:2],1);
end;
close;subplot(2,1,1);plot(w1');grid;subplot(2,1,2);plot(w2');grid
```

Excess Mean-square Error

$$e(n) = d(n) - \mathbf{w}_n^T \mathbf{x}(n) = d(n) - (\mathbf{w} + \mathbf{c}_n)^T \mathbf{x}(n) = e_{min}(n) - \mathbf{c}_n^T \mathbf{x}(n)$$

$\xi(n) = E\{|e(n)|^2\} = \xi_{min} + \xi_{cx}(n)$

Using the relations $\mu \sum_{k=0}^{p} \frac{\lambda_k}{2-\mu\lambda_k} < 1$ and $0 < \mu < \frac{2}{\lambda_{max}}$

$$\xi(\infty) = \xi_{min} + \xi_{ex}(\infty) = \xi_{min} / \left(1 - \mu \sum_{k=0}^{p} \frac{\lambda_k}{2 - \mu\lambda_k}\right)$$

$\xi_{ex}(\infty) = \mu \xi_{min} \sum_{k=0}^{p} \frac{\lambda_k}{2-\mu\lambda_k} / \left(1 - \mu \sum_{k=0}^{p} \frac{\lambda_k}{2-\mu\lambda_k}\right)$

If $\mu << 2/\lambda_{max}$, $\quad \frac{1}{2}\mu \sum_{k=0}^{p} \lambda_k < 1$ or $\mu < 2/Tr(\mathbf{R}_x)$

$$\xi(\infty) \approx \xi_{min} / (1 - \frac{1}{2}\mu Tr(\mathbf{R}_x))$$

$$\xi_{ex}(\infty) = \mu \xi_{min} \frac{\frac{1}{2} Tr(\mathbf{R}_x)}{1 - \frac{1}{2}\mu Tr(\mathbf{R}_x)} \approx \frac{1}{2}\mu \xi_{min} Tr(\mathbf{R}_x)$$

Misadjustment : $\mathcal{M} = \frac{\xi_{ex}(\infty)}{\xi_{min}} \approx \frac{1}{2}\mu Tr(\mathbf{R}_x)$

$$\xi_{min} = 1,\ \lambda_1 = 9.7924 \text{ and } \lambda_2 = 1.7073,\ \xi(\infty) = \begin{cases} 1.1942 & ; \quad \mu = 0.02 \\ 1.0155 & ; \quad \mu = 0.004 \end{cases}$$

LMS Misadjustment

```
N=1000;
a1 = 1.2827; a2=-0.81; sigma_v2 =1; mu=[0.02; 0.004];
for K=1:200,
v=sqrt(sigma_v2)*randn(N,1); v=(v-mean(v))/std(v);
x=filter(1,[1 -a1 -a2],v);
w1(:,1)=zeros(2,1); w2(:,1)=zeros(2,1);
for n=3:N,
y = w1(:,n-2)'*x(n-[1:2]);
e= x(n)-y; E1(n-2,K)=e;
w1(:,n-1) = w1(:,n-2) + mu(1)*e*x(n-[1:2],1);
y = w2(:,n-2)'*x(n-[1:2]);
e= x(n)-y; E2(n-2,K)=e;
w2(:,n-1) = w2(:,n-2) + mu(2)*e*x(n-[1:2],1);
end; end;
close;subplot(2,1,1);plot(mean(E1.^2,2));grid;
subplot(2,1,2);plot(mean(E2.^2,2));grid;
mean([mean(E1(900:end,:).^2,2)  mean(E2(900:end,:).^2,2) ])
```

Normalized LMS

λ_{max} or $\mathbf{R}_x$ must be known for convergence and step size selection.

$$\hat{E}\left\{|x(n)|^2\right\} = \frac{1}{p+1}\sum_{k=0}^{p}|x(n-k)|^2$$

$$0 < \mu < 2/((p+1)E\{|x(n)|^2\}), \quad 0 < \mu < \frac{2}{\mathbf{x}(n)^H\mathbf{x}(n)}$$

$$\mu(n) = \frac{\beta}{|\mathbf{x}(n)|^2}$$

β : Normalized step size $0 < \beta < 2$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \beta\frac{\mathbf{x}^*(n)}{|\mathbf{x}(n)|^2}e(n)$$

Normalization by $|\mathbf{x}(n)|^2$ eliminates the *gradient noise amplification* but for small values of $|\mathbf{x}(n)|^2$ a modified form is

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \beta\frac{\mathbf{x}^*(n)}{\epsilon + |\mathbf{x}(n)|^2}e(n)$$

For speeding up the NLMS algorithm

$$|\mathbf{x}(n+1)|^2 = |\mathbf{x}(n)|^2 + |x(n+1)|^2 - |x(n-p)|^2$$

Noise Cancellation

$$x(n) = d(n) + v_1(n)$$

A reference signal : $v_2(n)$ correlated with $v_1(n)$

$$\hat{d}(n) = x(n) - \hat{v}_1(n)$$

and $e(n) = v_1(n) - \hat{v}_1(n)$

$$d(n) = \sin(n\omega_0 + \phi), \qquad \omega_0 = 0.05\pi$$

$$\begin{aligned} v_1(n) &= 0.8v_1(n-1) + g(n) \\ v_2(n) &= -0.6v_2(n-1) + g(n) \end{aligned}$$

$g(n)$: unit variance white noise
Normalized LMS parameter : $\beta = 0.25$

Noise Cancellation with NMLS using a Reference Signal

```
N=1000;  omega=0.05*pi; beta=0.25; epsi=0.0001;
g=randn(N,1); g=(g-mean(g))/std(g);
v1=filter(1,[1 -0.8 ],g); v2=filter(1,[1 0.6 ],g);
d = sin([0:N-1]'*omega);
x=d+v1;
p=12;
w0(:,1)=zeros(p,1);
for n=p+1:N,
y = w0'*v2(n+1-[1:p]);
e= x(n)-y; E(n-p,1)=e;
w0 = w0 + beta*e*v2(n+1-[1:p],1)/(epsi+ norm(v2(n+1-[1:p]),2)^2 );
end;
close;subplot(2,1,1);plot( [E d(p+1:end) ]);grid;
subplot(2,1,2);plot([x d]);grid;
```

Noise Cancellation with NMLS under Nonstationary Environment

```
N=1000;  omega=0.05*pi; beta=0.25; epsi=0.0001;
g=randn(N,1); g=(g-mean(g))/std(g);
g=((linspace(0.25,6.25,1000).*(0.5)).*g')';
v1=filter(1,[1 -0.8 ],g); v2=filter(1,[1 0.6 ],g);
d = sin([0:N-1]'*omega);
x=d+v1;
p=12;
w0(:,1)=zeros(p,1);
for n=p+1:N,
y = w0'*v2(n+1-[1:p]);
e= x(n)-y; E(n-p,1)=e;
w0 = w0 + beta*e*v2(n+1-[1:p],1)/(epsi+ norm(v2(n+1-[1:p]),2)^2 );
end;
close;subplot(2,1,1);plot( [E d(p+1:end) ]);grid;
subplot(2,1,2);plot([x d]);grid;
```

Noise Cancellation with NMLS without a Reference Signal

```
n0=25;
N=1000+n0;  omega=0.05*pi; beta=0.25; epsi=0.0001;
g=randn(N,1); g=sqrt(0.25)*(g-mean(g))/std(g);
v1=filter(1,[1 -0.8 ],g);
d = sin([0:N-1]'*omega);
x=d+v1;
p=20;
w0(:,1)=zeros(p,1);
for n=[p+n0:N],
k=n-n0+1 - [1:p];
y = w0'*x(k);
Y(n-p,1) = y;
e= x(n)-y; E(n-p,1)=e;
w0 = w0 + beta*e*x(k,1)/(epsi+ norm(x(k),2)^2 );
end;
close;subplot(2,1,1);plot( [Y  d(p+1:end) ]);grid; axis([0 1000 -4 4])
subplot(2,1,2);plot([x d]);grid;axis([0 1000 -4 4]);
```

Recursive Least Squares

Minimizing $\xi(n) = E\{|e(n)|^2\}$ requires $r_x(k)$ and $r_{dx}(k)$ so that they are estimated using their instantaneous values in LMS. This can somteimes bring slow convergence and large excess mean square error.

Alternatively, $\mathcal{E} = \sum_{i=0}^{n} |e(i)|^2$ may be evaluated.

The difference is that minimizing $E\{|e(n)|^2\}$ yields the same $\mathbf{w}$ for all sequences having the same statistics.

On the other hand, $\sum_{i=0}^{n} |e(i)|^2$ must be optimized for each signal individually.

Exponentially Weighted RLS

$$\mathcal{E}(n) = \sum_{i=0}^{n} \lambda^{n-i} |e(i)|^2$$

$0 < \lambda < 1$: Forgetting Factor

$$e(i) = d(i) - y(i) = d(i) - \mathbf{w}_n^T \mathbf{x}(i)$$

The $\mathbf{w}_n$ are held constant in $[0, n]$.

for$k = 0, 1, \ldots, p$

$$\frac{\partial \mathcal{E}(n)}{\partial w_n^*(k)} = \sum_{i=0}^{n} \lambda^{n-i} e(i) \frac{\partial e^*(i)}{\partial w_n^*(k)} = -\sum_{i=0}^{n} \lambda^{n-i} e(i) x^*(i-k) = 0$$

$$\sum_{i=0}^{n} \lambda^{n-i} \left\{ d(i) - \sum_{l=0}^{p} w_n(l) x(i-l) \right\} x^*(i-k) = 0$$

$$\sum_{l=0}^{p} w_n(l) \left[\sum_{i=0}^{n} \lambda^{n-i} x(i-l) x^{*(i-k)} \right] = \sum_{i=0}^{n} \lambda^{n-i} d(i) x^*(i-k)$$

Deterministic Normal Equations

$$\mathbf{R}_x(n) \mathbf{w}_n = \mathbf{r}_{dx}(n)$$

$\mathbf{R}_x(n) = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}^*(i) \mathbf{x}^T(i)$ and $\mathbf{x}(i) = [x(i), x(i-1), \ldots, x(i-p)]^T$

$$\mathbf{r}_{dx}(n) = \sum_{i=0}^{n} \lambda^{n-i} d(i) \mathbf{x}^*(i)$$

$$
\begin{aligned}
\mathcal{E}(n) &= \sum_{i=0}^{n} \lambda^{n-i} e(i)e^*(i) = \sum_{i=0}^{n} e(i) \left\{ d(i) - \sum_{l=0}^{p} w_n(l)x(i-l) \right\}^* \\
&= \sum_{i=0}^{n} \lambda^{n-i} e(i)d^*(i) - \sum_{l=0}^{n} w_n^*(l) \underbrace{\sum_{i=0}^{n} \lambda^{n-i} e(i)x^*(i-l)}_{0 \textit{ for optimal } \mathbf{w}_n}
\end{aligned}
$$

$$
\{\mathcal{E}(n)\}_{min} = \sum_{i=0}^{n} \lambda^{n-i} |d(i)|^2 - \sum_{l=0}^{p} w_n(l) \sum_{i=0}^{n} \lambda^{n-i} x(i-l)d^*(i)
$$

$\{\mathcal{E}(n)\}_{min} = |\mathbf{d}(n)|_\lambda^2 - \mathbf{r}_{dx}^H \mathbf{w}_n$

Instead of solving *deterministic normal equation*, we solve

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \Delta \mathbf{w}_{n-1}$$

where $\mathbf{w}_n = \mathbf{R}_x^{-1}(n)\mathbf{r}_{dx}(n)$

Recursive Equations for $\mathbf{r}_{dx}(n)$ and $\mathbf{R}_x(n)$ are

$$
\begin{aligned}
\mathbf{r}_{dx}(n) &= \lambda \mathbf{r}_{dx}(n-1) + d(n)\mathbf{x}^*(n) \\
\mathbf{R}_x(n) &= \lambda \mathbf{R}_x(n-1) + \mathbf{x}^*(n)\mathbf{x}^T(n)
\end{aligned}
$$

Remembering *Woodbury Identity*

$$(\mathbf{A}+\mathbf{u}\mathbf{v}^H)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^H\mathbf{A}^{-1}}{1+\mathbf{v}^H\mathbf{A}^{-1}\mathbf{u}}$$

$$\mathbf{R_x}^{-1}(\mathbf{n}) = \lambda^{-1}\mathbf{R_x}^{-1}(n-1) - \frac{\lambda^{-2}\mathbf{R}_x^{-1}(n-1)\mathbf{x}^*(n)\mathbf{x}^T(n)\mathbf{R}_x^{-1}(n-1)}{1+\lambda^{-1}\mathbf{x}^T(n)\mathbf{R}_x^{-1}(n-1)\mathbf{x}^*(n)}$$

Defining $\mathbf{P}(n) = \mathbf{R}_x^{-1}$ and $\mathbf{g}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}^*(n)}{1+\lambda^{-1}\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}^*(n)}$

$$\mathbf{P}(n) = \lambda^{-1}\left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right]$$

Gain vector $\mathbf{g}(n)$ has the following property :

$$\mathbf{g}(n) + \lambda^{-1}\mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}^*(n) = \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}^*(n)$$

$$\mathbf{g}(n) = \lambda^{-1}\left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right]\mathbf{x}^*(n) = \mathbf{P}(n)\mathbf{x}^*(n)$$

$$\mathbf{g}(\mathbf{n}) = \mathbf{P}(n)\mathbf{x}^*(n) \longleftrightarrow \mathbf{w_n} = \mathbf{R}_x^{-1}(n)\mathbf{r}_{dx}(n)$$

Update for $\mathbf{w}_n$

$$\mathbf{w}_n = \mathbf{P}(n)\mathbf{r}_{dx}(n)$$

Remembering that $\mathbf{r}_{dx}(n) = \sum_{i=0}^{n} \lambda^{n-i} d(i)\mathbf{x}^*(i) = d(n)\mathbf{x}^*(n) + \sum_{i=0}^{n-1} \lambda^{n-i} d(i)\mathbf{x}^*(i)$

$$\mathbf{w}_n = \lambda \mathbf{P}(n)\mathbf{r}_{dx}(n-1) + d(n)\mathbf{P}(n-1)\mathbf{x}^*(n)$$

$$\mathbf{w}_n = \left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T\mathbf{P}(n-1)\right]\mathbf{r}_{dx}(n-1) + d(n)\mathbf{g}(n)$$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mathbf{g}(n)\left[d(n) - \mathbf{w}_{n-1}^T\mathbf{x}(n)\right]$$

a priori error: $\alpha(n) = d(n) - \mathbf{w}_{n-1}^T \mathbf{x}(n)$
a posteriori error: $e(n) = d(n) - \mathbf{w}_n^T \mathbf{x}(n)$

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha(n)\mathbf{g}(n)$$

Initialization

1) $\mathbf{P}(0) = \left[\sum_{i=-p}^{0} \lambda^{-i} \mathbf{x}^*(i)\mathbf{x}^T(i)\right]^{-1}$, $\mathbf{r}_{dx}(0) = \sum_{i=-p}^{0} \lambda^{-i} d(i)\mathbf{x}^*(i)$, $\mathbf{w}_0 = \mathbf{P}(0)\mathbf{r}_{dx}(0)$
optimal but requires inversion of $\mathbf{R}_x$ and a delay of $p+1$ samples.
2) $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$ and $\mathbf{w}_0 = \mathbf{0}$,
may introduce bias with a decay if $\lambda < 1$.

RLS Algorithm

```
Input :          Order of the filter : $p$
                 Exponential weighting factor : $\lambda$
                 Value to initialize $\mathbf{P}(0)$ : $\delta$
Initialize :     $\mathbf{w}_0 = \mathbf{0}$
                 $\mathbf{P}(0) = \delta^{-1}\mathbf{I}$
Computation :    For $n = 1, 2, \ldots$
                 $z(n) = \mathbf{P}(n-1)\mathbf{x}^*(n)$
                 $g(n) = \frac{1}{\lambda + \mathbf{x}^T(n)\mathbf{z}(n)}\mathbf{z}(n)$
                 $\alpha_n = d(n) - \mathbf{w}^T(n-1)\mathbf{x}(n)$
                 $\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha(n)\mathbf{g}(n)$
                 $\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{z}^H(n)\right]$
```

Linear Prediction Using RLS

$$x(n) = 1.2728x(n-1) - 0.81x(n-2) + v(n)$$

$\hat{x}(n) = w_n(1)(n-1) + w_n(2)x(n-2)$ $\lambda = 1$ and $\lambda = 0.95$

Time varying AR(1) model

$$x(n) = a_n(1)x(n-1) - 0.81x(n-2) + v(n)$$

$a_n(1) = \begin{cases} 1.2728 & ; \quad 0 \leq n < 100 \\ 0 & ; \quad 100 \leq n \leq 200 \end{cases}$ $\lambda = 1$ and $\lambda = 0.9$

Linear Prediction using growing window RLS ($\lambda = 1$)

```
N=200;
v=randn(N,1); v=(v-mean(v))/std(v);
x=[ filter(1,[1 -1.2728 0.81  ],v(1:N))  ];
delta = 0.001; lambda= [1 0.9];
w0(:,1)=zeros(2,1);
P0=eye(2)/delta;
for n=3:N,
W(n-2,:) = w0';
z = P0*x(n-[1:2],1);
g = (1/( lambda(1) +  x(n-[1:2])'*z )) * z;
alpha_n = x(n) - w0'*x(n-[1:2],1);
w0 = w0 + alpha_n * g;
P0 = (1/lambda(1)) * (P0 - g*z' ) ;
end;
close;plot(W);grid
```

Time varying AR(1) model using Exponentially Weighted RLS ($\lambda = 0.95$)

```
N=200;
v=randn(N,1); v=(v-mean(v))/std(v);
x=[ filter(1,[1 -1.2728 0.81 ], v(1:100)); filter(1,[1 0 0.81 ],v(101:200)) ];
delta = 0.001; lambda= [1 0.95];
w0(:,1)=zeros(2,1);
P0=eye(2)/delta;
for n=3:N,
W(n-2,:) = w0';
z = P0*x(n-[1:2],1);
g = (1/( lambda(2) + x(n-[1:2])'*z )) * z;
alpha_n = x(n) - w0'*x(n-[1:2],1);
w0 = w0 + alpha_n * g;
P0 = (1/lambda(2)) * (P0 - g*z' ) ;
end;
close;plot(W);grid
```

Sliding Window RLS

Both RLS and its wighted version require infinite memory.
In nonstationary environments, the sum of squares of error can be minimized over a finite window

$$\mathcal{E}_L(n) = \sum_{i=n-L}^{n} |e(i)|^2$$

The outliers are forgotten after a finite number of iterations.

$$\mathbf{R}_x(n)\mathbf{w}_n = \mathbf{r}_{dx}(n)$$

where $\mathbf{R}_x(n) = \sum_{i=n-L}^{n} \mathbf{x}(i)^* \mathbf{x}^T(i)$ and $\mathbf{r}_{dx}(n) = \sum_{i=n-L}^{n} d(i)\mathbf{x}(i)^*$.

1. Given $\mathbf{w}_{n-1}$ at $n-1$, $\tilde{\mathbf{w}}_n$ is found by minimizing

$$\mathcal{E}_{L+1} = \sum_{i=n-L-1}^{n} |e(i)|^2$$

2. $\mathbf{w}_n$ that minimizes $\sum \mathcal{E}_L$ is determined by discarding the last data point $x(n-L-1)$.

First Step

We can find $\tilde{\mathbf{w}}_n$ from $\mathbf{w}_{n-1}$ as

$$\begin{aligned}\mathbf{g}(n+1) &= \frac{1}{1+\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)}\mathbf{P}(n-1)\mathbf{x}^*(n) \\ \tilde{\mathbf{w}}_n &= \tilde{\mathbf{w}}_{n-1} + \mathbf{g}(n)[d(n) - \mathbf{w}_{n-1}\mathbf{x}^*(n)] \\ \tilde{\mathbf{P}}(n) &= \mathbf{P}(n-1) - \mathbf{g}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\end{aligned}$$

$$\tilde{\mathbf{P}}(n)^{-1} = \tilde{\mathbf{R}}_x(n) = \sum_{k=n-L-1}^{n} \mathbf{x}^*(k)\mathbf{x}^T(k)$$

$$\tilde{\mathbf{R}}_x(n)\tilde{\mathbf{w}}_n = \tilde{\mathbf{r}}_{dx}(n)$$

where $\tilde{\mathbf{r}}_{dx}(n) = \sum_{k=n-L-1}^{n+1} d(k)\mathbf{x}^*(k)$

Second Step

The last data point $x(n-L-1)$ is discarded by

$$\begin{aligned}\mathbf{R}_x(n) &= \tilde{\mathbf{R}}_x(n) - \mathbf{x}^*(n-L)\mathbf{x}^T(n-L-1)\\ \mathbf{r}_{dx}(n) &= \tilde{\mathbf{r}}_{dx}(n) - d(n-L-1)\mathbf{x}^*(n-L-1)\end{aligned}$$

$$\begin{aligned}\tilde{\mathbf{g}}(n+1) &= \frac{1}{1-\mathbf{x}^T(n-L-1)\tilde{\mathbf{P}}(n)\mathbf{x}^*(n-L-1)}\tilde{\mathbf{P}}(n)\mathbf{x}^*(n-L-1)\\ \mathbf{w}(n) &= \tilde{\mathbf{w}}(n) - \tilde{\mathbf{g}}(n)[d(n-L-1) - \tilde{\mathbf{w}}_n\mathbf{x}^*(n-L-1)]\\ \mathbf{P}(n) &= \tilde{\mathbf{P}}(n) - \tilde{\mathbf{g}}(n)\mathbf{x}^T(n-L-1)\tilde{\mathbf{P}}(n)\end{aligned}$$