LAB 4: WIENER FILTER

Introducció al Processament Audiovisual

Lucía De Pineda
Claudia Len

1. FILTRE DE WIENER PER ELIMINAR UNA INTERFERÈNCIA DE BANDA ESTRETA

En aquesta part, utilitzarem un filtre de Wiener per eliminar una interferència sinusoidal x[n] que corromp un senyal de parla w[n].

1.1. Implementació del Filtre de Wiener quan la funció d'autocorrelació és coneguda

Per començar, suposarem que w[n] és un procés blanc Gaussià de mitjana zero i variància σ_w^2 . Mentre que x[n] és una sinusoide amb amplitud $\sqrt{2P}$ i freqüència Ω_0 conegudes i amb fase θ, que és una variable aleatòria distribuida uniformement entre (0,2π).

La senyal que tenim és: $v[n] = x[n] + w[n] = \sqrt{2P}\, cos(2\pi F_0 m + \theta) + w[n]$

i) Matriu d'autocorrelació R_x i vector de correlació creuada r_{xd} :

La funció d'autocorrelació per v[n] és:

$R_v[n] = P\, cos(2\pi F_0 m) + \sigma_w^2 \cdot \delta[m]$

En el codi podem veure com utilitza aquesta fórmula per calcular el vector de correlació de x. Una vegada el troba, construeix la matriu d'autocorrelació amb la funció toeplitz, que agafa els dos elements que passem com a paràmetres (el vector de correlació dues vegades) i els utilitza com a primera fila i primera columna, construint una matriu simètrica amb un valor fixe a la diagonal ($r_x[0]$, el primer valor dels vectors). Per tant, ens queda la matriu d'autocorrelació de x, tal com la calculem teòricament.

Després, el codi calcula el vector de correlació creuada utilitzant el vector de correlació calculat anteriorment i agafant els valors retardats D mostres (en el nostre cas, D=1). Per tant, agafa des de la segona mostra fins el final.

ii) Resposta impulsional del filtre de Wiener $h_{opt}[n]$ i del filtre global $A(z)$:

Per calcular les respostes impulsionals, utilitzem la solució del filtre de Wiener, que és:

$h_{opt} = R_x^{-1} r_{xd}$.

Com ja hem calculat aquests dos valors, només fa falta fer l'operació. Ens queden els següents coeficients del filtre de Wiener $h_{opt}[n]$ (10 valors ja que Q=10):

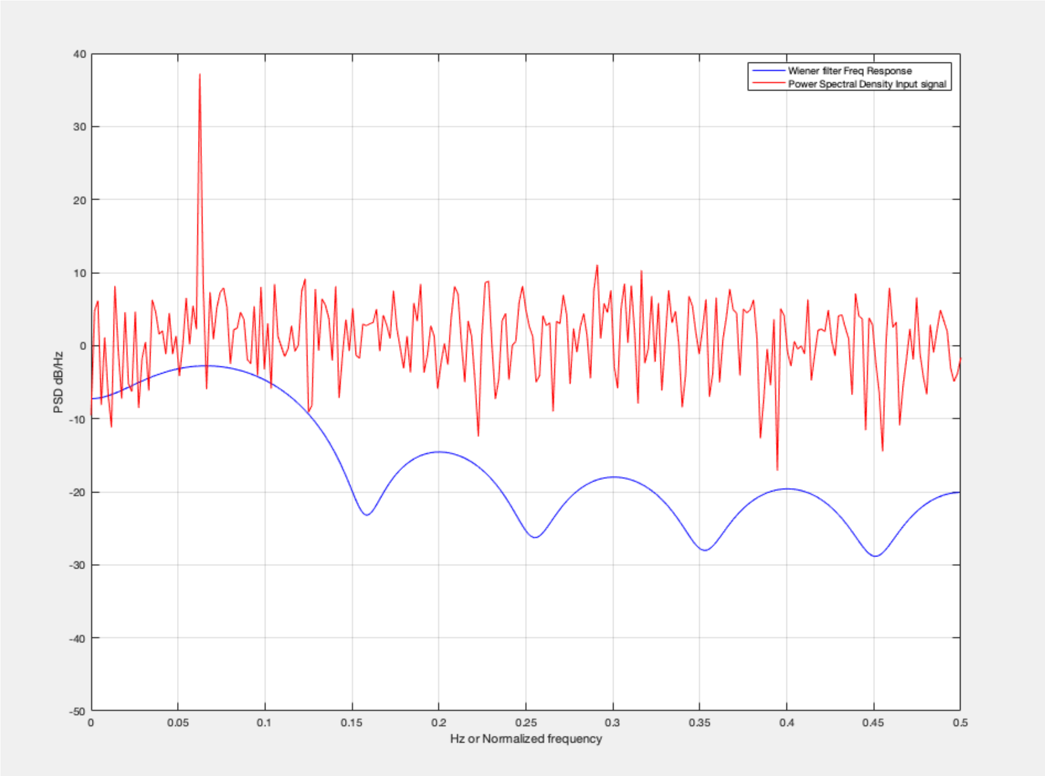
[0.1214, 0.0855, 0.0366, -0.0179, -0.0697, -0.1109, -0.1351, -0.1388, -0.1214, -0.0855]

Pel cas del filtre global $A(z)$, la resposta impulsional és $h_a[n] = \delta[n] - h_{opt}[n - D]$, i els coeficients calculats amb el codi són:

[1, -0.1214, -0.0855, -0.0366, 0.0179, 0.0697, 0.1109, 0.1351, 0.1388, 0.1214, 0.0855]

iii) <u>Resposta freqüencial del filtre de Wiener:</u>

La representació gràfica de la resposta freqüencial del filtre de Wiener és la següent:

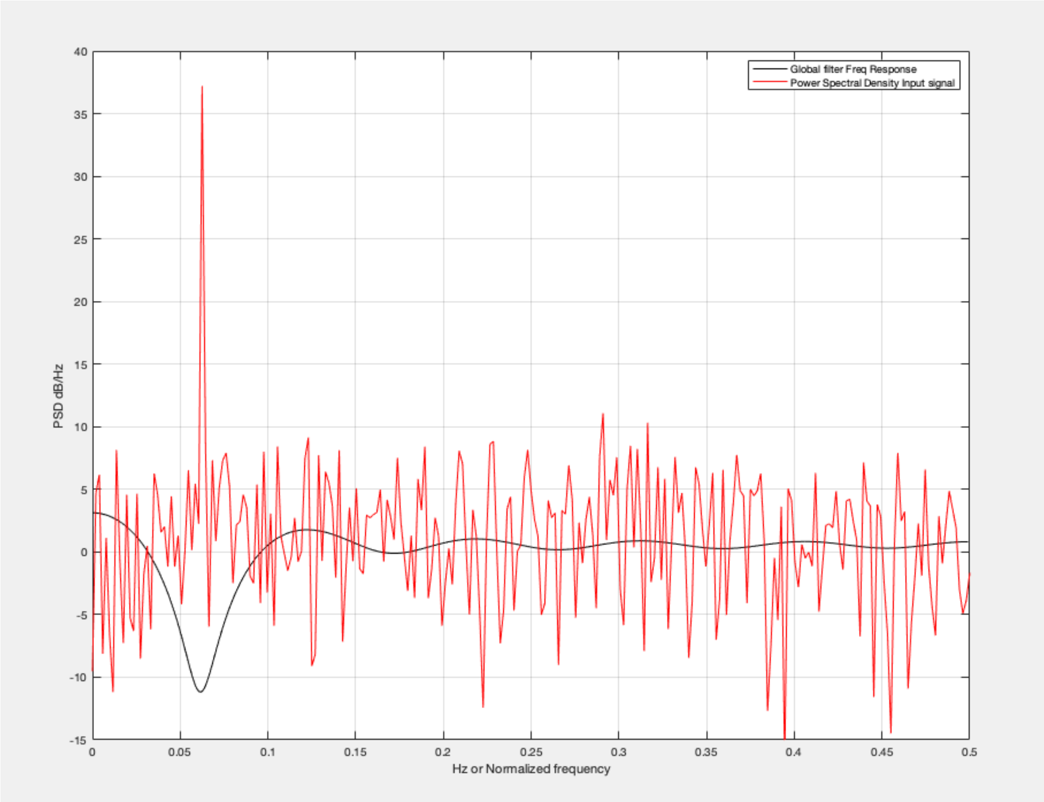


Sabem que la sortida del filtre serà el producte entre la resposta freqüencial $H(z)$ del filtre de Wiener (línia blava) i la power spectral density del senyal d'entrada $v[n]$ (línia vermella). Utilitzem la PSD de $v[n]$ com aproximació de la seva transformada $V(z)$. Per tant, la multiplicació d'aquests dos senyals ens donarà l'estimació de x que volem (senyal de sortida).

Per veure si la gràfica ens ha donat el resultat esperat comencem fixant-nos en el màxim de la power spectral density de $v[n]$. Veiem que el pic es troba aproximadament en la freqüència 0.06, que encaixa amb la freqüència d'interferència que hem fixat $F_0 = 1/16 = 0.0625$, així que és el que esperàvem. A més a més, podem observar com aquest pic coincideix amb el màxim de $H(z)$, el qual és el que volem, ja que així al fer el producte podem obtenir bé la freqüència F_0.

iv) <u>Resposta freqüencial del filtre global:</u>

La representació gràfica del filtre global és:

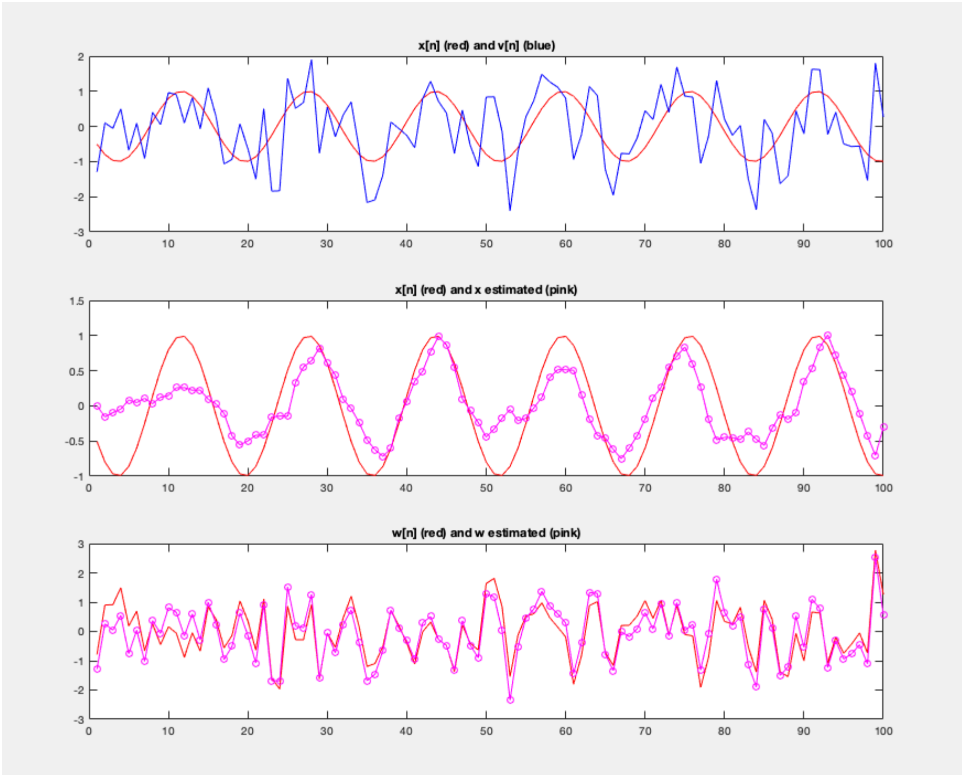


En el cas del filtre global ens trobem una situació bastant semblant. Ara volem estimar $w[n]$, per fer-ho utilitzem la resposta freqüencial del filtre global i la PSD de $v[n]$, com abans. El producte d'aquests dos senyals ens donarà l'estimació de $w[n]$ que volem.

En aquesta gràfica tornem a obtenir el resultat esperat. Podem veure com tenim un mínim del filtre global, que coincideix amb el màxim de la PSD. Com hem dit abans, es troben en F_0 . Això és exactament el que volem, ja que l'objectiu és aïllar $w[n]$, el qual és un procés blanc Gaussià.

v) <u>Representació gràfica dels senyals d'entrada i sortida:</u>

Ara representarem gràficament els senyals d'entrada i els de sortida.



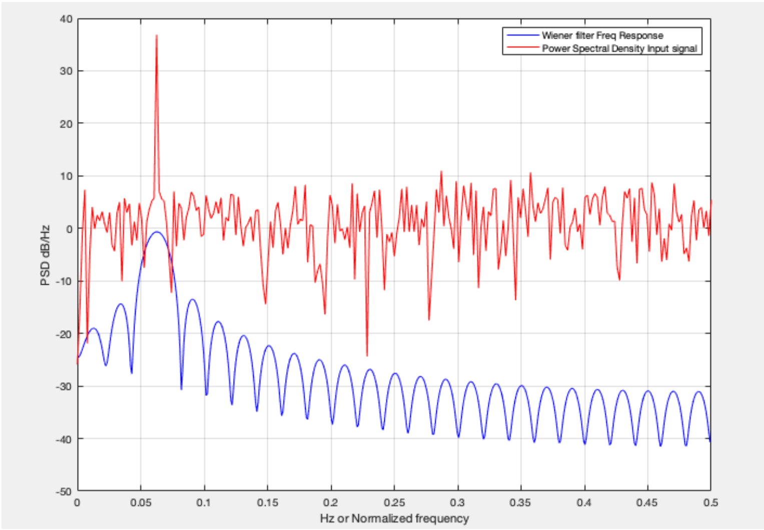
En la imatge anterior trobem tres gràfiques. La primera representa el senyal interferit $v[n]$ amb la interferència $x[n]$, la segona compara el senyal d'entrada $x[n]$ amb el senyal de sortida $\widehat{x}[n]$ i, per últim, la tercera compara el senyal d'entrada $w[n]$ amb la senyal de sortida $\widehat{w}[n]$.

En la primera observem la sinusoide $x[n]$ (línia vermella) i com afecta al senyal $w[n]$, donant com a resultat el senyal interferit $v[n]$ (línia blava). La interferència ha provocat que la forma del senyal s'assembli més a una sinusoide, encara que amb molts més pics.

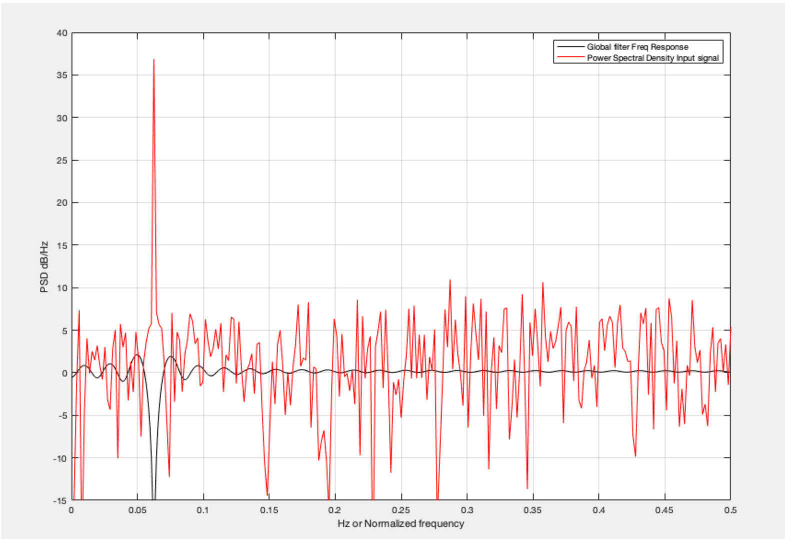
Ara ens fixem en les altres dos gràfiques, on trobem les comparacions dels dos senyals d'entrada amb les seves estimacions. Veiem com l'estimació de $w[n]$ s'aproxima bastant més al senyal original. L'estimació de $\widehat{x}[n]$ té més o menys la mateixa forma però hi ha punts fortament desviats, sobretot al principi, després va millorant. Això pot ser perquè $w[n]$ és un soroll blanc i és més fàcil d'estimar.

vi) <u>Canvi de nombre de coeficients en el filtre de Wiener:</u>

En els apartats anteriors hem utilitzat 10 coeficients per construir el filtre de Wiener. Ara provarem amb 50 coeficients per veure com afecta al sistema.
Comencem representant la resposta freqüencial del filtre de Wiener:

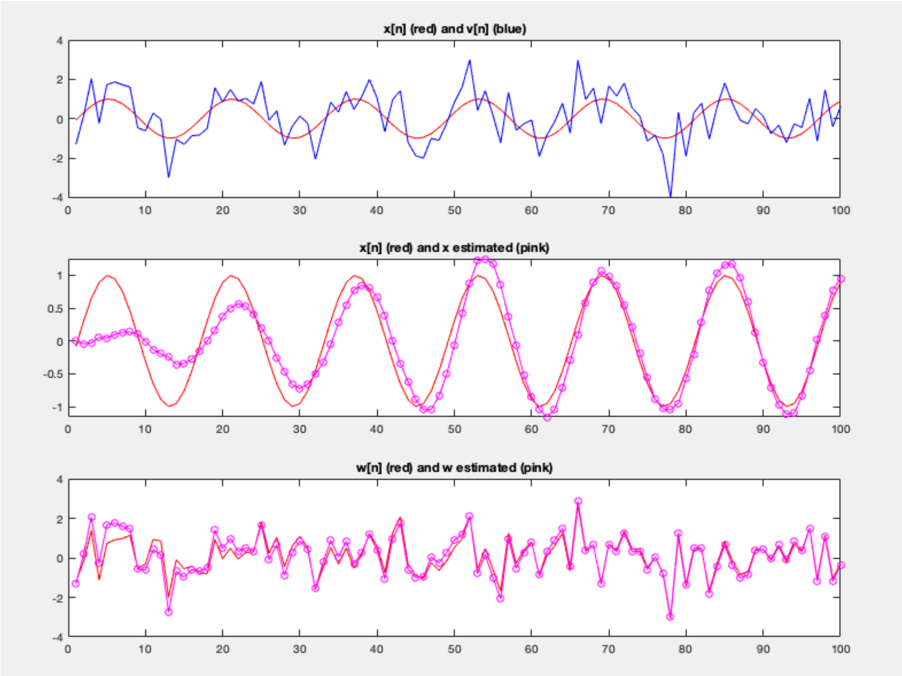


Podem observar com canvia la resposta freqüencial del filtre de Wiener quan utilitzem més coeficients. Ara tenim 25 pics (és a dir, la meitat del nombre de coeficients), quan abans en teníem només 5. Igualment, es segueix complint que el màxim del filtre coincideix amb el màxim de la PSD del senyal d'entrada.
Ara mirem el filtre global:



En el filtre global passa el mateix, podem veure com el filtre té molts més pics (encara que no s'aprecien cap al final del filtre). També trobem que el mínim del filtre coincideix amb el màxim de la PSD del senyal d'entrada.
Per tant, quants més coeficients utilitzem al filtre de Wiener, més oscil·lacions hi haurà a la resposta freqüencial, però el màxim/mínim seguirà estant al mateix lloc.

Per últim, també volem veure com canvien les estimacions dels senyals d'entrada quan tenim més coeficients. Representem els senyals d'entrada i sortida com abans:



Ens fixem en la segona i tercera gràfica per veure com les estimacions s'ajusten als senyals originals. En l'estimació de x observem com al principi es desvia bastant del senyal original, però després s'ajusta quasi perfectament. Amb 10 coeficients també havíem vist com al principi l'error era major, però amb 50 hi ha una millora molt més considerable. En el cas de l'estimació de w, podem veure com és molt bona, ja que s'ajusta molt bé al senyal original.

Per tant, podem concloure que, quants més coeficients tingui el filtre, millor seran les aproximacions dels senyals d'entrada, és a dir, millor serà el filtre. Però quants més coeficients tingui també serà més complex i potser és més complicat de tractar.

1.2. Implementació del filtre de Wiener amb l'algorisme LMS

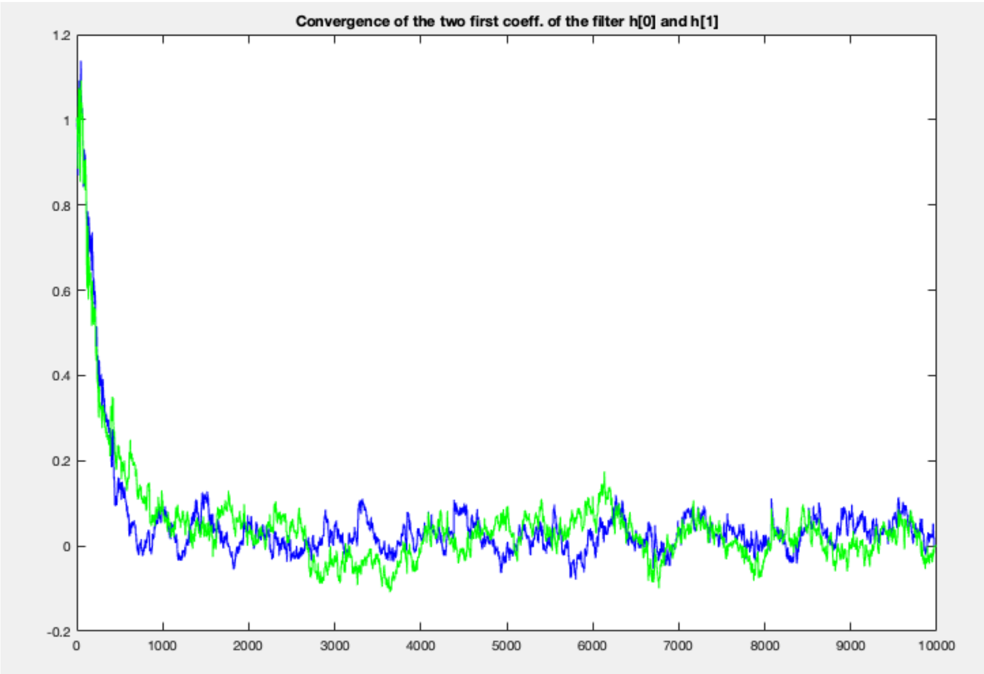
En aquesta secció, suposem que no coneixem la matriu d'autocorrelació ni el vector de correlació creuada, com passa en aplicacions reals. A més a més, necessitarem utilitzar un filtre adaptatiu ja que seran senyals no estacionaris. Per implementar un filtre adaptatiu ens basarem en l'algorisme LMS.

1.2.1. L'algorisme LMS i el seu anàlisi de convergència

Començarem amb un cas igual que en el primer apartat, és a dir, tenim un procés blanc Gaussià de mitjana zero interferit per una sinusoide.

i) Execució del codi i anàlisi de convergència:

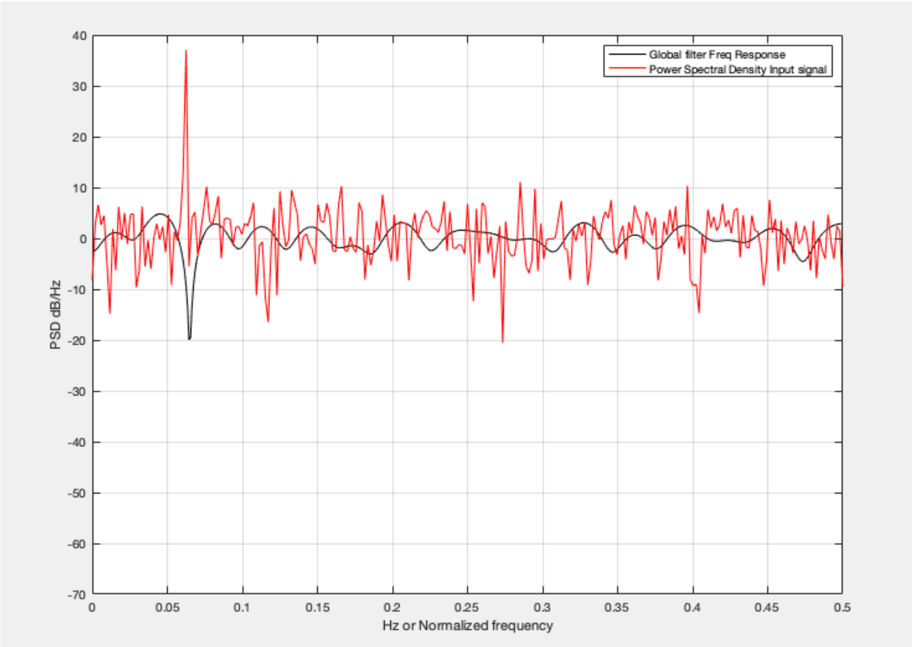
Primer fem la representació gràfica dels dos primers coeficients del filtre en funció del nombre d'iteracions.



Com podem veure a la gràfica, els dos coeficientes comencen a 1, i a mesura que augmenta el nombre d'iteracions el valor dels coeficients es va aproximant a zero.
Si mirem els valors dels coeficients podem veure com els dos acaben convergint, però triguen una mica en estabilitzar-se. Veiem com l'algorisme s'estabilitza en valors pròxims a zero, per tant significa que ha trobat els coeficients òptims i que els dos coeficients convergeixen. Si mirem els valors del vector de coeficients creat pel codi, veiem com el primer coeficient s'estabilitza en aproximadament 0.0153 i el segon en -0.0155. També com el valor dels coeficients fluctua entre un rang de valors bastant petit, per tant no hi ha molt desajust. Això pot ser perquè el valor de mu és proper a zero (μ = 0.004).

També mirem les respostes freqüencials del filtre de Wiener i del filtre global després de 3N/4 iteracions (ja s'ha aconseguit la convergència en aquest instant).

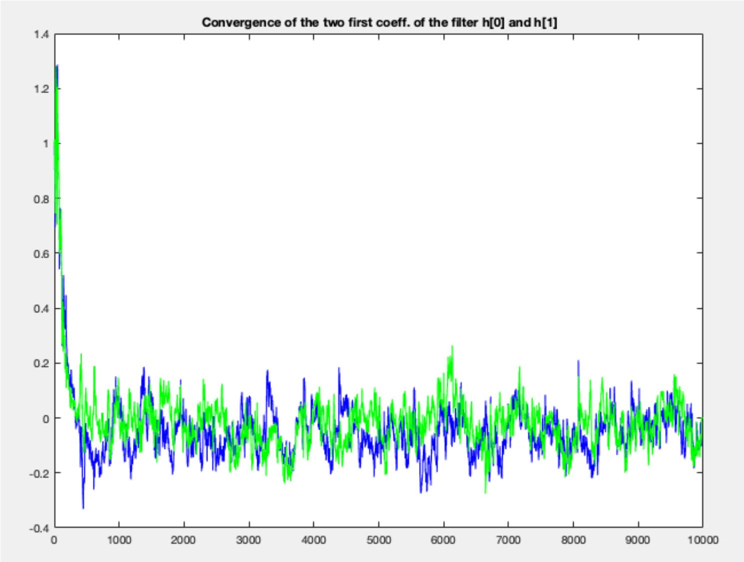
Pel filtre de Wiener obtenim resultat similars als que haviem obtingut en el primer apartat. La forma de la gràfica ha canviat una mica, però segueix havent un màxim que coincideix amb el pic de la PSD. Una altra observació és que en aquest cas sembla que el filtre es mou per un rang més petit de valors.



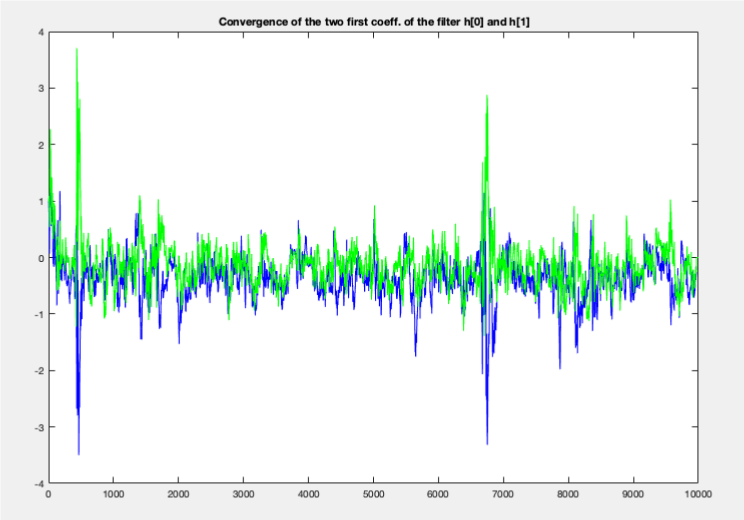
Pel filtre global si que obtenim resultats bastant semblants als primers, tenim un pic negatiu al començament de la gràfica i després va oscil·lant però té un comportament bastant lineal.

ii) <u>Anàlisi del paràmetre μ:</u>

Calculem la constant mu de referència, que en el nostre cas és 0.0413. Aquest és el nostre límit superior, i a partir d'aquest valor, anem canviant mu per tal de veure com afecta a la convergència i al desajust. Teòricament, amb tots els valors de mu entre 0 i aquesta fita, l'algorisme convergeix.
El valor que hem utilitzat en l'apartat anterior és μ = 0.004, amb el qual els coeficients convergeixen en la iteració 9968 i hi ha molt poc desajust.
Ara provarem amb μ = 0.01. Representem la convergència dels coeficients:



Podem observar com el desajust és major amb un valor de μ més gran, ja que hi ha més fluctuacions dels valors. Però, també veiem com els valors arriben ràpidament a valors propers a zero. És a dir, s'estabilitza abans, però amb un major desajust.
Per últim, provem amb μ = 0.03:



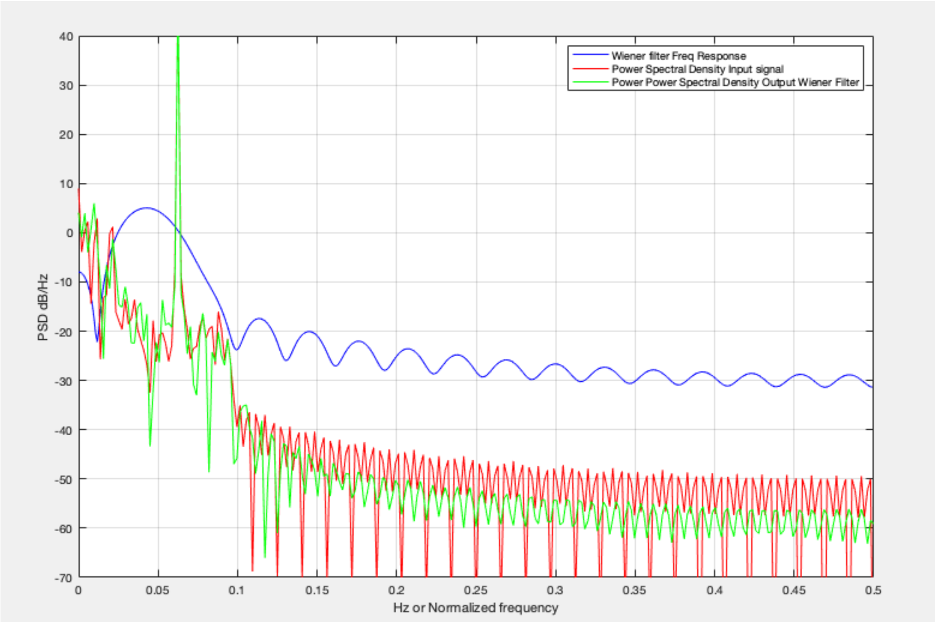
Observem com, a mesura que el valor de μ s'aproxima al límit superior, el desajust és més gran (veiem més fluctuacions dels coeficients), però, per altra banda, els coeficients convergeixen més ràpidament.

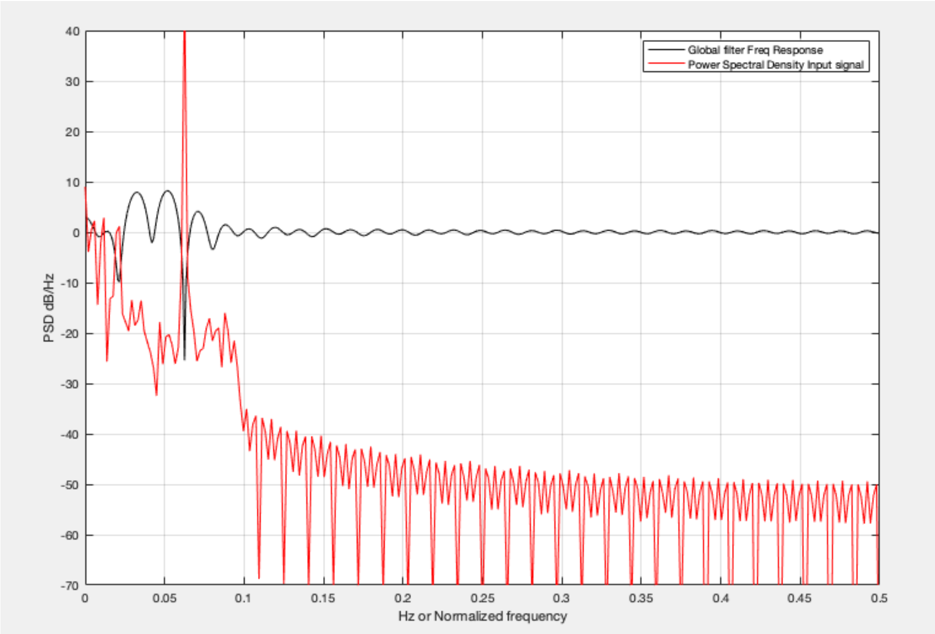
1.2.2. Filtre d'un senyal de parla

Ara analitzarem un cas més realista, on el senyal interferit és un senyal de parla. Per tant, ara no tindrem un procés blanc Gaussià.

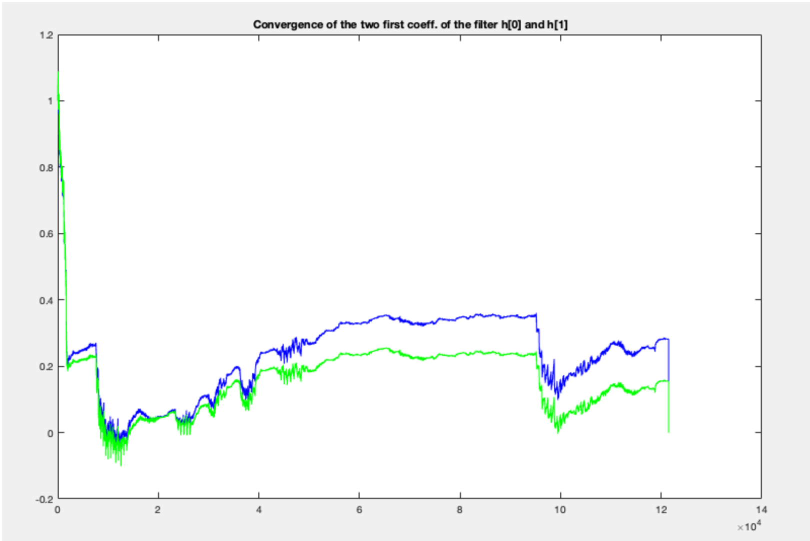
i) Execució del codi

Ara tornem a dissenyar el filtre de Wiener amb l'algorisme LMS. Representem gràficament les respostes freqüencials, la convergència dels dos primers coeficients del filtre, les power spectral densities i la comparació dels senyals d'entrada i de sortida.
Les respostes freqüencials del filtre de Wiener i el filtre global són:

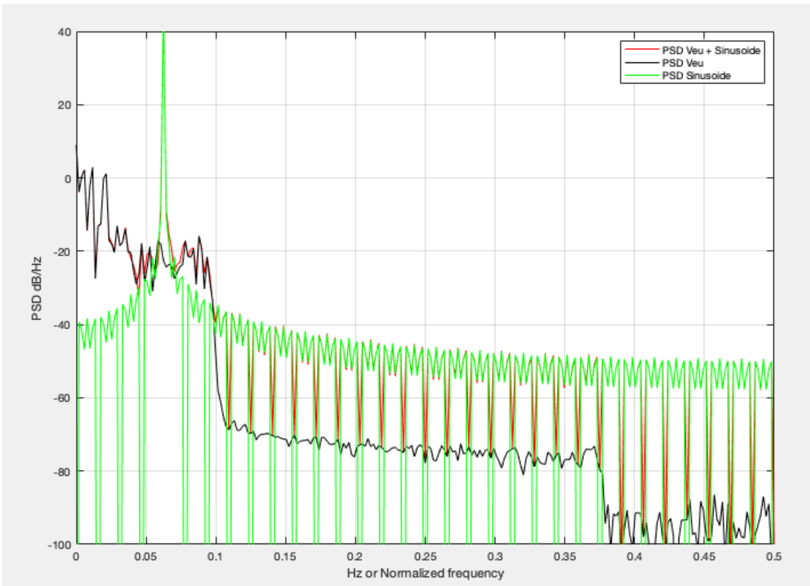




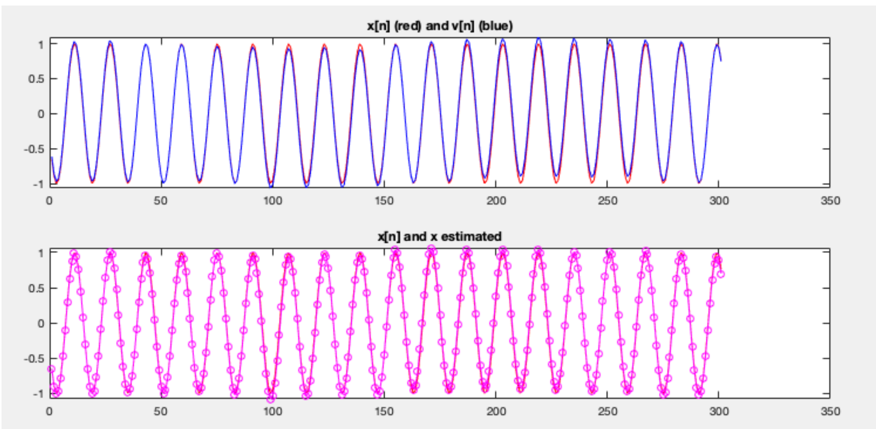
Convergència dels dos primers coeficients del filtre de Wiener:



Power spectral densities:



Senyals d'entrada i de sortida:



Una vegada executem el codi i tenim el senyal filtrat, el podem escoltar i comparar-lo amb el senyal de parla original. Quan escoltem v, s'escolta una interferència molt forta, però quan escoltem només w, es pot apreciar com la interferència s'ha eliminat.

ii) Anàlisi de les respostes freqüencials dels filtres

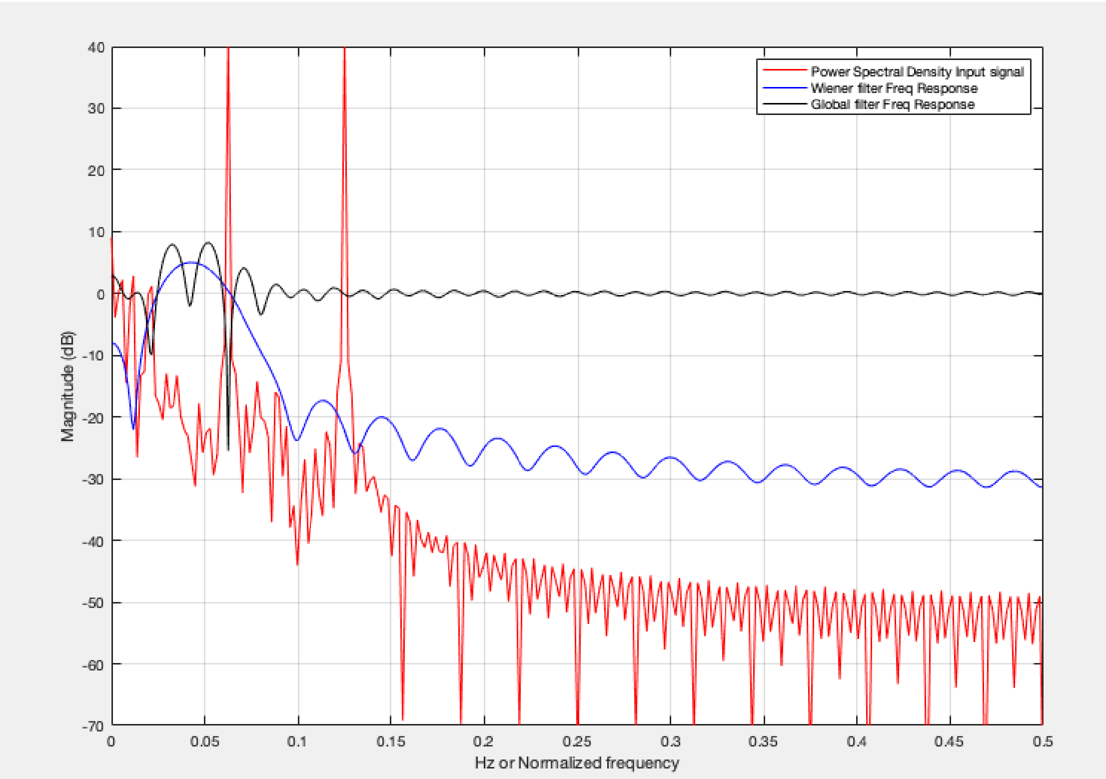
En l'apartat anterior hem representat gràficament les respostes freqüencials tant del filtre de Wiener com del filtre global, ara anem a analitzar-les.

El filtre de Wiener detecta la sinusoide i com es pot veure en el gràfic, fa un atenuament d'aquesta per intentar eliminar-la del senyal, el qual es veu perquè pren valors negatius durant la majoria de la senyal.

En la gràfica del filtre global podem veure que aquest té un pic clar en valors negatius, per tant podriem deduir que la interferencia del senyal es troba en aquell punt, ja que és el que més s'atenua. Aquest mínim coincideix amb el màxim de la PSD del senyal d'entrada.

iii) Modificació de la freqüència de la sinusoide després de N/2 iteracions

Ara canviem el codi perquè després de N/2 iteracions de l'algorisme, canviï la freqüència de la sinusoide.



En el gràfic veiem com el filtre es va adaptant als canvis. Ara tenim més d’un pic en la PSD, a més a més d’un canvi brusc de magnituds. Encara així, podem veure com el filtre segueix complint la seva funció i s’adapta als canvis que hi ha.

iv) Explicació del senyal de sortida amb D=1

Si determinem D=1, quan escoltem el senyal filtrar (westimate), pràcticament no s’escolta res. Això és perquè si la D és 1, el senyal considera que la interferència és com el senyal, i quan apliquem el filtre es resten i per tant s’eliminen pràcticament. Així que ens quedem amb un senyal que és quasi zero.