

Laboration i TSKS10

VT-2022

Den här laborationen går ut på att kommunicera audiosignaler med hjälp av I/Q-modulering över en kanal som skalar och fördröjer signalen, och där du har ett givet frekvensband att använda. Uppgiften är att konstruera en sändare och mottagare i Matlab, som också tar reda på fördröjningen och skalningen, samt kompenserar för dem.

Scenario

Du har i uppgift att konstruera en sändare och en mottagare som kommunicerar två audiosignaler över en delad kanal. Kanalen är uppdelad i flera frekvensband, och du har blivit tilldelad ett av dem. Andra tänkta användare kommunicerar på övriga frekvensband. Detta innebär att din signal i frekvensled inte får störa utanför ditt frekvensband.

I instruktionerna ligger att de två audiosignalerna ska kommuniceras samtidigt, varför det är rimligt att använda I/Q-modulation. Du bör alltså sända signalen

$$x(t) = x_I(t) \cos(2\pi f_c t) - x_Q(t) \sin(2\pi f_c t),$$

där de ingående beteckningarna har följande innebörd:

- t är tid.
- f_c är en bärfrekvens.
- $x_I(t)$ och $x_Q(t)$ innehåller audiosignalerna som ska kommuniceras. De kan innehålla annat också, som du väljer att inkludera.

Kanalen fördröjer signalen och amplitudskalar den, samt adderar övriga användares signaler. Kanalens utsignal ges alltså av

$$y(t) = Ax(t - \tau) + z(t),$$

där de ingående variablerna har följande innebörd:

- τ är fördröjningen.
- A är amplitudskalningen, som kan vara såväl positiv som negativ.
- $z(t)$ är en summa av alla andra användares signaler på angränsande frekvensband.

Din mottagare måste kunna bestämma fördröjningen och amplitudskalningen. Det innebär att minst en av audiosignalerna bör innehålla en väldefinierad del som är avsedd att bestämma dessa två parametrar.

Uppgift

Laborationen utförs i Matlab. Du får via kursrummet (Kursdokument/Laborationer/Labfiler) ett zip-arkiv, vars filnamn innehåller ditt student-ID. I detta arkiv finns följande filer:

- **xI.wav** och **xQ.wav**

Dessa wav-filer är exakt 5 sekunder långa och innehåller de audiosignaler som ska kommuniceras. Sampelfrekvensen är 20 kHz, och signalernas bandbredd är c:a 5 kHz.

- **TSKS10channel.p**

Detta är kanalen i form av en kompilerad Matlab-fil. Den tar ett argument, insignalen i form av en kolumnvektor, och ger ett svar, utsignalen i form av en kolumnvektor. Utsignalen är något längre än insignalen, men notera att det inte motsvarar fördröjningen, och att denna förlängning inte är garanterad att vara lika lång varje gång. Insignalen ska ha sampelfrekvens 400 kHz, vilket då också är den sampelfrekvens som gäller för utsignalen.

- **parametrar.txt**

I denna fil anges vilket frekvensband du ska använda.

Du ska skriva två matlabfunktioner, en sändare och en mottagare, som kommunicerar de två audiosignalerna över kanalen, och i samband med det identifierar skalningen och fördröjningen, samt kompenserar för dessa justeringar. De två utsignalerna från mottagaren ska vara väsentligen lika med de två insignalerna till sändaren. Sändaren ska ta två argument, de två audiosignalerna och generera signalen som ska skickas över kanalen. Mottagaren får inte använda någon annan information än mottagen signal. De parametrar som du fått i filen parametrar.txt och den pulsform du valt för att bestämma fördröjning och amplitudskalning ska hårdkodas i sändare och mottagare.

Det hela ska kunna anropas exakt på följande sätt:

```
[xI,fs] = audioread('xI.wav');
[xQ,fs] = audioread('xQ.wav');
x = sender(xI,xQ);
y = TSKS10channel(x);
[zI,zQ,A,tau] = receiver(y);
```

Här är `sender` och `receiver` de två matlabfunktionerna som du ska skriva.

Mottagaren `receiver` ska ge följande som svar:

- `zI` och `zQ`
Dessa kolumnvektorer ska vara väsentligen lika med `xI` respektive `xQ`. Specifikt ska de motsvara sampelfrekvens 20 kHz, och ha exakt samma längd som `xI` och `xQ`. Denna längd får vara hårdkodad i dina funktioner.
- `A`
Kanalens amplitudskalning. Den är nollskild, men kan vara såväl positiv som negativ. Detta ska din mottagare ta reda på utifrån den mottagna signalen. *Den ska anges med en decimal.*
- `tau`
Kanalens fördröjning i sekunder. Den är inte större än 500 μ s. Detta ska din mottagare ta reda på utifrån den mottagna signalen. *Den ska anges i μ s med en decimal.*

Du ska alltså inte manuellt ta reda på kanalens fördröjning och skalning och sedan hårdkoda det i mottagaren. Mottagaren ska extrahera det ur den mottagna signalen `y`. Vi ska kunna använda din kod med en annan kanal som använder samma frekvensband, och då ska din mottagare svara med de parametrar som gäller för den kanalen. Att `zI` och `zQ` är väsentligen lika med `xI` respektive `xQ` ska kontrolleras med följande matlabuttryck, som bestämmer signal-brus-förhållandena för de två signalerna i dB. Båda måste vara minst 25 dB.

$$\text{SNRzI} = 20 * \log_{10}(\text{norm}(xI) / \text{norm}(zI - xI));$$
$$\text{SNRzQ} = 20 * \log_{10}(\text{norm}(xQ) / \text{norm}(zQ - xQ));$$

Sändaren ska alltså göra följande:

1. Uppsampla `xI` och `xQ` från 20 kHz till 400 kHz sampelfrekvens.
2. Införa en lämplig pulsform för att mottagaren ska kunna bestämma fördröjning och amplitudskalning.
3. Generera utifrån det den signal som ska skickas över kanalen.

Mottagaren ska alltså göra följande:

1. Filtrera ut önskat frekvensband.
2. Bestämna kanalens fördröjning och amplitudskalning, samt kompensera för dem. Det är lämpligt att först bestämma fördröjningen och kompensera för den, och sedan bestämma amplitudskalningen och kompensera för den.
3. Demodulera den kompenserade versionen av mottagen signal.

Genomförande och redovisning av laborationen

- Det går bra att diskutera och arbeta med uppgiften parvis eller i små grupper, men programkod och rapport skrivs och lämnas in **individuellt**. Kopiering av text, figurer eller programkod från Internet, andra källor eller andra studenter är inte tillåtet.
- Varje kursdeltagare får ett unikt zip-arkiv med några filer enligt ovan. Det hittar du i kursrummet under Kursdokument/Laborationer/Labfiler.
- Labben genomförs i Matlab.
- Skriv en rapport som berättar hur Du har tänkt och gått tillväga för att lösa uppgiften och vad Du har kommit fram till.
 - Rapporten ska vara på maximalt två sidor (dubbelspaltigt format, 10pt typsnitt). Valfri ordbehandlare kan användas. En mall i L^AT_EX finns i kursrummet under Kursdokument/Laborationer.
 - Rapporten måste presentera och förklara relevanta samband som Du har använt. Använd gärna ekvationer. Skriv koncist och noggrant, så att en tilltänkt läsare som tillgodogjort sig kapitlen 2–4 i *Signals, Information and Communications* kan förstå innehållet och reproducera resultatet utifrån Din beskrivning. Inkludera gärna illustrationer eller plottar av relevanta signaler. Eventuella figurer ska i så fall ingå i de två sidorna.
 - Rapporten ska vara fristående från labb-handledningen. Det vill säga rapporten ska kunna förstås av någon som inte har sett labb-handledningen.
 - Rapporten ska skrivas på svenska. Adekvat terminologi ska användas, ingen “svengelska” accepteras. Till exempel heter det ett *sampel*, flera *sampel*.
 - Programkoden du har skrivit ska bifogas rapporten i separata filer (se nedan), och räknas förstås inte in i de två sidorna.
- Rapporten kommer att bedömas enligt den bedömningsmall som finns i kursrummet. Denna mall ger en del tips om vanliga fel som enkelt kan undvikas. Observera att alla punkter måste bedömas med “ja” på rättningsmallen för att rapporten ska godkännas.
- Detta ska lämnas in i kursrummet via Inlämningar, *vid varje inlämning*:
 1. Försättsbladet med ifyllda personuppgifter och underskrift som intygar att du gjort labben själv, inskannad till en PDF-fil.
 2. Rapporten, som en PDF-fil.
 3. All matlabkod som .m-filer. Den ska vara körbar enligt beskrivning ovan. Alltså måste koden för funktionerna `sender` och `receiver` finnas i separata filer, `sender.m` respektive `receiver.m`.
 4. Kopior av dina matlab-filer som .txt-filer. Detta för att Ouriginal (f.d. Urkund) inte hanterar .m-filer.

Vi använder Ouriginal som plagiatkontroll. Plagiat accepteras inte. Ifall det ändå skulle förekomma, så skickar vi det till disciplinnämnden.

- Deadline för inlämning av en betaversion av redovisningen är **fredagen den 13 maj 2022 kl. 17.00**. Vi medger inga undantag från detta. Börja därför arbetet i god tid!

Denna inlämning är inte till för godkännande, utan för att du ska få återkoppling på det du gjort. Rapporten läses av kursens assistenter/examinator, som ger skriftlig återkoppling i Lisam.

- Deadline för slutlig inlämning av redovisningen är **fredagen den 10 juni 2022 kl. 17.00**. I denna version av redovisningen ska alla kommentarer på betaversionen ha adresserats och eventuella fel åtgärdats.
- Så snart rapporterna är bedömda, så får du återkoppling via kursrummets Inlämningar.
- I den händelse att den slutliga versionen av rapporten skulle underkännas, så ges möjlighet till en ny labexamen under hösten. Du får i så fall utgå ifrån en ny uppsättning filer. Observera att i detta fall kommer redovisningen inte att bedömas som en komplettering, utan vi gör då en ny oberoende bedömning av redovisningen. (Exakt information om datum kommer senare, i förekommande fall.)
- Labhandledning sker på plats i schemalagd datorsal, som du anmäler dig till i kursrummet via Anmälan. Det är ett inledande tvåtimmarspass för att komma igång, och ett avslutande tvåtimmarspass för eventuella sena frågor.

Närvaro på labb-passen är inte obligatorisk men vi rekommenderar att Du åtminstone kommer på ett inledande pass.

Om du anmält dig till ett pass, och senare kommer på att du inte tänker gå på det passet, så ber vi dig att avanmäla dig från passet. Detta handlar om att kunna använda våra resurser effektivt.

- Det går utmärkt att genomföra laborationen på en egen, personlig dator. Enda systemkravet är att datorn har Matlab installerat. LiU tillhandahåller en studentversion av Matlab. Eftersom Octave är väldigt likt Matlab, så kan det gå att göra labben i Octave också. Du bör då provköra din kod i Matlab innan inlämning, eftersom vi kör din kod i Matlab som en del av bedömningen.

Rekommenderad arbetsgång

Det är olämpligt att försöka lösa hela problemet på en gång. Försök att dela upp problemet i delproblem, och lös dem var för sig. Följande förslag på arbetsgång utgår ifrån den tanken. Lämna inte ett steg förrän det är klart och fungerar. Först då är det vettigt att gå vidare till nästa steg. Får du problem i ett steg, så kan det vara idé att backa i listan.

I listan nedan återkommer uppmaningen att försäkra dig om att \mathbf{zI} och \mathbf{zQ} är väsentligen lika insignalerna \mathbf{xI} respektive \mathbf{xQ} . Med det menar vi att SNR-kraven på sidan 3 är uppfyllda.

- Börja med att skriva en sändare som I/Q-modulerar de två signalerna och en mottagare som I/Q-demodulerar, och som inte försöker ta reda på kanalens parametrar. Kommunicera direkt från sändare till mottagare utan att blanda in någon kanal, och försäkra dig om att resulterande utsignaler zI och zQ är väsentligen lika insignalerna xI respektive xQ .
- Låt nu utignalen från sändaren tillfälligtvis vara insignal till den kanal du har fått. Försäkra dig om att kanalen inte klagar på att din signal stör intilliggande frekvensband. *Om du i det fortsatta arbetet ändrar något i sändaren, upprepa då detta test för att försäkra dig om att din signal fortfarande ligger rätt till i frekvensled.*
- Komplettera mottagaren med ett inledande BP-filter för det frekvensband du ska använda. Det är här lämpligt att ha en liten marginal till angränsande frekvensband, så att inte kommunikation på angränsande frekvensband kan störa din kommunikation. Försäkra dig om att resulterande utsignaler zI och zQ är väsentligen lika insignalerna xI respektive xQ , när du kommunicerar utan kanal, alltså direkt från sändare till mottagare.
- Justera nu sändare och mottagare i syfte att bestämma kanalens fördröjning. Här ska du i sändaren komplettera signalen med en lämplig pulsform, så att mottagaren kan estimeras fördröjningen med hjälp av korrelation. För att göra detta någorlunda praktiskt rimligt, välj amplituden på denna pulsform så att amplituden hos sändarens utsignal är väsentligen oförändrad jämfört med innan du lade till detta. När du kommunicerar utan någon kanal, så ska mottagaren rapportera att fördröjningen är noll.
- Komplettera mottagaren med kod som korrigerar för fördröjningen. Försäkra dig om att zI och zQ är väsentligen lika xI respektive xQ , fortfarande utan att använda någon kanal.
- Skapa en egen kanal, som fördröjer signalen, så att du har kontroll över det när du utvecklar koden. Detta går att skriva som en en-radare i Matlab. Kommunicera nu med din sändare och mottagare över denna kanal. Prova olika fördröjningar, och försäkra dig om att mottagaren rapporterar rätt fördröjning, samt att zI och zQ fortfarande är väsentligen lika xI respektive xQ för olika fördröjningar.
- Justera nu sändare och mottagare i syfte att bestämma kanalens skalning. När du kommunicerar utan någon kanal, alltså direkt från sändare till mottagare, så ska mottagaren rapportera att skalningen är ett.
- Komplettera mottagaren med kod som korrigerar för skalningen. Försäkra dig om att zI och zQ fortfarande är väsentligen lika xI respektive xQ , utan att använda någon kanal.
- Uppdatera din egen kanal så att den också skalar signalen, så att du har kontroll över det när du utvecklar koden. Kommunicera nu med din sändare och mottagare över

denna kanal. Prova olika kombinationer av fördröjning och skalning, och försäkra dig om att mottagaren rapporterar rätt fördröjning och skalning, samt att z_I och z_Q fortfarande är väsentligen lika x_I respektive x_Q för alla kombinationer. Prova specifikt *både positiva och negativa* skalningar.

- Mottagaren ska inte förvänta sig att mottagen signal har en viss längd. Uppdatera därför din kanal så att den, förutom skalning och fördröjning, också lägger till ett antal extra sampel sist i kanalens utsignal. Detta kan antingen vara ett slumpmässigt antal eller att du kan välja det med en parameter. Försäkra dig om att mottagaren fortfarande rapporterar rätt fördröjning och skalning, samt att z_I och z_Q fortfarande är väsentligen lika x_I respektive x_Q för alla kombinationer.
- Först nu är det lämpligt att kommunicera över den kanal som du har fått. Försäkra dig om att z_I och z_Q fortfarande är väsentligen lika x_I respektive x_Q . Då är det realistiskt att anta att mottagaren rapporterar rätt fördröjning och skalning för den kanalen.
- När det är dags att skriva rapport, se till att den beskrivning du ger där stämmer överens med vad din matlabkod faktiskt gör, och att den beskrivningen är tydlig.

Tips

Det är lämpligt att använda FIR-filer till all filtrering. Anledningen till det är att FIR-filer kan fås med linjär fasgång, vilket innebär att alla delar av signalen fördröjs lika mycket. Närmare bestämt, om filtret har gradtal N , så är fördröjningen $N/2$ sampel. Vi vill gärna att alla fördröjningar ska vara hela sampel. Därför bör N vara ett jämnt tal. Vidare kräver FIR-filer relativt höga gradtal, jämfört med många andra typer av filter. Det kan vara lämpligt att låta gradtalet vara t.ex. 100.

Några användbara matlabfunktioner:

- **upsample**
Detta ökar samplingsfrekvensen genom nollinbakning, vilket komprimerar spektrum i frekvensled uttryckt i normerad frekvens. Det behöver följas av LP-filtrering för att åstadkomma en interpolering mellan de ursprungliga samplen. Notera att detta resulterar i en effektskalning. För att kompensera för denna effektskalning, multiplicera signalen med uppsamlingsfaktorn.
- **fir1**
Konstruerar ett FIR-filer med linjär fasgång. Se sidan 5 i *Short Matlab Manual* för närmare beskrivning.
- **filter**
Filtrerar en signal med hjälp av ett filter skapat med `fir1`. Se sidan 5 i *Short Matlab Manual* för närmare beskrivning.

- **norm**
Bestämmer normen av en signal (vektor). Kvadraten av detta är motsvarande signalenergi.
- **downsample**
Detta minskar samplingsfrekvensen genom att ta bort sampel, vilket expanderar spektrum i frekvensled uttryckt i normerad frekvens. En variant av Poissons summationsformel gäller i detta fall, varför det är lämpligt att det föregås av ett LP-filter, för att undvika vikning.
- **xcorr**
Detta korrelerar två vektorer.
- **findpeaks**
Detta hittar lokala maxima i en vektor. Det kan vara användbart för att hitta en korrelationstopp. Läs på om alla varianter att anropa denna funktion om du tänker använda den.
- **audioread**
Detta läser in en audiofil (exempelvis wav) och returnerar signalen som en kolumnvektor, samt sampelfrekvensen.
- **soundsc**
Detta spelar upp en vektor som ljud.

Dokumentet *Short Matlab Manual* (introToMatlab.pdf) finns i kursrummet under Kursdokument/Laborationer. Det innehåller ytterligare tips kring Matlab. Vidare har Matlab en mycket generös hjälpfunktion.

Notera att vissa av ovanstående funktioner hör till Signal Processing Toolbox. Det enklaste sättet att försäkra dig om att du har denna toolbox är att helt enkelt försöka anropa funktionerna. Så länge Matlab inte svarar att funktionen är okänd, så är allt som det ska.

- Kapitel 2–4 i *Signals, Information and Communications* samt kompletteringsmaterialet behandlar relevant teori.
- Innan Du börjar arbeta med laborationen, lös lektionsuppgift 2-13. Den behandlar I/Q-(de)modulation.

Om du har problem med att få utsignalerna att uppfylla SNR-kraven på sidan 3, så kan följande vara användbart:

- Om du ligger nära 25 dB, men inte riktigt når dit, så är nog det mesta rätt. Se då över dina filter, alltså gränshfrekvenser och kanske prova högre gradtal. Gradtal under 100 är antagligen olämpligt och över 500 ska normalt inte behövas.

- Om du ligger långt från 25 dB, så kan det ha flera orsaker. Det kan bero på felaktig korrigering av skalningen. Det kan bero på felaktig korrigering av fördröjningen. Det kan bero på att I- och Q-delarna har blandats ihop. Det kan bero på att du använder fel bärfrekvens i sändaren och/eller mottagaren.
- Plotta zI mot xI eller zQ mot xQ . Om allt är som det ska, så ska detta ge upphov till en graf som är nära en rät linje med lutning 1. Små avvikelser kan förekomma, utan att det är något större problem. En rät linje med annan lutning än 1 motsvarar felaktigt korrigerad skalning. Felaktig fördröjning och/eller sammanblandning av de två delsignalerna ger upphov till en graf som ser ut som ett slarvigt upprullat nystan.
- Plotta zI och xI mot tiden i samma graf, alternativt zQ och xQ , så att du ser hela signalerna. Om de inte ser väsentligen likadana ut, så har du stora fel. Om de ser väsentligen likadana ut, zooma in i horisontalled så att du ser några tiotal sampel. Ser signalerna fortfarande likadana ut? Här kan du se om signalerna har olika fördröjning. En skillnad i tidsled på ett enda sampel gör att du inte har en chans att nå 25 dB.

Några vanliga fel och hur de undviks

Reproducerbarhet. Kravet på att rapporten ska vara reproducerbar innebär att någon annan ska kunna lösa uppgiften utgående ifrån Din rapport. Det är t.ex. inte acceptabelt är att skriva något i stil med “Jag testade några olika fördröjningar och hittade vilken som var rätt”. För att resultaten ska anses vara reproducerbara, vill vi att Du beskriver hur sagda test gick till. Detta gäller alla parametrar, som Du har testat Dig fram till.

Stavfel. Stavfel är aldrig acceptabla. Enda sättet att helt undvika dem är att korrekturläsa den text Du har skrivit. Detta gäller alltid, även om ordbehandlingsprogrammet Du använder har rättstavningskontroll. Tänk på att det är möjligt att stava fel men ändå få ett ”korrekt” ord. Om Du är osäker på stavningen av något ord, bör Du slå upp det i en ordlista. Svenska Akademiens ordlista (SAOL) är utmärkt. SAOL finns tillgänglig gratis både på Internet och i applikationer för smarta mobiltelefoner.

Typsättning. Tabell 1 visar några vanligt förekommande fel i typsättningen. Om Du skriver rapporten i Microsoft Word, LibreOffice Writer eller liknande så bör Microsoft Equation Editor, LibreOffice Formula Editor eller liknande verktyg användas för att få till en proper typsättning.

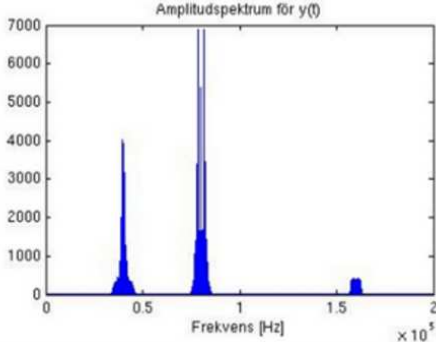
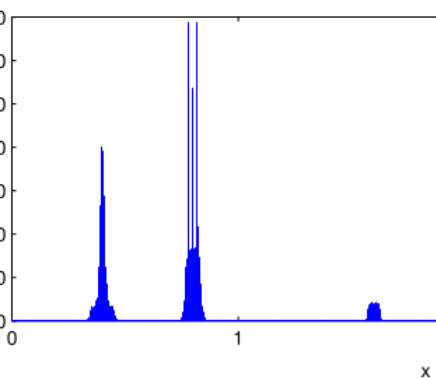
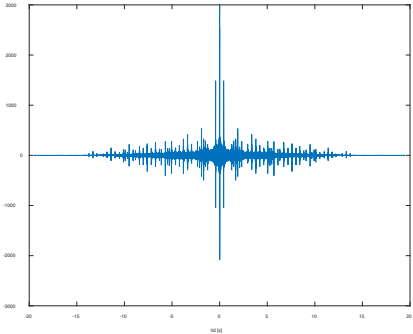
Variabeldefinition. Variabler ska definieras före eller i direkt anslutning till det ställe, där de används. Ett undantag är vedertagen notation, till exempel \sin , \log , π , som kan användas utan definition. Samma variabel får ej användas till olika storheter. Eftersom rapporten är fristående från labb-handledningen, måste eventuell notation från labb-handledningen definieras i rapporten.

Kod. Läsaren av rapporten ska kunna använda Din rapport när hon löser liknande problem, även om det sker i ett annat programmeringsspråk än Matlab. Rapporten ska därför inte innehålla matlabspecifika beskrivningar. Förklara lösningsgången med hjälp av matematiska samband istället.

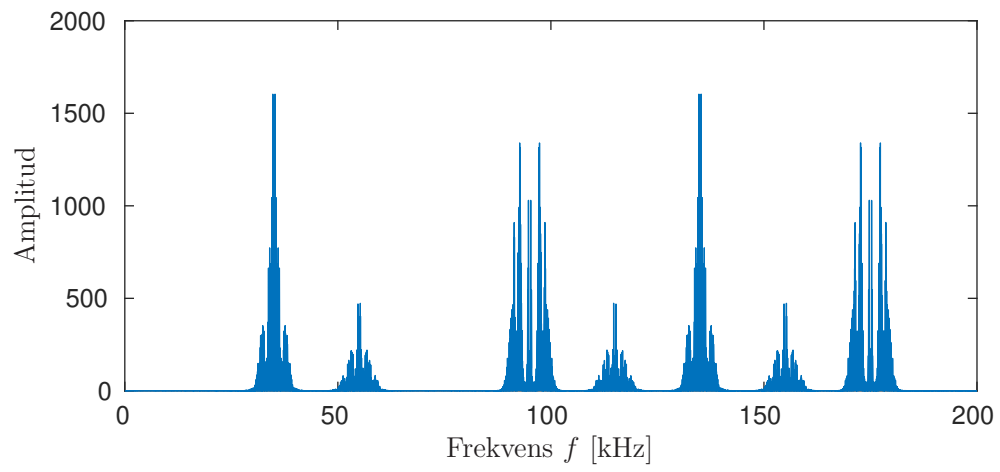
Figurer. Figurer ritas lämpligen i Matlab, eftersom det är det verktyget vi använder. Tabell 2 visar några vanligt förekommande fel i figurerna. Figur 1 visar hur en korrekt formatterad figur kan se ut.

Vad?	Ej OK	OK	Gör så här (om Du använder \LaTeX)
Funktioner ej korrekt typsatta	$\cos(x)$, $\sin(x)$	$\cos(x)$, $\sin(x)$	$\cos(x)$ och $\sin(x)$. Notera att det samma gäller för exempelvis max, min och log.
Nedsänkta tecken är inte korrekt typsatta	$xI(t)$	$x_I(t)$	$x_I(t)$
Fel typsättning av matematisk notation	$fc=100$ Hz	$f_c = 100$ Hz	$f_c=100\text{ Hz}$, $f_c=100$ Hz
Enheter (Hz, sekunder m.m.) typsatta som matematisk notation	$100Hz$	100 Hz	100 Hz
Okonventionella multiplikationstecken.	$2 * \pi * fc * t$	$2\pi f_c t$	$2\pi f_c t$. Notera att vid korrekt typsättning behövs inget tecken för multiplikation
Allt kursiverat.	$y(t) = 2x(t)$	$y(t) = 2x(t)$	$y(t) = 2x(t)$. Använd Equation Editor även i löpande text.

Tabell 1: Exempel på vanliga fel i typsättningen.

Fel	Ej OK	Lösning
<p>Figurer inklippta som bitmappar (ex. .bmp och .jpg), vilket gör att de ser gryniga ut.</p>	 <p>Amplitudspektrum för y(t)</p> <p>7000 6000 5000 4000 3000 2000 1000 0</p> <p>0 0.5 1 1.5 2</p> <p>Frekvens [Hz] $\times 10^5$</p>	<p>Använd pdf, eps eller möjligtvis png med hög upplösning</p>
<p>Axlar saknar gradering eller storhet och enhet</p>	 <p>7000 6000 5000 4000 3000 2000 1000 0</p> <p>0 1 2</p> <p>$\times 10^5$</p>	<p>I Matlab, lägg till figurtitel och text på axlar med kommandona <code>title</code> respektive <code>xlabel</code> och <code>ylabel</code>. För att ändra graderingen på den horisontella axeln på aktiv figur (klicka på figuren för att göra den aktiv), skriv in <code>set(gca,'xTick',gradering)</code> där vektorn <code>gradering</code> innehåller de värden som ska markeras. Använd <code>'yTick'</code> för den vertikala axeln.</p>
<p>Oläsbart liten text eller en figur som är så utzoomad att det är svårt att utläsa relevant information</p>	 <p>3000 2000 1000 0 -1000 -2000 -3000</p> <p>-20 -15 -10 -5 0 5 10 15 20</p>	<p>När figuren sparas i Matlab, ska det aktuella fönstret inte vara för stort. När fönstret skalas upp, så är det bara grafen och inte texten som skalas. Därför blir texten relativt sett mindre.</p>

Tabell 2: Exempel på vanliga fel i figurer.



Figur 1: Exempel på korrekt formatterad figur.