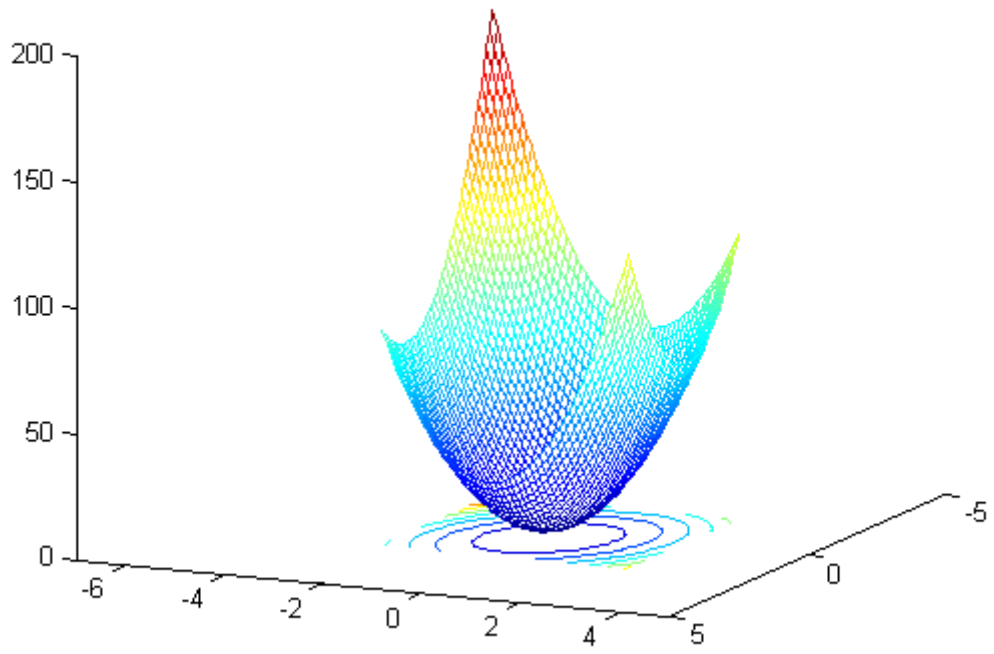
Johan Haarala
2002-12-11
Signaler och System

Adaptiva Filter



Abstract

Målet med den här rapporten är att ge en introduktion samt översikt till adaptiva filter. I den beskrivs några av de algoritmer som kan användas och deras egenskaper. Vilka fördelar och nackdelar som kommer med de olika algoritmerna tas upp samt hur man mäter prestanda på algoritmer generellt och problemet med att välja rätt algoritm.
Beskrivning av några applikationer och användningsområden för adaptiva filter tas också upp

Inledning

En kategori av signalbehandling är adaptiv signalbehandling, den är relativt ny och hela tiden ökar användandet av det i applikationer. Adaptiv signalbehandling uppstod genom att man ville kontrollera tidsvarianta system och har ökat beräkningskapaciteten hos digital teknologi.

Den största skillnaden mellan signalbehandling och adaptiv sådan är att i adaptiv behandlar man tidsvarierande digitala system. När adaptiva filter används på ickestationära signaler så krävs det mindre information om signalerna än om man skulle behandla dem med ett vanligt digitalt filter.

Adaptiva filter kan ändra på sig själva via sina insignaler. De används t.ex. i applikationer som kräver förändring i systemet beroende på signalförhållanden. Adaptiva filter behöver två olika indata, en insignal *x(n)* och en referenssignal *d(n)*.

Ett adaptivt filter har alltså förmågan att förändra sig genom att ändra på filterkoefficienterna. Nya koefficienter skickas från en koefficientgenerator vilken egentligen är en adaptiv algoritm som modifierar koefficienterna i förhållande till en insignal. Det adaptiva digitala filtret är vanligtvis en typ av FIR-filter (Finite Impulse Response) men kan också vara ett IIR-filter (Infinite Impulse Response). Adaptiva filter används i en rad applikationer och exempel på dessa är brusreducering, källseparering, signalprediktion och systemidentifiering.

Metod

För att skriva den här sammanfattningen av adaptiva filter och deras användningsområde har jag studerat litteratur i området signalbehandling och adaptiv sådan, speciellt då med inriktning mot adaptiva filter. Har även studerat ett liknande projekt med mer praktisk tillämpning och använt mig av slutsatser från det.
Arbetet kan ses som en introduktion till adaptiva filter och går inte så djupt in på problemen matematiskt utan är mer en översikt om egenskaper och funktioner hos adaptiva filter.

Adaptiva filter

Signaler som skapas i de flesta applikationer kommer att vara ickestationära och med icke adaptiva metoder skulle man behöva dela upp dessa ickestationära signaler i mindre stationära delar, d.v.s. man approximerar ett kortare tidsintervall av den ickestationära processen som att vara stationär.

Denna metod blir dock inte särskilt effektiv, t.ex. för snabbt varierande processer så måste tidsintervallet, för vilken processen antas vara stationär vara så kort att estimeringen blir dålig på grund av att man vill ha så hög upplösning. Dessutom skulle metoden inte fungera så bra om steglängden ändras under analysintervallet.

Det viktigaste är troligen att denna lösning inför en felaktig modell på de data man har erhållit, d.v.s. att de skulle vara stationära bitvis. Därför är det bättre att försöka använda sig av något som är ickestationärt från början.

Om $\mathbf{w}(n)$ är impulssvaret för ett FIR Wienerfilter som ger det skattade värdet $d_1(n)$ av minsta kvadratlösningen av en önskad process d(n),

$$d_1(n) = \sum_{k=0}^{p} w(k)x(n-k)$$

och om x(n) och d(n) är ”wide-sense stationary” (WSS) processer, d.v.s. processerna har medelvärde noll, och $e(n) = d(n) - d_1(n)$ så kan man finna koefficienterna som minimerar kvadratfelet med hjälp av Wiener-Hopf ekvationerna.

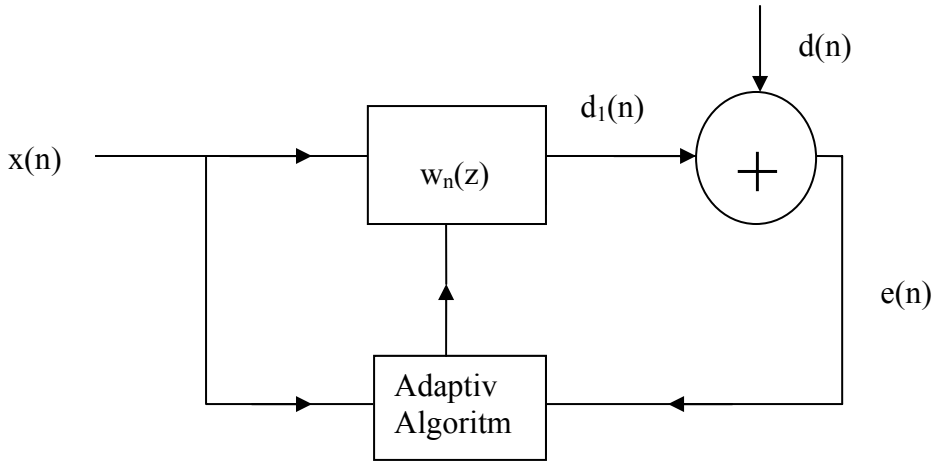
$$\mathbf{R}_x\mathbf{w} = \mathbf{r}_{dx}$$

Men om d(n) och x(n) är ickestationära så kommer filterkoefficienterna som minimerar $E\{|e(n)|^2\}$ att bero på n, d.v.s. filtret kommer att vara tidskiftsvariant. På många sätt är det svårare att göra ett tidskiftsvariant filter än att göra ett tidskiftsinvariant Wienerfilter, eftersom för varje värde på n måste man hitta de optimala filterkoefficienterna $\mathbf{w}_n(k)$.

Problemet kan göras enklare genom att låta $\mathbf{w}_n$ minimera minsta kvadratfelet vid varje tidpunkt n och istället titta på formeln för att uppdatera koefficienterna på formen:

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

Där $\Delta\mathbf{w}_n$ är uppdateringen som appliceras på filterkoefficienterna $\mathbf{w}_n$ vid tiden n för att få en ny uppsättning koefficienter $\mathbf{w}_{n+1}$ vid tiden n+1 (se bild nedan).



Den viktigaste delen med adaptiva filter är algoritmen för hur uppdateringen $\Delta\mathbf{w}_n$ ska utformas. Sekvensen av uppdateringar ska minska minsta kvadratfelet. Oavsett vilken algoritm som används så ska det adaptiva filtret ha följande egenskaper:

- I en stationär miljö ska ett adaptivt filter producera en sekvens av uppdateringar $\Delta\mathbf{w}_n$ så att $\mathbf{w}_n$ konvergerar mot lösningen av Wiener-Hopf ekvationerna.

- Man behöver inte känna till signalens statistik för att räkna fram $\Delta\mathbf{w}_n$. Estimeringen av auto- och korskorrelation ska vara inbyggda i det adaptiva filtret.

- För ickestationära signaler ska filtret kunna adaptera till förändringarna och följa lösningarna i tiden.

En viktig del i implementering av adaptiva filter är att felsignalen e(n) är tillgänglig för den adaptiva algoritmen. Det är genom e(n) som filtret kan mäta hur det fungerar samt bestämma hur koefficienterna ska ändras. Utan e(n) skulle inte filtret kunna adaptera fram nya koefficienter. I vissa applikationer är det lätt att få fram d(n) vilket gör det enkelt att få fram e(n).

Ett adaptivt filter är självskapande som använder sig av en rekursiv algoritm för att designa sig själv. Algoritmen startar från en gissning som baseras på information som finns om systemet. Sedan förfinas gissningen successivt i iterationer och förhoppningsvis konvergerar en optimal lösning fram.

Adaptiva filtertekniker används i många olika applikationer exempel på sådana är eko- och brusreducering och utjämning. Dessa applikationer innebär behandling av signaler som genereras av system vars karakteristik inte är känd på förhand. Det är på grund av detta som adaptiva filter kan ge en ökad prestanda om man jämför med de icke-adaptiva filtren.

Ett adaptivt filters prestanda mäts bland annat på dessa kriterier:

- *Konvergenshastighet*: Definierat som antalet iterationer som krävs för att algoritmen ska konvergera till en optimal lösning under stationära förhållanden.

- *Antalet beräkningar*: En viktig parameter i praktisk synpunkt som anger antalet operationer för en fullständig iteration. Påverkar i sin tur kostnaden att bygga det adaptiva filtret beroende på hur mycket datorkraft som behövs.

- *Numerisk robusthet*: Implementeringen av en adaptiv filteralgoritm på en dator medför kvantiseringsfel, vilka kan orsaka numerisk instabilitet. En adaptiv filtreringsalgoritm är numeriskt robust om den digitala implementationen är stabil.

Ett annat bra mått på prestanda är hur många beräkningar som krävs för att det adaptiva filtret ska uppnå en optimal lösning i stabil miljö. Detta mått kombinerar konvergenshastigheten och antalet beräkningar genom att ta produkten av dem.

Exempel på adaptiva filteralgoritmer

Följande är några av de vanligaste adaptiva algoritmerna som används i adaptiv signalbehandling.

- Least Mean Square (LMS)
- Normaliserad LMS (NLMS)
- Recursive Least Square (RLS)

Adaptiva FIR-filter

FIR-filter används t.ex. i adaptiva system som utjämnare i digitala kommunikationssystem och bruskontrollerande system. Anledningen till att FIR är så populärt är att stabiliteten är lätt att kontrollera genom att se till att koefficienterna är begränsade. Dessutom finns det enkla och effektiva algoritmer för att beräkna koefficienterna samt att algoritmerna är lätta att förstå i termer av konvergens och stabilitet.

Det finns en mängd olika anledningar att använda adaptiva filter. För det första i det fall då processen som skall filtreras är icke-stationär. I det fallet så är inte filtrets koefficienter tidsinvarianta, vilket medför att det existerar en optimal lösning av filtrets koefficienter för varje sampeltillfälle.

Även för stationära processer så finns det fördelar med att använda sig av adaptiva filter. Om filtret innehåller ett stort antal koefficienter så kan det t.ex. vara svårt att lösa Wiener-Hopf ekvationer i realtid.

Wiener-Hopf ekvation:

$$\mathbf{R}_x w = \sum_{l=0}^{p-1} w(l) r_x(k-l) = r_{dx}(k)$$

Dessutom kan autokorrelationsmatrisen, $\mathbf{R}_x$ vara dåligt konditionerad vilket medför att Wiener-Hopf ekvationerna blir känsliga för numeriska fel. Wiener-Hopf ekvationerna kräver också att autokorrelationen $\mathbf{r}_x$ och korskorrelationen $\mathbf{r}_{dx}$ är kända. De två finns sällan till hands och även om det går att approximera dem så resulterar beräkningarna i långa fördröjningar.

Steepest descent algoritmen

Om man vill skapa ett adaptivt FIR-filter kan man ta fram vektorn $\mathbf{w}_n$ vid tiden n som minimerar den kvadratiska funktionen:

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

Man kan minimera funktionen genom att sätta derivatan av ξ(n) till 0 och söka efter minimum.

Ett annat sätt är att använda ”*steepest descent”*-metoden som är en iterativ metod. Man låter då $\mathbf{w}_n$ vara en estimering av vektorn som minimerar ξ(n) vid tiden n. Vid tiden n+1 bildar man ett nytt estimat genom att addera en uppdatering till $\mathbf{w}_n$ som för $\mathbf{w}_n$ närmare en lösning.

Uppdateringen innebär att man tar ett steg μ i riktning mot lägsta punkten i den kvadratiska funktionen. Riktningen mot den lägsta punkten i den kvadratiska funktionen ges av den negativa gradient vektorn, som man får från den partiella derivatan av ξ(n) med avseende på koefficienterna **w**(k) gånger -1. Uppdateringsekvationen för $\mathbf{w}_n$ är

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

Steglängden μ påverkar hastigheten som koefficient vektorn rör sig mot den lägsta punkten på den kvadratiska ytan. För att koefficient vektorn ska röra sig i rätt riktning måste μ vara positivt. Med små värden på μ kommer rörelsen mot den lägsta punkten på den kvadratiska ytan gå mycket långsamt, medan ett större värde på μ kommer att öka hastigheten.

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

Med en steglängd μ blir *steepest decsent* algoritmen

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

Least Mean Square (LMS)

Det finns alltså en metod som kallas ”steepest descent” som kan användas för att konstruera adaptiva filter men nackdelen med denna är att man måste känna till E{e(n)x*(n)}, vilket normalt sett är okänt. Genom att använda sig av ”steepest descent”-metoden och en estimering av E{e(n)x*(n)} så fås LMS algoritmen.

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

Enkelheten hos LMS kommer ifrån att uppdateringen av den k:te koefficienten

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

endast kräver en multiplikation och en addition, värdet av μe(n) behövs bara räknas ut en gång och kan användas för alla koefficienter. Ett adaptivt LMS filter som har p+1 koefficienter behöver således bara p+1 multiplikationer och p+1 additioner för att beräkna de nya filterkoefficienterna. Ytterligare en addition behövs för att beräkna felet e(n) = d(n) – y(n) och en multiplikation för produkten μe(n). För utsignalen krävs p+1 multiplikationer och p additioner så den totala beräkningen blir 2p+3 multiplikationer och 2p+2 additioner.

LMS algoritmen konvergerar beroende på om steglängden μ ligger inom vissa värden och med en större steglängd konvergerar LMS fortare men allt för stor steglängd kan orsaka divergens[3]. Nackdelen med LMS är att den konvergerar långsamt speciellt om insignalen är färgat brus.

Normaliserad LMS (NLMS)

Ett av problemen med att designa och konstruera ett adaptivt LMS-filter är att välja vilken steglängd μ man ska använda sig av. Om man har stationära processer kan man använda sig av en tidsvarierande steglängd på formen,

$$\mu(n) = \beta / ||x(n)||^2$$

där β är en normaliserad steglängd som uppfyller $0< \beta < 2$. Ersätter man μ i LMS med μ(n) så får man den normaliserade LMS algoritmen som ges av:

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

Jämfört med LMS algoritmen så kräver den normaliserade LMS algoritmen flera beräkningar för att få fram normaliseringstermen $||x(n)||^2$. Men genom att beräkna denna rekursivt kan man minska de extra beräkningarna till bara två kvadratiska operationer, en addition och en subtraktion.

$$||x(n+1)||^2 = ||x(n)||^2 + |x(n+1)|^2 - |x(n-p)|^2$$

Normaliseringen innebär endast att variera steglängden, inte riktningen på gradientvektorn. Om $\mathbf{x}(n)$ är stor får LMS algoritmen problem med gradient brusförstärkning.
Genom att använda en variabel steglängd försvinner detta problem. Istället kan man få problem med att $||\mathbf{x}(n)||$ blir för litet. Man kan då modifiera NLMS till:

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

där ε är ett litet positivt tal.

Förutom NLMS algoritmen så finns ett antal andra modifieringar av LMS algoritmen, där varje modifikation är ett försök att förbättra en eller flera egenskaper hos LMS algoritmen. Ibland medför även dessa förbättringar några nackdelar på andra aspekter av algoritmen. Ett sätt att förbättra algoritmen är att använda variabel steglängd.

Rekursiva filter (RLS)

Precis som med linjära tidsinvarianta filter så har rekursiva (IIR) adaptiva filter en fördel jämfört med ickerekursiva filter genom att ge bättre prestanda för en given grad (antal filterkoefficienter). Men med ett adaptivt rekursivt filter finns problemet med instabilitet som kan påverka konvergenstiden hos filtret. Trots dessa problem så finns det många applikationer som kräver rekursiva adaptiva filter.

RLS (Recursive Least Squares) algoritmen kräver p^2 operationer vilket kan jämföras med LMS som kräver p operationer, mer exakt behöver RLS $3(p+1)^2 +3(p+1)$ multiplikationer och ungefär lika många additioner. Det som åstadkoms med den ökade komplexiteten hos beräkningarna jämfört med LMS är ökad prestanda. Generellt konvergerar RLS snabbare än LMS.

RLS skiljer sig från de algoritmer som bygger på LMS i att man inte använder gradienten för att minimera *mean-square error*. I stället försöker man minimera *least-square error* eftersom denna inte kräver statistisk information om signalen utan kan beräknas direkt från $x(n)$ och $d(n)$. Filterkoefficienterna som minimerar *least-square*-felet kommer att vara optimala för de aktuella data i signalerna, medan koefficienter som minimerar *mean-sqare*-felet är optimala beroende på den statistiska informationen i signalen. Detta innebär att olika realisationer av $x(n)$ och $d(n)$ kan komma att se olika ut då man använder RLS, även om den statistiska informationen är den samma i de olika realisationerna.

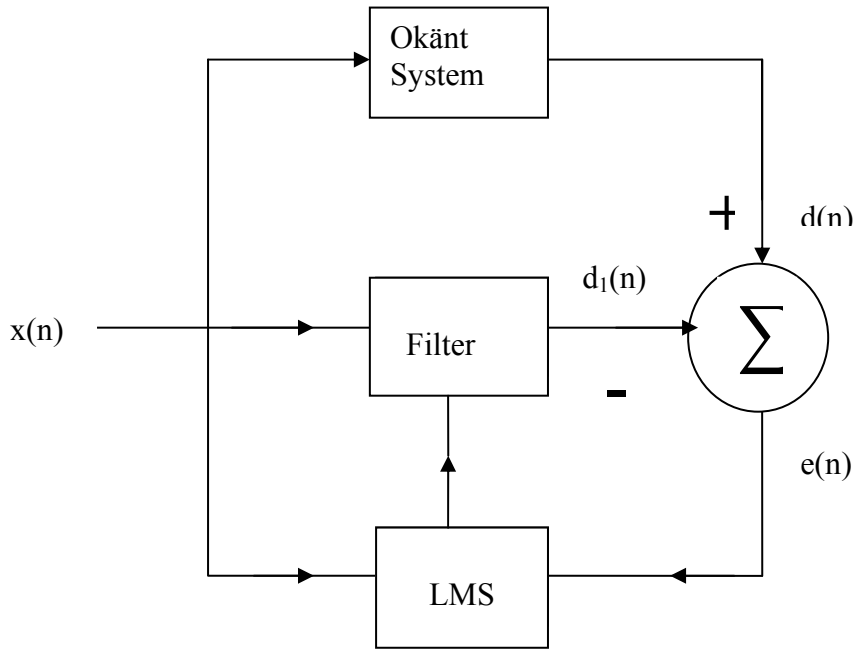
Exempel på användningsområden för adaptiva filter

Systemidentifiering

Ofta måste man skaffa sig en matematisk modell av ett okänt system. För att identifiera ett sådant system finns det två olika sätt att gå till väga. Det ena är att man använder sig av lagar från mekanik, fysik, ekonomi, biologi beroende på tillämpning. Då får man ett system av differentialekvationer. Den andra metoden är att använda sig av systemidentifiering, vilket innebär att konstruera en modell och estimera värden på de parametrar man har i modellen med hjälp av testdata.

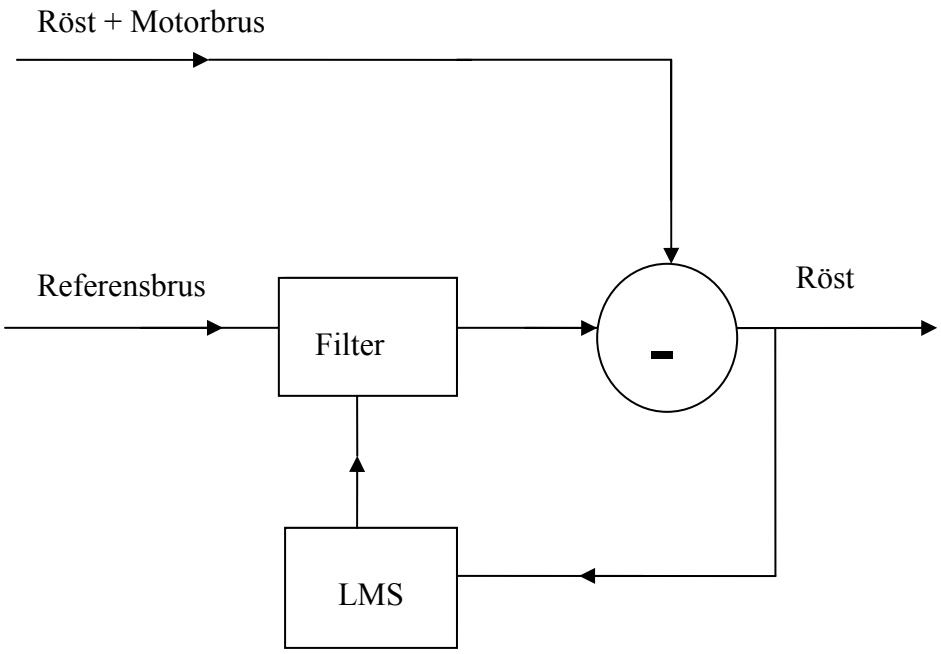
Adaptiva filter kan alltså användas till systemidentifiering. En signal x(n) matas in i både det okända systemet och filtret och utsignalerna från båda subtraheras till en felsignal e(n). Målet är att adaptera filterkoefficienterna så att felsignalen blir så nära noll som möjligt. När det adaptiva systemet slutligen konvergerar mot den sökta modellen så kan sedan det okända systemet identifieras med hjälp av det adaptiva filtrets koefficienter.

Det resulterande FIR-filtret representerar det tidigare okända systemet. För att nå en bra modell av alla intressanta frekvenser så måste de frekvenserna finnas i insignalen, vanligen används vitt brus som insignal x(n). Man kan t.ex. använda den beräkningseffektiva LMS algoritmen som adapteringsalgoritm.



Brusreducering

En av de vanligaste tillämpningarna av adaptiva filter är brusreducering. Ett exempel kan vara en person som pratar i mobiltelefon i bilen. Då rösten störs av bruset från bilens motor. LMS-algoritmen uppdaterar filterkoefficienterna så att felsignalen, mellan insignalen (Röst + Motorbrus) och det filtrerade bruset, blir så litet som möjligt, vilket förhoppningsvis resulterar i den riktiga röstsignalen.
Referensbruset kan vara en mikrofon som placerats nära motorn och insignalen mäts upp ifrån en mikrofon nära användarens mun så både tal och motorbrus fås med. En viktig detalj med att använda adaptiva filter vid brusreducering är att bruset inte får vara för likt den önskade signalen.



Slutsatser

Generellt så är adaptiva filter system som varierar med tiden på grund av att karakteristiken hos insignalerna varierar. Skillnaden mellan klassisk och adaptiv signalbehandling är att i adaptiv så förändras själva systemet med tiden.

Idealt så skulle man vilja ha ett adaptivt filter som kräver få beräkningar och är numeriskt robust. Det ska konvergera snabbt och enkelt kunna implementeras på en dator. Men som i många tekniska problem kan inte dessa krav uppfyllas samtidigt utan man får välja vad som är viktigast i situationen i fråga.

Många olika tekniker kan användas för att behandla ickestationära signaler med adaptiva filter. Dessa kan sedan användas i flera olika tillämpningar så som t.ex. systemidentifiering, signal modellering och brusreducering. En effektiv och enkel teknik är LMS algoritmen vilken inte behöver känna till några ensamblemedelvärden, nackdelen med den är att den konvergerar långsamt.

Den normaliserade LMS algoritmen gör valet av steglängd enklare vilket medför att filterkoefficienterna konvergerar. Om man använder variabel steglängd i LMS så är det bra att börja med en stor steglängd när koefficienterna är långt ifrån sina optimala värden och en kortare steglängd när de börjar närma sig sina optimala värden.

Rekursiva adaptiva filter är mycket mer komplexa än adaptiva FIR filter och eftersom de är rekursiva så kan de bli instabila på grund av detta är inte adaptiva rekursiva filter använda i lika många applikationer som de adaptiva FIR-filtrena.

En nackdel med RLS jämfört med LMS-baserade algoritmer är att den är beräkningsintensiv. RLS-algortimen konvergerar snabbare än LMS algoritmer dock kan den vara sämre på att följa en signal vid stora förändringar i signalen.

Referenser

1. Monson H. Hayes, *"Statistical Digital Signal Processing And Modelling"*, John Wiley & Sons.
2. Bernard Widrow, Samuel D. Stearns, *"Adaptive Signal Processing"*, Prentice Hall.
3. Adaptive Filters http://www.owlnet.rice.edu/~elec301/Projects00/site/.
4. David Crawford, *"Adaptive Filters"*, 1996.