

EN KORT SAMMANFATTNING AV TEORIN INOM

DIGITAL KOMMUNIKATION

Linjär kod

En *binär* linjär kod kännetecknas av att summan av två kodord också är ett kodord. Ett specialfall är summan av ett kodord med sig själv som blir bara nollor, eftersom man med addition menar bitvis addition. Notera att om man inte talar om en *binär* linjär kod, så är varje *linjärkombination* av kodord också ett kodord.

Effektspektrum, spektraltäthet

Effektspektrum (eller spektraltätheten) för en signal är fouriertransformen av signalens autokorrelationsfunktion. Ett viktigt specialfall är vitt brus som har ett konstant spektrum enligt:

$$R(f) = \frac{N_0}{2}$$

Additivt vitt gaussiskt brus (AWGN)

En enkel modell för en kanal innefattar additivt vitt gaussiskt brus (förkortat AWGN efter den engelska beteckningen). Detta är precis vad det låter som – vitt gaussiskt brus adderas till den signal som överförs på kanalen. Eftersom bruset är vitt har det konstant spektraltäthet:

$$R(f) = \frac{N_0}{2}$$

Dessutom är det gaussiskt, vilket innebär att dess amplitud är normalfördelad.

Signalanpassat filter

Ett filter sägs vara anpassat till signalen $s(t)$ om dess impulssvar har formen $h(t) = k \cdot s(T-t)$ där k och T är konstanter. Om man som insignal till detta filter skickar in $s(t) + \text{brus}$ så blir utsignalen i tidpunkten $t = T$ $k \int_{-\infty}^{\infty} s^2(t) dt + \text{brus}$. I precis denna tidpunkt är utsignalens signal-brus-förhållande maximalt och passar därför bra för sampling och vidare behandling.

Vektormodellen

De olika signalalternativen $s_i(t)$ med $i \in \{1, 2, \dots, \mu\}$ befinner sig i ett funktionsrum med N dimensioner, där $N \leq \mu$. Detta rum spänns upp av ON-basen Φ_j där $j \in \{1, 2, \dots, N\}$. Detta ger att signalen $s_i(t)$ kan skrivas som $s_i(t) = \sum_{j=1}^N s_{ij} \Phi_j(t)$, $i \in \{1, 2, \dots, \mu\}$ där s_{ij} definieras som $s_{ij} = \int_0^T s_i(t) \Phi_j(t) dt$. I vektormodellen representeras signalerna med vektorer \mathbf{s}_i där $i \in \{1, 2, \dots, \mu\}$. Den j :te komponenten i vektorn \mathbf{s}_i ges av uttrycket för s_{ij} ovan.

Även bruset tas omhand av vektormodellen. Naturligtvis har bruset en större dimension än de rena signalerna och man kan därför tycka att det inte skulle gå att representera det med hjälp av de N basfunktionerna. Det visar sig dock att man inte förlorar något på att bortse från det brus som inte ligger längs någon av basfunktionerna – varför skulle man ta hänsyn till en störning som uppenbart inte hör till den funktionsrymd som de intressanta signalerna lever i?

Vektormodellen är användbar vid beräkning av till exempel minimiavstånd och felsannolikhet, eftersom den ger upphov till smarta geometriska tolkningar som gör det mycket lättare att se signalernas orientering relativt varandra än om man befunnit sig i funktionsrymden. Det visar sig också att alla beräkningar som man kan göra i vektorrymden kan utföras i funktionsrymden med ekvivalent resultat. Här nedan beskrivs några viktiga geometriska beräkningar för signaler.

Normen av en vektor

Normen (längden) av vektorn \mathbf{s}_i betecknas $\|\mathbf{s}_i\|$ och beräknas enligt:

$$\|\mathbf{s}_i\|^2 = \sum_{j=1}^N s_{ij}^2 = \int_0^T s_i^2(t) dt = E_i$$

Vinkeln mellan två vektorer

Vinkeln α_{ik} mellan de två vektorerna \mathbf{s}_i och \mathbf{s}_k kan beräknas enligt:

$$\cos(\alpha_{ik}) = \frac{\mathbf{s}_i \cdot \mathbf{s}_k}{\|\mathbf{s}_i\| \|\mathbf{s}_k\|} = \frac{\int_0^T s_i(t) s_k(t) dt}{\sqrt{E_i} \sqrt{E_k}}$$

Avståndet mellan två vektorer

Avståndet d mellan de två vektorerna \mathbf{s}_i och \mathbf{s}_k kan beräknas enligt:

$$d^2 = \|\mathbf{s}_i - \mathbf{s}_k\|^2 = \int_0^T (s_i(t) - s_k(t))^2 dt = \int_0^T s_i^2(t) dt + \int_0^T s_k^2(t) dt - 2 \int_0^T s_i(t) s_k(t) dt$$

Ett stort avstånd mellan två basvektorer är bättre än ett litet, då det medför att det blir lättare att rekonstruera den utsända signalen på mottagarsidan.

Generatormatris

Varje blockkod har en *generatormatris* \mathbf{G} som fullständigt definierar koden. Kodorden fås genom att man bildar alla linjärkombinationer av raderna i \mathbf{G} . Uttryckt i algebra – om vi har en meddelandevektor $\mathbf{m}=[m_1 m_2 \dots m_k]$ så erhåller vi kodvektorn $\mathbf{c}=[c_1 c_2 \dots c_n]$ genom operationen $\mathbf{c}=\mathbf{mG}$. Ur detta ser vi att generatormatrisen måste ha k rader och n kolonner. Observera att raderna i generatormatrisen måste vara linjärt oberoende – annars skulle den producera flera likadana kodord för olika meddelanden, vilket ju är helt meningslöst.

En särskild generatormatris är en sådan som genererar en *systematisk* kod, det vill säga en kod där informationssymbolerna ingår i kodordet som de k sista (eller första) symbolerna. Formen för en sådan generatormatris är $\mathbf{G}=[\mathbf{P} \mathbf{I}_k]$ där \mathbf{P} är en kontrollmatris med dimension $(k, n-k)$ och \mathbf{I}_k är enhetsmatrisen med dimension (k, k) . En sådan generatormatris ger upphov till kodord på formen $\mathbf{c}=\mathbf{mG}=\mathbf{m}[\mathbf{P} \mathbf{I}_k]=[\mathbf{mP} \mathbf{mI}_k]=[\mathbf{mP} \mathbf{m}]$. Vi ser alltså att själva meddelandevektorn ligger med som sista biten i kodvektorn – precis som vi ville.

Paritetsmatris

Paritetsmatrisen \mathbf{H} är relaterad till generatormatrisen \mathbf{G} genom sambandet $\mathbf{HG}^T=\mathbf{0}$, där nollvektorn har dimension $(n-k, k)$. Det följer även att om vi har en kodvektor \mathbf{c} så gäller $\mathbf{Hc}^T=\mathbf{0}$ där nollvektorn är en kolonnvektor med $n-k$ nollor.

Om vi har en systematisk kod så kan vi bilda en paritetsmatris med hjälp av det vi vet om generatormatrisen. Eftersom denna definieras som $\mathbf{G}=[\mathbf{P} \mathbf{I}_k]$ kan paritetsmatrisen skrivas som $\mathbf{H}=[\mathbf{I}_{n-k} \mathbf{P}^T]$.

Syndromvektor

Vid överföringen av ett kodord kan det hända att fel inträffar, vilket i praktiken innebär att en symbol som skulle varit en nolla istället tas emot som en etta, eller tvärt om. För att beskriva detta kan man säga att om \mathbf{c} är den utsända kodvektorn så är den mottagna vektorn $\mathbf{r}=\mathbf{c}+\mathbf{e}$, där \mathbf{e} är en *felvektor* som har ettor på de positioner där ett fel uppstått. När mottagaren tar emot \mathbf{r} så multiplicerar den det med paritetsmatrisen och bildar *syndromvektorn* \mathbf{s} enligt $\mathbf{s}=\mathbf{Hr}^T$. Denna syndromvektor är en kolonnvektor med $n-k$ komponenter. Om man sätter in $\mathbf{r}=\mathbf{c}+\mathbf{e}$ i uttrycket för syndromvektorn så får man $\mathbf{s}=\mathbf{Hr}^T=\mathbf{H}(\mathbf{c}^T+\mathbf{e}^T)=\mathbf{Hc}^T+\mathbf{He}^T=\mathbf{He}^T$ och kan då inse den intressanta egenskapen att syndromet endast beror av felvektorn – den utsända kodvektorn spelar ingen roll.

Mottagarens uppgift är att utifrån det erhållna syndromet komma fram till vilket felmönster som är aktuellt, för att därefter kunna rätta till felen och producera ett mottaget meddelande som är detsamma som det utsända. Dessvärre är det inte så bekvämt att varje syndrom motsvarar ett felmönster – om k är antalet informationssymboler så finns det 2^k stycken felvektorer som ger upphov till samma syndromvektor. Avkodaren måste ändå välja ett felmönster som verkar sannolikt, och generellt väljs det mönster som har lägst *vikt*, det vill säga man gissar på att så få fel som möjligt har uppstått. Detta är den bästa gissningen under förutsättning att bitfels sannolikheten är mindre än $\frac{1}{2}$, vilket kan anses som rimligt.

Hammingavstånd

Koder kan, om de är smart konstruerade, användas till att rätta och/eller upptäcka överföringsfel. Hur många fel per kodord som koden kan upptäcka eller rätta avgörs av *minimivståndet* d_{min} , eller det minsta *Hammingavståndet*. Hammingavståndet mellan två kodord avses antalet positioner där orden skiljer sig åt. Vidare finns det två regler för hur detta påverkar kodens förmåga att rätta och upptäcka fel, enligt nedan.

För att en kod ska kunna rätta t fel per kodord krävs att:

$$d_{min} \geq 2t + 1$$

För att en kod ska kunna upptäcka v fel per kodord krävs att:

$$d_{min} \geq v + 1$$

Observera att dessa är beroende av varandra – om man vill att koden ska kunna både rätta t fel och samtidigt upptäcka v fel per kodord krävs istället att:

$$d_{min} \geq t + v + 1$$

Man kan beräkna minimivståndet för en kod på flera sätt. Ett är förstås att ställa upp alla kodord och beräkna avstånden mellan dem. Ett annat är att utnyttja att *Hammingvikten* (antalet ettor) av summan av två kodord är detsamma som avståndet mellan dem. Alltså är minimivståndet i en kod detsamma som minimivikten.

Gray-kodning

Idén med Gray-kodning är att kodorden ska arrangeras så att ett kodord bara skiljer sig från vart och ett av sina närmaste grannar med en bit. På det sättet minimeras skadan som ett överföringsfel skulle innebära. Gray-kodning fungerar inte med alla moduleringsmetoder – vid exempelvis M-FSK är alla signalavstånd lika stora vilket gör Gray-kodning meningslös.

Digitala moduleringsmetoder

Det finns en mängd metoder för digital modulering. Här är några av de vanligaste. I beskrivningen av var och en av metoderna antas kommunikationen ske över en AWGN-kanal med spektraltäthet $N_0/2$. Vidare antas E representera medelsignalenergin över alla signaler i den aktuella situationen.

On-off keying (OOK)

On-off keying (OOK) kallas på svenska till-från-signalering och går ut på att en binär nolla representeras av en nollsignal och en binär etta av en nollskild signal, vanligen någon cosinusformad sådan. Exempel:

$$s_0(t)=0$$

$$s_1(t)=\sqrt{\frac{4E}{T}}\cos(2\pi f_c t), \quad 0 \leq t < T$$

Avståndet mellan de två signalerna är $d=\sqrt{2E}$, vilket ger felsannolikheten $P_e=Q\left(\sqrt{\frac{E}{N_0}}\right)$.

Binary phase shift keying (BPSK)

Binary phase shift keying (eller binär fasskiftssignalering) (BPSK) innebär att de binära symbolerna 0 och 1 representeras av signalerna:

$$s_0(t)=\sqrt{\frac{2E}{T}}\cos(2\pi f_c t), \quad 0 \leq t < T$$

$$s_1(t)=\sqrt{\frac{2E}{T}}\cos(2\pi f_c t + \pi) = -s_0(t), \quad 0 \leq t < T$$

Avståndet mellan signalerna fås till $d=2\sqrt{E}$, vilket innebär att felsannolikheten kan beräknas till $P_e=Q\left(\sqrt{\frac{2E}{N_0}}\right)$.

Binary frequency shift keying (BFSK)

Binary frequency shift keying (binär frekvensskiftssignalering) (BFSK) kallas även ortogonal signalering och är en moduleringsmetod där symbolerna 0 och 1 representeras av signalerna:

$$s_0(t)=\sqrt{\frac{2E}{T}}\cos(2\pi f_1 t), \quad 0 \leq t < T$$

$$s_1(t)=\sqrt{\frac{2E}{T}}\cos(2\pi f_2 t), \quad 0 \leq t < T$$

Frekvenserna f_1 och f_2 väljs så att $2f_1T$ och $2f_2T$ är olika heltal, vilket medför att signalerna garanterat är ortogonala. Avståndet mellan signalerna är $d = \sqrt{2E}$ vilket ger felsannolikheten $P_e = Q\left(\sqrt{\frac{E}{N_0}}\right)$.

Amplitude shift keying (ASK)

En utökning av OOK är amplitude shift keying (ASK) eller amplitudskiftssignalering. Denna metod använder en endimensionell uppsättning signaler, oftast jämnt fördelade längs basfunktionen.

Quadrature phase-shift keying (QPSK/4-PSK)

Quadrature phase-shift keying (QPSK) eller rätt och slätt 4-PSK består av fyra signaler enligt:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + (2i-1)\frac{\pi}{4}), \quad 0 \leq t < T, \quad i=1,2,3,4$$

Signalerna är alltså jämnt fördelade på en cirkel med radien \sqrt{E} och avståndet mellan närliggande signaler är därför $d = \sqrt{2E}$ vilket ger att felsannolikheten kan approximeras med närmsta-granne-metoden till $P_e = 2Q\left(\sqrt{\frac{E}{N_0}}\right)$.

8-Phase shift keying (8-PSK)

8-Phase-shift keying (8-PSK) användes i en del gamla modem och ser ut som 4-PSK fast med åtta signaler jämnt fördelade på cirkeln. Signalerna kan alltså beskrivas med:

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + (2i-1)\frac{\pi}{8}), \quad 0 \leq t < T, \quad i=1,2,3,4,5,6,7,8$$

Med några geometriska manövrar kan man komma fram till att minimiavståndet mellan signalerna är $d = 2\sqrt{E} \sin\left(\frac{\pi}{8}\right)$. Detta betyder att symbolfelsannolikheten kan approximeras till $P_e = 2Q\left(\sqrt{\frac{2E}{N_0}} \sin\left(\frac{\pi}{8}\right)\right)$.

Quadrature amplitude modulation (QAM)

När man använder PSK har alla signaler samma amplitud och därmed samma energi. I quadrature amplitude modulation (QAM) är signalerna istället fördelade över ett rutmönster. Oftast är antalet signaler en jämn fyrapotens. Exempelvis vid 16-QAM så kan man komma fram till att minimiavståndet är $d_{\min} = \sqrt{2E/5}$ och symbolfelsannolikheten enligt närmsta-granne-approximering $P_e = 3Q\left(\sqrt{\frac{E}{5N_0}}\right)$.