APPENDIX H
THE DEMONSTRATED EXAMPLE CODE

```
signal = sin(2*pi*0.055*[0:1000-1]');
noise=randn(1,1000);
nfilt=fir1(11,0.4); % Eleventh order lowpass filter.
fnoise=filter(nfilt,1,noise); % Correlated noise data.
d=signal'+fnoise; % disered
coeffs = zeros(12,1); % Set the filter initial conditions.
mu = 0.04;            % Set step size for algorithm update.
ha = adaptfilt.lms(12,mu);
set(ha,'coefficients',coeffs);
set(ha,'persistentmemory',true); % Prevent filter reset.
[ya,ea] = filter(ha,noise,d);
subplot(2,1,1),plot(0:199,signal(1:200),0:199,ea(1:200));grid
legend('Original','filtered by LMS')
axis([0 200 -2 2])
title(' clean signal vs filtered signal (LMS)')
xlabel('time index')
ylabel(' Magnitude')
subplot(2,1,2),plot(fnoise(1:200)); grid
title( 'The noisy input')
axis([0 200 -2 2])
xlabel('time index')
ylabel(' Magnitude')
lammda = 1;            % Set weighting factor for algorithm update.
hb = adaptfilt.rls(12,lam);
set(hb,'coefficients',coeffs);
set(hb,'persistentmemory',true); % Prevent filter reset.
[yb,eb] = filter(hb,noise,d);
subplot(2,1,2),plot(0:199,signal(1:200),0:199,eb(1:200));grid;
legend('Original','filtered by RLS')
axis([0 200 -2 2])
title(' clean signal vs filtered signal(RLS)')
xlabel('time index')
ylabel(' Magnitude')
figure,plot(fnoise(1:200)); grid
title( 'The noisy input')
axis([0 200 -2 2])
xlabel('time index')
ylabel(' Magnitude')
```