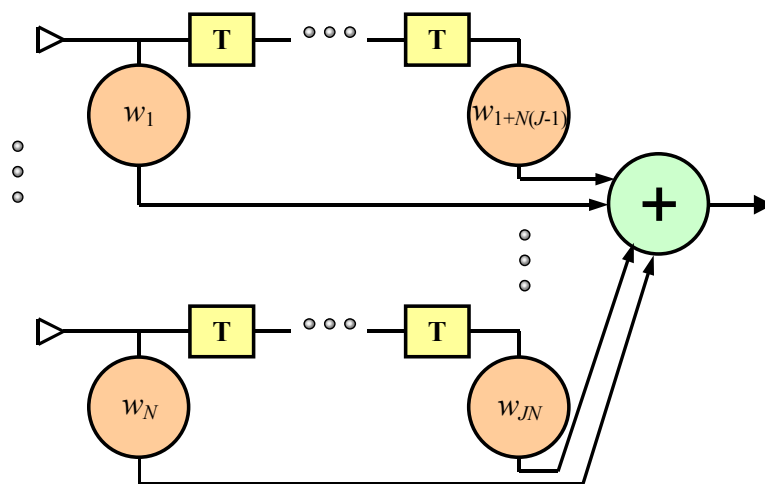


Lars Pettersson

## WBBF

Ett simuleringsverktyg för bredbandig adaptiv lobformning



TOTALFÖRSVARETS FORSKNINGSINSTITUT

Sensorteknik

Box 1165

581 11 Linköping

FOI-R--0605--SE

Oktober 2002

ISSN 1650-1942

**Teknisk rapport**

Lars Pettersson

**WBBF**

**Ett simuleringsverktyg för bredbandig adaptiv lobformning**

<b>Utgivare</b> Totalförsvarets Forskningsinstitut - FOI Sensorteknik Box 1165 581 11 Linköping	<b>Rapportnummer, ISRN</b> FOI-R--0605--SE	<b>Klassificering</b> Teknisk rapport
	<b>Forskningsområde</b> 4. Spaning och ledning	
	<b>Månad, år</b> Oktober 2002	<b>Projektnummer</b> E7759
	<b>Verksamhetsgren</b> 5. Uppdragsfinansierad verksamhet	
	<b>Delområde</b> 41 Ledning med samband och telekom och IT- system	
	<b>Författare/redaktör</b> Lars Pettersson	
<b>Projektledare</b> Fredrik Eklöf		
<b>Godkänd av</b>		
<b>Uppdragsgivare/kundbeteckning</b>		
<b>Tekniskt och/eller vetenskapligt ansvarig</b>		
<b>Rapportens titel</b> WBBF: Ett simuleringsverktyg för bredbandig adaptiv lobformning		
<b>Sammanfattning (högst 200 ord)</b> <p>I denna rapport beskrivs ett Matlab-baserat simuleringsverktyg kallat WBBF (Wide Band Beam Forming) för bredbandiga adaptiva gruppantennor. Verktöget är huvudsakligen gjort med tanke på kommunikationstillämpningar men även algoritmer mer passande för t.ex. radar är inkluderade. Signalerna byggs upp av nyttosignal och störare, vilka beskrivs med riktning, styrka och vågform, samt av brus. Antennen beskrivs med en modell som, även om den är ganska enkel, innehåller eventuella ofullkomligheter hos såväl antennelement som mottagare. Elementen kan beskrivas antingen med någon standardtyp eller med verkliga, mätta eller beräknade, data.</p> <p>Stor bandbredd vid lobformningen uppnås antingen med ett FIR-filter i varje elementkanal eller genom en uppdelning av processningen i delband. För cirkulära gruppantennor kan vi också arbeta med de s.k. fasskiftsmoderna vilket kan ge fördelar med bl.a. större bandbredd. Ett antal adaptiva algoritmer är implementerade såsom "Minimum Mean Square Error", "Constrained Power Minimization", "Maximum SINR", "Adaptive-adaptive beamforming", "Generering av 0-djup i skattade riktningar" och "Side Lobe Canceller". Ett antal metoder för att minska känsligheten för ofullkomligheter är också implementerade såsom diagonalbelastning, styrvektorprojektion och egenvärdesmodifiering samt även breddning av 0-djupen genom augmentering av kovariansmatrisen.</p> <p>Resultat ges i form av resulterande SINR-mått, samt resulterande frysta antenndiagram och frekvensgång. Med utgångspunkt från denna rapport ska det vara möjligt att använda simuleringsverktyget på ett effektivt sätt och att vidareutveckla det.</p>		
<b>Nyckelord</b> adaptiva antenner, simuleringsverktyg, MMSE, MVDR, Applebaum, Frost, Max SINR, störundertryckning, cirkulär array, STAP, flervägsutbredning		
<b>Övriga bibliografiska uppgifter</b>	<b>Språk</b> Svenska	
<b>ISSN</b> 1650-1942	<b>Antal sidor:</b> 60 s.	
<b>Distribution enligt missiv</b>	<b>Pris:</b> Enligt prislista	

<b>Issuing organization</b> FOI – Swedish Defence Research Agency Sensor Technology P.O. Box 1165 SE-581 11 Linköping	<b>Report number, ISRN</b> FOI-R--0605--SE	<b>Report type</b> Technical report
	<b>Research area code</b> 4. C4ISR	
	<b>Month year</b> October 2002	<b>Project no.</b> E7759
	<b>Customers code</b> 5. Contracted Research	
	<b>Sub area code</b> 41 C4I	
<b>Author/s (editor/s)</b> Lars Petterson	<b>Project manager</b> Fredrik Eklöf	
	<b>Approved by</b>	
	<b>Sponsoring agency</b>	
	<b>Scientifically and technically responsible</b>	
<b>Report title (In translation)</b> WBBF: A simulation tool for wide band adaptive beamforming		
<b>Abstract (not more than 200 words)</b> <p>In this report a Matlab-based simulation tool named WBBF (Wide Band Beam Forming) for wide band adaptive arrays is described. The tool is designed mainly with communication applications in mind but algorithms more suitable for e.g. radar are also included. The signals are composed of information signal and jammer or interference which are described by their direction, strength and wave form, plus noise. The antenna is described with a model which, even though it is relatively simple, include imperfections in the antenna elements as well as in the receivers. The elements can be described by a standardtype or by real, measured or computed, data.</p> <p>Wide bandwidth for the beamforming can be achieved either by a FIR-filter in each element channel or by division of the processing into several subbands. For circular arrays we may also work with phase shift modes which may give advantages such as wider bandwidths. A number of different adaptive algorithms are implemented such as "Minimum Mean Square Error", "Constrained Power Minimization", "Maximum SINR", "Adaptive-adaptive beamforming", "Formation of nulls in estimated directions" and "Side Lobe Canceller". A number of methods to decrease the sensitivity to imperfections are also implemented such as diagonal loading, steering vector projection and noise whitening and also widening of the adaptive nulls by covariance matrix augmentation.</p> <p>Results are given in the form of resulting SINR-measures and as resulting frozen antenna patterns and frequency behaviour.</p> <p>By using this report it should be possible to use the simulation tool in an efficient way and to develop it further.</p>		
<b>Keywords</b> adaptive antennas, simulation tool, MMSE, MVDR, Applebaum, Frost, Max SINR, interference suppression, circular array, STAP, multiple signal paths		
<b>Further bibliographic information</b>	<b>Language</b> Swedish	
<b>ISSN</b> 1650-1942	<b>Pages</b> 60 p.	
	<b>Price acc. to pricelist</b>	

## Innehållsförteckning

1. Inledning .....	7
2. Grundläggande teori.....	8
2.1 Antenn- och signalmodeller.....	8
2.1.1 Scenario.....	8
2.1.2 Kortfattad signal- och gruppantenn-teori .....	9
2.1.3 Antennbeskrivning .....	12
2.1.4 Signalbeskrivning.....	13
2.2 För de implementerade algoritmerna gemensam teori .....	14
2.3 Några korrektionsmetoder .....	15
2.4 Minimum Mean Square Error.....	17
2.5 Constrained Power Minimization.....	18
2.6 Maximum SNR.....	19
2.7 Side Lobe Canceller.....	19
2.8 Adaptiv-adaptiv processning .....	20
2.9 Generering av 0-djup i skattade riktningar .....	20
2.10 Spatiell filtrering.....	21
2.11 Uppdelning i delband.....	21
2.12 Sammanläggning av flervägs-komponenter .....	22
3. Simuleringsflöde.....	24
4. Konfigureringsprogram: WBkonf.....	26
5. Huvudprogram: WBSim3.....	30
5.1 Generering av datasampel .....	30
5.2 Beräkning av adaptiva vikter.....	31
5.2.1 Minimum Mean Square Error .....	31
5.2.2 Constrained Power Minimization.....	32
5.2.3 Maximum SNR .....	32
5.2.4 Adaptiv-adaptiv processning.....	32
5.2.5 Side Lobe Canceller .....	33
5.2.6 Generering av 0-djup i skattade riktningar.....	33
5.3 Generering av utdata.....	33
5.4 Utdata.....	34
6. Använda funktioner .....	36
6.1 adaptfun .....	36
6.2 adaptnorm .....	37
6.3 adaptnull .....	37
6.4 adaptstap .....	38

6.5 butinvfilt3.....	38
6.6 butler3 .....	39
6.7 cheby .....	39
6.8 combine.....	39
6.9 constraugment .....	40
6.10 constrdirmod .....	40
6.11 constrmat3.....	41
6.12 doa.....	41
6.13 doadir .....	42
6.14 doamaxfind .....	43
6.15 fftcperm.....	43
6.16 findsig .....	44
6.17 geometry .....	44
6.18 heffrippel.....	44
6.19 heffun3 .....	45
6.20 Hmatfilt.....	46
6.21 kanalS21.....	46
6.22 korrdelay .....	47
6.23 korrmaxfind.....	48
6.24 noisepar .....	48
6.25 Propmodel.....	48
6.26 Propsig .....	49
6.27 radpattern .....	49
6.28 randncmplx .....	50
6.29 raugmentBut.....	50
6.30 raugmentlin .....	50
6.31 rektfilt.....	51
6.32 RmatFB .....	51
6.33 Rssmooth .....	51
6.34 Rstap .....	52
6.35 Rtimestap .....	52
6.36 sdstap3.....	52
6.37 signals .....	53
6.38 specsignals .....	53
6.39 taylor .....	53
6.40 taylorcont .....	54

6.41 vdelay.....	54
6.42 WBxyvektor.....	54
6.43 weight0 .....	54
6.44 weights.....	54
7. Referenser .....	56
APPENDIX A. Alla indatavariabler i alfabetisk ordning .....	57

## 1. Inledning

Denna rapport beskriver ett Matlab-baserat simuleringsverktyg kallat WBBF (Wide Band Beam Forming) för bredbandiga adaptiva gruppantenner, specifikt version 14. Verktöget är huvudsakligen gjort inom projektet ELSA (Elektriskt styrbara kommunikationsantennor) med tanke på kommunikationstillämpningar där nyttsignalen är relativt stark och ett högt "signal-till-stör-plus-brus" förhållande SINR ("Signal to Interference plus Noise Ratio") önskas. Även algoritmer som är mer passande för t.ex. radar, där signalen är svag men ett snyggt antenndiagram med riktninginformation önskas, är dock också inkluderade. Bredbandighet i lobformningen uppnås antingen med hjälp av ett FIR-filter i varje antennkanal eller genom en uppdelning av processningen i delband.

Med hjälp av denna rapport ska läsaren förhoppningsvis dels kunna förstå de grundläggande teorierna, simuleringsprogrammets principiella uppbyggnad och in- och utdatavariablernas betydelse. Dessutom bör läsaren få förståelse för de olika algoritmernas karaktäristika och vilka parametrar som påverkar resultaten och hur resultaten kan förbättras med olika korrektionsmetoder. Några simuleringsresultat redovisas dock inte här utan det överlätes helt på användaren. Dessutom ska det vara möjligt att kunna rätta "buggar" (vilka naturligtvis finns) och att vidareutveckla simuleringsverktyget.

I kapitel 2 går den grundläggande teorin och huvudstrukturen för programmet igenom kortfattat. Där beskrivs hur signalen byggs upp av nyttsignal, störare och brus. Vidare beskrivs den fysikaliska modellen för gruppantennen. Denna modell är ganska approximativ men innehåller ändå de viktigaste parametrarna, och också eventuella ofullkomligheter hos såväl antennelement som mottagare, för att med dess hjälp kunna bedömma systemprestanda och de olika algoritmernas fördelar och nackdelar. Teorin för ett antal olika algoritmer vilka utnyttjar olika kunskap om nyttsignalen för att finna optimala parametrar beskrivs kortfattat. Några namn på dessa algoritmer är "Minimum Mean Square Error", "Constrained Power Minimization", "Maximum SNR", "Adaptive-adaptive beamforming", "Generering av noll-djup i skattade riktningar" och "Side Lobe Cancellation", men dessa namn är dock långt ifrån enhetliga i litteraturen.

Dessutom beskrivs ett antal implementerade metoder för att minska känsligheten för ofullkomligheter såsom diagonalbelastning, styrvektorprojektion och egenvärdesmodifiering samt även breddning av noll-djupen genom augmentering av kovariansmatrisen.

I kapitel 3 ges ett principiellt flödesdiagram för hela simuleringen medan kapitel 4 behandlar indatagenereringen i `WBkonf`.

Kapitel 5 behandlar huvudprogrammet `WBSim3` med dess olika delblock för datagenerering, till viss del identisk men till viss del olika för de olika algoritmerna, samt genereringen av resultaten. Dessa ges i form av resulterande SINR, samt resulterande antenndiagram och frekvensgång.

I kapitel 6 slutligen beskrivs alla de olika funktionsprogrammen.

Den ovanstående indelningen medför att vissa punkter delvis beskrivs i såväl kapitel 2, 5 och 6 med olika vinkling och djup.

Namn på programvariabler och Matlab-funktioner är skrivna i typsnittet `Courier`, specifika variabelalternativ samt matematiska variabler med *kursiv text* och vektorer och matriser med **fet skrift**.



## 2. Grundläggande teori

I detta kapitel beskrivs de modeller för signaler, antenn och algoritmer som ligger till grund för simuleringsverktyget. Beskrivningen har avsiktligt inte gjorts alltför detaljerad. För vissa ytterligare detaljer hänvisas till referenserna i kapitel 7 och till programbeskrivningarna i kapitel 4, 5 och 6.

Det råder en viss språkförbistring mellan olika discipliner. Med *kanal* menas här den del av signalvägarna som går från den till antennelementen infallande vågen, via ett antennelement, mottagare med förstärkare, filter och blandare till AD-omvandlare och digitala filter (och eventuell transformering till fasskiftsmoder som kan ske digitalt eller analogt via en Butlermatris). Vågutbredningen från sändare till mottagare, via eventuell flervägsutbredning, respektive den del av radiomottagaren som finns efter den adaptivt formade signalen ingår alltså inte.

Namnen på de olika adaptiva algoritmerna är inte heller entydigt definierad i litteraturen. I själva verket förekommer en mängd olika alternativ, ofta hänvisande till det sätt på vilket högerledet i ekvationen för de adaptiva vikterna (styrvektorn) väljs eller efter någon viss upphovsman, [1], [2], [3], [4], [5], [6].

Då man försöker minimera avvikelsen mellan en referenssignal och antennens utsignal består styrvektorn av korrelationen mellan signalen i kanalerna och denna referenssignal och några gångbara namn är "Minimum Mean Square Error" (MMSE), "Least Mean Square" (LMS) och Widrow's algoritm, se vidare 2.4.

Om algoritmen består av att minimera total uteffekt med ett antal bivillkor för känsligheten för vissa riktningar och frekvenser blir styrvektorn ett mer komplicerat uttryck i vilket korrelationsmatrisen själv ingår. Använda namn är då "Constrained Power Minimization", Frost's algoritm [9], "Minimum Variance Distorsionless Response" (MVDR) eller "Linearly Constrained Minimum Variance" (LCMV), se vidare 2.5.

Om styrvektorn består av en a priori antagen signalriktning är gångbara namn "Steering Vector Arrays", "Maximum SNR", Applebaum's [8] eller Howell-Applebaums algoritm, se vidare 2.6.

### 2.1 Antenn- och signalmodeller

#### 2.1.1 Scenario

De mot antennen infallande elektromagnetiska signalerna antas bestå av ett antal vågor, plana eller icke-plana, med ett relativt brett frekvensspektrum. Väl framme vid ett antennelement inducerar signalerna en spänning över en belastning, t.ex. ingången på en mottagarförstärkare. Signalerna förstärks, filtreras och blandas eventuellt ned för att så småningom hamna i en AD-omvandlare där den samplas till en digital signal för varje antennelement. Denna samplade signal translateras sedan till en komplex basbandssignal för att användas i den adaptiva processningen. Sampelfrekvensen tänkes här vara vald så att, efter decimering och generering av en komplex basbandssignal, signalens (dubbelsidiga) bandbredd och samplingsfrekvensen är lika, dvs.  $f_{\text{samp}} = B$ . Smalbandigare signaler kan genereras genom en filtrering men samplen blir då inte okorrelerade (vilket antas i vissa teoretiska satser inom området). I varje kanal finns också i huvudsak okorrelerat additivt vitt gaussiskt egenbrus (AWGN) med en total, nominell, bruseffekt  $P_{\text{noise}} = k_B T_{\text{ekv}} B$ .

Ett tänkt scenario med ett antal olika infallande vågtyper, såväl direkt som fördröjd nyttsignal, nyttsignal med distorderad fasfront, planvågsstörare och störare med distorderad fasfront visas i

Fig. 2.1. Där antyds också att antennelementen kan ha något olika elementdiagram och mottagarna något olika förstärkning. Effekten av dessa vågtyper diskuteras kortfattat i avsnitt 2.3 samt också i de specifika metodavsnitten 2.4 - 2.10.

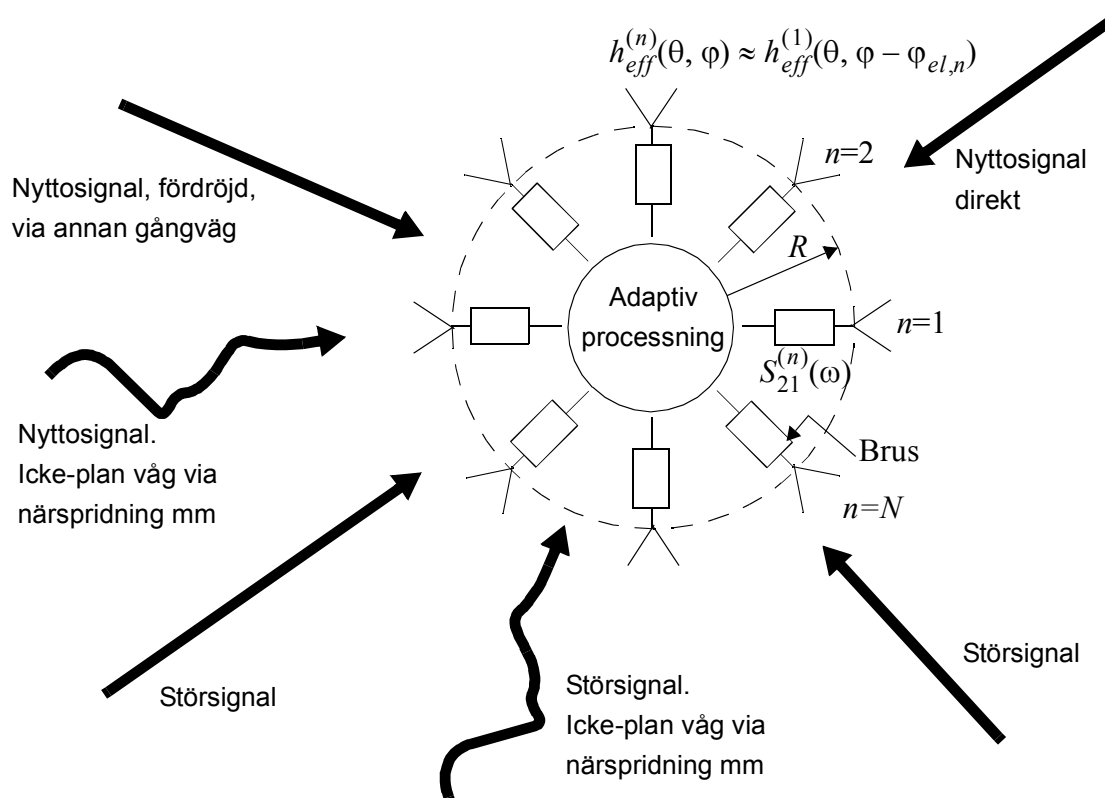


Fig. 2.1. Ett tänkt scenario med en cirkulär gruppantenn.

### 2.1.2 Kortfattad signal- och gruppantenn teori

De infallande vågorna träffar så antennen. Eftersom antennkanalernas känslighet är frekvensberoende transformerar vi signalerna till frekvensdomän via en Fouriertransform:

$$\tilde{\mathbf{E}}(\mathbf{k}) = \mathcal{F}(\mathbf{E}(t, \hat{k})) \quad (2.1)$$

där  $\mathbf{k}$  är vågvektorn och enhetsvektorn  $\hat{k}$  betecknar vågens riktning ( $\mathbf{k} = \frac{\omega}{c}\hat{k}$ ), som alternativt kan ges via vinklar  $\theta$  och  $\varphi$ . Signalerna i varje antennelementkanal,  $n$ , i position  $\mathbf{r}_n$  kan nu för en infallande plan våg beskrivas med hjälp av överföringsfunktioner för antennelementen,  $\mathbf{T}^{(n)}(\mathbf{k})$ , och mottagarna,  $S_{21}^{(n)}(\omega)$ , enligt:

$$\tilde{v}_n(\omega) = S_{21}^{(n)}(\omega)\mathbf{T}^{(n)}(\mathbf{k}) \cdot \tilde{\mathbf{E}}_{inc}(\mathbf{k})\exp(-j\mathbf{k} \cdot \mathbf{r}_n) \quad (2.2)$$

där effekttätheten för den plana vågen ges av  $S_{inc}(\mathbf{k}) = \frac{1}{2Z_0} |\tilde{\mathbf{E}}_{inc}(\mathbf{k})|^2$ . Överföringsfunktionen

$\mathbf{T}^{(n)}(\mathbf{k})$  kan ses som effektiv antennlängd, motsvarande den normala  $\mathbf{h}_{eff}^{(n)}$ , om vi inkluderar belastningsimpedansen i antennen. Funktionen  $\mathbf{T}^{(n)}(\mathbf{k})$  är vektoriell (signalen beror på vågens polarisation) och är en funktion av vågtalet  $\mathbf{k}$  dvs. av frekvens och infallsriktning  $(\omega, \hat{k})$  samt dessutom, i ett generellt fall, olika för alla element,  $n$ . Vi utgår i det följande från att alla signaler har samma polarisation varvid vi formellt kan glömma vektorkaraktären hos  $\mathbf{T}^{(n)}(\mathbf{k})$  (denna antas då vara given för "korrekt" polarisation). Om vi har en icke-plan våg, t.ex. p.g.a. spridning i antennens närområde) kan vi formellt tänka oss denna uppdelad i planvågskomponenter, vilket vi dock bortser ifrån här.

De mottagna tidssignalerna fås slutligen via en invers Fouriertransform:

$$v_n(t) = \mathcal{F}^{-1}(\tilde{v}_n(\omega)). \quad (2.3)$$

Om vi organiserar, de skalära, överföringsfunktionerna för vågornas polarisation  $T^{(n)}$  och de effektiva antennlängderna  $h_{eff}^{(n)}$  i Fig. 2.1 som vektorer  $\mathbf{T}$  och  $\mathbf{h}$  med längderna  $N$ , med ett element per antennelement, kan vi skriva följande samband mellan dessa vektorer, se t.ex. [16]:

$$\mathbf{T}(\omega, \hat{k}) = \mathbf{Z}_L(\mathbf{Z}_A + \mathbf{Z}_L)^{-1} \mathbf{h}(\omega, \hat{k}) \quad (2.4)$$

där  $\mathbf{Z}_A$  är gruppantennens impedansmatris och  $\mathbf{Z}_L$  dess belastningsimpedansmatris (ofta en skalär).

Om  $G_{el,r}^{(n)}(\omega, \hat{k})$  är realiserade antennvinsten för element  $n$ , i sin omgivning och med alla belastningsimpedanser, fås dess realiserade effektiva antennarea enligt (där vi, liksom nedan, utelämnar frekvens och riktningsargumenten  $\omega$  och  $\hat{k}$ ):

$$A_{eff,r}^{(n)} = G_{el,r}^{(n)} \frac{\lambda^2}{4\pi}. \quad (2.5)$$

Då mottagen effekt är  $P_r = S_{inc} A_{eff,r}$  får vi också ett samband mellan  $|T^{(n)}|$  och  $G_{el,r}^{(n)}$  respektive  $A_{eff,r}^{(n)}$  enligt:

$$|T^{(n)}| = \lambda \sqrt{\frac{G_{el,r}^{(n)} R_L}{4\pi Z_0}} \quad (2.6)$$

och

$$A_{eff,r}^{(n)} = |T^{(n)}|^2 \frac{Z_0}{R_L} \quad (2.7)$$

där vi också antagit att alla belastningsimpedanser är lika (ofta är  $Z_L = R_L = 50\Omega$ ).

Vi får nu också enkelt hela gruppantennens effektiva antenntyta refererat till förstärkarnas ingång. Om elementens positionsvektorer är  $\mathbf{r}_n$  och deras viktkoefficienter är  $w_n$  får vi

$$A_{eff}^{(A)} = \frac{Z_0}{R_L} \frac{\left| \sum_n w_n T^{(n)} e^{-jk\hat{k} \cdot \mathbf{r}_n} \right|^2}{\sum_n |w_n|^2} = \frac{Z_0}{R_L} \frac{|\mathbf{w}^T \text{diag}(e^{-jk\hat{k} \cdot \mathbf{r}}) \mathbf{T}|^2}{|\mathbf{w}|^2}. \quad (2.8)$$

Denna blir, enligt Schwarz olikhet, maximal då  $\mathbf{w}$  är konjugatet av produkten av de andra faktorerna i täljaren, vilka beror av både frekvens och infallsriktning, dvs.

$$\mathbf{w}_{opt}^* = \text{diag}(e^{-jk\hat{k} \cdot \mathbf{r}}) \mathbf{T}. \quad (2.9)$$

Med (2.6) får vi nu (med återinförande av  $\omega$  och  $\hat{k}$ )

$$A_{eff,max}^{(A)}(\omega, \hat{k}) = \max_{\mathbf{w}} (A_{eff}^{(A)}(\omega, \hat{k}, \mathbf{w})) = \sum_n A_{eff,r}^{(n)}(\omega, \hat{k}) \quad (2.10)$$

vilket naturligtvis kunde förväntas från början. Gruppantennens maximala realiserade antennvinst blir p.s.s. lika med summan av alla antennelements realiserade antennvinster

$$G_{r,max}^{(A)}(\omega, \hat{k}) = \max_{\mathbf{w}} (G_r^{(A)}(\omega, \hat{k}, \mathbf{w})) = \sum_n G_{el,r}^{(n)}(\omega, \hat{k}). \quad (2.11)$$

Dessa maxnivåer kan också tjäna som referensvärden att jämföra de resulterande parametrarna för vår adaptiva antenn med, enligt avsnitt 5.4 och 2.1.4.

Man kan här också notera att antennens resulterande direktivitet,  $D^{(A)}$ , naturligtvis blir något högre än  $G_{r,max}^{(A)}$  eftersom vi för  $D^{(A)}$  normerar strålningstätheten till den utstrålade nettoeffekten, given av:

$$P_{rad} = \mathbf{w}^H (\mathbf{I} - \mathbf{S}_A^H \mathbf{S}_A) \mathbf{w} \quad (2.12)$$

istället för till den tillgängliga effekten,  $\mathbf{w}^H \mathbf{w}$ . Här är  $\mathbf{S}_A$  antennarrayens yttre kopplingsmatris, given av impedansmatriserna  $\mathbf{Z}_A$  och  $\mathbf{Z}_L$  i (2.4). Maximalt möjlig direktivitet,  $D_{max}^{(A)}$ , där vi alltså får en "skarp" lob men också högre reflektionsförluster, fås emellertid inte för vikterna enligt (2.9) utan, genom maximering av ett uttryck liknande (2.8) men med (2.12) i nämnaren, av

$$\mathbf{w}_{D,opt} = (\mathbf{I} - \mathbf{S}_A^H \mathbf{S}_A) \mathbf{w}_{G_r,opt} \quad (2.13)$$

där  $\mathbf{w}_{G_r,opt}$  fås via (2.9).

För att på ett enkelt sätt kunna specificera signalstyrkorna tänker vi oss nu ett referensfall med en enda anpassad isotrop antenn varvid  $G_{el,r}^{(n)} \equiv 1$  och

$$|T| = \lambda \sqrt{\frac{R_L}{4\pi Z_0}}. \quad (2.14)$$

Vår antenn kan nu i följande avsnitt beskrivas med funktionerna  $T^{(n)}$  normerat till  $|T|$  i (2.14), dvs. enligt (2.6) som  $\sqrt{G_{el,r}^{(n)}}$  med tillhörande fas, medan vågornas signalstyrkor kan beskrivas med hjälp av mottagen signalstyrka för ett referensfall givet av (2.14). Som vi kommer att se tvingas vi dock oftast till vissa förenklingar.

### 2.1.3 Antennbeskrivning

Den antenn som vi tänker oss i denna rapport är en gruppantenn vars elementplaceringar kan vara godtyckliga. Det förutsättes dock att de ingående antennenelementen har identiska nominella strålningsdiagram relaterat till sitt lokala koordinatsystem. Detta blir speciellt enkelt då vi simulerar en linjär array med ekvidistanta element, dvs. en ULA (uniform linear array), eller en cirkulär array med lika elementvinklar, dvs. en UCA (uniform circular array). Vi kan dock ge antennenelementens strålningsdiagram individuella, slumpmässiga, rippel. Även andra antennformer kan simuleras, se 6.17, men elementens pekriktning är f.n. begränsad till en rotation i xy-planet (givet av en vinkel  $\varphi_{el}$ ). Det bör framhållas att antagandet om identiska elementdiagram är en approximation, speciellt för små icke-cirkulära arrayer. Elementdiagrammens form kan vi f.n. välja att ge via en indatafil, vilken helst då ska ha beräknats med någon elektromagnetisk simuleringsmetod eller vara mätt så att det därvid uppfyller fysikens lagar. Alternativt kan vi ange elementdiagrammen som någon enkel standardform, med den direktivitet som gäller för motsvarande isolerade (ensamma) antennenelement. Det senare innebär att vi därmed mycket väl kan bygga in en fysikalisk felaktighet i simuleringarna eftersom den ömsesidiga impedansen (kopplingen) och impedansvariationerna kan ha stor betydelse, t.ex. finns en naturlig begränsning på  $A_{eff,r}^{(n)}$  i (2.7) som den yta som upptas av elementet.

Ett exempel på ett fall där denna ömsesidiga påverkan dock kamoufleras och får mindre betydelse är om vi har en missanpassad dipolarray med mycket hög ingångsimpedans på förstärkarna vilka sitter direkt på dipolanslutningarna dvs., något oegentligt,  $Z_L \gg Z_A$  i (2.4). Detta kommer dock att kraftigt försämra kanalernas känslighet såvida inte det utifrån kommande, externa, bruset dominerar såsom i t.ex. kortvågsområdet. Missanpassningen (och det stora yttre bruset) i ett sådant fall måste man dock här inkludera själv via t.ex. signalstyrkorna.

För plana gruppantenner som är väl anpassade med elementavstånd mindre än  $\lambda/2$  är  $G_{el,r} \propto \cos\theta$ , där  $\theta$  är vinkeln från ytnormalen, med maxvärde givet av cellstorleken ( $A_{eff,r}^{(n)}$  motsvarar projicerad yta), så att ett amplitud-diagram av typen  $\sqrt{\cos\theta}$  kan anses mest fysikaliskt.

Det faktum att vi inte känner formen på de korrekta elementdiagrammen, vilket kan ha stor betydelse för vissa algoritmer, kan lätt simuleras genom att använda två olika diagram, ett "verkligt" (sant) som används vid simulering av scenariot och ett "antaget" (vad vi tror att det är) som används i signalbehandlingen vid formering av vilodiagram, bivillkor o.dyl.

För cirkulära gruppantenner, en UCA enligt Fig. 2.1, kan man utnyttja den inbyggda symmetrin genom att, i stället för att använda elementsignalerna direkt, transformera signalerna till fasskiftsmoder genom en diskret Fouriertransform i elementledd. Detta kan leda till vissa speciella användbara bredbandsegenskaper. Om elementen inte ligger för glest eller vi använder för höga

modnummer kommer alla moder att ha nästan rundstrålande frekvensoberoende diagramform med en fassfaktor  $\exp(jn\phi)$ . Genom att använda detta kan man generera nästan frekvensoberoende strålningsdiagram med en enda viktfaktor och ett korrektionsfilter per fasskiftsmod som kompenserar för modernas känslighetsvariationer i frekvens. I analog teknik kan sådana moder genereras med en s.k. Butlermatris där man dock kan notera att en port i Butlermatrisen korresponderande mot en fasskiftsmod  $n$  vid sändning kommer att (p.g.a. komplexkonjugering) svara mot en mod  $-n$  vid mottagning. Eftersom sådana Butlermatriser oundvikligen innebär förluster tänker vi oss här dessa, liksom naturligtvis dess digitala motsvarighet, vara placerad efter mottagarförstärkarna med deras brusgenerering. Se mer i t.ex. [17], [18] och avsnitt 6.6.

### 2.1.4 Signalbeskrivning

Vi definierar nu signalernas effektnivå relativt kanalmottagarnas nominella egenbrus, plus yttre brus, för en isotrop och impedans- och polarisationsanpassad förlustfri antenn enligt avsnitt 2.1.2. Bruset antas här vara vitt och okorrelerat mellan kanalerna. Signalstyrka  $P$  betyder alltså att  $SNR=P$  för ett sådant fall men att det är större eller mindre beroende på elementets realiserade antennvinst och verkligt mottagarbrus. Maximala  $SNR$  för hela arrayen blir då lika med  $P$  multiplicerat med maximal antennvinst enligt (2.11) (om  $G_{el,r}$  kan antas vara frekvensoberoende). Om man genererar ett vitt signalspektrum betyder det alltså, enligt denna definition samt (2.2) och (2.14), att spektrats magnitud före mottagarelementen (dvs.  $\tilde{E}_{inc}$ ) är linjärt ökande med frekvensen. Å andra sidan är ofta, i en väl dimensionerad bredbandig gruppantenn där elementavståndet är högst  $\lambda/2$ , elementens effektiva antennarea  $A_{eff,r}$ , och därmed  $|T|$  enligt (2.7), ungefär konstant med frekvensen så att om man väljer ett "enkelt" antennelement, se avsnitt 6.19, dvs. antennelementens strålningsdiagram och antennvinst vid en frekvens (gärna i övre bandkanten där kopplingseffekterna normalt är minst) så tenderar dessa effekter att ta ut varandra.

Sammanfattningsvis finner vi dock att, om vi inte har en rimlig modell av elementens mottagarfunktion, vi får akta oss för att dra alltför långtgående slutsatser vad gäller kritiska signalnivåer. Om vi har en bättre elementmodell, enligt 2.1.2 och 6.19, där elementdata anges över hela det använda bandbredden får vi naturligtvis bättre data.

Med ovanstående signalnormering genererar vi alltså den totala signalen i varje antennelement i frekvensdomän enligt:

$$v_n = \sum_m \sqrt{P_m} \cdot \sqrt{G_{el}^{(n)}(\omega, \hat{k})} \cdot \exp(-j\mathbf{k} \cdot \mathbf{r}_n + j\angle T^{(n)}(\omega, \hat{k})) + v_{noise} \quad (2.15)$$

där vi också inkluderat elementdiagrammets fas. Denna fas sätts dock till 0 i de standardtyper som inkluderats men tas med då vi använder riktiga, beräknade eller mätta, strålningsdiagram.

Brussignalen genereras i tidsdomän som en komplex gaussisk slumpvalssekvens med variansen 1 där vi har möjlighet till vissa variationer enligt 6.24 och 6.21, medan signalerna kan genereras som antingen en komplex slumpvalsfrekvens, som en ren sinussignal med slumpmässig startfas, som en BPSK-kod eller som en QPSK-kod enligt 6.38. Dessa lagras i en frekvensdomän- och en tidsdomänmatris för alla frekvens- respektive tidpunkter och alla kanaler. Frekvensdata används företrädesvis för att generera simuleringsdata och slutresultat samt vid eventuell uppdelning i delband medan tidsdata används huvudsakligen vid själva den adaptiva processningen. Detta betyder bl.a. att vi inte avsiktligt gör något som inte vore realistiskt i ett verkligt fall.

Sammanfattningsvis måste dock betonas att resultat, främst absoluta *SINR*-värden från enkla standardiserade elementdiagram, måste behandlas med försiktighet.

## 2.2 För de implementerade algoritmerna gemensam teori

Syftet med de adaptiva algoritmerna är att hitta en uppsättning optimala viktcoefficients som bildar en, eller flera, viktvektorer  $\mathbf{w}_{opt}$  vilka, då de appliceras på en signalvektor  $\mathbf{x}(t_k)$ , ger en i någon mening optimal utsignal för ett scenario enligt Fig. 2.1. Vikterna är i denna rapport definierade på det sätt som är standard i antensammanhang dvs. så att utsignalen  $s(t_k)$  ges av:

$$s(t_k) = \sum_n w_n x_n(t_k) = \mathbf{w}^T \mathbf{x}(t_k) \tag{2.16}$$

vilket skiljer sig med en komplexkonjugering av  $\mathbf{w}$  från den definition som är vanlig i signalbehandlingsammanhang (varvid vi skriver  $s(t_k) = \mathbf{w}^H \mathbf{x}(t_k)$  där  $H$  står för hermite-konjugering). För den som hellre följer den senare definitionen torde nödvändiga modifieringar vara uppenbara.

Summationen i (2.16) tänker vi oss gjord som i Fig. 2.2 med ett FIR-filter med  $J$  tappar i var och en av de  $N$  antennelementen med signalvektorn  $\mathbf{x}(t_k)$  av längd  $NJ$  bestående av, de samplade och digitaliserade och fördröjda, signalerna  $v_n(t_k)$  enligt (2.3) och där  $t_k = t_0 + kT$  med samlingsintervallet  $T = 1/f_{samp}$ .

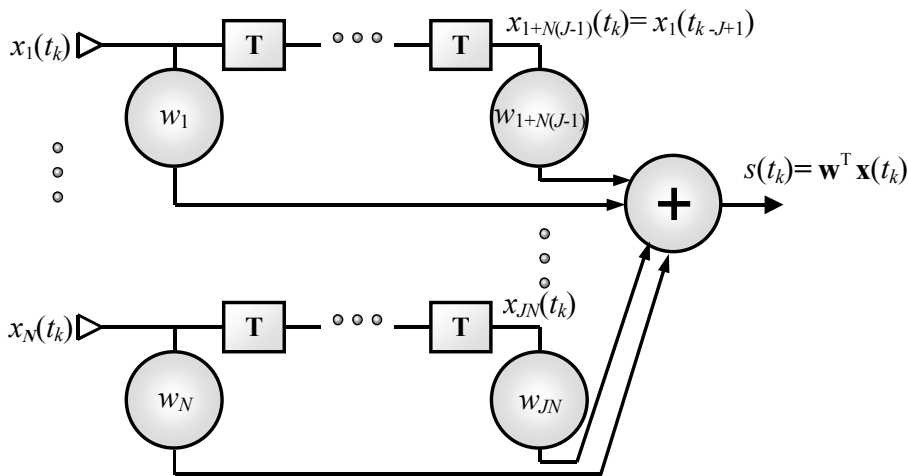


Fig. 2.2. Summering av signaler från  $N$  antennelement och  $J$  filtertappar med “Space Time Adaptive Processing”, STAP. T= fördröjning med ett samplingsintervall. Figuren motsvarar den inre cirkeln i Fig. 2.1.

De allra flesta av de här implementerade algoritmerna leder fram till en ekvation för den optimala viktvektorn enligt

$$\mathbf{w}_{opt}^* \propto \mathbf{R}^{-1} \mathbf{s} \tag{2.17}$$

där  $\mathbf{s}$  är en styrvektor som beror av algoritmen men som i huvudsak beskriver hur (konjugatet av) viktvektorn ser ut då signaler, nytto- såväl som stör-, saknas och vi bara har lika och okorrelerat vitt brus (AWGN) i alla antenkanaler och filtertappar. Matrisen  $\mathbf{R}$  är korrelationsmatrisen<sup>1</sup> mel-

lan alla signaler,  $\mathbf{x}(t)$  enligt ovan, vars matriselement ska bestå av ett statistiskt väntevärde men som vi, eftersom vi bara har ett ändligt antal sampel, skattar som ett tidsmedelvärde, dvs.

$$\mathbf{R} = E[\mathbf{x}(t)\mathbf{x}^H(t)] \approx \frac{1}{K} \sum_{k=1}^K \mathbf{x}(t_k)\mathbf{x}^H(t_k). \quad (2.18)$$

Man kan visa att för vitt okorrelerat gaussiskt brus kommer de skattade matriselementens standardavvikelse att asymptotiskt avta som  $1/\sqrt{K}$  och brusegenvärdernas relativa standardavvikelse som  $\sqrt{NJ/K}$ , se t.ex. [14] (generaliserat till  $J$  tappar).

### 2.3 Några korrektionsmetoder

Om ekvation (2.18) ger en bra skattning av den riktiga korrelationsmatrisen och om vi känner en "korrekt" styrvektor  $\mathbf{s}$  i (2.17) kommer de simulerade algoritmerna att ge ett kraftigt förbättrat *SINR*, såvida inte antalet störare är större än antalet antennelement.

Naturligtvis förekommer dock en hel del fel och ofullkomligheter som gör att resultatet inte blir optimalt men några av dessa går att minska även då de inte är explicit kända (i vilket fall de naturligtvis bör elimineras). Dessa kan t.ex. bestå av felaktigt angiven nyttsignalriktning, felaktigt antagande om antenmodell, distorderad fasfront, eller för få sampel vid skattning av korrelationsmatrisen. De olika algoritmerna reagerar olika för dessa s.k. modellfel.

Algoritmen "Minimum Mean Square Error" i 2.4 är relativt okänslig för sådana fel förutsatt att vi verkligen hittar träningssekvensen. "Constrained Power Minimization" i 2.5 kan däremot vara mycket känslig då en distorderad nyttsignal kan tolkas som en störare och kan komma att undertryckas kraftigt. "Maximum SNR" i 2.6 har liknande egenskaper som "Constrained Power Minimization" eftersom den motsvarar ett fall med bara ett bivillkor. Dock tänker man sig oftast denna algoritm tillämpad på det fall att nyttsignalen saknas (eller är mycket svag som i radar) vid bildandet av korrelationsmatrisen varvid algoritmen blir okänslig för dessa modellfel. "Sidelobe Canceller" i 2.7 kan liknas med "Maximum SNR" men är ofta mindre känslig för få sampel eftersom antalet frihetsgrader är mindre. Detsamma gäller också "Adaptiv-adaptiv processning" i 2.8. Vid "Generering av 0-djup i skattade riktningar" enligt 2.9 försöker man skapa 0-djup i de skattade störriktningarna varför denna algoritm blir känslig för distorsion av dessa störvågor men mindre känslig för distorderade nyttoågor förutsatt att man verkligen lyckats identifiera dessa korrekt och att distorsionen inte är så stor att antennvinsten i denna riktning blir låg. "Spatiell filterning" slutligen är inte en adaptiv algoritm i vanlig mening och är också relativt okänslig för modellfel på samma sätt som konventionell lobformning.

Det finns i huvudsak tre metoder att modifiera den samplade korrelationsmatrisen  $\mathbf{R}$  enligt (2.18) för att förbättra resultaten, se mer i t.ex. [14].

En metod, kallad diagonal-belastning (diagonal loading) eller artificiell brusaddition, består av att addera en diagonalmatris med identiska element,  $D_{last}\mathbf{I}$  (normerat till egenbruset), till  $\mathbf{R}$ . En sådan matris är lika med den asymptotiska bruskorrelationsmatrisen för ett idealt fall. Detta ger två olika effekter. Dels kommer de statistiska variationerna p.g.a. brussamplingen (som annars har en relativ standardavvikelse av  $K^{-1/2}$ ) att minska med en faktor  $1 + D_{last}$  och dels kommer signalernas relativa styrka att minska med samma faktor. Det första kommer att stabilisera alla fluk-

1. Matrisen  $\mathbf{R}$  kallas oftast kovariansmatris i litteraturen. I en kovariansmatris ska dock avvikelsen av  $\mathbf{x}(t)$  från dess väntevärde ingå och om denna är noll, men inte annars, är korrelationsmatris och kovariansmatris identiska. Detta väntevärde är naturligen noll för RF-signalerna men inte nödvändigtvis för de komplexa, nedblandade, basbandssignalerna.



tuationer som beror på att vi försöker minimera det specifika bruset i det samplade intervallet istället för ett statistiskt medelvärde. Det andra kommer att minska känsligheten för de fall då ett felaktigt antaget utseende på nyttsignalvektorn betraktas som en störare av algoritmen. Alla egenvärden, men inga egenvektorer, kommer att påverkas av denna diagonalbelastning. Man kan också låta diagonalelementen ha olika värden varvid en taperad viloviktvektor med vikter lika med (något förenklat)  $1/(D_{last,n} + 1)$  fås, [14] och [15], vilket fungerar också då nyttsignalen är inkluderad i  $\mathbf{R}$  enligt 2.5, där en taperad styrvektor som i 2.6 inte fungerar eftersom nyttsignalen då kommer att undertryckas.

En annan metod, kallad brusvitning ("noise whitening") eller egenvärdesmodifiering, består av att identifiera de egenvärden till  $\mathbf{R}$  som hör till bruset (brusunderrummet) samt att sätta dessa lika innan  $\mathbf{R}$  återskapas (i praktiken behöver  $\mathbf{R}$  oftast inte alls återskapas utan man bildar inversen direkt). Denna metod kommer också att minska variationen orsakade av för få brussampel men inte att påverka de relativa signalstyrkorna.

En tredje metod, kallad styrvektorprojektion eller "rank-reduktion", grundar sig på det förhållandet att om nyttsignalen finns närvarande vi skattningen av  $\mathbf{R}$ , och eftersom styrvektorn för ett bra resultat i någon mening ska vara en god representation för denna nyttsignal, så bör denna styrvektor helt ligga i signalunderrummet till  $\mathbf{R}$ . Om så inte är fallet finns stor risk att nyttsignalen tolkas som en störare och kraftigt undertrycks. Om vi alltså ser till att styrvektorn  $\mathbf{s}$  är en projektion på  $\mathbf{R}$ 's signalunderrum eller, vilket ger samma resultat, att vi endast använder en pseudoinvers till  $\mathbf{R}$  med brusegenvektorerna borttagna kommer resultatet ofta att kraftigt förbättras. Som exempel se [20].

Att kombinera dessa olika metoder synes inte vara av större mening. Den tredje metoden ovan förtar helt effekten av den andra medan den både tredje och den andra till stor del, men inte helt, minskar effekten av den första.

För de algoritmer som ska ha en styrvektor som beskriver en infallande våg, eller ett snyggt vilodiagram, finns två alternativ om de olika antennelementen och/eller förstärkarna har olika överföringsfunktioner,  $T^{(n)}$  respektive  $S_{21}^{(n)}$  enligt 2.1.2. Vi kan dels försöka avbilda en inkommande planvåg som den ser ut efter samplingen eller vi kan generera en vektor som motverkar distorsionen och, tillsammans med ett störfritt scenario (med bara brus, t.ex. kraftig diagonallast) ger ett önskat vilodiagram. I ett idealt fall med en linjär uniform array (ULA) ger dessa identiska resultat men inte annars. Se även appendix H i [16] och delvis 6.44.

Speciellt för smalbandig adaptivitet kommer de resulterande 0-djupen i störriktningarna att bli mycket smala vilket kommer att accentuera den smala bandbredden. Bredare, och därmed bredbandigare, 0-djup kan åstadkommas om man noterar att korrelationsmatrisen för en ULA med smalbandig planvågsstörare har en speciell struktur,  $R_{n,m} \propto \exp(jk(x_n - x_m)\sin\varphi)$ . En motsvarande struktur finns även för en cirkulär array, en UCA, med Butlermatris, förutsatt att elementen inte sitter alltför glest och att vi inte använder moder av alltför hög ordning ([17], [18]), där  $R_{n,m} \propto \exp(j(n-m)\varphi)$ . Härigenom kan vi lätt bilda nya artificiella planvågssignaler, och därmed 0-djup, vid sidan av de verkliga genom att multiplicera matriselementen i  $\mathbf{R}$  med ovanstående faktorer med  $\sin\varphi$  respektive  $\varphi$  svarande mot avståndet från de verkliga signalerna och addera till det verkliga  $\mathbf{R}$ . Se mera i [10] samt i avsnitten 6.29 och 6.30. Eftersom extrasignaler också kommer att bildas runt nyttsignalen kan huvudloben komma att påverkas kraftigt. Vid MVDR, avsnitt 2.5, kan vi för att motverka detta lägga till extra bivillkor runt denna nyttsignal. För övriga algoritmer bör nyttsignalen helst vara svag så att även extrasignalerna blir svaga. Denna metod bör därför användas med viss försiktighet.

## 2.4 Minimum Mean Square Error

I denna algoritm (MMSE) försöker vi se till att den resulterande utsignalen  $\mathbf{w}^T \mathbf{x}(t_k)$  är så lika en känd referenssignal, eller träningssekvens,  $d(t)$  som möjligt varvid vi också tillåter referenssignalen att ha en godtycklig fördröjning  $\tau$ .

Vi söker alltså den viktvektor  $\mathbf{w}$  och den fördröjning  $\tau$  som minimerar

$$|e(\tau)|^2 = E \left\{ \left\| d(t - \tau) - \mathbf{w}^T \mathbf{x}(t) \right\|^2 \right\}. \quad (2.19)$$

Teorin behandlas i t.ex. [1], [2], [3] och [4] och resultatet är att optimal viktvektor fås som (jfr. (2.17)):

$$\mathbf{w}_{opt}^* = \mathbf{R}^{-1} \mathbf{s}_d \quad (2.20)$$

där styrvektorn  $\mathbf{s}_d$  nu är korrelationen mellan vår signalvektor, med ett vektor-element per antenn-element och filtertapp, och vår referenssignal  $d(t)$  med lämplig fördröjning  $\tau$  enligt

$$\mathbf{s}_d(\tau) = E \{ \mathbf{x}(t) d^*(t - \tau) \} \approx \frac{1}{K} \sum_{k=1}^K \mathbf{x}(t_k) d^*(t_k - \tau) \quad (2.21)$$

där referenssignalen antas innehålla  $K_{train}$  sampel med  $K_{train} < K$  så att hela referenssignalen  $d(t - \tau)$  ryms inom de  $K$  datasampelen ( $0 \leq K_{train} \leq K - \tau_{max}/T_{samp}$ ).

Vi kan också formulera algoritmen som att, givet den optimala fördröjningen  $\tau$ , söka en minsta-kvadratlösning till det överbestämde problemet (jfr. (2.16))

$$\mathbf{w}^T \mathbf{X} = \mathbf{d} \quad (2.22)$$

där  $\mathbf{X}$  är en matris och  $\mathbf{d}$  en radvektor bildad av de  $K$  sampel i  $\mathbf{x}(t_k)$  respektive  $d(t_k - \tau)$ . Minsta-kvadratlösningen till (2.22) är

$$\mathbf{w}_{LS}^* = (\mathbf{X}\mathbf{X}^H)^{-1} \mathbf{X}\mathbf{d}^H \quad (2.23)$$

vilket är identiskt med (2.20) med de skattade  $\mathbf{R}$  och  $\mathbf{s}_d$  enligt (2.18) och (2.21) insatta.

Den minimerade kvadratsumman i (2.19) blir, med den optimala viktvektorn enligt (2.20):

$$|e(\tau)|_{min}^2 = E \{ |d(t)|^2 \} - \mathbf{s}_d^H(\tau) \mathbf{R}^{-1} \mathbf{s}_d(\tau) \approx \frac{1}{K} \sum_{k=1}^K |d(t_k)|^2 - \mathbf{s}_d^H(\tau) \mathbf{R}^{-1} \mathbf{s}_d(\tau). \quad (2.24)$$

Förutom inversen till korrelationsmatrisen behöver vi här alltså finna den fördröjning,  $\tau$ , som ger det bästa resultatet. (Vid flervägsutbredning söker vi ett antal "bästa" fördröjningar  $\tau_n$ .) Detta kan göras på några olika sätt varav tre är implementerade här enligt avsnitt 6.22. En metod består av att minimera (2.24) m.a.p.  $\tau$  och de två andra av att söka maximum till beloppet av något av

elementen i (2.21) respektive till en linjärkombination av dessa element (vilken bildar en konventionell antennlob). Som exempel se [21].

Man kan visa att det, se kap. 6 i [1] (generaliserat till  $J$  tappar), utan någon korrektion enligt avsnitt 2.3 behövs approximativt

$$K \geq 2NJ \left( 1 + \frac{1}{2 \cdot SINR_{opt}} \right) \quad (2.25)$$

sampel för att förväntat  $SINR$  ska vara inom 3 dB från dess asymptotiska, optimala, värde  $SINR_{opt}$  (approximativt  $PG_{r,max}^{(A)}$  enligt 2.1).

## 2.5 Constrained Power Minimization

I denna algoritim, som också kallas "Minimum Variance Distorsionless Responce" (MVDR) mm., försöker vi att minimera den totala uteffekten under ett antal bivillkor. Dessa bivillkor säger att systemets känslighet för vissa signaler från given riktning och vid given frekvens skall vara lika med förutbestämde värden. Det bör understrykas att vi inte får förväxla känslighet med antennvinst (gain). En hög "känslighet" kan t.ex. fås genom höga viktcoefficienter och låg direktivitet. Eftersom den adaptiva algoritmen försöker att trycka ner allt som inte ser ut som bivillkorsvektorerna är det här mycket viktigt att dessa bivillkor är korrekt valda. En riktning som ligger något snett i förhållande till den verkliga nyttosignalen kan lätt resultera i stark undertryckning av denna, särskilt om signalen är stark. Likaså kan amplitud- och fasfel, eller att vi ej känner antennelementens strålningsdiagram, ge samma resultat. Teorin beskriv utförligt i t.ex. [9] och [2] (där vissa definitioner skiljer sig på en komplexkonjugering mot de här) och resultatet kan skrivas som:

$$\mathbf{w}_{opt}^* = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{f}_c^* \quad (2.26)$$

där  $\mathbf{C}$  är en matris som beskriver bivillkorsvektorerna, med utsignalen vid ett bivillkor i varje kolumn, och  $\mathbf{f}_c$  är en vektor som beskriver själva bivillkorens värden.

Man kan även tänka sig att ha andra typer av bivillkor, t.ex. att derivatan m.a.p. en vinkel ska vara noll. Sådant är inte implementerat här (blir bl.a. besvärligt i ett generellt antennfall, dvs. icke-ULA). Sådana bivillkor kan dock lätt approximeras m.h.a. två närliggande bivillkor.

Om alltså signalvektorn  $\mathbf{x}$  är lika med en av kolumnerna (nummer  $n$ ) i  $\mathbf{C}$  fås utsignalen:

$$v_{ut} = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{C}_{kol(n)} = [\mathbf{w}^T \mathbf{C}]_{kol(n)} = [\mathbf{f}_c^T (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})]_{kol(n)} = f_{c,n}. \quad (2.27)$$

Vi ser här också att bivillkoren blir uppfyllda oavsett vilka manipulationer vi gjort på matrisen  $\mathbf{R}$  förutsatt att den fortfarande är Hermitsk.

Om vi endast har ett enda bivillkor blir  $\mathbf{C}$  en kolumnvektor,  $\mathbf{f}_c$  ett tal, och (2.26) blir formellt identiskt med övriga algoritmer, t.ex. (2.29) med  $\mathbf{s}_d \rightarrow \mathbf{C}$ .

Man kan visa att, se kap. 6 i [1] (generaliserat till  $J$  tappar), om nyttosignalen och antennen är helt kända, och utan någon korrektion enligt avsnitt 2.3, det behövs approximativt

$$K \geq 2NJ(1 + SINR_{opt}/2) \quad (2.28)$$

sampel för att förväntat  $SINR$  ska vara inom 3 dB från dess asymptotiska, optimala, värde  $SINR_{opt}$  (approximativt  $PG_{r,max}^{(A)}$  enligt 2.1).

## 2.6 Maximum SNR

I denna algoritm försöker vi optimera det resulterande signalbrusförhållandet, SNR (egentligen  $SINR$ ), givet att signalen a priori definieras som en våg kommande från en given riktning. Den går under diverse olika namn såsom “Maximum SNR”, “Applebaum’s” och “Howell-Applebaum’s” algoritm. I härledningen av de optimala vikterna förutsätts att den nyttiga signalen inte är närvarande under bildandet av korrelationsmatrisen  $\mathbf{R}$  och de optimala vikterna ges då av:

$$\mathbf{w}_{opt}^* \propto \mathbf{R}^{-1} \mathbf{s}_d \quad (2.29)$$

där styrvektorn  $\mathbf{s}_d$  definierar nyttig signal. Ofta, som här, normeras också viktvektorn så att utsignalens styrka blir ett om nyttosignalen verkligen vore  $\mathbf{s}_d$ , dvs så att

$$\mathbf{s}_d^H \mathbf{w}_{opt} = 1. \quad (2.30)$$

Genom att låta styrvektorn  $\mathbf{s}_d$  ha taperad magnitud kan vilodiagram, dvs. då endast termiskt vitt brus finns och  $\mathbf{R}$  blir en enhetsmatris, med t.ex. låga sidolober fås. Om vi härvid fortfarande vill tolka algoritmen som maximum SNR får vi definiera nyttig signal som havande just denna amplitudtapering.

Antagandet att den nyttiga signalen inte är närvarande då man bildar korrelationsmatrisen är oftast korrekt i radartillämpningar men kanske inte alltid annars. Om  $\mathbf{s}_d$  verkligen är en avbild av inkommande signalvektor, och antalet sampel  $K$  är stort, saknar dock detta betydelse, [13] [14], och algoritmen blir identisk med den enklaste formen av “Constrained Power Minimization” enligt 2.5.

Man kan visa att, se kap. 6 i [1] (generaliserat till  $J$  tappar), utan någon korrektion enligt 2.3, det behövs approximativt

$$K \geq 2N \quad (2.31)$$

sampel för att förväntat  $SINR$  ska vara inom 3 dB från dess asymptotiska, optimala, värde  $SINR_{opt}$  (approximativt  $PG_{r,max}^{(A)}$  enligt 2.1).

## 2.7 Side Lobe Canceller

Med “Side Lobe Canceller” menar vi här en algoritm som efterliknar ett antensystem med en direktiv antenn i önskad signalriktning samt innehåller ett antal adaptiva hjälpkanaler. Den direkta antennen skapas här genom att specificera lobriktning och icke-adaptiv viktvektorform och den adaptiva styrvektorn ges värdet ett för denna lob och värdet noll för hjälpanternerna. Det optimala resultatet blir här inte lika bra som för ett fullständigt adaptivt system med en frihetsgrad per antennelement men risken för problem av typen höga sidolober och nyttosignalundertryckning minskar. Kravet på antal sampel för att man ska närma sig asymptotiska prestanda minskar också då antalet frihetsgrader blir mindre. Denna algoritm liknar i viss mån både algoritmen i 2.6

och även en enklare variant av den adaptiva-adaptiva i 2.8 med skillnaden att loberna till nyttig signal här är bestämd i förväg och de adaptiva kanalerna består av enskilda element.

## 2.8 Adaptiv-adaptiv processning

Huvudsyftet med algoritmen "Adaptiv-adaptiv processning" är att undvika de problem som kan uppstå då antalet element (frihetsgrader) är mycket större än antalet störare som ska elimineras, [11]. Dessa problem består främst av att insamlandet av korrelationsmatrisen kräver många sampel (lång tid), att beräkningsbördan för att beräkna optimala viktcoefficienter blir stor vid många frihetsgrader och att kvaliteten på antenndiagrammen, då man tillämpar en rättfram matrisinvertering, kan bli dålig vid för få sampel.

Ett sätt att undvika dessa problem är att först identifiera nyttsignal och störare och deras riktningar, vilket inte kräver så många sampel, och att sedan generera en riktad antennlob i dessa (relativt få) riktningar samt att slutligen låta signalerna från dessa antennlobber vara indata till den adaptiva störundertryckningen. Viktvektorerna för loberna till alla identifierade störare och nyttsignaler bildar en transformationsmatris,  $\mathbf{w}_T$ , med en viktvektor per kolumn. Elementen i den adaptiva processens styrvektor (matris med en kolumn per flervägskomponent) sätter vi nu till noll utom för de element som motsvarar de lobber som pekar i de nyttiga signalernas riktning, dvs för ett element per kolumn, medan den tillhörande korrelationsmatrisen fås som:

$$\mathbf{R}_{Ad} = \mathbf{w}_T^T \mathbf{R} \mathbf{w}_T^* \quad (2.32)$$

Den resulterande totala viktmatrisen,  $\mathbf{w}$ , för alla antennelement och för varje separat flervägskomponent, fås genom multiplikationen av  $\mathbf{w}_T$  av med den resulterande adaptiva viktmatrisen  $\mathbf{w}_{Ad}$ , som fås ur (2.29), genom

$$\mathbf{w} = \mathbf{w}_T \mathbf{w}_{Ad} \quad (2.33)$$

Algoritmen är i grunden en smalbandsalgoritm men genom att dela upp signalerna och processningen i delband som sedan läggs ihop, se 2.11, kan även bredbandiga scenarier behandlas förutsatt att lobernas känslighet i nyttsignalens riktning är normerad på ett enhetligt sätt.

Som antytts ovan kommer denna algoritm bäst till sin rätt då antalet element är mycket stort, som i en radartillämpning.

## 2.9 Generering av 0-djup i skattade riktningar

Antag att vi, t.ex. utgående från uppmätt korrelationsmatris  $\mathbf{R}$ , har skattat riktningarna till  $ndir$  stycken planvågs signalkällor och att vi önskar generera ett antenndiagram som har genuina smalbandiga 0-djup i dessa riktningar. Det innebär t.ex. att vi nu inte längre försöker att maximera  $SINR$ . Det förutsätter också att vi antar att de skattade signalriktningarna är korrekta, att signalerna verkligen kommer in som plana vågor, att en smalbandsapproximation är tillräcklig samt att vår antennarray är väl känd eller, uttryckt på annat sätt, att våra signal- och antennmodeller är korrekta.

Vi kan då sätta upp en matris  $\mathbf{A}$  med dimension  $[nel \times ndir]$ , där  $ndir < nel$ , som utgående från vår antennmodell beskriver de  $nel$  antennelementsignalerna för de  $ndir$  källorna där varje matriselement beskriver signalen för tillhörande element och källa. Den totala utsignalen i varje kanal för ett sådant scenario blir då:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \quad (2.34)$$

där  $\mathbf{s}(t)$  är den tidsberoende signalvektorn av längd  $n_{dir}$  med element  $s_j(t)$ . Vi önskar nu bestämma en viktvektor  $\mathbf{w}_j$  för varje källa  $j$  så att  $\mathbf{w}_j^T \mathbf{x}(t) = s_j(t)$  eller, sammanfattat för alla källor, en viktmatris  $\mathbf{W}$  så att

$$\mathbf{W}^T \mathbf{x}(t) = \mathbf{W}^T \mathbf{A}\mathbf{s}(t) = \mathbf{s}(t) \quad (2.35)$$

vilket betyder att vi får en lösning som  $\mathbf{A}$ :s pseudoinvers, t.ex. [25]:

$$\mathbf{W}^T = \mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H. \quad (2.36)$$

För att få rätt lösning måste vi naturligtvis också identifiera nyttosignalen bland de  $n_{dir}$  funna signalkällorna så att vi kan plocka ut rätt kolumn  $\mathbf{w}_j$  ur matrisen  $\mathbf{W}$ .

Denna lösning är inte unik (antalet källor är mindre än antalet antennelement) utan vi kan till varje viktvektor  $\mathbf{w}_j$  addera alla viktvektorer som ger 0-djup i alla signalriktningar (uttryckt matematiskt kan vi lägga till vektorer som inte ligger i  $\mathbf{A}$ :s kolumn-rum, kallat det vänstra nollrummet). Lösningen enligt (2.36) har dock minst norm vilket innebär att vi får minsta bidrag från antennens egenbrus, vilket annars försummas här.

Denna algoritm är i grunden smalbandig men om vi delar upp signalerna i delband kan den appliceras på var och en av dessa varpå utsignalen fås genom att lägga samman delbanden, se 2.11.

## 2.10 Spatiell filtrering

Termen "Spatiell filtrering" används ofta i signalbehandlingsbranschen för att beteckna vanlig konventionell lobformning med en eller många samtidiga lober. Uttrycket kommer sig av att en antennlob filtrerar ut ett vinkelspektrum p.s.s. som ett vanligt filter filtrerar ut ett frekvensspektrum och om antennen är en uniform linjär array, ULA, (vilka naturligtvis inte förekommer i verkligheten men ibland är en god approximation) blir diagramsyntes identisk med vanlig filtersyntes. Detta är i sig ingen adaptiv procedur men en sådan, egentligen diversitets-, procedur fås om man på detta sätt bildar ett antal samtidiga lober och väljer ut den som ger bäst signal. Detta val görs f.n. inte i simuleringsprogrammet. *SINR* för varje dellob ges dock så att detta val lätt kan göras i efterhand, t.ex. med hjälp av skattat *SINR*-mått enligt avsnitt 5.4. Detta förutsätter naturligtvis att vi kan göra "rätt" val även i ett verkligt fall.

## 2.11 Uppdelning i delband

Några av de ovan beskrivna algoritmerna är egentligen avsedda för smalbandssignaler. Det gäller framför allt vid generering av 0-djup i 2.9 men även den adaptiva-adaptiva processningen i 2.8, side lobe canceller i 2.7 och maximalt signal-brus förhållande i 2.6. För dessa men även för de andra kan ett alternativ till en bredbandig processning, vilket kan kräva många sampel och många filtertappar och därmed stora matriser att jobba med, vara att dela upp signalen i flera delband och att processa dessa var för sig för att slutligen lägga ihop dem till en total signal. Sammanfogningen görs lättast i frekvensdomänen genom att lägga spektra bredvid varandra (förfarandet kräver alltså, som det är implementerat här, ett antal FFT-operationer). För alla algoritmer utom MMSE i 2.4 är detta relativt rättframt (mest en fråga om korrekt bokföring). Det enda man måste tänka på är att de olika delbanden skarvas ihop på ett korrekt och kontinuerligt sätt.

Detta sker dock automatiskt i de flesta fall genom att vi har normerat på ett enhetligt sätt enligt ekv. (2.30) och avsnitt 6.2 eller för MVDR genom att se till att bivillkoren, som nu måste finnas i varje delband, inte har någon inbyggd diskontinuitet.

För MMSE, där vi ju har en träningssekvens som täcker hela bandbredden synes det inte vara någon bra ide att dela upp processningen i delband eftersom vi då får mindre träningsdata i varje delband med sämre synkronisering som följd, vilken dessutom helst bör vara densamma i varje delband. Att göra uppdelningen i delband efter synkroniseringen skulle kunna vara en möjlighet som dock inte implementerats här.

## 2.12 Sammanläggning av flervägskomponenter

Om nyttosignalen utbreder sig längs flera gångvägar fram till vår mottagare kan vi öka *SINR* genom att lägga samman dessa flervägsutbredda signaler på ett lämpligt sätt. I alla de här behandlade algoritmerna, avsnitt 2.4 - 2.10, har vi möjlighet att alstra flera samtidiga antennlob mot de olika flervägskomponenterna. I alla utom MMSE, 2.4, måste vi dock ha åtminstone en approximativ kännedom om dessa komponenters infallsriktning.

På sina olika vägar fram till mottagarantennen har de olika komponenterna fått dels olika fördröjningar och dels olika dämpning. Dessutom bör man också tänka sig att de fått olika fasskift. Det senare kan tyckas vara onödigt med tanke på att en fördröjning innebär en med frekvensen linjärt varierande fasskift. Att tänka sig både fasskift och fördröjning betyder dock att vi kan inkludera ett frekvensoberoende fasskift i t.ex. antennen, att vi får en term till i en tänkt serieutveckling av fasgången (avvikande från rent linjärt) samt att vi, framför allt för smalbandigare men även för bredbandiga system, får en betydligt större noggrannhet då ett samplingsintervall motsvarar ett mycket stort fasskift.

Såväl fördröjning, amplitudrelationer som fasskift för de olika flervägskomponenterna fås genom att söka maximal korskorrelation mellan signalerna, precis som vid MMSE, dvs. med ekvation (2.21) där  $\mathbf{x}$  är utsignalerna från de olika delloberna och  $d(t)$  en referenssignal, se också 6.22. För MMSE-algoritmen väljer vi att som referenssignal  $d(t)$  använda samma träningssekvens som vid synkroniseringen (då translatterade vi träningssekvensen medan vi nu translatterar utsignalerna) medan vi för de övriga algoritmerna, där vi ju saknar träningssekvens, använder en av dellobsignalerna själv.

Då vi hittat signalernas relativa fördröjningar och komplexa amplitudrelationer återstår att lägga samman dem till en total signal. Först translatterar vi därvid alla signaler i tiden enligt de funna fördröjningarna.

För "vanliga" icke-adaptiva antenner där vi t.ex. har en rundstrålande antenn eller en huvudlob i signalriktningarna finns det i huvudsak två metoder att lägga samman dessa signalkomponenter, "equal ratio combining" (eller "equal gain combining") respektive "maximum ratio combining", se t.ex. [23], [24].

I den första korrigerar vi endast för signalernas relativa fas, dvs. viktar med

$$\alpha_n \propto \exp(-j\angle v_{korr,n}) \quad (2.37)$$

där  $v_{korr,n}$  är maximal korrelation enligt (2.21).

I den andra korrigerar vi även för signalernas olika magnitud och viktar med

$$\alpha_n \propto v_{korr,n}^* \quad (2.38)$$

Om alla delsignalerna har lika brus innebär detta att totalsignalen får maximalt  $SINR$ .

I en adaptiv antenn är det sista antagandet, om lika brus i alla utsignalerna, tveksamt. Om de adaptiva loberna har en snygg huvudlob i respektive nyttsignals riktning och för övrigt låga sidolober (dvs. hög aperturverkningsgrad) kan antagandet vara bra men om vi har en stark störare nära nyttsignalen kan antennloben få en stark lutning i nyttsignalens riktning och antennvinsten blir låg. Observera att känsligheten för antennen, som den definieras för adaptiva antenner, ändå kan vara hög. Om vi antar att vi har lika, okorrelerat, och vitt brus i alla antennkanalerna kommer det inre bruset att bli proportionellt mot de adaptiva vikternas kvadratsumma och vi finner att maximalt  $SINR$  fås då de olika flervägskomponenterna viktas med

$$\alpha_n \propto \frac{v_{korr,n}^*}{\|\mathbf{w}_n\|^2} \quad (2.39)$$

vilket vi kan kalla “maximum ratio combining typ 2”.

Om det yttre bruset är i form av en “brusmatta”, med icke-lokaliserat brus som inte kan undertryckas i den adaptiva antennen, är det svårare att generellt säga hur flervägskomponenternas viktning,  $\alpha_n$ , optimalt ska vara uttryckt i de adaptiva vikterna. Om detta yttre brus är isotropt fördelat och antennelementen inte hade någon ömsesidig resistans (vilket de dock har, jämför ekv. (2.4)), så att total utstrålad effekt vid ett tänkt sändningsfall är lika med vikternas kvadratsumma, gäller dock (2.39) fortfarande.



### 3. Simuleringsflöde

Flödet för simuleringsberäkningarna visas mycket schematiskt i nedanstående figur.

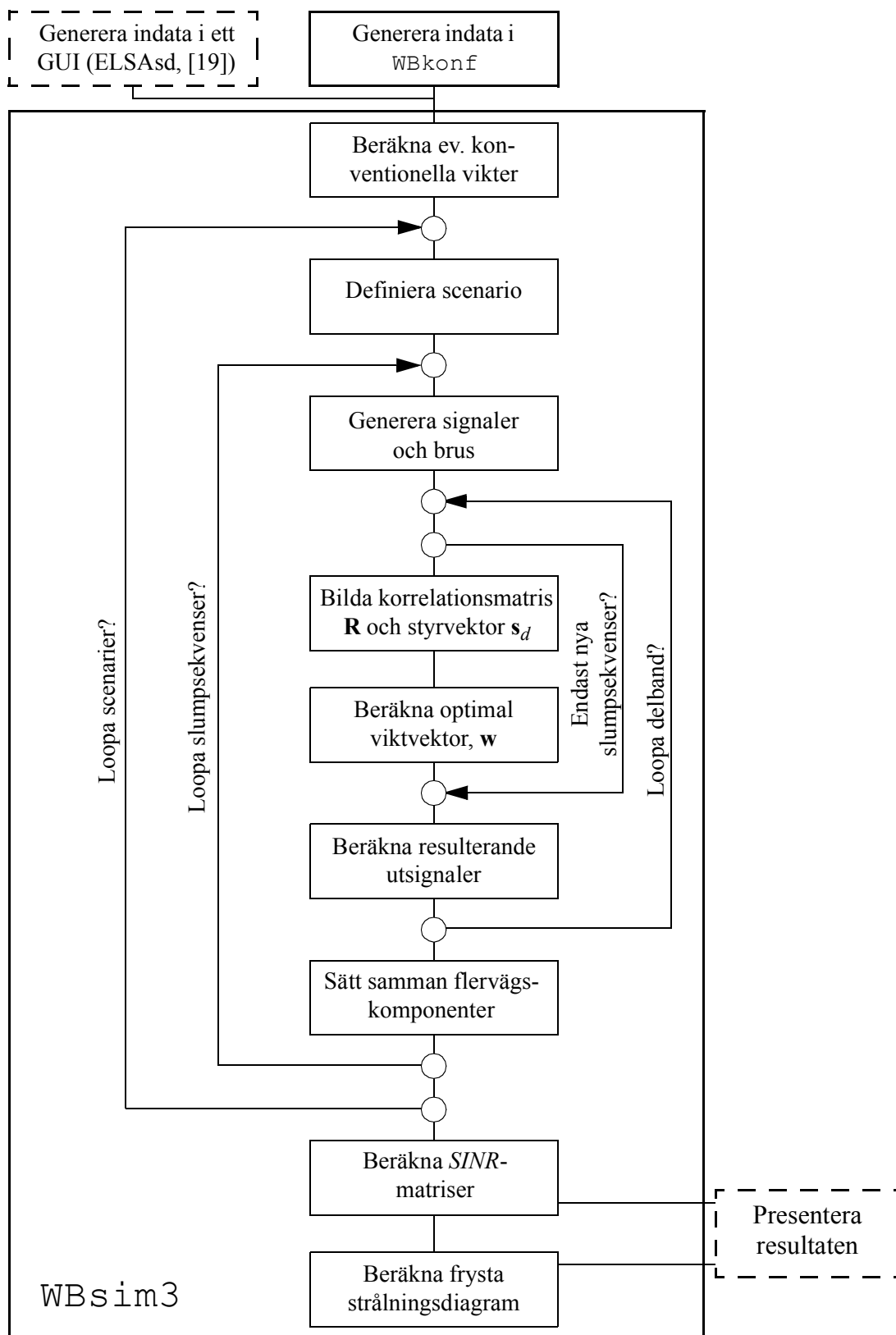


Fig. 3.1. Förenklat flödesdiagram för simuleringarna.

I denna rapport beskrivs delarna `WBkonf` och `WBSim3` medan de block som är inramade med streckade linjer, GUI och plottning, inte ingår. `WBkonf` beskrivs i kapitel 4, `WBSim3` beskrivs i kapitel 5 medan de olika ingående funktionerna beskrivs i kapitel 6.

Om man skulle vilja variera en annan parameter än vad som visas i Fig. 3.1. eller vad som beskrivs i kapitel 5 kan detta göras relativt lätt genom att man kör `WBSim3` via ett egenhändigt skrivet program, `WLoopX`. Parametrarna genererade av `WBkonf` ändras inte av `WBSim3` så `WLoopX` behöver i princip bara bestå av en slinga i vilket man uppdaterar den varierade parametern (och eventuella beroende storheter) och sparar mellanresultat i en extra utdatamatrix med en dimension mer än motsvarande `WBSim3`-matrix.

## 4. Konfigureringsprogram: WBkonf

WBkonf är konfigureringsdelen till simuleringsverktyget, i vilket alla indata ges och viss datakontroll utförs (dock långt ifrån någon fullständig kontroll). Det är rent textbaserat och genererade indata är grupperade i vissa huvudblock vilka beskrivs i det följande. Användandet av WBkonf förutsätter att användaren är relativt väl förtrogen med vissa interna variabelnamn o.dyl. För att kunna följa den nedanstående texten är källkoden till WBkonf till stor hjälp.

Ett grafiskt användargränssnitt (GUI) kallat ELSAsd i vilket de viktigaste och oftast ändrade variablerna kan anges finns dock utvecklat inom projektet och beskrivs översiktligt i [19] men behandlas inte vidare i denna rapport.

En lista över alla indatavariabler som kan eller ska ges av en användare redovisas i Appendix A. De olika beskrivningarna över alla funktioner i kapitel 6 innehåller också viss, mer detaljerad, parameterbeskrivning.

### Initiering:

Före tilldelningen av indata sätts ett antal default-variabler och världskonstanter.

Därefter anges några initieringsvariabler för slumpalsgenereringen (`randstate`, `randnstate`) liksom några parametrar för metodval och interpolering (`vrandmet`, `maxfindmet`, `Omitsyntlog`, `kintp`). Dessa behöver normalt inte ändras utom vid vissa tester.

### Allmänna parametrar för processningen:

Sedan anges några allmänna parametrar för angivande av signalbehandling (`Adapt`). Vidare hur många gånger (`Nrun`) vi ska simulera med samma adapteringsvikter, men olika slumpalsgenereringar, för att härigenom få ett större statistiskt utfall. Här anges också antal sampel i signalerna och för generering av korrelationsmatris och träningssekvens (`ksamp`, `ksampcov`, `ktrain`)

### Geometri och antennelement:

Antennens geometri, FIR-filtrens längd och antennelementen definieras sedan (`geom`, `nel`, `Jtap`, `elpatt`, `elpattsynt`, `nerrheff`, `dhdBF`, `dphdegF`).

Om antennen är en uniform cirkulär array (UCA) anges dess radie till elementens fasreferenspunkter (`Rel`). För dessa arrayer kan vi arbeta med signalernas fasskiftsmoder istället för själva elementsignalerna. I så fall anges detta och eventuellt tillhörande parametrar (`Butlerlog`, `invfiltlog`, `elevinvfilt`).

För övriga geometrier anges geometri och pekriktning (`del`, `nelrekt`, `phielrekt`).

### Signaler och brus:

Därefter definieras brus (`Brussamplog`, `Ortolog`) och signalernas bandbredd, styrka, vågform, infallsriktning samt vilka signaler som är nyttsignaler och fördröjningarna för dessa (`fmin`, `fmax`, `phisig`, `elevsig`, `psigdB`, `truesig`, `truedelay`, `sigmode`, `Fsinusreal`, `Flowreal`, `Fhighreal`). Dessutom anges om eventuell extern vågutbredningsmodell ska användas (`GeomPropMod`). I så fall kommer senare nya signaler, med eventuell flervägsutbredning, att genereras (via funktionen `Propmodel`), varvid dock källornas styrka och vågform är de som anges här.

Om simuleringarna ska utföras för varierande scenario beskrivs detta genom att låta varje delscenario vara en ny rad i variablerna `phisig`, `elevsig`, och `psigdB`.

Eventuell korskorrelation mellan signalerna kan också anges ( $S_{cov}$ ) vilket dock fungerar bra endast för brus-signaler, i annat fall kommer signalformerna att distorderas.

### Mottagare:

Mottagarförstärkarnas, för varje antennelement, inbördes olikheter definieras sedan i form av variationer i förstärkning, fasvridning och brusfaktor samt inkluderande eventuella trasiga förstärkare ( $nerrterm$ ,  $dgdB0$ ,  $dphdeg0$ ,  $dgdBF$ ,  $dphdegF$ ,  $indS210$ ,  $noisevar$ ,  $noisesp$ ,  $foreft$ ).

### Korrektionsmetod med parametrar:

De adaptiva algoritmerna behandlade i denna rapport har vissa gemensamma korrektionsmetoder, enligt avsnitt 2.3, vilka innebär modifieringar av själva korrelationsmatrisen för rådata innan man applicerar den egentliga adaptiva algoritmen. Dessa parametrar ( $Tprelog$ ,  $sdproj$ ,  $sdprojmet$ ,  $Lprojpar$ ,  $evm$ ,  $Tmin$ ,  $Tegv$ ,  $wDiagtype$ ,  $Tartrel$ ,  $r0dBa$ ,  $nbara$ ) anges nu före valet av algoritmen.

För att i viss mån minska känsligheten för störundertryckningen kan man i vissa fall bredda nollstupen enligt metoderna i avsnitten 6.29 och 6.30. Signalen i MVDR kan dessutom skyddas genom att, om önskat, lägga till extra bivillkor kring dessa signaler. Tillhörande parametrar ( $Raugmentlog$ ,  $nnls$ ,  $Wpar$ ,  $Constraugmentlog$ ) ges nu.

Om man vill generera icke-adaptiva antenndiagram för ren konventionell lobformning, för en synkroniseringssignal i MMSE eller som ett vilodiagram i den adaptiva processningen, behövs också ett antal parametrar. Dessa innefattar både lobform, lobriktning och syntesfrekvens ( $wtype$ ,  $wButel$ ,  $Antdiagtyp$ ,  $elpattkor$ ,  $r0dB$ ,  $nbar$ ,  $Scanvinkel$ ,  $Elevsynt$ ,  $Fsyntrel$ ).

Dessutom görs ett val av aktiva kanaler, eller fasskiftsmoder, eftersom inte alla element/fasskiftsmoder behöver användas ( $aktelem$ ,  $nelstart$ ,  $nelw0$ ,  $nmod0$ ).

Ett liknande val av aktiva element, eller fasskiftsmoder, och även aktiva filtertappar görs också för själva den adaptiva processningen ( $nelstartA$ ,  $nelA$ ,  $nmod0adapt$ ,  $Indtaps$ ).

### Algoritmval:

Därefter anges valet av adaptiv algoritmen ( $sdlog$ ).

För att utröna algoritmernas känslighet för nyttsignalens närvaro (som kan vara stor) har vi också möjlighet att välja att plocka bort denna signal från korrelationsmatrisen ( $signinRmat$ ). Detta simulerar också ett verkligt fall där man adapterar i ett tidsintervall där nyttsignalen är tyst.

### Flervägsutbredning:

Om flervägsutbredning förekommer väljer vi parametrar för att hitta dessa och hur de skall kombineras ihop ( $Nsignin$ ,  $combmet$ ,  $comb$ ,  $Vsigrel$ ).

### Parametrar för de olika adaptiva algoritmerna:

Efter detta kan vi göra en uppdelning av indatagenerering beroende på algoritmen.

För Maximum SNR enligt avsnitt 2.6 (här kallat *styrv*) och för Spatiell filtrering enligt avsnitt 2.10 (då *Adapt* är lika med *spfilt*) anges hur antenndiagrammen ska syntetiseras då vi har kanalolikheter ( $sdmet$ ) som *Vilodiagram*, vilken i viss mån korrigerar för kanalolikheter, eller *Planvåg*, vilken avbildar signalerna från en plan våg, enligt 6.44 och 2.3.

För Minimum mean square error (MMSE) enligt avsnitt 2.4 (här kallad *korr*) anger vi parametrar för hur vi skall hitta träningssekvensen i insignalerna och vilka sampel som ska användas i denna process (sdRlog, Jtapkorr, Beamkorrlog, sdet, korrkanal, korrfold, Rmatktrainsamp).

För generering av 0-djup i skattade riktningar enligt avsnitt 2.9 (här kallad *NBnull*) anger vi var vi tror att våra nyttsignaler finns, så att vi bättre kan välja ut rätt signaler (phisigNBnull, elevsigNBnull).

För Adaptiv-adaptiv processning enligt avsnitt 2.8 (här kallad *AdAd*) anges typ av korrektion för eventuella kanalolikheter, enligt ovan och 2.3, samt nyttsignalernas riktning om dessa plockas bort från korrelationsmatrisen enligt ovan, vilka annars kommer att skattas i WBSim3 (sdmet, phisigAdAd, elevsigAdAd).

För Side Lobe Canceller enligt avsnitt 2.7 (här kallad *SLC*) anges också typ av korrektion för eventuella kanalolikheter, enligt ovan och 2.3, samt nyttsignalernas riktning och vilka antennelement som skall ingå i den adaptiva processningen (sdmet, adaptel, Scanvinkel, Elevsynt).

För Constrained Power Minimization, eller MVDR, enligt avsnitt 2.5 (här kallad *riktn*) är antalet parametrar större än för övriga algoritmer. Här anges riktningarna och frekvenserna för alla potentiella bivillkor (phiconstr, elevconstr, FTcreal). Dessa kommer att bilda en matris *C* men vi har också möjlighet att bara låta vissa av dessa vara verkliga, aktiva, bivillkor (indconstr). Dessutom anges bivillkorens värden (fconstr). Här kan vi också, som vid signalgenereringen ovan, variera scenariet genom att ge olika bivillkors-scenarier i olika rader av phiconstr och elevconstr. Signal- och bivillkors-variationerna kan antingen vara koplade med lika många rader (scenarier) eller också varierar endast den ena.

Vi har också möjlighet att välja att, i WBSim3, modifiera de till nyttsignalerna hörande bivillkorsriktningarna till motsvarande skattade riktningar (phisigmodlog). Denna möjlighet minskar risken att nyttsignalerna, p.g.a. fel hos angivna riktningar, tolkas som störare och därför undertrycks. Detta kan vara mycket viktigt om vi inte är helt säkra på nyttsignalernas riktningar. Om flervägsutbredning förekommer (fler än en nyttsignal) och vi inte önskar skatta signalriktningar och modifiera bivillkorsriktningarna enligt ovan måste vi också tala om vilka bivillkor som hör till nyttsignaler (Imod) så att endast en av dessa används för varje flervägskomponent.

### **Uppdelning i delband:**

Istället för att processa hela bandbredden i ett steg har vi möjlighet att dela in signalerna, och processningen, i ett antal delband, som sedan slås ihop till en total signal efteråt enligt avsnitt 2.11. Detta val görs nu varvid vi fritt kan välja antalet filtertappar i varje sådant delband (Nsubband, Jtaptot). En sådan uppdelning kan dock inte väljas för MMSE-algoritmen ty då skulle vi behöva en träningssekvens för varje delband och att dela upp den befintliga förefaller inte som en bra lösning.

### **Parametrar för riktningsskattning:**

I flera av algoritmerna ska man skatta riktningarna till signalerna (med doadir och doa) och för detta krävs några speciella indata nämligen vilken av ett antal algoritmer som ska användas samt några speciella parametrar för denna, enligt avsnitt 6.12. Där ingår bl.a. hur många signaler vi maximalt önskar hitta och hur starka, i någon mening, dessa måste vara för att klassas som signaler samt önskade testriktningar i sökningsproceduren (sigbeh, nsource, Tdoamax, Mxekolumn, spsmooth, FBaver, phidoatest, elevdoatest, Pdoarel).

**Frysta strålningsdiagram:**

Slutligen ges ett antal data för att kunna generera frysta antenndiagram, dvs. diagram med de beräknade adaptiva vikterna fixerade. Dessa är önskade riktningar och frekvenser samt hur diagrammen ska beräknas (Radpattlog, Deltaf, Ftestreal, phitest, elevtest, t<sub>f</sub>BF).

## 5. Huvudprogram: WBSim3

WBSim3 är simuleringsverktygets huvudprogram. Indata är alla de parametrar som genererats i WBkonf enligt kapitel 4 (eller något annat användargränssnitt), vilka beskriver scenarier, signaler, algoritmval m.m., medan resultatet består av resulterande *SINR*, viktvektorer, mm. I avsnitt 5.1 nedan beskrivs kortfattat generering av scenario och signaler, de tre översta blocken i WBSim3-rutan enligt Fig. 3.1. Beräkningen av de adaptiva vikterna, specifika för de olika algoritmerna, motsvarande block 4 och 5 i Fig. 3.1, beskrivs i avsnitt 5.2. Generering av resultat utgående från beräknade adaptiva vikter, block 6 och 7 i Fig. 3.1, beskrivs i avsnitt 5.3. De resulterande utdata, block 8 och 9 i Fig. 3.1, beskrivs slutligen i avsnitt 5.4.

### 5.1 Generering av datasampel

WBSim3 börjar med att definiera några grundläggande variabler för geometri (via funktionen *geometry*) och frekvenspunkter (variabeln *k*). Eftersom vi gör en bredbandig simulering och transformerar signalerna mellan tids- och frekvens- domän blir antalet frekvenspunkter lika med antalet tidssampel (snapshot).

Om någon form av icke adaptivt antenndiagram krävs beräknas sådana viktvektorer sedan med *weights*. Dessutom definieras antenkanalernas förstärkning,  $S_{21}$  i Fig. 2.1, med sina kanalolikheter, fel och rippel.

Om endast icke adaptiva antenndiagram ska beräknas går vi direkt till slutet och beräknar strålningsdiagram med *radpattern* och plottar dessa (plottning är dock ej inkluderat i WBSim3), annars genomförs det följande en gång för varje specificerat scenario. Detta börjar med att definiera aktuella infallsriktningar, och eventuella bivillkorsriktningar, varvid en extern vågutbredningsmodell, via *Propmodel*, kan komma till användning.

Om flera scenarier med varierande signal- och/eller bivillkors- riktningar ska simuleras, genomlöps det följande i ett antal scenariesteg.

Signalernas verkliga form, i tids och frekvensdomän, genereras så via *signals* och dämpning och fördröjning (fram till antensystemets origo) för signaler som genererats med *Propmodel* läggs på via *Propsig*.

Antennelementens överföringsfunktion enligt 2.1, inkluderande eventuellt rippel, beräknas så för alla signalriktningar och frekvenser och appliceras, inkluderande tidsskift beroende på elementposition, tillsammans med kanalförstärkarna så att vi får signaler för alla elementkanaler och frekvenser samt, via en FFT, för alla tidssampel. Egenbrus i mottagarna läggs här också på, före eller efter förstärkarvariationerna, såvida vi inte valt att använda en separat asymptotisk bruskorrelationsmatris för vissa teständamål. Om vi har en cirkulär array och har valt att arbeta med dess fasskiftsmoder, t.ex. för att kunna utnyttja moddiagrammens inneboende bandbredd, transformerar vi nu till dessa. Därefter extraherar vi signalerna i de verkliga aktiva kanalerna om inte alla ska användas.

Efter detta har vi nu de totala signalerna i varje adaptiv kanal som vi har att arbeta med. Härfter återstår själva den adaptiva processningen för att finna de optimala viktcoeffienterna.

För flera av de alternativa algoritmerna har vi möjlighet att dela upp signalerna i flera delband enligt 2.11 för att processa dem var för sig och lägga ihop dem efteråt, varvid vi naturligtvis måste vara noga med att skarva ihop dem med rätt amplitud och fas. Denna eventuella uppdelning görs nu.

Den adaptiva processningen börjar därefter. Denna processning behöver inte använda alla signalsampel men resultaten såsom  $SINR$  m.m. enligt 5.4 baseras dock på alla sampel, även de utanför själva adapteringsintervallet. För att få goda och representativa data behövs dock ofta en relativt lång tidssekvens. Då detta lätt kommer att ta mycket lång tid har vi också möjligheten att (med  $N_{run} > 1$ ) löpa igenom en stor del av simuleringen, bl.a. hela signalmodelleringen, flera varv men endast beräkna adaptiva vikter det första varvet vilka sedan används för att beräkna resulterande  $SINR$  mm. även vid alla följande varv. Med en sådan scenarieupprepning utan upprepade adaptering kan man, med relativt lite arbete, få ett mycket större statistiskt utfall och därigenom större noggrannhet i medelvärdesbildade resultat. Om sådan upprepning pågår hoppar vi alltså över nedanstående del och går direkt till resultatberäkningen enligt 5.3.

## 5.2 Beräkning av adaptiva vikter

Denna procedur börjar med att beräkna en korrelationsmatris  $\mathbf{R}$  enligt (2.18) vilken alla algoritmer utom `spfilt` i 2.10 använder. Om en asymptotisk bruskorrelationsmatris för vitt brus (AWGN), som är okorrelerat mellan såväl tidssampelen som mellan kanalerna, ska användas beräknas den här och adderas till  $\mathbf{R}$ .

Eftersom vissa algoritmer, vid vissa scenarier, fungerar väsentligt bättre om man kan generera sin korrelationsmatris då den egna nyttsignalen saknas kan en sådan korrelationsmatris alternativt genereras vid behov.

Ytterligare en modifiering av  $\mathbf{R}$  som införs redan här är en "augmentering" med `raugmentlin` eller `raugmentbut`, avsnitt 2.3, 6.29 och 6.30, varvid en breddning av resulterande noll-djup ska fås, åtminstone för de tillämpliga (ideala) antennmodellerna.

### 5.2.1 Minimum Mean Square Error

I denna algoritm, se 2.4, behöver vi inte anta någonting om nyttsignalens infallsriktning, eller ens att den kommer in som en plan våg, även om ett sådant antagande ändå används i vissa fall (se nedan). I stället för riktningen används att vi har viss kunskap om signalens form. Här kommer vi att anta att en viss del av signalen består av en träningssekvens, med längden  $k_{train}$ , som vi känner eller åtminstone har en god uppfattning om.

En viktig del av sökandet efter optimal lösning består av att hitta träningssekvensen i signalen, dvs. de fördröjningar (en för varje flervägskomponent) av träningssekvensen som ger bäst överensstämmelse mellan denna och signalen. Detta görs i `korrdelay` på ett av tre olika sätt och tidsupplösningen förbättras genom interpolation av de (bandbegränsade) signalerna till att vara en faktor  $k_{intp}$  tätare än samplingsintervallet. Två sätt består av att söka de fördröjningar som ger högst korrelation antingen med antennementsignalerna direkt, varvid det bästa av några utvalda element väljes, eller genom att först forma en antennlob i nyttsignalernas antagna riktning. Ett annat sätt, som oftast ger bättre resultat men också är mer beräkningskrävande, är att söka den fördröjning som ger det minsta skillnaden mellan träningssekvensen och utsignalen efter den adaptiva lobformningen, dvs (2.24), vilken ju algoritmen ska minimera. Se mer i avsnitt 6.22.

Styrmatrisen  $\mathbf{s}_d$ , med en kolumn per flervägskomponent, ges så av korrelationen mellan de, enligt ovan, optimalt fördröjda träningssekvenserna och signalen i varje kanal och filtertapp. Korrelationsmatrisen  $\mathbf{R}$  beräknas antingen som vanligt eller alternativt genom att bara använda de  $k_{train}$  sampel som passade bäst överens med träningssekvensen (olika sampel för de olika fler-



vägskomponenterna). Detta senare alternativ ger ofta stabilare resultat eftersom vi då använder i princip samma data vid genereringen av styrmatrix och korrelationsmatrix.

Optimala vikter fås därefter som vanligt från `adaptfun`.

Dessutom skattas riktningarna till de funna signalerna med `doadir`, vilka kan vara av visst intresse men dock inte krävs av algoritmen.

### 5.2.2 Constrained Power Minimization

I denna algoritm, se 2.5, minimerar vi den totala uteffekten, givet ett antal bivillkor. Det enklaste, och vanligaste, är att som bivillkor fixera antennens känslighet i en given nyttsignalriktning. Prestanda för denna algoritm kan då försämrats avsevärt om vi har angivit (gissat) denna riktning felaktigt, eller om den inte ser ut som en plan väg, särskilt om nyttsignalen är stark. För att om möjligt förbättra detta börjar vi här, om så önskas, med att skatta riktningen till dessa nyttsignaler med `doadir` så att motsvarande bivillkorsriktningar kan modifieras. Vid själva riktningssmodifieringen, som görs i `constrdirmod`, är det naturligtvis viktigt att vi parar ihop de skattade signalriktningarna med rätt nyttsignal och rätt bivillkor, se vidare 6.10.

Utgående från riktning, frekvens och antennelementens antagna känslighet för bivillkorsriktningarna och frekvenserna beräknas bivillkorsmatrisen  $\mathbf{C}$ , [2], i `constrmat3`. De optimala viktvektorena beräknas slutligen enligt ekv. (2.26) m.h.a. `adaptstap`, utom då vi bara har ett bivillkor ( $\mathbf{C}$  är en vektor) då vi (något godtyckligt) använder `adaptfun` som för alla andra algoritmer.

Vid flervägsutbredning, med separat adaptering i `adaptstap` till var och en av dessa signalvägar, ser vi också till att i tur och ordning inaktivera bivillkoren till övriga flervägsriktningar, vilka då naturligtvis måste vara korrekt identifierade.

### 5.2.3 Maximum SNR

I denna algoritm, se 2.6, förutsätter man egentligen att vår nyttsignal inte finns med i data för korrelationsmatrisen, eller att den är mycket svag. I implementeringen gjord här bildas först en nominell icke adaptiv (ev. taperad) viktvektor med `weights`, 6.44, svarande mot en lob i nyttsignalens antagna riktning och en styrvektor bildas med denna viktvektor i vektorelementen för den mittersta av de  $J_{\text{tap}}$  filtertapparna, och noll för övriga tappar. Frekvensen för denna väg antas vara  $F_{\text{syntreal}}$  såvida vi inte har en cirkulär array och vi använder fasskiftsmoder med korrektionsfilter då loben blir bredbandig. Vid flervägsutbredning bildas en matris med en sådan kolumn för varje flervägskomponent. Adaptiva viktvektorer, med element för alla antennelement och filtertappar, fås sedan via `adaptfun`. Denna algoritm fungerar, som påpekats tidigare, bäst då nyttsignalen saknas vid bildandet av korrelationsmatrisen  $\mathbf{R}$ , och passar därför bra i t.ex. radartillämpningar.

### 5.2.4 Adaptiv-adaptiv processning

Här utförs den adaptiva lobformningen genom att arbeta med ett antal hjälplobber i signalriktningarna, enligt 2.8. Först skattas då riktningarna till ett önskat antal källor med `doadir`, där vi använder (del-) bandets mittfrekvens i den smalbandiga algoritmen, varefter nyttsignalerna identifieras bland dessa med `findsig` och vanliga oadaptiva viktvektorer för lobber i de skattade riktningarna genereras med `weights`, med förutbestämda parametrar enligt 6.44 och 6.43. Dessa

viktvektorer bildar tillsammans en viktmatris med vars hjälp vi beräknar en ny korrelationsmatris svarande mot dessa lobsignaler enligt (2.32) och styrmatrix enligt 2.8 med hjälp av de identifierade nyttsignalriktningarna enligt ovan. Viktvektorer för lobsignalerna beräknas sedan på normalt sätt med `adaptfun` varefter elementvikter beräknas med (2.33). Detta är i grunden en smalbandig algoritm och inget FIR-filter tillåts för närvarande ( $\mathcal{J}_{\text{tap}}=1$ ) men algoritmen kan tillämpas på bredbandssignaler om en uppdelning i delband görs.

### 5.2.5 Side Lobe Canceller

Här bildar vi en konventionell lob för varje angiven nyttsignalriktning och använder ett antal adaptiva elementkanaler (hjälpkanaler) vid adapteringen, se 2.7. Vi bildar alltså från början oadaptiva viktvektorer för lobar  $i$ , från start angivna, riktningar med `weights` enligt förutbestämda parametrar enligt 6.44 och 6.43. Detta är i grunden en smalbandig algoritm och inget FIR-filter tillåts för närvarande ( $\mathcal{J}_{\text{tap}}=1$ ). Därefter fortsätter vi som i avsnitt 5.2.4

### 5.2.6 Generering av 0-djup i skattade riktningar

Givet att våra signal- och antenmodeller stämmer bildas här genuina 0-djup i störarnas riktningar, se 2.9. Först skattas härvid riktningarna till ett önskat antal källor med `doadir`, där vi använder (del-) bandets mittfrekvens i den smalbandiga algoritmen. Ett antal viktvektorer, en för varje nyttsignalriktning (flervägskomponent), beräknas sedan i `adaptnull` varvid nyttsignalerna identifieras (i `findsig`) som de skattade riktningarna som ligger närmast i förväg angivna signalriktningar, se 6.3 och 6.16. Detta är i grunden en smalbandig algoritm och inget FIR-filter tillåts ( $\mathcal{J}_{\text{tap}}=1$ ) men algoritmen kan tillämpas på bredbandssignaler om en uppdelning i delband har gjorts.

## 5.3 Generering av utdata

Oberoende av algoritm har vi nu de förhoppningsvis optimala viktvektorerna, dvs. en koefficient för varje antennelement, filtertapp och flervägskomponent. Utsignalerna för varje flervägskomponent fås nu från insignalen via funktionen `Hmatfilt` eller, om vi inte har några kanalfilter, direkt via en matrismultiplikation. Eftersom vi i dessa simuleringar, till skillnad mot i verkligheten, har tillgång till nyttsignaler, störsignaler och brussignaler separat beräknar vi även motsvarande ut signaler separat för kunna utvärdera simuleringen bättre, se även 5.4. Om en separat asymptotisk bruskorrelationsmatris använts i simuleringen beräknas asymptotiskt brus på utgången.

Om vi inte har delat upp processningen i delband kombinerar vi nu ihop eventuella flervägskomponenter med `combine` där hänsyn tas till dessa komponenters olika fördröjning respektive amplitud och fas.

Om vi däremot har delat upp processningen i delband kan vi välja att kombinera ihop flervägskomponenterna här eller alternativt efter hopslagningen till en total utsignal för varje flervägskomponent. Resultatet borde bli ungefär likvärdigt men om det görs efter hopslagningen är antalet datasampel större och möjligtvis synkroniseringen bättre, men också mer beräkningskrävande.

Sammanlagningen av de olika delbanden görs genom att bilda en lång total frekvensvektor som sedan transformeras till tidsplanet. Det är här naturligtvis viktigt att de olika delbanden normalerats på samma sätt så att man inte får någon stor diskontinuitet i amplitud eller fas i bandgrän-

serna. Detta sker direkt i `adaptstap` eller `adaptnorm` genom att vi väljer bivillkor och styrvektorer på ett "kontinuerligt" sätt (vi ska naturligtvis t.ex. inte sätta bivillkoren för viss riktning som någon diskontinuerlig frekvensfunktion) som diskuterats i 2.11.

Som ett alternativ till den nyttiga utsignalen, beräknad enligt ovan, beräknas även en korrelerad utsignal dvs. vi beräknar den del av den totala utsignalen som är korrelerad med vår, i simuleringen kända, nyttsignal varvid resten av utsignalen blir ortogonal mot denna nyttsignal. Om vi har perfekt okorrelerade signaler, störare och brus ska denna bli identisk med den signal som fås då vi följer nyttsignalen separat genom systemet men inte annars. Om vi har en skenbar korrelation, p.g.a. ändligt antal sampel, kommer det att påverka våra prestandamått.

För att i viss mån utröna vad distorsionen i antennelement och förstärkare, p.g.a. deras frekvensberoende, har för inverkan beräknas även korrelationen mellan insignalerna till processningen och sann nyttsignal.

## 5.4 Utdata

Eftersom `WBSim3` är en "matlab script-fil" är alla parametrar använda i där tillgängliga efter simuleringen. Några parametrar är dock primärt avsedda just som resultat. Dessa levereras som en uppsättning (flerdimensionella) matriser med data för varje genomlöst scenario (om flera) och scenarieupprepning utan upprepad adaptering (om flera) samt dessutom vid varje utstyrningsvinkel (om tillämpligt).

En parameterkategori är ett antal mått på "Signal-to-Interference-plus-Noise-Ratio", *SINR*. Om störarna, brus och signal är statistiskt okorrelerade och vi har oändligt många sampel är definitionen av detta mått entydigt men om något av dessa villkor inte är uppfyllt finns ett antal alternativa definitioner på vad vi klassar som nyttsignal respektive brus och störning, vilka kan ge olika resultat. De som beräknas i `WBSim3` är:

`SINRkor`: Här räknas signaleffekten ut genom att, med `korrdelay`, beräkna den till verklig nyttsignal (vilken vi känner här men naturligtvis inte i verkligheten) korrelerade delen av den totala utsignalen inkluderande lämplig fördröjning. Resteffekten beräknas sedan som skillnaden mellan totaleffekt och signaleffekt (vilken i detta fall blir lika med kvadraten på skillnadssignalen).

`SINRtrue`: Här räknas signaleffekten ut genom att följa nyttsignalen separat genom hela signalkedjan (vilket är möjligt i simuleringarna men naturligtvis inte i verkligheten). Resteffekten beräknas sedan som skillnaden mellan totaleffekt och signaleffekt.

`SINRtrued`: Signaleffekten beräknas som för `SINRtrue` medan resteffekten fås som kvadraten på skillnaden mellan total signal och nyttsignal.

`SINRtruejn`: Signaleffekten beräknas som för `SINRtrue` medan resteffekten fås som summan av kvadraterna på de (liksom nyttsignalen) separat beräknade stör- och brus- signalerna.

Eftersom en skenbar korrelation mellan signalerna kan ge både positivt och negativt bidrag kan storleksrelationerna mellan dessa mått variera. Speciellt för stora *SINR* kan `SINRtrue` ge dåliga, t.o.m. ofysikaliska, resultat då signaleffekten blir större än totaleffekten och nämnaren därmed negativ. Orsaken är antagligen att en kraftig undertryckning av störningen genom korrelationen också medför en undertryckning av nyttsignalen medan en ensam nyttsignal passerar mer opåverkad. Som ett enkelt exempel kan vi ta MVDR för smalbandiga signaler och utan FIR-filter, med ett bivillkor i nyttsignalriktningen, en störare och mycket lite (försumbart) egenbrus. Den "separata" signaleffekten är helt given av bivillkoret medan den totala effekten, som ska minimeras i den adaptiva processningen ges av signaleffekt, störeffekt plus korrelationseffekt. Störeffek-

ten kan då elimineras helt medan korrelationseffekten, genom korrekt hopfasning, kan göras negativ vilket alltså skulle ge ett negativt  $SINR_{true}$ .

Besläktad med  $SINR$ -måttan är bitfelssannolikheten (BER) vid BPSK- eller QPSK-modulation. Detta mått testas direkt efter antennutgången genom att jämföra utdata med den här kända nyttsignalen och inkluderar alltså ingen övrig demodulation eller kodning. Om BER blir mycket liten (god mottagning) får måttet lätt dålig tillförlitlighet. Denna kan då bättras på genom att köra många upprepade simuleringar ( $N_{run} > 1$ ) och beräkna medelvärdet.

Om endast ett scenario simulerats, och vi alltså bara har en uppsättning vikter, kan de frysta antenndiagrammen beräknas med `radpattern` för att senare kunna plottas. Med frysta diagram menas de normala antenndiagram, över hela bandbredden, som fås med de resulterande viktcoefficienterna. Även diagram i tidsdomän, dvs. pulssvarets vinkelberoende, levereras.

## 6. Använda funktioner

I detta kapitel beskrivs, i bokstavsordning, de Matlab-funktioner som används vid simuleringarna.

### 6.1 adaptfun

```
[wadapt, Rmatml]=adaptfun(Rmat, sd, adaptnorm, Jtap, indtaps, wDshift)
```

Anropas från: WBsim3

Anropar: adaptnorm, weight0

Här beräknas den adaptiva viktvektorn för de flesta av algoritmerna utom för MVDR, avsnitt 2.5, med fler än ett bivillkor då adaptstap enligt 6.4 används. De viktigaste indata är korrelationsmatrisen  $R_{mat}$  av storlek  $[nel \cdot J_{tap}, nel \cdot J_{tap}]$  och en styrvektor  $sd$  med  $nel \cdot J_{tap}$  rader. Förutom parametrarna i anropet ovan erhålls vissa speciella metodparametrar för den adaptiva processningen via mat-filen adaptpar.mat.

Ibland ska endast vissa av de  $J_{tap}$  filtertapparna, enligt indexvektorn  $indtaps$ , vara aktiva (dvs. existerar) och de andra kan då plockas bort genom att man tar bort motsvarande rader och kolumner i  $R_{mat}$  och  $sd$ , antingen före all annan processning eller efter det att en modifierad korrelationsmatris beräknats. Om man använder sig av en direkt metod med en rak invertering av  $R_{mat}$  ger dessa identiska resultat men inte annars.

Efter denna eventuella storleksreduktion beräknas så en modifierad korrelationsmatris,  $R_{mat}$ . Denna modifiering kan göras på tre olika sätt enligt avsnitt 2.3; genom modifiering av brusegenvärdena, ofta kallat "whitening" (som dock inte är detsamma som "pre-whitening" i funktionen `doa`), genom att artificiellt höja brusbidraget till korrelationsmatrisen genom att addera en diagonalmatris till  $R_{mat}$ , s.k. "diagonal loading", eller genom att använda en, på korrelationsmatrisens signalunderrum, projicerad styrvektor, ibland kallat "reduced rank method", som är användbar då nyttig signal är tillräckligt stark.

I den förstnämnda metoden anger vi i förväg, i `WBkonf`, det största brusegenvärde som ska modifieras relativt det minsta (brus-) egenvärdet samt vad dessa brusegenvärden ska modifieras till. Efter egenvärdesuppdelning, med "singular value decomposition" (`svd`), och modifiering av egenvärdena beräknas en ny korrelationsmatris helt enkelt genom hopmultiplikation av egenvektormatris,  $\mathbf{E}$ , och ny egenvärdesmatris  $\Lambda$  enligt:  $\mathbf{R} = \mathbf{E}\Lambda\mathbf{E}^H$ . (Om vi sedan bara ska invertera  $R_{mat}$  vore det naturligtvis beräkningsmässigt effektivare att göra detta direkt via  $\mathbf{R}^{-1} = \mathbf{E}\Lambda^{-1}\mathbf{E}^H$ .)

I den andra metoden ovan väljer vi i förväg, i `WBkonf`, ett värde på hur stor den adderade diagonalmatrisens element ska vara relativt det minsta (brus-) egenvärdet. Normalt adderar man här en matris med identiska diagonalelement men det är ibland användbart att addera en matris som har större element för ändelementen. Detta kommer nämligen, i MVDR, att resultera i ett vilodiagram (då endast vitt brus finns) med ett taperat diagram med lägre sidolober, se avsnitt 2.3 och [14] eller [15]. Matriselementen väljs då med hjälp av någon standard-tapering, specificerad i `WBkonf`, enligt 2.3. För att få taperingen rätt placerad, speciellt för cirkulära arrayer, skiftas viktningen till rätt plats via parametern `wDshift`.

I den tredje metoden ovan slutligen måste vi först klassificera egenvärdena som brus- eller signal- egenvärden. Detta kan f.n. göras på tre olika sätt: genom att i förväg, i `WBkonf`, ange hur stort det minsta signalegenvärdet får vara relativt det minsta (brus-) egenvärdet, genom att ange

hur litet det minsta signalegenvärdet får vara relativt det största (signal-) egenvärdet eller genom att helt enkelt ange hur många signalegenvärden vi ska använda. Man bör härvid tänka på att det oftast är bättre att överskatta än att underskatta antalet signaler.

Efter att en modifierad korrelationsmatris så erhållits kan data för aktiva tappar plockas ut, om detta inte skett tidigare, enligt ovan.

Därefter beräknas den modifierade korrelationsmatrisens invers, eller pseudo-invers om styrvektorprojektion (den tredje metoden ovan) ska användas.

Slutligen beräknas de adaptiva viktvektorerna för varje önskad nyttosignalriktning ut via, se (2.17):

$$\mathbf{w}_{adapt} = (\mathbf{R}^{-1} \mathbf{s}_d)^* \quad (6.1)$$

Om önskat (normalt) normeras så viktvektorerna med `adaptnorm` i 6.2.

Utdata är nu resulterande adaptiva viktvektorer i `wadapt`, med en kolumn per antennlob, där de inaktiva filtertapparna fyllts i som 0-or, samt den resulterande inversen av korrelationsmatrisen (inkluderande manipulationer), `Rmatm1`.

## 6.2 adaptnorm

```
wadapt=adaptnorm(wadapt, sd)
```

Anropas från: `adaptfun`

Normerar den adaptiva viktvektorn `wadapt` så att  $\mathbf{s}_d^H \mathbf{w}_{adapt} = 1$ . Om flera viktvektorer samlats i en matris så normeras varje kolumn för sig.

## 6.3 adaptnull

```
[w, Imin]=adaptnull(phisig, elevsig, phidoa, elevdoa, phiel, ...
kdoa, rel, elpattsynt, Butlerlog, invfiltlog, krow, Achan)
```

Anropas från: `WBSim3`

Anropar: `butler3, heffun3, findsig`

Här beräknas viktvektorer för vad vi kan kalla "Adaptiv nollning" enligt 2.9. Vi alstrar härvid en viktvektor för varje nyttosignal som har riktiga 0-djup i alla andra skattade signalriktningar, förutsatt att vår antennmodell är riktig (en viktig förutsättning). Detta sker väsentligen genom att först bilda styrmatrisen  $A$  för lobar i alla skattade riktningar (`phidoa, elevdoa`) och beräkna dess pseudoinvers (applicera Matlab-operationen `pinv(A)`). Parametrarna `kdoa, rel, elpattsynt, Butlerlog, invfiltlog, krow` och `Achan` används alla för att bilda denna styrmatris  $A$ .

Riktningsskattningarna (`phidoa, elevdoa`) är gjorda smalbandigt m.h.a. funktionen `doa`, vid en frekvens svarande mot `kdoa`, och själva styrmatrisen  $A$  svarar mot en smalbandig lobformning ( $J_{tap}$  ska vara 1) varför denna algoritm med fördel kombineras med en uppdelning i delband.

Vi måste också välja ut de av de skattade riktningarna, och därmed kolumner i  $A$  och  $A^\dagger$ , som svarar mot nyttosignalerna (`phisig, elevsig`). Detta görs genom att identifiera nyttosignalerna bland de skattade signalriktningarna (`phidoa, elevdoa`) med `findsig`.

## 6.4 adaptstap

```
[wadapt, Rmatm1]=adaptstap(Rmat, Cc, fc, Jtap, Rideal, ...
    indtaps, indconstr, wDshift)
```

Anropas från: WBSim3

Anropar: weight0

Här beräknas den adaptiva viktvektorn för algoritmen MVDR, även kallad "Constrained Power Minimization" eller Frost's algoritm m.m. enligt 2.5, då vi har fler än ett bivillkor. De viktigaste indata är korrelationsmatrisen  $R_{mat}$  av storlek  $[n_{el} \cdot J_{tap}, n_{el} \cdot J_{tap}]$ , en matris  $C_c$  av storlek  $[n_{el} \cdot J_{tap}, \text{antal bivillkor}]$  skapad i `constrmat3` som innehåller data vilka definierar bivillkorens riktningar och frekvenser och en vektor  $f_c$  av längd  $[\text{antal bivillkor}]$ , som innehåller bivillkorens värden. Dessutom erhålls vissa speciella parametrar för den adaptiva processningen via mat-filen `adaptpar.mat`.

Utsorteringen av aktiva filtertappar samt manipulerandet av korrelationsmatrisen  $R_{mat}$  görs här precis som för de övriga adaptiva algoritmerna behandlade i `adaptfun`, avsnitt 6.1.

Efter att en modifierad korrelationsmatris så erhållits kan data för aktiva tappar plockas ut, om detta inte skett tidigare, enligt ovan.

Därefter beräknas den modifierade korrelationsmatrisens invers, eller pseudo-invers om styrvektorprojektion ska användas.

Det är inte nödvändigt att alla kombinationer av bivillkorsriktningar och -frekvenser är aktiva. Speciellt gäller detta vid flervägsutbredning då vi antagligen har bivillkor i önskad nyttsignalriktning men inte för övriga nyttsignaler som gått annan utbredningsväg. Om alltså endast vissa bivillkor ska vara aktiva plockas motsvarande kolumner i  $C_c$  och värden i  $f_c$  bort.

Slutligen räknas de adaptiva viktvektorena för varje önskad nyttsignalriktning ut via (2.26), se vidare [2]:

$$\mathbf{w}_{adapt} = (\mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C})^{-1})^* \mathbf{f}_c. \quad (6.2)$$

Här kontrolleras också att bivillkoren är oberoende genom att beräkna konditionstalet för matrisen  $\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}$ .

Utdata är nu resulterande adaptiva viktvektorer i `wadapt`, med en kolumn per flervägsutbredningskomponent och där de inaktiva filtertapparna fyllts i som 0-or, samt den resulterande inversen av korrelationsmatrisen (inkluderande manipulationer), i `Rmatm1`.

## 6.5 butinvfilt3

```
[HinvfiltN, Hinvnorm, Hinvfilt]=butinvfilt3(elpatt, the1, k, R, nel)
```

Anropas från: butler3, weights

Anropar: heffun3

Här syntetiseras de filter vilka återställer fasskiftsmodernas (för en cirkulär array) olika känslighet. För en mer detaljerad teori hänvisas till [17] eller [18].

Utgående från aktuellt elementdiagram (i många vinkelpunkter), vid elevationsvinkeln `the1`, beräknas detta diagrams uppdelning i fasskiftsmoder med en FFT i vinkelledd över hela det aktu-

ella frekvensbandet givet av  $k$  varefter de  $n_{el}$  lägsta (lika med antal element och "tillgängligt" antal moder i arrayen) extraheras. Filterfunktionen `Hinvfilt` ges nu av inversen till dessas komplexa amplitud medan `Hinvnorm` är varje mods RMS-värde över frekvensbandet och `HinvfiltN` är de inversa filterfunktionerna normerade med `Hinvnorm`.

## 6.6 butler3

```
[vutf, H0]=butler3(vinf, invfiltlog, amodes, elpatt, thel, k, R, krow)
```

Anropas från: `adaptnull`, `constrmat3`, `doadir`, `radpattern`, `WBSim3`

Anropar: `butinvfilt3`

Här beräknas fasskiftmodernas (Butler-lobernas) komplexa amplitud genom att Fourier-transformera signalerna i elementledd, från `vinf` till `vutf`.

Eftersom denna process ger en frekvensberoende utsignal även om insignalen är frekvensoberoende finns möjlighet att, via `invfiltlog`, lägga på ett korrektionsfilter (invers-filter) vilken motverkar denna automatiska filtrering. Denna filterfunktion ges också som utdata via `H0`. Eventuellt används endast filterfunktionens värde för en, eller några, utvald frekvenspunkt, `krow`.

Filterfunktion `H0` används också i `WBSim3` till att skala eventuellt asymptotiskt brus.

Det inversa filtret syntetiseras via `butinvfilt3` första gången `butler3` genomlöps.

## 6.7 cheby

```
an=cheby(nn, r0dB)
```

Anropas från: `weight0`

Ger viktvektor för en konventionell Chebyshev-fördelning för antenn med `nn` stycken element och sidolobsnivån `r0dB`.

## 6.8 combine

```
[vut, vutttrue, vutjam, vutnoise, v2noiseA, delay, rcw]=...
  combine(vtrue, vut, vutttrue, vutjam, vutnoise, v2noiseA, ...
  ktrain, ksampcov, kintp, sdlog, combmet, Nsig, w2adapt)
```

Anropas från: `WBSim3`

Anropar: `korrdelay`, `vdelay`

Då flervägsutbredning förekommer kan man bilda separata utsignaler från olika antennlobar, med separata viktvektorer, vilka var och en tar emot signalen från en utbredningsväg enligt 2.12. Dessa delsignaler kan sedan kombineras ihop i `combine`, till en totalsignal. Därigenom ökar totala signalstyrkan och *SINR* med en faktor upp till antalet flervägskomponenter. Tre kombineringsmetoder (`combmet`) är implementerade, "equal ratio combining", *erc*, där de olika delsignalerna fördröjs och fhasas ihop med samma amplitudviktnings samt två typer av "maximum ratio combining" enligt 2.12, dels *mrc1* där amplitudviktnings läggs på i förhållande till signalernas individuella styrka, dels *mrc2* där hänsyn även tagits till dellobernas olika förväntade brusbidrag, i `w2adapt`, för att därmed försöka få maximalt resulterande *SINR*.



Bästa amplitudfaktorer, relativa fördröjning och fasskift fås från `korrdelay` genom att korrelera delsignalerna (sampel `1:ksampcov`) med träningssekvensen, `vtrue(1:ktrain)`, om detta är tillämpligt eller med en av delsignalerna som då bör vara av god kvalitet.

Utdata och delvis indata utgörs av totalsignal, sann signal, störsignal, brus (vut, vutttrue, vutjam, vutnoise) efter respektive före kombineringsen, samt asymptotiskt brus om tillämpligt (`v2noiseA`) samt använda fördröjningar (`delay`) och komplexa vikt faktorer (`rcw`).

## 6.9 constraugment

```
phiconstr=constraugment(phiconstr, nnls, Wpar, geom, Iconstr)
```

Anropas från: `WBkonf`

Då breddning av 0-riktningarna görs genom s.k. "augmentering", enligt [10], lägger man artificiellt till ett antal signaler vid sidan av de verkliga, se även 2.3, 6.29 och 6.30. Detta kommer då också att ske runt nyttosignalerna. Dessa "skyddas" i MVDR, avsnitt 2.5, normalt från undertryckning genom ett bivillkor och om vi vill undvika sådan undertryckning av de artificiella signalerna nära nyttosignalen, vilket bl.a. kommer att påverka antennvinsten för nyttosignalen (även om "känsligheten" som den är definierad här är fixerad), måste vi därför alstra nya bivillkor i dessa nya, artificiella, signalriktningar.

I `constraugment` beräknas, om så önskas, dessa nya bivillkorsriktningar för nyttosignalerna givna av index i `Iconstr`.

Detta kan tillämpas både för linjär och cirkulär array och de nya riktningarna definieras genom `nnls` eller `Wpar` på samma sätt som i `raugmentlin` och `raugmentBut` i 6.29 och 6.30. För att detta ska fungera bra för cirkulära arrayer krävs att faskiftsmoder (Butlermatris) används vid processningen och att antalet moder begränsas, se vidare 6.29 och 6.30. Som antytts i 2.3 bör detta användas med viss försiktighet.

## 6.10 constrdirmod

```
[phimod, elevmod, Imin]=constrdirmod(phidoa, elevdoa, phiconstr, ...
    elevconstr, phisig, elevsig)
```

Anropas från: `WBSim3`

Om vi använder en algoritm som definierar den nyttiga signalen genom dess infallsriktning, och därvid ansätter ett bivillkor i denna riktning, kan ett felaktigt antagande leda till kraftigt försämrade prestanda, i synnerhet om signalen är stark. Funktionen `constrdirmod` kan då användas för att modifiera denna bivillkorsriktning. Detta görs genom att beräkna vinkelavstånden från alla bivillkorsriktningar (`phiconstr`, `elevconstr`) till dels alla skattade riktningar (`phidoa`, `elevdoa`) och dels till alla antagna signalriktningar (`phisig`, `elevsig`) som kan vara fler än en vid flervägsutbredning.

Först identifieras härvid de till nyttosignalerna hörande bivillkorsriktningarna, dvs. de som ger minst vinkelskillnad, och därefter identifieras på samma sätt de till dessa bivillkorsriktningar hörande verkliga, skattade, signalriktningarna. Dessa identifierade skattade riktningar (med index `Jmin`) får sedan, i `WBSim3`, vara de nya modifierade bivillkorsriktningarna som ersätter de gamla, med index `Imin`.

## 6.11 constrmat3

```
Cctot=constrmat3(heff, k, fT, Jtap, khatrel, ksamp, Butlerlog, ...
    invfiltlog, Achan, krow0)
```

Anropas från: WBSim3

Anropar: butler3

Denna funktion beräknar den matris `Cctot` som beskriver hur bivillkoren kommer in i den adaptiva processningen, se 2.5. Den grundläggande teorin finns beskriven i [2] (sid. 361 och framåt) vars bivillkorsmatris är komplexkonjugatet av vår matris. Antennelementens känslighet i alla `Kdir` bivillkorsriktningar (enligt `khatrel`) och `Kstap` bivillkorsfrekvenser (enligt `fT`) ges i `heff` och om vi arbetar med fasskiftsmoder, enligt `Butlerlog`, räknas dessa först om till känslighet för moderna. Inte alla element/moder behöver vara aktiva vilket anges via `Achan`.

Varje kolumn i matrisen `Cctot` består av utsignalerna för var och en av de `nel` elementen och de `Jtap` filtertapparna för respektive bivillkors riktning och frekvens, travade på varandra. För var och en av de `Kdir` bivillkorsriktningarna och de `Kstap` bivillkorsfrekvenserna genereras så en ny kolumn. Dimensionen på `Cctot` blir då  $[nel \cdot Jtap, Kdir \cdot Kstap]$ .

Parametrarna `Kdir` och `Kstap` beskriver det totala antalet bivillkor. Alla kombinationer av riktningar och frekvenser behöver inte vara aktiva varvid de inaktiva sållas ut i funktionen `adaptstap` via parametern `indconstr`.

## 6.12 doa

```
[vut, Rm1doa, lambdaut, nsourceest]=doa(Rmat, eintest)
```

Anropas från: doadir

Anropar: RmatFB, Rsmooth

Utgående från signalernas korrelationsmatris `Rmat` och en styrmatrix `eintest`, bestående av signalernas ideala utseende i varje kolumn och med ett antal testriktningar i de olika kolumnerna, beräknas här ett vinkelspektrum (eller pseudospektrum). För detta använder `doa`, förutom ovan nämnda `indata`, ett antal algoritmparametrar som ligger i en mat-fil med namnet `doapar.mat`. Typen av signalbehandlingsmetod bestäms av parametern `sigbeh`.

Många estimeringsmetoder bygger på att antensystemets egenbrus är identiskt och okorrelerat mellan kanalerna (spatiellt vitt). Om så inte är fallet kan förbättrade prestanda fås om man först bestämmer, och sparar, antennens bruskorrelationsmatris. Om `sigbeh = Rnoise` betyder det att den levererade `Rmat` är just en sådan uppmätt bruskorrelationsmatris,  $\mathbf{R}_{noise}$ , som då kommer att sparas. Om man därefter åter kör programmet och har angivit ett namn på en sådan matris-fil, `Rnoisefile`, i `WBkonf` kommer denna att läsas in för användning i en korrektionsprocess, s.k. "pre-whitening". Detta sker konkret genom att pre- och post -multiplicera `Rmat` samt pre-multiplicera styrmatrisen `eintest` med matrisen:  $\mathbf{R}_{pw} = \mathbf{R}_{noise}^{-1/2}$ .

Därefter modifieras `Rmat` eventuellt för att kunna ta hand om korrelerade signaler enligt `RmatFB` respektive `Rsmooth` i 6.32 och 6.33.

Beroende på vald metod görs följande:

Om `sigbeh = con` görs en konventionell lobformning, dvs. vi beräknar utsignalen för en svept lob med konstant amplitudviktning enligt `eintest`.

Annars bildas en inversmatris  $\mathbf{R}_{\text{m1doa}}$  på några olika sätt beroende på den metod som används, se t.ex. [7] och [12]. De vanligast använda av dessa torde vara Capon, MUSIC och Minimum Norm.

Vid Capon's metod bildas helt enkelt inversen  $\mathbf{R}^{-1}$ . Denna metod är ekvivalent med adaptiv störundertryckning enligt MVDR, avsnitt 2.5, med ett bivillkor i en varierande testriktning, dvs. att vi beräknar uteffekten för scenariet som funktion av bivillkorsriktning.

Vid Maximum Entropi plockas en specificerad kolumn av denna inversa matris ut och en ny matris bildas som yttre produkten av denna vektor med sig själv.

Vid övriga (underrums-) metoder beräknas först egenvärden och tillhörande egenvektorer varefter de underrum som vi definierar som signal- respektive brus- underrum extraheras. Signalunderrummet definierar vi här via de egenvärden som är minst en faktor  $T_{\text{doamax}}$  större än det minsta egenvärdet (dock väljs högst  $n_{\text{source}}$  egenvärden ut).

I Pisarenko's metod alstrar vi en matris utgående från endast egenvektorn hörande till det lägsta egenvärdet (alla egenvärden utom det lägsta anses tillhöra signalunderrummet). Detta är ekvivalent med att låta alla egenvärden utom det minsta gå mot  $\infty$ . Härigenom kommer alla signalvektorer som är ortogonala mot denna enda brusegenvektor att ge en kraftig ( $\infty$  hög) topp i pseudospektrat.

I MUSIC alstrar vi på motsvarande en matris utgående från alla, ovan identifierade, brusegenvektorer. Det är ekvivalent med att låta alla signalvärden gå mot  $\infty$  och sätta alla brusegenvärden lika.

I Minimum Norm bildar vi först en ny "brusegenvektor" som är ett viktat medelvärde av alla brusegenvektorerna där viktfaktorerna, något godtyckligt, väljs som komponenterna i en viss, i förväg vald, rad av brusegenvektormatrisen (ofta den första raden). Denna ger t.ex. högre upplösning än MUSIC men också sämre varians.

I Egenvektor-metoden gör vi ungefär som i MUSIC men vi behåller brusegenvärdena, dvs. de sätts inte lika.

I Egenvärdesmodifiering-metoden (skulle även kunna kallas "Whitening"-metoden) slutligen, jfr. `adaptfun`, sätter vi först alla brusegenvärden lika för att sedan återskapa en ny, modifierad invers av denna modifierade korrelationsmatris.

Flera av dessa algoritmer är numera av sekundärt intresse men är här inkluderade bl.a. eftersom de utförs på nästan samma sätt (dessutom har funktionen `doa` ärvts från ett annat projekt). De mest relevanta och använda torde vara MUSIC, Capon och Minimum Norm.

För alla metoder (utom den "konventionella") bildas slutligen ett pseudo-spektra,  $\mathbf{v}_{ut}$ , genom projektionen (med beteckningar från Capon):

$$\mathbf{v}_{ut} = \frac{1}{\sqrt{\mathbf{e}_{test}^H \mathbf{R}^{-1} \mathbf{e}_{test}}} \quad (6.3)$$

där rot-funktionen har inkluderats här för att resultatet i viss mening, åtminstone för Capon, ska få dimensionen signalstyrka som vi kan plotta på samma sätt som övriga signaler, (för övrigt har rot-funktionen ingen inverkan).

### 6.13 doadir

```
[phidoa, elevdoa, vutdoamax, vutdoa]=...
```

```
doadir (Rdoa, phitest, elevtest, phiel, kdoa, rel, elpattsynt, ...
Butlerlog, invfiltlog, Achan, krow, Ndir0, Vdoarel)
```

Anropas från: WBSim3

Anropar: butler3, doa, heffun3, doamaxfind

Denna funktion organiserar riktningbestämningen till de olika signalkällorna genom att generera nominella antensignaler, `vftest`, i ett antal testriktningar givna av `phitest` och `elevtest` och därur generera (pseudo-)vinkelspektra. Förutom att dessa riktningar kan ha ett egenintresse även i kommunikationssammanhang bygger vissa adaptiva algoritmer på att åtminstone några av dem är kända, se avsnitt 2.2.

Dessa vinkelspektra beräknas normalt i funktionen `doa`, m.h.a. korrelationsmatrisen `Rdoa`, med någon vald metod. För MMSE används dock projektionen av ovanstående testsignaler på styrvektorerna  $s_d(\tau)$  enligt (2.21) (levereras i `Rdoa`-variabeln). Det senare motiveras av att korrelationen mellan träningssekvensen och andra signaler för just den funna optimala fördröjningen  $\tau$  antagligen är mycket låg så att vi får en situation med en enda signal plus brus varvid styrvektorn blir lika med den enda signalvektorn.

Andra metoder för riktningsskattning, ofta mer optimala men också beräkningskrävande eller byggande på idealiserade symmetriantaganden, vilka ger riktningar utan att gå omvägen via vinkelspektra har ej implementerats.

Som indata levereras bl.a. signalernas korrelationsmatris, `Rdoa`, eller styrvektorn för MMSE-algoritmen, och ett antal testriktningar, `phitest`, `elevtest`.

Levererade utdata är spektraltopparnas maxvärde, `vutdoamax`, sorterade i storleksordning samt tillhörande riktningar, `phidoa` och `elevdoa`, samt hela spektrum i `vutdoa`.

## 6.14 doamaxfind

```
[phidoa, elevdoa, vutdoamax, Ndir]=
doamaxfind(vutdoa, phitest, elevtest, Ndir0, Vdoarel)
```

Anropas från: `doadir`

Här väljer vi ut de relevanta maxima i de (psuedo-) spektra `vutdoa` som beräknats i `doadir` i riktningarna [`phitest`, `elevtest`]. Funktionen söker först upp alla lokala maxima, i en eller två dimensioner, och väljer ut de `Ndir0` högsta, eller så många den hittar som är högre än en faktor `Vdoarel` gånger den högsta. Slutligen skattas noggrannare värden på riktningarna genom att anpassa en andragsgradsfunktion runt varje maxima och bestämma dess maxvärde.

Levererade utdata är spektraltopparnas maxvärde, `vutdoamax`, sorterade i storleksordning samt tillhörande riktningar, `phidoa` och `elevdoa`, och antalet utvalda maxima, `Ndir`.

## 6.15 fftcperm

```
y=fftcperm(x, drow)
```

Anropas från många andra program.

Denna rutin är av generell typ. Den permuterar raderna i en, flerdimensionell, matris `x` `drow` rader nedåt (`drow` kan också vara negativt). Matlab's `fft`-rutin ger ett spektrum med DC som första punkt, och transformen tillbaka till tidsdomän, `ifft`, kräver sådana data. Vi önskar dock ofta behandla spektra där frekvensaxeln ökar monotont. Denna funktion gör, med det vanligaste

valet av `drow`, som är halva kolumnlängden, samma sak som `fftshift(x,1)` i Matlab6, (fanns inte i tidigare Matlab-versioner för annat än vektorer).

## 6.16 findsig

```
Imin=findsig(phisig,elevsig,phidoa,elevdoa)
```

Anropas från: `Wbsim3`, `adaptnull`

Här identifieras nyttosignalriktningarna med nominella riktningar `phisig` och `elevsig` bland ett antal skattade signalriktningar `phidoa` och `elevdoa`. Resultatet ges som en indexvektor `Imin`.

## 6.17 geometry

```
[rel,phiel]=geometry(geom,nel,Rel,del,nelrekt,phielrekt)
```

Anropas från: `Wbsim3`

Här genereras relevanta geometridata för arrayen, dvs. positionsvektorena `rel` och pekriktningarna `phiel` för alla `nel` element.

Fyra typer enligt parametern `geom` finns f.n. implementerade:

*linj*: En linjär array längs y-axeln (element 1 i  $y=0$ ) med elementavståndet `del` och elementen pekande i x-led.

*cirk*: En cirkulär array i xy-planet med radien `Rel`, Fig. 2.1, och elementen pekande radiellt (element 1 på x-axeln). Det ska här observeras att radien `Rel` refererar till elementens fasreferenspunkt från vilket diagrammens fas också refererar i `heffun3`.

*rekt*: En 3-dimensionell rektangulär array med antalet element i x, y respektive z-ledd givet av de tre elementen i vektorn `nelrekt` och med elementavstånden via de tre elementen i `del`. Elementens individuella, eller gemensamma, pekriktningar i xy-planet ges direkt via `phielrekt`. Normalt är här en komponent i `nelrekt` ett för en plan 2D-array. Observera också att viktning, annat än uniform, f.n. inte fungerar bra här.

Ett filnamn: Parametrarna `rel` och `phiel` ges då direkt som data i en mat-fil, filnamn `.geo`.

Det kan noteras att alla antennelement alltid förutsätts ha samma nominella elementdiagram, se `heffun3`, vilket ibland kan vara en grov approximation. Det finns dock möjlighet att via `heffrippel` lägga på individuella elementdiagram-rippel, se 6.18.

## 6.18 heffrippel

```
[heffut,heffact]=heffrippel(heffin,phi,nerrheff)
```

Anropas från: `heffun3`

Vid första anropet genereras här amplituderna på de slumpmässiga rippelfunktioner för fas och amplitud som, om önskat, ska appliceras individuellt på alla nominella elementdiagram `heffin` enligt 6.19 för att ge resulterande elementdiagrammen `heffut` för vinklarna `phi`.

Parametrar för rippet är antalet rippelperioder per varv, `nerrheff`, (dvs. fas- och amplitud-ripplet beskrivs av vardera `nerrheff` cos- och sin- funktioner) medan RMS-värdena för rippet amplitud och fas ges via mat-filen `heffrippelpar.mat`.

Därefter appliceras dessa rippelfunktioner på de nominella diagrammen i dess riktningar  $\phi_i$ . Varje element får härvid alltså ett individuellt rippel. Detta är fixt för alla anrop i en simulering men appliceras bara på de verkliga, fysikaliska, diagrammen och inte på de vid den adaptiva signalbehandlingen använda "antagna" diagrammen. Härigenom kan vi simulera okända diagramvariationer.

### 6.19 heffun3

`heff=heffun3(phi, thel, phielem, k, elpatt, nerrheff)`

Anropas från: `adaptnull, butinvfilt3, doadir, radpattern, WBSim3, weights.`

Anropar: `heffrippel`

Här genereras parametern `heff` som beskriver (är proportionell mot) antennelementens effektiva längd. Specifikt beräknas, se avsnitt 2.1.2 och 2.1.3,  $\sqrt{G_{el,r}^{(n)}(\theta, \varphi)}$  med tillhörande fas och läggs i en 3-dimensionell matris med dimension [antal vinkelpunkter, antal element, antal frekvenspunkter].

I huvudprogrammet `WBSim3` har vi möjlighet att använda två typer av elementdiagram, ett som motsvarar det verkliga fysikaliska diagrammet vilket används vid simuleringen av mottagning och av frysta diagram och ett annat antaget som motsvarar vad vi "tror" att diagrammet är vilket används vid signalbehandlingen. Härigenom kan vi simulera förhållandet att vi inte har fullständig kännedom om antennens egenskaper.

Inparametrarna till `heffun3` är:

`phi`: en vektor med alla azimutvinklar,

`thel`: elevation, antingen en skalär för gemensam elevation eller en vektor med samma längd som `phi` för individuella elevationsvinklar,

`phielem`: pekriktningarna för alla element, i xy-planet.

`k`: en vektor med vågtalen för infallande vågor ( $k = 2\pi f/c$ ).

`elpatt`: beskriver elementtyp, den verkliga eller den som vi tror att det är.

`nerrheff`: högsta Fourierkomponenten i `heff`:s rippel, se 6.18, eller 0 då rippel saknas.

Ett alternativ är att ge `elpatt` som något av ett antal standard-diagram. Om  $\theta$  betecknar vinkeln från elementets pekriktning och  $\theta_{elev}$  vinkeln från horisontal- (xy-) planet finns följande `elpatt`-alternativ:

$\cos (h_{eff} \propto \cos \theta)$ ;  $sq\cos (h_{eff} \propto \sqrt{\cos \theta})$ ;  $1+\cos (h_{eff} \propto 1 + \cos \theta)$ ;  $omni (h_{eff} \propto \cos \theta_{elev})$  och  $iso (h_{eff} \propto \text{konstant})$ .

Maxvärdet sätts som roten ur elementdiagrammens direktivitet, vilken är 6, 4, 3, 1.5 och 1 respektive. Se även avsnitt 2.1. Om andra maxvärden för dessa elements realiserade antennvinst önskas får man f.n. simulera detta genom att ändra signalstyrkorna.

I konfigureringsprogrammet `WBkonf`, kapitel 4, ges signalstyrkan för signalerna. Dessa ska då tolkas som resulterande *SNR* om antennen i fråga hade varit en isotrop antenn, dvs. `elpatt = iso`. Eftersom, i verkligheten,  $h_{eff} \propto \lambda \sqrt{G}$  innebär detta att vi simulerar med en inneboende, med frekvensen ökande, infallande effekttäthet. Se vidare avsnitt 2.1 och kapitel 4.

Dessa ovanstående alternativ av såväl form som maxvärde skall betraktas som en grov approximation till verkligheten som mycket väl kan strida mot fysikens lagar. I verkligheten påverkas elementdiagrammen i högsta grad av de andra elementen, särskilt om dessa ligger tätt räknat i våglängder. Normalt ska elementen ligga på högst 1/2 våglängds intervall varvid  $h_{eff} \propto \sqrt{\cos\theta}$  kan synas vara det mest realistiska för stora linjära arrayer där vi dock fortfarande försummat missanpassningens inverkan på vinkel och frekvensberoendet. För bredbandiga arrayer ligger elementen därför också ofta tätare varvid  $h_{eff}$  sjunker, något som inte heller tagits hänsyn till här. Se vidare avsnitt 2.1.

Ett annat alternativ är att ge `elpatt` som ett filnamn. Data för  $h_{eff}$  ska då ligga i en 3-dimensionell matris i filen `filnamn.pat`, helst i form av  $\sqrt{G_{el}}$  med strålningens fas inkluderad enligt 2.1.2, tillsammans med vektorer för aktuella azimut-, elevations- och frekvens- punkter. Önskade data för punkterna i `phi`, `thel`, `phielem` och `k` interpoleras sedan fram. Denna interpolationsprocedur tar f.n. så lång tid att utföra att det dominerar den totala simuleringstiden varför det ofta är fördelaktigt att göra så många studier som möjligt med de tidigare nämnda standardtyperna. (En snabbare funktion i t.ex. programmeringsspråket C vore önskvärt.)

## 6.20 Hmatfilt

```
vut=Hmatfilt(vin,Hmat,Jtap)
```

Anropas från: `radpattern`, `WBSim3`

Givet indata matrisen `vin` med dimension [`ksamp`,`nel`] och koefficientmatrisen `Hmat` med dimension [`nel · Jtap`, Antal lober] beräknas utsignalerna från lobformning med ett FIR-filtter med `Jtap` tappar i varje kanal. Det antas här att signalen i `vin` fortsätter periodiskt (försumbart om `Jtap`  $\ll$  `ksamp`) så att alla fördröjningar lätt kan realiseras. Utdatamatrisen `vut` får då också samma dimension som `vin`.

## 6.21 kanalS21

```
[S21,S2mean,S21std]=...
```

```
kanalS21(nerr,dgdB0,dphdeg0,dgdBF,dphdegF,nel,ksamp,inds210)
```

Anropas från: `WBSim3`

Här genereras data för kanalernas överföringsfunktion, `S21`, samt deras kvadratiska medelvärde och standardavvikelse. Medelvärdet över alla element och alla frekvenser är alltid lika med ett.

Parametern `nerr` beskriver hur många Fourier-termer som beskriver amplitud- och fas -ripplet, `nerr=[]` betyder inget fel alls, `nerr=0` betyder frekvensoberoende fel med de för alla kanaler oberoende och normalfördelade felens RMS-värden givna av `dgdB0` och `dphdeg0`. Om `nerr>0` betyder det att både amplitud och fas har, förutom de frekvensoberoende felen, ett oberoende rippel i varje kanal som beskrivs via sina RMS-värden `dgdBF` och `dphdegF`, fördelade över vardera `nerr` `cos`- och `sin`- funktioner.

Indexvektorn `inds210` används för att kunna nollställa vissa `S21` så att helt trasiga mottagare kan simuleras.

## 6.22 korrdelay

```
[vkorr, vrefdelay, delay, indrowmax, indcolmax, vkorrM, E2]=...
    korrdelay(v, vref, kintp, maxtotal, Nsig0, Vsigrel, sdRlog, ...
    indcolkorrr, Jtap, korrfold)
```

Anropas från: combine, WBSim3

Anropar: korrmaxfind

Denna funktion används till tre saker, dels för att finna den bästa synkroniseringen av styrvektorn för MMSE-algoritmen, dels att finna bästa synkronisering av de olika signalerna vid flervägsutbredning och dels att finna den nyttiga signalen i utdatasignalen vid utvärderingen av simuleringsresultatet. I de två senare fallen är endast de tre första indataparametrarna satta och endast en del av funktionen utnyttjas. I det första fallet består indataparametern  $v$  antingen av signalerna i alla antennelement eller av utsignalen från en i förväg bildad konventionell antennlob, vilket kan vara användbart om vi vet ungefär var nyttosignalen finns, medan i de två senare fallen  $v$  består av utsignalen från adaptivt formade lobber.

Övriga indata består av referenssignalen  $vref$ , en interpolationsfaktor  $kintp$ , en logisk variabel  $maxtotal$  som är 1 för det första fallet ovan, antalet sökta maxima  $Nsig0$  för varje signal, relativ nivå  $Vsigrel$  för godkända maxima, metodval  $sdRlog$ , aktiva kanaler  $indcolkorrr$  för sökning, antal filtertappar  $Jtap$  för eventuellt FIR-filter vid sökning av minstakvadratfel och den logiska variabeln  $korrfold$  som beskriver eventuell periodisk fortsättning av signalerna.

Utdata är de funna maxvärdena  $vkorr$ , fördröjda referenssignaler  $vrefdelay$ , fördröjningarna  $delay$  själva, index för korrelationsmaxima  $indrowmax$  och  $indcolmax$ , den totala korrelationsmatrisen  $vkorrM$  och resulterande minstakvadratfelet  $E2$ .

Korrelationsberäkningen görs, istället för med faltning i tidsdomänen, genom en multiplikation av  $v$  och konjugatet av  $vref$  i frekvensdomänen följt av en inverstransform tillbaka till tidsdomänen till funktionen  $vkorr$ . Den normalt kortare  $vref$  har då först fyllts ut med 0-or i tidsdomänen för att få samma längd som  $v$ . Med denna metod kommer signalerna att permuteras cykliskt dvs. det antas att de upprepas periodiskt. Genom parametern  $korrfold$  kan vi emellertid kräva att de resulterande maxima inte ligger så nära signalvektoreernas ändpunkter att någon vinkning av signalen skett (förutom eventuell extra fördröjning av hela vektorns längd).

Eftersom vi önskar en betydligt bättre skattning av optimal fördröjning än vad som ges av samplingstätheten, vilken förutsätts vara bestämd enligt samplingsteoremet enligt avsnitt 2.1 dvs. med tidsintervallet  $1/B$ , interpoleras också korrelationsfunktionen med en faktor  $kintp$  genom att lägga till 0-sampel utanför nominella bandbredden.

I de två senare användningsfallen ovan eller om bästa fördröjning ska sökas direkt på element eller lobdata ( $sdRlog=0$ ) sorteras först data nära änden bort om vinkning ska undvikas enligt ovan. Därefter sökes helt enkelt maximum av beloppet av korrelationsfunktionen och om data från många antennelement finns (specifika val kan göras i  $indcolkorrr$ ) söker man därefter upp det antennelement ( $indcolmax$ ) som gav det högsta maximumet.

Som ett alternativ till detta ( $sdRlog=1$ ) kan vi, vid första användningsfallet ovan, söka den fördröjning som ger det minsta värdet på minsta kvadratfelet, som ju är vad den adaptiva MMSE-algoritmen ska minimera. Det innebär alltså att vi, för varje fördröjning räknar ut detta minsta kvadratfel (och inverterar då vi söker maximum). Vi räknar då ut en total signalmatris,  $v$ , giltig för alla kanaler och alla  $Jtap$  filtertappar och bildar den totala korrelationsmatrisen  $\mathbf{R}$  som vi inverterar. Den resulterande felsignalen ( $E2$ ) ges, för varje fördröjningssteg, nu enligt ekv. (2.24) av:



$$|e(\tau)|^2 = |v_{ref}|^2 - (\mathbf{v}_{korr}^H(\tau) \mathbf{R}^{-1} \mathbf{v}_{korr}(\tau)) / K. \quad (6.4)$$

Det har visat sig att denna senare metod ger mycket goda prestanda men att den också är mycket beräkningskrävande. (Antalet filtertappar i denna sökning,  $J_{tapkorr}$  enligt Appendix A, kan oftast väljas till 1 för bra och stabila resultat.)

Om fler än ett korrelationsmaxima ska hittas för en korrelationsfunktion, dvs. sökning av bästa fördröjningar av referenssignalen vid flervägsutbredning, söks dessa därefter upp i `korrmaxfind` och sorteras varefter de `Nsig0` högsta maxima extraheras.

När vi slutligen hittat de bästa fördröjningarna, i `delay`, fördröjs kopior av referenssignalen med respektive tidsfördröjning och ges via utdataparametern `vrefdelay`. Resultterande korrelation vid optimal fördröjning ges i utdata via `vkorr` (behöver inte vara reell) och används vid hopkombineringen vid flervägsutbredning, enligt 2.12.

## 6.23 korrmaxfind

```
[vabsmax, indrowmax]=korrmaxfind(vabs, kintp)
```

Anropas från: `korrdelay`

Denna funktion letar upp relevanta maxima i beloppet av den korrelationsfunktion som bildats i `korrdelay`. Vi vill här alltså hitta de "riktiga" korrelationsmaxima men sälla bort sekundära maxima som uppkommer p.g.a. sidolobsstrukturer runt huvudmaxima och brusliknande maxima. Man kan här välja mellan tre olika metoder via parametern `maxfindmet` (global variabel). Dessa har något olika prestanda vid speciella fall men är i huvudsak likvärdiga. I samtliga tre metoder utgår vi från en reell positiv vektor, `vabs`, där antalet datapunkter utökats från originaldata genom interpolation med en faktor `kintp`. Max-värdena och index ges i `vabsmax` och `indrowmax`.

I metod "1" söker vi först de "originalpunkter", dvs. punkter som skiljer sig med `kintp` steg, som är större än sina grannar och när sådana hittats sökes maximum i ett  $2 * kintp$  brett intervall.

I metod "2" söker vi först maxima i `kintp` breda intervall och därefter söker vi det största av varje sådant maximum och de två närmaste på varje sida.

I metod "3" söker vi direkt upp maxima i  $4 * kintp$  breda löpande intervall.

## 6.24 noisepar

```
pnoise=noisepar(noisevar, noisesp, nel)
```

Anropas från: `WBsim3`

Beräknar bruseffekten `pnoise` i var och en av de `nel` mottagarkanalerna, givet bruseffektens variationsegenskaper i variationstypen `noisevar` ( $n$ ,  $u$  eller  $g$ ) och RMS-värdet `noisesp`.

## 6.25 Propmodel

```
[phisigi, elevsigi, numsig, Delaysigi, Magsigi, npaths, truesigi]=...
```

```
Propmodel(truesigin)
```

Anropas från: `WBsim3`

Denna funktion är ett mellansteg mellan `WBSim3` och den specifika vågutbredningsmodellen `Propmodel_72` (vilken inte behandlas i denna rapport). Resulterande utdata är `phisigi`, `elevsigi`, `Delaysigi`, `Magsigi` som är radvektorer med resulterande signalriktningar i azimuth och elevation, signalernas fördröjningar i sekunder och signalstyrkor, vilka motsvarar *SNR* för ett isotropt antennelement enligt avsnitt 2.1, `numsig` som är antalet signaler (längden av föregående radvektorer), `npaths` som är en vektor med antalet flervägskomponenter för varje signalkälla med ett element per signalkälla (summan av elementen i `npaths` ska vara lika med `numsig`) samt `truesig` som ger vilka källor (ofta bara en) som ska betraktas som vår nyttiga signal.

## 6.26 Propsig

```
[vsigf, vsigt]=Propsig(vsigt, delaysigi, magsigi, npaths, f0Ts)
```

Anropas från: `WBSim3`

Anropar: `vdelay`

Denna funktion används om den specifika vågutbredningsmodellen `Propmodel` i 6.25 används. Utgående från sändarsignaler `vsigt`, fördröjningar `delaysigi`, signalstyrkor `magsigi`, flervägskomponenter `npaths` samt centerfrekvensens fasskift per sampelsteg `f0Ts` beräknas här de totala infallande signalerna i antennsystemets origo i både frekvensdomän (`vsigf`) och tidsdomän (`vsigt`). Data ligger i matriser med frekvens/tids-sampel i den ena och totala antalet signaler (riktningar) i den andra dimensionen.

## 6.27 radpattern

```
[vutf, vutt]=radpattern(wadapt, wadaptf, Jtap, phitest, elevtest, ...
    phiel, k, krowtest, Nsubband, Subbandindex, elpatt, elpattsynt, ...
    tfBF, rel, Rel, S21, Butlerlog, invfiltlog, elevinvfilt, ...
```

`Achan, nerrheff`)

Anropas från: `WBSim3`

Anropar: `butler3, heffun3, Hmatfilt`

Efter det att de adaptiva viktcoefficienterna, och diverse andra parametrar, beräknats i `WBSim3` kan den adaptiva antennens strålningsdiagram, med de antagna elementdiagrammen enligt 6.19, beräknas. Detta görs i riktningar som definieras av vinklarna `phitest` och `elevtest`. Beräkningarna kan göras antingen i tidsplanet genom faltning med kanalfiltrens `Jtap` tappar, varvid viktcoefficienterna i `wadapt` används, eller i frekvensdomänen, varvid coefficienterna för alla `ksamp` frekvenspunkter i `wadaptf` används. Dessa två metoder ska ge identiska resultat men eftersom vi oftast endast önskar strålningsdiagram för ett begränsat antal frekvenspunkter, definierade av `krowtest`, är analysen i frekvensplanet ofta betydligt snabbare.

Utdata är två matriser, `vutf` och `vutt`, som innehåller komplexa strålningsfält med frekvens respektive tidssteg längs varje kolumn och eventuellt olika lober, t.ex. för olika flervägskomponenter eller utstyrningsvinklar, för olika kolumner. Insignalen antas vara en plan våg från testriktningarna med konstant frekvensspektrum (efter en isotrop mottagarantenn med  $G_{el} = 1$  enligt 2.1) i frekvenspunkterna `krowtest`, så att (om alla `ksamp` frekvenspunkter är medtagna) tidsignalen blir av typen modulerad  $\sin x/x$ -funktion (andra vågformer kan lätt implementeras).

Först genereras insignalen i varje antennelement i frekvensdomän, med frekvenspunkter givna av `krowtest`, genom multiplikation med elementens "verkliga", eventuellt rippliga, överföringsfunktion hämtade från `heffun3` och med eventuellt mottagarrippel i `S21` följt av eventuell transformation till fasskiftsmoder via `butler3`. Om inte alla element eller fasskiftsmoder är aktiva (ges av `Achan`) extraheras dessa därefter.

Om lobformningen ska göras i frekvensdomän multipliceras så helt enkelt signalspektrum efter kanalmottagarna, för alla indatariktningar, skalärt med vikterna `wadaptf` för aktuella frekvenspunkter varefter tidssvaret beräknas med en invers FFT.

Om lobformningen ska göras i tidsdomän, och vi inte delat upp den adaptiva processningen i delband, räknas frekvensspektrat efter kanalmottagarna om till tidsdomän med en invers FFT varefter utsignalen genereras via skalärmultiplikation med viktvektorn `wadapt` eller via anrop till filterlobformaren i `Hmatfilt`. Därefter beräknas utsignalen i frekvensdomän med en FFT och de efterfrågade frekvenspunkterna extraheras.

Om den adaptiva processningen delats upp i flera delband (`Nsubband > 1`) görs lobformning i frekvensdomän precis som ovan medan lobformning i tidsdomän måste modifieras något. Detta görs genom att utföra processningen ovan för varje delband för sig, dvs. för ett intervall av frekvensindex, vilket processas i tidsdomän och transformeras tillbaka till frekvensdomän för att tillsammans generera ett totalt frekvenssvar. Härvid används olika viktcoefficienter, och kanske också olika antal filtertapper, i de olika delbanden. Det totala tidssvaret fås sedan med en invers FFT varefter de efterfrågade frekvenspunkterna extraheras ur det totala spektrat.

## 6.28 `randncmplx`

```
v=randncmplx(ntot,ksamp)
```

Anropas från: `signals`

Här genereras komplexa slumpade normalfördelade signaler, med variansen ett, i alla kanaler och för alla signalkällor. För teständamål kan två olika metoder väljas i `WBkonf` (variabeln `vrandmet`) för att få viss reproducerbarhet mellan olika simuleringar.

För deterministiska signaler ändras senare dessa slumpsignaler i `specsignals`.

## 6.29 `raugmentBut`

```
Raug=raugmentBut(nnls,Wpar,nel,amodes)
```

Anropas från: `WBSim3`

Denna funktion är nära besläktad med `raugmentlin` i 6.30. Den senare bygger emellertid på förutsättningen att antennen är en linjär ULA medan vi här tänker oss en uniform cirkulär array (UCA). Som beskrivits på annan plats, t.ex. avsnitt 2.3, är emellertid likheterna mellan en ULA och en UCA stora om vi arbetar med fasskiftsmoderna istället för med elementsignalerna direkt. Den enda skillnaden blir att parametern `Wpar` nu beskriver vinkelavståndet  $\delta$  från verklig signal direkt (istället för  $\sin\Delta$ ). Vektorn `amodes` beskriver vilka fasskiftsmoder som ska användas och om det hela ska fungera bra får man inte använda för höga modnummer, dvs. för nära  $\pm nel/2$ .

## 6.30 `raugmentlin`

```
Raug=raugmentlin(nnls,Wpar,xel,k0)
```

Anropas från: `WBSim3`

De resulterande minima för en adaptiv gruppantenn är vanligen mycket smala och ibland, t.ex. för att få större effektiv bandbredd, önskar man bredda dessa. Detta kan göras genom att artificiellt lägga till ett antal extra störare runt de verkliga. Det är naturligtvis mycket svårt i ett generellt fall men för en likformig linjär array, s.k. ULA, finns så stora symmetrier att det blir relativt enkelt, se [10] och avsnitt 2.3. Man finner nämligen i detta fall att om varje element  $(i,j)$  i korrelationsmatrisen  $\mathbf{R}$  multipliceras med en faktor

$$R_{aug}(i,j) = \exp(jk_0(x_i - x_j)\delta), \quad (6.5)$$

där  $x_i$  och  $x_j$  (i vektorn `xel`) är positionerna för element  $i$  och  $j$ , så är detta ekvivalent med att varje planvågssignal (antas okorrelerade med varandra) kommer att förflyttas en vinkel  $\Delta$ , där  $\sin\Delta = \delta$ , från den ursprungliga. Den totala  $\mathbf{R}_{aug}$  fås som en summa av bidragen och individuella  $\delta$  kan ges i vektorn `wpar` varvid  $\delta=0$  bör vara ett av värdena (ger en matris med bara ettor). Om man önskar extra minima på ekvidistanta avstånd från de verkliga minima kan proceduren förenklas genom att summationen kan utföras analytiskt, förutom att indata blir enklare att generera. Detta sker genom att ge ett enda värde på `wpar`, svarande mot halva bredden, samt antalet extra minima på varje sida i parametern `nnls`.

Förutom att införa nya minima kommer denna procedur att öka egenbrusbidraget som finns i korrelationsmatrisens diagonal med en faktor lika med totala antalet termer i `wpar`, respektive  $2 \cdot nnls + 1$ , vilket dock oftast saknar större betydelse.

Vi kommer dock också automatiskt att lägga till extra signaler runt nyttsignalen, vilket kan vara en nackdel om inte denna är mycket svag. Se vidare 2.3.

### 6.31 rektfilt

```
vut=rektfilt(vin, flow, fhigh)
```

Anropas från: `specsigs`

Denna funktion lägger på ett enkelt filter med rektangulär överföringsfunktion genom att transformera insignalen `vin` till frekvensplanet och multiplicera med ett rektangulärt fönster, skilt från noll mellan `flow` och `fhigh`, samt en transformation tillbaka till tidsplanet till variabeln `vut`.

### 6.32 RmatFB

```
RFB=RmatFB(R)
```

Anropas från: `doa`

För att kunna lösa upp två koherenta signaler appliceras s.k. "Forward-Backward averaging" på kovariansmatrisen  $\mathbf{R}$ , se t.ex. [7]. Metoden fungerar dock bara för linjära symmetriska arrayer, t.ex. en ULA, eller då signalerna korrigerats till "ULA-signaler" eller för en UCA med fasskiftsmoder av inte alltför hög ordning (dvs. då modernas rippel är litet enligt t.ex. [17] [18]), dvs. där en spegelvändning och komplexkonjugering ger samma resultat.

### 6.33 Rsmooth

```
[Rss, esteer]=Rsmooth(R, esteer, msub)
```

Anropas från: `doa`

För att kunna lösa upp koherenta signaler appliceras här s.k. "Spatial Smoothing" på kovariansmatrisen  $\mathbf{R}$ , se t.ex. [7]. Här tänkes arrayen uppdelad i ett antal,  $K_{\text{tot}} = n_{\text{el}} - m_{\text{sub}} + 1$ , överlappande och identiska subarrayer, vardera med storleken  $m_{\text{sub}}$  element. En mindre korrelationsmatris bildas så från  $\mathbf{R}$  som medelvärde av  $K_{\text{tot}}$  delmatriser. Samtidigt beräknas också nya styrvektorer i `esteer` med längd,  $m_{\text{sub}}$ . Fungerar bara för ULA eller då signalerna korrigerats till "ULA-signaler" eller för en UCA, dvs. som vid `RmatFB` enligt 6.32.

### 6.34 Rstap

`[R, vfiltstap]=Rstap(vfilt, Jtap)`

Anropas från: `WBSim3`

Givet datamatrisen `vfilt`, med data i tidsledd, med dimension  $[k_{\text{samp}}, n_{\text{el}}]$ , beräknas här den direkta korrelationsmatrisen för "Space Time Adaptive Processing" (STAP) enligt (2.18) dvs. då vi har ett FIR-filer med `Jtap` tappar i varje kanal.

Detta görs genom att först bilda en större datamatrix  $\mathbf{v}$  för signalerna i alla element och alla filtertappar. Denna får dimension  $[k_{\text{samp}} - J_{\text{tap}} + 1, n_{\text{el}} \cdot J_{\text{tap}}]$  och minskningen i kolumnlängd orsakas av att vi inte vill extrapolera fram några nya data. Varje fördröjningssteg betraktas alltså precis som ett nytt element ty inget principiellt skiljer dessa åt ur signalbehandlingsynpunkt. Korrelationsmatrisen bildas nu på vanligt vis dvs., med vårt utseende på  $\mathbf{v}$ , enligt:

$$\mathbf{R} = \mathbf{v}^T \mathbf{v}^* / (k_{\text{samp}} - J_{\text{tap}} + 1) \quad (6.6)$$

### 6.35 Rtimestap

`Rtot=Rtimestap(Rspat, Rtime, Jtap, Achan)`

Anropas från: `WBSim3`

Här bildas en asymptotisk bruskorrelationsmatris,  $\mathbf{R}_{\text{tot}}$ , utgående från kanalernas bruseffekter i `Rspat`-vektorn och kanalfiltrens autokorrelationsmatris `Rtime`. Kanalerna antas här vara helt okorrelerade.

Parametern `Jtap` är antalet filtertappar och `Achan` innehåller index på de aktiva kanalerna.

### 6.36 sdstap3

`sd=sdstap3(vref, vin, Jtap)`

Anropas från: `WBSim3`

Här beräknas den korrelationsvektor  $\mathbf{sd}$  mot vilken MMSE-algoritmen ska försöka att styra in signalen. Denna funktion är besläktad med `Rstap`, 6.34, för skattning av korrelationsmatrisen  $\mathbf{R}$ . Vektorn  $\mathbf{sd}$  är skattad korrelation mellan i `korrdelay`, 6.22, tidigare fördröjd referenssignal `vref` (flera signaler vid flervägsutbredning) och mätt signal `vin` i alla kanaler och alla `Jtap` filtertappar enligt (2.21). För att undvika att behöva skapa nya data genom t.ex. periodisk extrapolation minskas antalet använda sampel vid varje filtertapp med `Jtap - 1`.

Eftersom vi förväntar oss bäst resultat om de starkaste filterkoefficienterna hamnar ungefär mitt i filtret, och sedan avtar mot dess ändar, translateras först `vref` med halva filterlängden innan

korrelationen beräknas genom en enkel skalärmultiplikation. Korrelationsvektorn  $sd$ , med längd  $nel \cdot J_{tap}$  (där elementnummret varierar snabbast), normeras också så att elementens kvadratsumma blir  $nel$  så att ett "idealt" fall ger koefficienterna 1 för filtrens mitt-tappar.

### 6.37 signals

```
[vsigf, vsigt, vtrue, vnoise]=...
    signals(numsig, nel, ksamp, vsource, truesig, truedelay, ...
    f0Ts, sigmode, fsig, flow, fhigh, Ortolog, Scov, Brussamplog)
```

Anropas från: `WBSim3`

Anropar: `randncmplx, vdelay`

Här bildas alla `numsig` signaler med `ksamp` sampel, var och en med form enligt parametervektorn `sigmode`, styrka `vsource` och fördröjning `truedelay`. Dessutom bildas brussignaler, dvs. egenbrus, med variansen ett för de `nel` kanalerna. Funktionens utdata är signalerna i frekvensdomän `vsigf`, signalerna i tidsdomän `vsigt`, den verkliga sanna informationssignalen `vtrue` och brussignalerna `vnoise`.

Signalen `vtrue` (eller en del av `vtrue`) kommer i `WBSim3` att användas dels som en träningssekvens vid korrelationsdetektering och dels vid själva utvärderingen av resultatet för att få ett mått på adaptivitetens prestanda i form av *SINR* och *BER*.

För vissa teständamål finns här också möjlighet att, via `Ortolog`, se till att de normalfördelade slumpsekvenserna verkligen är helt okorrelerade och också att bilda signaler med given korrelationsmatris `Scov`, varvid dock de ursprungliga vågformerna delvis förstörs såvida de inte också är rena brussignaler. Dessutom kan man, via `Brussamplog`, låta bli att generera egenbrus varvid beräkningarna i `WBSim3` kommer att ske med en asymptotisk bruskorrelationsmatris.

### 6.38 specsignals

```
vrand=specsignals(vrand, numsig, ksamp, sigmode, fsig, flow, fhigh)
```

Anropas från: `signals`

Anropar: `rektfilt`

Här genereras, eventuellt med hjälp av tidigare genererade slumpsignaler i `vrand`, upp till `numsig` speciella vågformer enligt önskemål i `sigmode`, var och en med `ksamp` sampel. Dessutom filtreras signalerna via `rektfilt` efter önskemål enligt `flow` och `fhigh`.

### 6.39 taylor

```
an=taylor(nn, r0dB, nbar)
```

Anropas från: `weight0`

Ger viktvektor för en konventionell Taylor-fördelning givet antal element `nn`, sidolobsnivå `r0dB` och den speciella parametern `nbar` som i huvudsak beskriver efter hur många sidolober dessa ska börja minska, se t.ex. [22]. Beräkningen baseras här på placering av diagrammets 0-ställen och en rekursiv beräkning av antenndiagram-polynomets koefficienter utgående från dessa 0-ställen. Då antalet element är stort kan numeriska onoggrannheter uppstå varvid man kan använda `taylorcont` istället.

### 6.40 taylorcont

```
an=taylorcont(nn,r0dB,nbar)
```

Anropas från: `weight0`

Ger viktvektor för en konventionell Taylor-fördelning liksom `taylor`. Funktionen `taylorcont` baseras på sampling av en kontinuerlig fördelning. Det ger sämre resultat för små och medelstora antenner men ger mindre numeriska problem för stora antenner. Se t.ex. [22].

### 6.41 vdelay

```
vtdelay=vdelay(vtin,delay,f0Ts)
```

Anropas från: `combine`, `Propsig`, `signals`

Här fördröjs signalerna i indatamatriken `vtin` med `delay` antal sampelsteg (ej nödvändigtvis ett heltal) genom multiplikation med  $\exp(-j\omega\tau)$ . Parametern  $f0Ts = f_0 \cdot T_{samp}$ , där  $f_0$  är signalens mitt- (bärvågs-) frekvens, är centerfrekvensens fasskift per sampelsteg, (som är lika med 0 efter sampling).

### 6.42 WBxyvektor

```
WBxyvektor
```

Anropas från: `Wbsim3`

Detta är ett enkelt hjälpprogram för generering av utdatamatriernas rad- och kolumn- variabler och deras namn. Att användas vid plottning.

### 6.43 weight0

```
[w0,etaa0,w2aver]=weight0(wtype,nel,r0dB,nbar,aktelem)
```

Anropas från: `adaptfun`, `adaptstap`, `weights`

Anropar: `cheby`, `taylor`, `taylorcont`

Organiserar anropen till andra funktioner som beräknar viktvektorn `w0` för konventionella antennlobber av typ givna i `wtype`. För närvarande finns `enkelement`, `uniform`, `Taylor` och `Chebyshev` implementerade men man kan lätt utöka till fler. Viktvektorn normeras till  $\max(w0) = 1$  och dessutom beräknas konventionell aperturverkningsgrad `etaa0` och kvadratiska medelvärdet `w2aver`.

### 6.44 weights

```
[wtot,wtot1,wfilt]=...
```

```
weights(geom,wtype,r0dB,nbar,aktelem,nel,nelw0,nelstart,...
Scanvinkel,Elevsynt,rel,Rel,phiel,elpattsynt,elpattkorr,...
k,ksynt,fsynt,Butlerlog,wButEl,invfiltlog,Achan)
```

Anropas från: `Wbsim3`

Anropar: `butinvfilt3`, `heffun3`, `weight0`

Här beräknas de viktvektorer som, då de appliceras på signalerna från antennelementen eller fasskiftsmoderna, ger önskade vilodiagram. Sådana vilodiagram används då man vill generera icke adaptiva antenndiagram och då man använder de adaptiva metoderna (givna av indatavariabeln `sdlog` i `WBkonf`) `styrv` (dvs. "Maximum SNR" eller "Applebaums" algoritm, avsnitt 2.6), `SLC` (dvs. "Sidelobe Cancellor", avsnitt 2.7), `AdAd` (dvs. vi bildar nya adaptiva kanaler från lobdata, avsnitt 2.8) eller `korr` (dvs. "Minimum Mean Square Error", avsnitt 2.4) om bästa synkronisering av träningssekvensen ska ske på data från en konventionell lob.

Först beräknas en konventionell viktvektor med något standardutseende i funktionen `weight0`.

Förutsatt att antennarrayen inte är en cirkulär array och att vi därvid ska använda fasskiftsmoder (dvs. Butlermatris) vid syntesen (men dock om vi syntetiserar en cirkulär array direkt) räknar vi ut fasen för alla elementvikter som fasen fram till en plan våg i syntesriktningen (`Scanvinkel`, `Elevsynt`) med ombytt tecken, dvs. faktorn  $\exp(-jkx_{el} \cdot \hat{r})$ . Dessutom ser vi till att rätt vikter tillskrivs rätt element vilket speciellt för en cirkulär array kan variera.

Om elementen inte har samma elementdiagram, räknat i ett antennfast koordinatsystem, finns möjlighet att, via `elpattkorr`, ta hänsyn till detta genom att multiplicera vikterna med elementdiagrammens (konjugerade) styrka i lobriktningen, varigenom den realiserade antennvinsten hos vilodiagrammet maximeras (jfr. avsnitt 2.1.2) och vikterna ser ut som konjugatet av planvågssignalerna, eller med elementdiagrammens invers, vilket kompenserar för felet så att ett "bättre" vilodiagram fås, (`expfact1` resp. `expfact`). Detta har olika inverkan vid den adaptiva lobformningen och den signalbehandling som sker där, se vidare avsnitt 2.2 och appendix H i [16].

Om antennarrayen är en cirkulär array och fasskiftsmoder ska användas i syntesen beräknas fasskiftet för moderna i syntesriktningen och viktcoefficienterna och fasskiftet sorteras om så att vikter, fasskift och moder passar ihop. Vikternas fasskift ska, enligt reciprocitet, vara "elementens" negativa sändningsfas i lobriktningen. Här måste man dock tänka på att dessa "element" är portarna till en tänkt Butlermatris (fasskiftsmoderna är inga antennportar i egentlig mening) och att en sådan port  $n$  tar emot en fasskiftsmod med index  $-n$  varför fassfaktorn för mod  $n$  förblir  $\exp(jn\phi_{scan})$ . Se vidare [17], [18] och avsnitt 2.1. Därefter beräknas filterkoefficienter, `wfilt`, och normeringskoefficienter, `wnorm`, ut i funktionen `butinvfilt3`.

Om inversfilter inte ska användas beräknas modsignalerna (konjugerade) ut för ett planvågssinfall vid den valda syntesfrekvensen, med index `krow`, respektive de koefficienter, `rad krow` i `wfilt`, vilka motverkar dessa modernas amplitud- och fasskift vid denna frekvens, ekvivalent med planvågssyntesen ovan (`expfact1` och `expfact`). Vikterna korrigeras så med dessa koefficienter för att bli korrekta vid just syntesfrekvensen.

Om inversfilter (`invfiltlog`), dvs. ett filter för varje fasskiftsmod vilken motverkar modens frekvensgång, ska användas återställer detta filter signalen och vi korrigerar då endast för medeldämpningarna i varje mod, i `wnorm`, vilken filtren själva också är normerade med. Inverkan av olika korrigeringar beskrivs mera i appendix H i [16] och delvis i avsnitt 2.3. Användandet av sådana filter gör att antennens strålningsdiagram blir mycket bredbandigt även då endast rena kanalvikter (`Jtap=1`) används. Om fasskiftsmoderna endast ska användas vid själva syntesen men inte vid lobformningen, återförs vikterna till elementdomän genom en Fouriertransform.

Om endast vissa (`Achan`) element eller fasskiftsmoder ska användas väljer vi sedan ut dessa.

Resultatet returneras så i viktvektorerna `wtot` och `wtot1` samt eventuellt också filterkoefficienterna i `wfilt`.



## 7. Referenser

- [1] R. A. Monzingo, T. W. Miller, *Introduction to Adaptive Arrays*, John Wiley & Sons, 1980.
- [2] R. T. Compton, *Adaptive Antennas - Concepts and Performance*, Prentice Hall, 1988.
- [3] J. Litva, T. K.-Y. Lo, *Digital Beamforming in Wireless Communications*, Artech House, 1996.
- [4] J. C. Liberti Jr., T. S. Rappaport, *Smart Antennas for Wireless Communications*, Prentice Hall 1999.
- [5] J. E. Hudson, *Adaptive Array Principles*, IEE Electromagnetic Waves Series 11, Peter Peregrinus LTD, 1981.
- [6] E. Nicolau, D. Zaharia, *Adaptive Arrays*, Studies in Electrical and Electronic Engineering 35, Elsevier, 1989.
- [7] H. Krim, M. Viberg, "Two Decades of Array Signal Processing Research", *IEEE Signal processing Magazine*, pp. 67-94, Juli 1996.
- [8] S. P. Applebaum, "Adaptive Arrays", *IEEE Trans. on Antennas and Prop.*, pp. 585-598, Sept. 1976.
- [9] O. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing", *Proc. IEEE*, pp.926-935, Aug. 1972.
- [10] R. J. Mailloux, "Covariance matrix augmentation to produce adaptive array pattern troughs", *Electr. Letters*, No. 10, pp. 771-772, Maj 1995.
- [11] E. Brookner, J. M. Howell, "Adaptive-adaptive array processing", *Proc. IEEE*, April 1986, pp. 602-604.
- [12] D. H. Johnson, D. E. Dudgeon, *Array Signal Processing. Concepts and techniques*, Prentice Hall, 1993.
- [13] H. Steyskal, "Array Error Effects in Adaptive Beamforming", *Microwave Journal*, pp. 101-112, Sept. 1991.
- [14] L. Pettersson, "Adaptive beamforming with imperfect arrays: Pattern effects and their partial correction", Rome Lab. report, RL-TR-92-341, Dec. 1992.
- [15] L. Pettersson, "Adaptive beamforming with imperfect arrays: Pattern effects and their partial correction", *Proc. Int. Symp. on Antennas (JINA92)*, Nice, Nov. 1992, pp. 621-624.
- [16] L. Pettersson, "An S-band Digital Beamforming Antenna: Design, Procedures and Performance", FOA-R--99-01162-408--SE, Dec. 1999.
- [17] T. Martin, L. Pettersson, J. Lorén, R. Erickson, J. Rahm, "A Broadband Circular Antenna Array", FOA-R--99-01321-504--SE, Dec. 1999.
- [18] L. Pettersson, "Cirkulär gruppantenn för Aerostatradar", FOI-R--0245--SE, Okt. 2001.
- [19] F. Eklöf, L. Pettersson, J. Rantakokko, O. Tronarp, "ELSAsd - en mjukvarudemonstrator för adaptiva gruppantenn", FOI Memo 01-3224, Okt. 2001.
- [20] F. Eklöf, L. Pettersson, J. Rantakokko, "On Broadband Adaptive Beamforming for Tactical Radio Systems", FOI-S--0149--SE, Presenterad på MILCOM, McLean, USA, Oktober 2001.
- [21] J. Rantakokko, L. Pettersson, F. Eklöf, "Robust Broadband Adaptive Beamforming for Tactical Communication Systems", FOI-S--0225--SE, Presenterad på Information Systems Technology Panel Symposium (NATO/PfP), Warszawa, Polen, Oktober 2001.
- [22] R. S. Elliot, *Antenna Theory and Design*, Prentice Hall, 1981.
- [23] T. S. Rappaport, *Wireless Communications. Principles & Practice*, Prentice Hall, 1996.
- [24] S. R. Saunders, *Antennas and Propagation for Wireless Communication Systems*, Wiley, 1999.
- [25] G. Strang, *Linear Algebra and its Applications*, Saunders Publishing, Harcourt Brace Jovanovich College Publishers, 1988.

## APPENDIX A. Alla indatavariabler i alfabetisk ordning

- Adapt: Styr typ av processning: adaptering (*adapt*), spatiell filtrering (*spfilt*) eller bara antenndiagram (*antdiag*).
- adaptel: Adaptiva element vid  $sdlog=SLC$ , avsnitt 2.7.
- aktelem: Aktivt element eller aktiv fasskiftsmod (Butlerlob) (från  $-n\ell/2$ ) vid  $wtype=sing$ .
- Antdiagtyp: Vilken vikt-typ ska väljas för icke-adaptiva diagram, *wtot* eller *wtot1*? Se vidare 2.3 och 6.44.
- Beamkorrllog: Ska korrelation i *korrdelay* ske med signal i konventionell lob, (1/0)? Förutsätter  $sdRlog=0$ .
- Brussamplog: Slumpa brussampel eller addera en asymptotisk bruskorrelationsmatris, (1/0)?
- Butlerlog: Butlermatris eller inte för cirkulär array, (1/0)?
- comb: Ska flervägsutbredningskomponenterna hopkombineras *fore* eller *efter* sammanläggningen av eventuella delband?
- combmet: Kombineringsmetod av flervägsutbredningskomponenter. "Maximum Ratio Combining" typ1, *mrc1*, resp. typ2, *mrc2*, eller "Equal Ratio Combining", *erc*. Se 2.12.
- Constraugmentlog: Ska nya bivillkorsriktningar skapas runt signalerna, (1/0)? Bra vid *Raugmentlog*. Se vidare 6.10.
- del: Elementavstånd i linjär array (skalär) eller i rektangulär array (3 elements vektor), [m]?
- Deltaf: Frekvenssteg vid strålningsdiagramberäkning. (Där ofta färre frekvenspunkter önskas.)
- dgdBF: RMS-värde för stokastiska förstärkningsrippel i kanalförstärkarna [dB].
- dgdB0: RMS-värde för stokastiska frekvensoberoende förstärkningsfel i kanalförstärkarna [dB].
- dhdBF: RMS-värde för stokastiska amplitudrippel i elementdiagrammen [dB].
- dhpdegF: RMS-värde för stokastiska fasrippel i elementdiagrammen [grader].
- dphdegF: RMS-värde för stokastiska fasrippel i kanalförstärkarna [grader].
- dphdeg0: RMS-värde för stokastiska frekvensoberoende fasmusfel i kanalförstärkarna [grader].
- elevconstr: Elevationsvinklar för bivillkoren. Eventuellt med en ny rad för varje scenario.
- elevdoatest: Testelevationer för riktningsskattning i *doadir*.
- elevinvfilt: Giltig elevationsvinkel för utjämningsfiltret, vid *invfiltlog*.
- elevsig: Elevationsvinkel för signal/störare. Skalär, radvektor eller matris med en rad för varje adapteringsfall (scenario).
- elevsigAdAd: Antagna nyttosignal-elevationer att användas om  $siginRmat=0$  och  $sdlog=AdAd$ .
- elevsigNBnull: Antagna nytto-signalelevationer då  $sdlog=NBnull$ .
- Elevsynt: Elevationsvinkel vid diagramsyntes.
- elevtest: Elevationsvinklar vid strålningsdiagramberäkning av frysta diagram. Ett värde eller lika lång vektor som *phitest*.
- elpatt: Verkligt elementdiagram, används för simuleringsdata. Alternativ: *iso*, *omni*, *cos*, *sqcos*, *1+cos* eller ett filnamn (exkl. extension \*.pat).
- elpattkorrl: Ska viktvektorn modifieras med hänsyn till varierande elementdiagram vid element- (dvs. ej Butler-) syntes, (1/0)? Se vidare *Antdiagtyp*.
- elpattsynt: Antaget elementdiagram. Används vid syntes och signalbehandling. För övrigt som *elpatt*.
- evm: Ska egenvärdesmodifiering användas, (1/0)? Avsnitt 2.3.
- FBaver: "Forward/Backward averaging" i *doa*, (1/0)? Endast för geometrierna *linj* och *cirk* med *Butlerlog=1*.
- fconstr: Bivillkorens värden,  $f_c$  i (2.26).

**F<sub>highreal</sub>**: Vektor med övre gränsfrekvenser för rektangulära filter för respektive signal.  
**F<sub>lowreal</sub>**: Vektor med undre gränsfrekvenser för rektangulära filter för respektive signal.  
**f<sub>max</sub>**: Övre bandgränsen för simuleringen (inkluderas ej).  
**f<sub>min</sub>**: Undre bandgränsen för simuleringen (inkluderas).  
**foreft**: Sitter eventuella kanalfel före eller efter egenbrusgenereringen, *fore* eller *after*?  
**F<sub>sinusreal</sub>**: Frekvenser för eventuella sinussignaler för respektive signal. Se **sigmode**.  
**F<sub>syntreal</sub>**: Frekvens för diagramsyntes.  
**F<sub>Tcreal</sub>**: Bivillkorsfrekvenser.  
**F<sub>ttestreal</sub>**: Frekvenser för strålningsdiagramberäkning.  
**geom**: Typ av array-geometri, *cirk*, *linj*, *rekt* eller ett filnamn (exkl. extension \*.geo).  
**GeomPropMod**: Signaler via separat geometrisk vågutbredningsmodell? (1/0).  
**Imod**: Vilka bivillkor hör till nyttsignalriktningar? Används endast om **phisigmodlog=0** och vi har flervägsutbredning, annars beräknas **Imod** i **WBSim3**.  
**indconstr**: Logisk matris för gällande bivillkor. Av storlek [antal riktningar, antal frekvenser]. [] betyder att alla gäller.  
**indS210**: Index för trasiga mottagare. [] innebär: ingen trasig.  
**Indtaps**: Logisk array av storlek [**nelA**, **Jtap**] som beskriver vilka filtertappar som är aktiva. [] betyder att alla är aktiva.  
**invfiltlog**: Ska utjämningsfilter för fasskiftsmoderna i cirkulär array användas då **Butlerlog=1**, (1/0)?  
**Jtap**: Antal filtertappar i kanalfiltren. Max-antal om de ska vara olika i olika delband, se **JtapTot**.  
**Jtapkorr**: Antal filtertappar vid sökandet efter vid minsta fel i **korrdelay** då **sdRlog=1**.  
**JtapTot**: Vektor med antal filtertappar i de olika delbanden. Endast om alla filtertappar ska vara aktiva enligt **Indtaps**.  
**kintp**: Interpolationsfaktor till **korrdelay**.  
**korrfold**: Ska korrelationsmax i **korrdelay** sökas med periodiskt **vref**, (1/0)?  
**korrkanal**: Vilka elementkanaler ska användas i **korrdelay** (om **sdRlog=0** och **Beamkorrlog=0**)? [] betyder alla kanaler.  
**ksamp**: Antal datasampel, och frekvenspunkter. Ett jämnt tal.  
**ksampcov**: Antal datasampel i kovariansmatrisen **R** och i korrelationsstyrvektorn **s<sub>d</sub>** vid MMSE.  
**ktrain**: Antal datasampel i träningssekvensen vid MMSE.  
**lprojpar**: Parameter vid **sdproj**. Se **sdprojmet**.  
**maxfindmet**: Metodvalsparameter till **korrmaxfind**.  
**Mxekolumn**: Kolumnnummer att användas i **doa** för *mxe* och *mnn*.  
**nbar**: Taylor-parameter, se [22], (< **nel/2**).  
**nbara**: Taylor-parameter för diagonallast enligt **wDiagtype**, se [22], (< **nel/2**).  
**Ndir**: Max antal estimerade signalriktningar i **doadir**.  
**nel**: Antal antennelement.  
**nelA**: Antal aktiva adaptiva antennelement för cirkulär array utan Butlermatris (**Butlerlog=0**). För övriga fall räknas **nelA** ut.  
**nelrekt**: Vektor med antal element i x-, y- respektive z- ledd för rektangulär array.  
**nelstart**: Första aktiva element för vilodiagram vid syntes på elementnivå.  
**nelstartA**: Första aktiva adaptiva elementet för cirkulär array utan Butlermatris (**Butlerlog=0**). För övriga fall räknas **nelstartA** ut.  
**nelw0**: Antal aktiva element för vilodiagram vid syntes på elementnivå.  
**nerrheff**: Max antal perioder i diagrammens vinkel-rippel, 0 ger inget rippel.

nerrterm: Max antal perioder i kanalfelens frekvens-ripple. 0 ger endast frekvensoberoende fel, [] betyder: inga fel.

nmod0: Antal 0-ställda moder vid syntes av vilodiagram på Butlermatrisnivå.

nmod0adapt: Antal 0-ställda adaptiva moder för cirkulär array med Butlermatris.

nnls: Antal artificiella störare på varje sida om verklig, jfr. [10]. Se även Raugmentlog och Wpar.

noisesp: RMS-värde för brusfaktorernas variation [dB], se noisevar.

noisevar: Brusfaktorernas variationstyp mellan kanalerna:  $n=nej$ ,  $u=uniform$  eller  $g=gaussisk$ .

Nrun: Antal repeterade körningar med samma adaptiva koefficienter men olika slumpsekvenser för signaler och brus.

Nsig: Antal flervägsutbredningskomponenter som vi ska leta efter vid adapteringen.

nsource: Antal antagna källor för riktningsskattningen i doa. Används för *mus*, *mnn*, *egv* och *evm*.

Nsubband: Antal delband som den adaptiva processningen ska delas in i.

Omitsyntlog: Logisk variabel för att kunna undertrycka diagramsyntes. För debug-ändamål.

Ortolog: Ortogonalisering av brus och signaler, (1/0)? Fungerar bra endast för *randn*-signaler.

Pdoarel: Den lägsta relativa nivån i pseudospektrum i doadir för klassning som signal.

phiconstr: Azimutvinklar för bivillkoren. Eventuellt med en ny rad för varje scenario.

phidoatest: Testriktningar för riktningsskattning i doadir.

phielrekt: Pekriktning för alla element i rektangulär array, vektor eller array.

phisig: Azimutvinkel för signal/störare, matris med en rad för varje adapteringsfall (scenario).

phisigAdAd: Antagna nytto-signalriktningar att användas om  $signRmat=0$  och  $sdlog=AdAd$ .

psigdB: Styrka för signalerna. SNR per kanal i dB för isotropt antennelement, radvektor eller matris med en rad för varje adapteringsfall (scenario). Se också avsnitt 2.1.

phisigmodlog: Ska bivillkorsriktningar till nyttosignalerna modifieras till, av  $constrdirmod$ , skattade riktningar, (1/0)?

phisigNBnull: Antagna nytto-signalriktningar då  $sdlog=NBnull$ .

phitest: Vinklar för strålningsdiagramberäkning av frysta diagram.

Radpattlog: Ska strålningsdiagram beräknas, (1/0)? (Det krävs dock fler villkor uppfyllda.)

randnstate: Generator-variabel till Matlab-funktionen *randn*. Används för att kunna ändra utfallet av slumpalsgenereringarna.

randstate: Generator-variabel till Matlab-funktionen *rand*. Ekvivalent med *randnstate*.

Raugmentlog: Ska nollställena breddas genom "augmentering", (1/0)? Endast för *linj* eller *cirk* med  $Butlerlog=1$ . Se vidare avsnitt 6.29, 6.30 och [10]. Se upp med nya 0-ställen runt nyttosignal. Se även *Constraugmentlog*.

Rel: Radie till elementens fasreferenspunkt för cirkulär array [m].

Rmatktrainsamp: Ska endast *ktrain* sampel användas i **R** i MMSE, (1/0)? (Ger stabilare lösning.) Se även avsnitt 2.4 och 5.2.1.

Rnoisefile: Filnamn för eventuell fil med bruskorrelationsmatris för "pre-whitening" i doa, [] betyder: ingen "pre-whitening".

r0dB: Nominell sidolobsnivå enligt *wtype* (för *cheb*, *tayl* och *taylc*).

r0dBa: Nominell sidolobsnivå för diagonallast enligt *wDiagtype* (för *cheb*, *tayl* och *taylc*).

Scanvinkel: Nominell utstyrningsvinkel. Får ofta vara flera i en vektor för  $sdmet=styrv$  eller *SLC*.

Scov: Matris för signalernas eventuella korrelationskoefficienter, om skilt från en enhetsmatris.

sdlog: Val av adaptiv algoritm. *riktn*: Constrained Power Minimization, 2.5 och 5.2.2, *korr*: Minimum Mean Square Error, 2.4 och 5.2.1, *styrv*: Max signal-brus-förhållande, 2.6 och

5.2.3, *NBnull*: 0-djup i skattade riktningar, 2.9 och 5.2.6, *AdAd*: Adaptive-adaptive processing, 2.8 och 5.2.4, *SLC*: Side Lobe Canceller, 2.7 och 5.2.5.

*sdmet*: Val av viktvektoralternativ: *Vilodiagram* (ger *wtot*) eller *Planvåg* (ger *wtot1*). Se vidare 2.3 och 6.44.

*sdproj*: Ska styrvektorprojektion på signalunderrummet användas, (1/0)? Se avsnitt 2.3.

*sdprojmet*: Vilken metod för val av signal- och brus- underrummen ska användas, *maxnoise*, *minsig* eller *antal*, se vidare avsnitt 2.3?

*sdRlog*: Ska optimal fördröjning av träningssekvens hittas med "Minimum-Error" i *korrdelay*, (1/0)? Detta ger bättre resultat men också längre beräkningstid. Se avsnitt 2.4, speciellt (2.24), och 6.22.

*sigbeh*: Algoritm för sökandet av signalriktningar i *doa*: *capon*, *mus*, *mnn*, *egv*, *evm*, *pis*, *mxe*, *con* eller *Rnoise*. Avsnitt 6.12.

*siginRmat*: Ska nyttig signal vara med vid adapteringen, (1/0)? För vissa teständamål.

*sigmode*: Vågform för alla signaler: *randn* (default), *sinus*, *BPSK*, *QPSK* eller *MLSBPSK*. (Den senare är en funktion, ej inkluderad i denna rapport.)

*spsmooth*: Antal subarrayer vid "spatial smoothing" i *doa*. Endast för geometrierna *linj* och *cirk* med *Butlerlog*=1.

*Tartrel*: Relativ diagonallast, relativt det lägsta egenvärdet.

*Tdoamax*: Definition av högsta brusegenvärde, relativt det minsta, i *doa*. Större egenvärden definierar signaler och det villkor som ger flest signaler av detta och *nsource* används.

*Tegv*: Ersättningsegenvärde för små egenvärden (se *Tmin*) vid egenvärdesmodifiering, *evm*, i *adaptfun* och *adaptstap*.

*tfBF*: Ska beräkning av frysta strålningsdiagram utföras i frekvens- eller tids- domän, *frek* eller *time*? (Ska ge identiska resultat.)

*Tmin*: Minsta använda brusegenvärde, relativt det lägsta egenvärdet, vid egenvärdesmodifiering, *evm*.

*Tprelog*: Ska matriselementen för använda filter-tappar extraheras ur **R** redan före den adaptiva processningen, (1/0)?

*truedelay*: Fördröjningar, i antal sampelintervall (=1/B), för nyttsignalerna i ordning enligt *truesig*.

*truesig*: Nyttosignaler, fler vid flervägsutbredning, angivna enligt deras kolumnposition i *phisig* och *psigdB*.

*vrandmet*: Metodvalsparameter till *randcplx*. Huvudsakligen för debug-ändamål.

*Vsigrel*: Lägsta korrelation, relativt den högsta, för att en flervägsutbredningskomponent ska inkluderas i *korrdelay*.

*wButEl*: Ska diagramsyntesen ske på *element* eller *butler*- nivå för cirkulär array då *Butlerlog*=0, dvs. då fasskiftsmoder inte används vid själva analysen?

*wDiagtype*: Vilket typ av diagonallast ska appliceras: *nej* (ingen diagonal last), *unif*, *cheb*, *tayl* eller *taylc*? Se mera i avsnitt 2.3.

*wpar*: Positionen för artificiella störare. Se även *Raugmentlog*, *nnls*, i 6.29 och 6.30.

*wtype*: Vilken typ av elementviktning ska användas till ordinärt antenndiagram: *unif*, *cheb*, *tayl*, *taylc* eller *sing* (ensamt element eller fasskiftsmod)?

