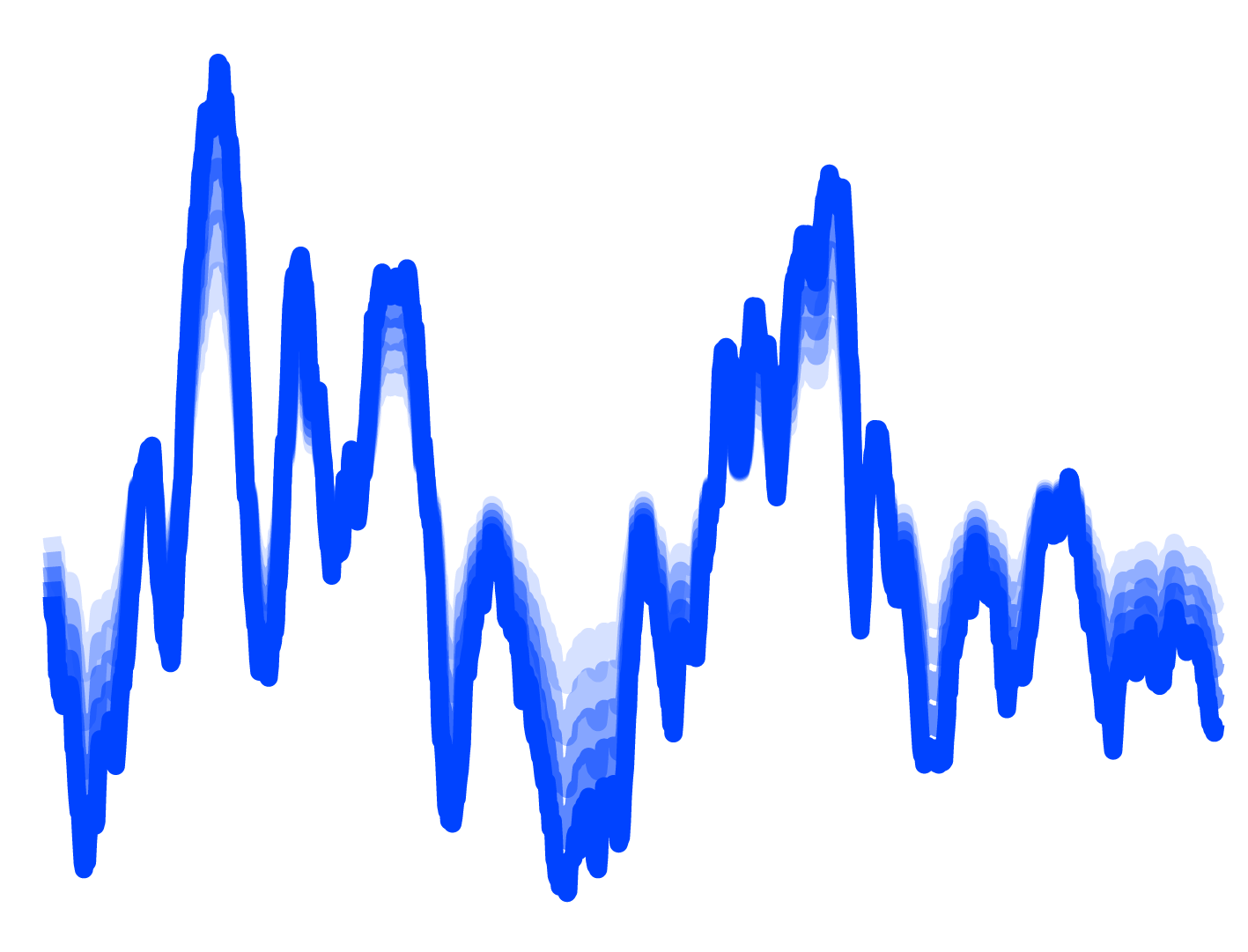
Signaler och System

**LABORATIONSKOMPENDIUM MED HANDLEDNINGAR**

**TILL SAMTLIGA 3 LABORATIONER**

* **Version 11.0 – mars 2023**



Institutionen för Teknik och Naturvetenskap, Campus Norrköping

Klas Tybrandt, Ole Pedersen, mars 2023

|  |  |  |
| --- | --- | --- |
| **Namn** | **Personnr. (år-månad-dag)** | **Studentepost** |
| Magnus Kling | 00-12-01 | Magkl572@student.liu.se |
| Max Wiklund | 01-07-11 | Maxwi824@student.liu.se |

|  |  |  |
| --- | --- | --- |
| **Laboration 1** | **Laboration 2** | **Laboration 3** |
| Komplettera och lämna  in igen: | Komplettera och lämna  in igen: | Komplettera och lämna  in igen: |
|  |  |  |
|  |  |  |
| Godkänd: | Godkänd: | Godkänd: |

LABORATIONSHANDLEDNING

FÖR

SIGNALER OCH SYSTEM

Laboration 1: Faltning och tillämpningar för frekvensanalys

Klas Tybrandt, Ole Pedersen

Version 11.0 / mars 2023

Jag intygar att jag deltagit i genomförandet av alla moment i laborationen:

|  |
| --- |
| Magnus Kling, Magkl572@student.liu.se |
| Max Wiklundh, Magkl572@student.liu.se |

# INLEDANDE ORD OM LABORATIONEN

För kursens samtliga laborationer gäller att du framförallt ska satsa på att förstå frågeställningar och metoder. Ett tips är att du arbetar i din egen takt. Experimentera gärna om du får några uppslag under laborationens gång.

Denna laborationen är indelad i två delar. Den första, kortare delen, behandlar några tillämpningar med differensekvationer och faltning.

I den andra delen får du arbeta med problem som tillämpar Fouriermetoder. I det sammanhanget är det vanligt att man gör någon form av frekvensanalys på tillgängliga data.

Redovisningen av resultat görs på anvisade platser i laborationshandledningen. Platser där grafer ska redovisas är markerade, men man kan även inkludera grafer på andra ställen vid behov.

FÖRBEREDELSEUPPGIFTER

- Läs igenom labhandledningen och relevanta avsnitt i kurslitteraturen innan du kommer till laborationen.

- Uppgift 2: Beräkna den ”teoretiska” lösningen till differentialekvationen, med stegsignalen som insignal.

- För uppgift 8 och framåt: Utred hur en korrekt frekvensaxel skapas med enheten Hz, samt

vilken frekvensupplösning de följande försöken bör få. Redogör för ett generellt uttryck

som du kan använda till alla de följande uppgifterna.

UPPGIFTER SOM INGÅR I LABORATIONEN

* Uppgifterna 1 – 8
* Valbara uppgifter 9 eller 10 (du gör en av dessa)
* Uppgift 11 (sista uppgiften)

Totalt gör du 10 st av laborationens 11 uppgifter.1. Differentialekvationer, differensekvationer och faltning

Antag att du behöver göra en analys av ett system som kan karakteriseras av en differentialekvation av första ordningen. Enklare system kan oftast beskrivas med första ordningens differentialekvationer. Ekvationen eller sambandet mellan insignal x(t) och utsignal y(t) kan t.ex. se ut som:



Du vet också att systemet saknar energi initialt, dvs  om .

1a. Skriv om differentialekvationen som en differensekvation. Använd samplingsintervallet 0.02 sekunder.

1b. Skapa ett MATLAB-program som beräknar de tidsdiskreta värdena y[n] för differensekvationen. Beräkna 40 värden totalt så att du får en tidsaxel från t = 0 till t = 0.8 sekunder. Välj en stegsignal u[n] som insignal. Rita ut sampelvärden för y[n].

1c. Differensekvationens graf konvergerar mot ett gränsvärde. Är gränsvärdet rimligt? Studera den ursprungliga differentialekvationen och fundera på om den ger dig någon ledtråd.

Redovisning:

En bild som visar diagram

Automatiskt genererad beskrivning1c) konvergerar mot 0,25 vilket är rimligt då 0.08/0.02 = 0,25

En bild som visar text

Automatiskt genererad beskrivning

En bild som visar text, brev

Automatiskt genererad beskrivning

2a. Differentialekvationen kan naturligtvis lösas direkt med papper

och penna (förberedelseuppgift). Använd samma insignal som tidigare (stegsignalen) och rita grafen för lösningen y(t). För att lösa diffferentialekvationen ska du använda standardformeln:

2b. Utgående från lösningen y(t) enligt ovan, visa att gränsvärdet för grafen är rimligt.

Redovisning:

En bild som visar diagram

Automatiskt genererad beskrivning En bild som visar diagram

Automatiskt genererad beskrivning

3a. Skriv ett MATLAB-program som utgår från den teoretiska lösningen i uppgift 2 och som ritar grafen då sekunder.

3b. Jämför graferna för den tidsdiskreta- och den tidskontinuerliga lösningen.

3c. Är det så att 3a ger en tidskontinuerlig graf och 1b en tidsdiskret

graf ? Eller är det i praktiken samma sak? Kommentera!

Redovisning:

3c) 3A är diskret men i praktiken är den kontinuerlig jämfört med 1A som är diskret och behandlas som den är diskret.

En bild som visar diagram

Automatiskt genererad beskrivningMan kan också beräkna utsignalen från ett system om man känner insignalen och systemets impulssvar. Du ska nu försöka beräkna det tidigare systemets utsignal genom att använda faltning istället för lösning av differentialekvation / differensekvation.

För att faltningen ska kunna genomföras behöver du först ha tillgång till systemets impulssvar h(t).

Impulssvaret får du genom att använda impulsen som insignal och beräkna utsignalen h(t) med samma integralformel som tidigare:

 (impulssvaret för systemet)

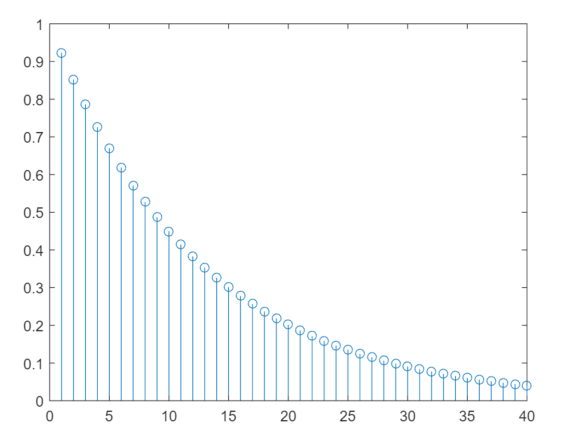
4a. Skriv ett MATLAB-program som skapar ett tidsdiskret impulssvar h[n]. Observera att värdena ska vara tidsdiskreta vilket innebär att du ersätter variabeln t med motsvarande tidsdiskreta uttryck. Samplingsintervall som tidigare. Beräkna 40 värden (motsvarar t = 0 till t = 0.8 sekunder).

4b. Komplettera programmet med faltningsoperationen. Insignalen ska nu vara en stegsignal, som tidigare. Skapa 40 sampel även för insignalen.

Observera att utsignalen y[n] efter faltningen får fler sampel, men det är bara de 40 första samplen av dessa som är relevanta för oss i denna uppgift.

4c. Observera att utsignalen inte har rätt nivå, den är oskalad, jämfört med lösningen till differentialekvationen y(t) eller differensekvationen y[n]. Rita grafen till den oskalade utsignalen.

Redovisning:



En bild som visar diagram

Automatiskt genererad beskrivning 5a. Resonera dig fram till hur faltningsresultatet i uppgift 4 bör skalas om för att

y[n]-värdena ska bli korrekta. Som tips kan du jämföra den tidskontinuerliga

faltningsformeln med den motsvarande tidsdiskreta. Studera samtliga variabler

och jämför variablerna i de båda uttrycken. Något som saknas ??

Tidskontinuerlig faltning:

Tidsdiskret faltning:

5b. Beräkna faltningsresultatet igen genom att använda korrekt skalning (väl

motiverad). Använd samma samplingsintervall som tidigare. Rita en graf.

Redovisning:

6a. Faltning av signal och impulssvar kan t.ex. användas för att manipulera ljud.

Genom att välja ett lämpligt impulssvar kan man åstadkomma olika ljudeffekter,

som t.ex. eko som är enkelt att förstå rent fysikaliskt.

Välj någon av ljudfilerna (.wav) som finns under filkatalogen

S:\ TN \ G \ 015\_Signaler\_och\_System för ett faltningsexperiment.

Ladda in den valda filen till en variabel i MATLAB på följande sätt:

**[y,fs]=audioread('namn.wav');**

Variabeln “y” är nu din ljudvektor, variabeln “fs” innehåller värdet på samplingstakten.

6b. Skapa ett impulssvar som har följande struktur:

h[n] = [1 , 0….0 , 1 , 0…..0 , 1]

Förklaring:

Första samplet har värdet ”ett”, därefter följer ett antal nollor som ska motsvara en lagom tidsfördröjning på 0.3 sekunder (godtyckligt vald), därefter kommer en ny ”etta”, ett antal nollor, och så vidare ……

För att få en fördröjning på 0.3 sekunder måste du ta hänsyn till samplingsintervallet och räkna ut hur många nollor som krävs för att åstadkomma ett tidsintervall på 0.3 sekunder. I MATLAB kan du skapa nollor med funktionen ”zeros (1, antal\_nollor)”. Funktionen har två argument. Det första argumentet talar om att det ska vara en vektor (1). Det andra argumentet är antal nollor.

Utför en faltning i MATLAB mellan ljudfilen och impulssvaret h[n].

6c. Lyssna på resultatet via MATLAB-funktionen **soundsc(y,fs)** .

6d. Hur bör impulssvaret se ut för att faltningen ska ge ett fysikaliskt realistiskt eko? Tips: vad händer med ett naturligt eko och dess styrka, allteftersom?

Redovisning:

2. TILLÄMPNINGAR MED FREKVENSANALYS

Du ska nu studera några inspelade datafiler. Följande exempel visar på ytterligare möjligheter med MATLAB och ljud-/datafiler.

Filerna finns på filkatalogen S:\TN\G\015\_signaler\_och\_system.

2.1 Ljudsignaler

Filen ”piano1.mat” har en inspelning av ettstrukna C från ett piano.

7a. Analys av filen ”piano1.mat”. Börja med att ladda in filen till MATLAB.

**load piano1;** (det här är ett alternativt sätt att ladda in en fil med annat format än tidigare ”.wav”.

Ljudet finns i vektorn ”y1” och samplingstakten finns i variabeln ”fs”.

Spela upp ljudvektorn via datorns ljudkort som är åtkomligt via MATLAB.

**soundsc(y1,fs);**

Du bör höra en närmast ren ton via hörlurarna (använd hörlurar för att inte störa

övriga).

7b. På vilket sätt ändras tonen om du väljer att spela upp det redan inspelade ljudet med en dubbelt så hög samplingstakt (22050 Sa/s) och hälften så hög samplingstakt (5512.5 Sa/s) ?

Förklara vad som händer.

Redovisning:

8a. Ettstrukna C ska vara ungefär 523 Hz. Din uppgift blir att ta reda på om

pianot är stämt. För att avgöra detta måste du göra en frekvensanalys

av ljudfilen och rita upp ett frekvensspektrum.

Skriv lämpliga MATLAB-kommandon och exekvera dessa för att skapa ett frekvensspektrum med korrekt enhet på den horisontella axeln (frekvensaxeln).

När du hittat den högsta (starkaste) frekvenskomponenten i frekvensspektrum

ska du också beräkna ”frekvensfelet”, dvs frekvensupplösningen

Man kan anse att pianot är stämt om frekvensen 523 Hz ligger inom intervallet

8b. Vilket värde har 1:a och 2:a övertonen i ditt frekvensspektrum?

Redovisning:

2.2 Resonansfrekvens för en lyftkran (välj denna uppgift eller uppgift 10)

För att testa hur vibrationskänslig en stor lyftkran är, har man limmat fast accelerometrar på olika platser på kranen. Accelerometrarna ger en signal (acceleration) så fort den delen där sensorn sitter, börjar vibrera. I det här fallet vill man veta om lyftkranens konstruktion är vibrationskänslig för den stora kranmotor som sitter längst ner och som roterar (vibrerar) med ett visst varvtal, beroende på hur fort den körs.

Du ska nu titta närmare på några datavektorer som kommer från en motordriven lyftkran. Datafilen heter ”balk.mat” och finns på samma plats som tidigare datafiler.

Signalerna y11, y12, y21, y22, y31 och y32 (totalt 6 st), är signaler från olika platser på lyftkranen. Vektorn ”t” innehåller tiden och ”fs” anger samplingsfrekvensen.

9a. Utför en frekvensanalys av alla 6 datafilerna. Låt MATLAB rita frekvensspektrat för respektive datafil. Var noga med att det blir rätt frekvensaxel horisontellt med enheten Hz.

9b. En av frekvensstaplarna har den största amplituden. Vilken frekvens har den (motsvarar det motorvarvtal som är mest kritiskt för lyftkranen)?

9c. Om du ritar alla frekvensspektra ovanpå varandra kan du se att det finns ett frekvensområde där den sammanlagda effekten av alla vibrationer blir störst. Vilket frekvensområde är det?

2.3 Trafikradar (välj denna uppgift eller uppgift 9)

Redovisning av samtliga 6 frekvensspektra och slutsatser:

När man kontrollerar hastigheten hos bilar med radar används dopplereffekten, dvs bilens hastighet ger upphov till en förskjutning i frekvens mellan utsänd och mottagen signal. Den mottagna signalen kan beräknas med formeln:



där

v är bilens hastighet i enheten m/s

f0 är utsänd frekvens från radarutrustningen (polisens trafikradar)

c är ljushastigheten (den elektromagnetiska vågens hastighet = här radarsignalens hastighet)

På filen ”radar.mat” finns en signal y (ekot från den aktuella bilen) som kommer från en trafikradarmätning i Sverige. Dessutom finns samplingsfrekvensen fs i filen. Den utsända signalen har frekvensen Hz och ljushastigheten kan sättas till m/s.

10a. Beräkna frekvensspektrum för signalen y i radar-filen. Skapa korrekt frekvensaxel med enheten Hz så att ett frekvensspektrum kan ritas av MATLAB. Den högsta toppen i frekvensspektrat motsvarar ekot från radarn.

10b. Beräkna bilens hastighet i km/tim och uppskatta felmarginalen i dina beräkningar.

Redovisning:

2.4 Jakten på det försvunna telefonnumret

Bakgrund: Du arbetar som teknisk konsult åt polisen. Genom en skarpsinnig insats har en grupp kriminalare lyckats spela in ett telefonsamtal genom att ”bugga” mobiltelefoner för två av den undre världens största vapenhandlare, när de diskuterar en vapenaffär. Kvalitén på ljudet är inte särskilt bra men det gör nu inte så mycket i detta fall. Här vill man istället att du som är teknisk konsult, ska ta tag i problemet med att identifiera **vilket telefonnummer** som slagits av den uppringande parten i sin mobiltelefon.

11a. Du ska via datafilen "phonenumber.mat" försöka att hitta telefonnumret. Filen innehåller den inspelade uppringningen (nummerslagningen) i vektorn y och värdet på samplingsfrekvensen i variabeln fs. Du känner till att en telefonsystemet fungerar på följande sätt:

Varje siffra i telefonnumret svarar mot två frekvenser enligt tabell 1 nedan.

|  |  |  |  |
| --- | --- | --- | --- |
|  | f2 (Hz) | | |
| f1 (Hz) | 1209 | 1336 | 1477 |
| 697 | **1** | **2** | **3** |
| 770 | **4** | **5** | **6** |
| 852 | **7** | **8** | **9** |
| 941 | **\*** | **0** | **#** |

Tabell 1: Sifferkodning i telefonsystemet

Din uppgift blir att ta reda på telefonnumret genom att studera datafilen. Gör analysen på det sätt du tycker är mest lämpligt. Rita frekvensspektrum i mallarna på nästa sida!

11b. Finns det några felkällor i din analys som är väsentliga för slutresultatet, dvs som skulle

kunna äventyra ditt resultat? Kom ihåg att uppgifter av den här arten används i bevisning

(beviskedjor) vid t.ex. en rättegång. Det är alltså viktigt att du som konsult kan stå för resultatet och påtala vilka eventuella fel som kan finnas i resultatet.

Mallar för frekvensspektrum till uppgift 11 samt plats för kortare kommentar:

Redovisning här och nästa sida!

5:e siffran

1:a siffran

2:a siffran

3:e siffran

4:e siffran

6:e siffran

7:e siffran

LABORATIONSHANDLEDNING

FÖR

SIGNALER OCH SYSTEM

Laboration 2: Övningar med Euler/Laplace och z-transform

Klas Tybrandt, Ole Pedersen

Version 11.0 / mars 2023

Jag intygar att jag deltagit i genomförandet av alla moment i laborationen:

|  |
| --- |
| Namn, epost |
| Namn, epost |

INLEDANDE ORD OM LABORATIONEN

Den här laborationen handlar om tidskontinuerliga- och tidsdiskreta system. Du får öva på att studera egenskaper hos system (impulssvar, stegsvar) samt att forma om en differential-ekvation till en mer hanterbar differensekvation.

Redovisningen av resultat görs på anvisade platser i laborationshandledningen. Platser där grafer ska redovisas är markerade, men man kan även inkludera grafer på andra ställen vid behov.

FÖRBEREDELSER

# Läs relevanta avsnitt om Laplace och z-transformen i kompendiet eller fördjupningsboken. Dessutom ingår avsnitt som handlar om digitala filter, vilket du bör kunna åtminstone

översiktligt.

UPPGIFTER SOM INGÅR I LABORATION 2

Obligatoriska uppgifter: 1 – 11

1. TIDSKONTINUERLIGA SYSTEM: Några övningar med Laplace-transformen

# 1. Ett tidskontinuerligt system kan beskrivas med differential-

# ekvationen (systembeskrivningen):

# ; x(0) = 0; y(0) = 0; y´(0) = 0

# 

# 

y(t)

x(t)

Systembeskrivning

# Ekvationen är tämligen komplicerad och ett sätt att beräkna

# utsignalen y(t) som funktion av insignalen x(t) är att lösa ekvationen.

# Det här kräver naturligtvis att du har goda kunskaper i lösning av

# differentialekvationer (vilket inte krävs i den här kursen!).

# En alternativ väg att gå för att analysera vad systembeskrivningen

# egentligen betyder, är att gå via Laplaceverktyget.

# Gör en transformering av differentialekvationen via Laplace-

# verktyget (Laplacetransformen) så att du får ett uttryck för överföringsfunktionen H(s).

Redovisning:

# 2. Överföringsfunktionen H(s), som också är en systembeskrivning, ger dig information om det tidskontinuerliga systemet på liknande sätt som differentialekvationen gör. Men en avgörande skillnad är att det nu går att se helt andra egenskaper för systemet. Bland annat är systemets poler och nollställen mer eller mindre uppenbara.

Ange systemets poler och nollställen.

3. Baserat på vad du vet om pol/nollställe placeringen, skissa på hur

frekvensgången och fasgången ser ut för systemet. Frekvensgångsgrafen får du fram om du analyserar vektorlängderna för nollställen och poler i Laplaceplanet.

På liknande sätt får du fram fasgången genom att analysera vektorvinklarna för nollställen och poler i Laplaceplanet.

Formel för frekvensgången:



Formel för fasgången:





ω



ω

4. Frekvensgång och fasgång kan hanteras i MATLAB på ett relativt enkelt sätt.

MATLAB behöver veta hur systemet beskrivs i termer av täljare och nämnare för överföringsfunktionen (se uppgift 1). Täljaren och nämnaren för överförings-funktionen H(s) ska först skrivas upp på formen:





där n,m = heltal, högsta potensen av ”s” för täljare resp. nämnare

= koefficienter (reella tal) framför resp. täljarterm och nämnarterm

OBSERVERA att termen  eller bara syftar på konstanter eftersom s-termen försvinner!

När täljaren och nämnaren är skrivna på ovanstående form, anger du täljarens och nämnarens koefficienter, som argument till MATLAB-funktionen *bode( ….)*

enligt följande:

*bode([],[]);*

**Skriv in detta i MATLAB och kör funktionen *bode****.* Det bör dyka upp två s.k. Bodediagram på datorskärmen. Den övre visar frekvensgången och den nedre visar fasgången för överföringsfunktionen H(s). Skalorna på tre av fyra axlar är logaritmiska.

5. Det är viktigt att kunna avläsa diagram, som t.ex. frekvensgång och fasgång, på ett korrekt sätt. Utgå från Bodediagrammet i föregående uppgift 4.

Omvandla decibel till ”linjära värden” där det behövs. Observera också att den horisontella skalan anger vinkelfrekvens (vinkelhastighet) med enheten rad/s. MATLAB anger skalan med ”frequency” men det borde rimligen stå ”angular frequency”.

Ange vilket värde de eftersökta storheterna har och fyll i tabell 1 på nästa sida!

Tabell 1: Avläsning från Bodediagrammet för H(s).

|  |  |
| --- | --- |
| STORHET | VÄRDE |
| rad per sek |  |
| Hz |  |
| rad per sek |  |
| H( då rad per sek |  |

6. Slutligen, vad gör systemet H(s), både med amplitud och fas, om insignalen exempelvis är en ljudsignal bestående av många olika toner (frekvenser)?

Du kan tänka dej en situation enligt figuren nedan.

H(s)

Utsignal som är ”bearbetad” av systemet H(s) på något sätt. Hur?

Ljudsignal bestående av ett flertal olika frekvenser, både låga och höga

Dina kommentarer:

2. BILINJÄR TRANSFORM: Laplace övergår till tidsdiskret z-uttryck

Utgå från differentialekvationen **.** När du ska söka en lösning till ekvationen kan du göra det på lite olika sätt. Här följer några alternativ som är tänkbara.

**I.** Givet en viss insignal kan du räkna fram lösningenmed papper och penna om du också känner till startvillkoren (initialvillkoren). Har du lösningen kan du alltid få fram siffervärden på utsignalen.

*Fördel: en metod som går direkt på uppgiften att hitta en lösning, dvs y(t)-värden.*

*Nackdel: kan bli svårt att räkna fram lösningen om differentialekv. är knepig.*

II. Skriv om differentialekvationen till en differensekvation och beräkna värden för

. Du måste veta insignal, initialvillkor och samplingsfrekvens.

*Fördel: Ger en enkel ekvation där y[n]-värden kan räknas fram i en tabell.*

*Nackdel: Hur väljer du bästa samplingstakt (samplingsfrekvens)?*

III. Ta fram Laplaceuttrycket för differentialekvationen och beräkna den inversa Laplace-transformen. Naturligtvis bör du veta vilken insignal du har och initialvillkoren. Den inversa Laplacetransformen är alltså lösningen .

*Fördel: Omvandling differentialekv. 🡪 Laplace är ofta problemfri.*

*Nackdel: Måste därefter beräkna invers-Laplace.*

IV. Ta fram Laplaceuttrycket för differentialekvationen, gör en bilinjär

transformation så att du får en differensekvation. Du måste veta insignal, initialvillkor och samplingsfrekvens.

*Fördel: En lösningsväg som liknar alternativ II.*

*Nackdel: Samma som alternativ II.*

V. Använd MATLABs möjligheter till symboliska beräkningar och beräkna

lösningen med siffervärden. Du måste veta insignal, initialvillkor och samplingsfrekvens.

*Fördel: Snabbt, relativt enkelt.*

*Nackdel: Du vet inte vad som händer i MATLAB. Finns det begränsningar du missat?*

För den enkla differentialekvation som vi har här ovanför i exemplet, skulle man kunna använda i stort sett vilken av alternativen **I -V** som helst. Som en övning skall du nu utföra alla fem varianterna i den här laborationen. Självklart så är vissa metoder att föredra

framför andra för den här enkla ekvationen, men av pedagogiska skäl får du göra samtliga.

**2.1 Fem olika alternativa metoder för att söka lösningen till en enkel differential-ekvation**

För uppgifterna 7-11 gäller att insignalen är,  för  och att samplingsfrekvensen sampel/sekund. Initialvillkor  om .

ALTERNATIV I:

7a. Beräkna lösningen till differentialekvationen manuellt genom att använda en standardformel för lösningen.

7b. Skriv ett MATLAB-program som beräknar utsignalen  för ett

tidsintervall sekunder och rita grafen för .

Redovisning:

ALTERNATIV II:

8. Skriv om differentialekvationen till en differensekvation och

beräkna 50 st värden på  med hjälp av MATLAB. Rita resultatet.

Redovisning:

ALTERNATIV III:

9a. Beräkna  med papper och penna via Laplacetransformering och sedan via invers-Laplacetransformering. Utgå från den ursprungliga differentialekvationen.

9b. Rita grafen för y(t) via MATLAB-kod. Jämför graferna från denna uppgift och uppgifterna 7 och 8.

Redovisning:

ALTERNATIV IV:

10. Via den bilinjära transformationen kan man också nå fram till en differensekvation. Börja med att ta fram uttrycket för H(s) = Y(s)/X(s).

Gör därefter en variabelsubstitution (bilinjär transform) enligt:

 , så att du kan skapa överföringsfunktionen H(z).

Slutligen bör du kunna skriva uttrycket för differensekvationen genom att använda förskjutningsregeln. Testkör differensekvationen i MATLAB och rita

en graf med 100 värden.

Jämför med övriga grafer.

Redovisning:

ALTERNATIV V:

11a. Till sist ska du använda MATLABs möjligheter att

räkna ”symboliskt” för att ta fram utsignalen . Du utgår i det här

exemplet från Laplace-uttrycket  som tidigare tagits fram.

**>> syms s;**

**>> Y=2/(s+ 1)^2;**

**>> y=ilaplace(Y);**

11b. För att få siffervärden från den symboliska variabeln **y** i MATLAB måste du konvertera symboluttrycket till ett numeriskt uttryck. Det görs med funktionen *eval(y)* enligt koden nedan:

**>> for n = 1:1:50**

**>> t = 0.1\*n; % Skapa t-värde**

**>> y\_numerisk(n)=eval(y); % Funk. “eval” utnyttjar t-värdet**

**>> t2(n) = t; % Spara t-värde**

**>> end**

**>> plot(t2,y\_numerisk); % Rita graf**

Jämför med tidigare systemsvar (y-signal)!

Redovisning:

LABORATIONSHANDLEDNING

FÖR

SIGNALER OCH SYSTEM

Laboration 3: Lite om bildfiltrering och adaptiva filter

Klas Tybrandt, Ole Pedersen

Version 11.0 / mars 2023

Jag intygar att jag deltagit i genomförandet av alla moment i laborationen:

|  |
| --- |
| Namn, epost |
| Namn, epost |

INLEDANDE ORD OM LABORATIONEN

Den här laborationen tar dels upp lite om bildbehandlande åtgärder som du kan göra med kunskaper om signaler och system, men också några övningar med adaptiva filter eller det närbesläktade begreppet artificiella neurala nätverk (ANN).

Redovisningen av resultat görs på anvisade platser i laborationshandledningen. Platser där grafer ska redovisas är markerade, men man kan även inkludera grafer på andra ställen vid behov.

FÖRBEREDELSER

Läs avsnitten i kurskompendiet som handlar om adaptiva filter. Det är en fördel om du hunnit titta på något övningsexempel i kompendiet som handlar om adaptiva filter.

UPPGIFTER SOM INGÅR I LABORATION 3

Obligatoriska uppgifter: 1 – 4

Välj mellan uppgifterna 5 – 8 eller uppgifterna 9 – 12

**1. LITE OM FILTRERING AV BILDER I MATLAB**

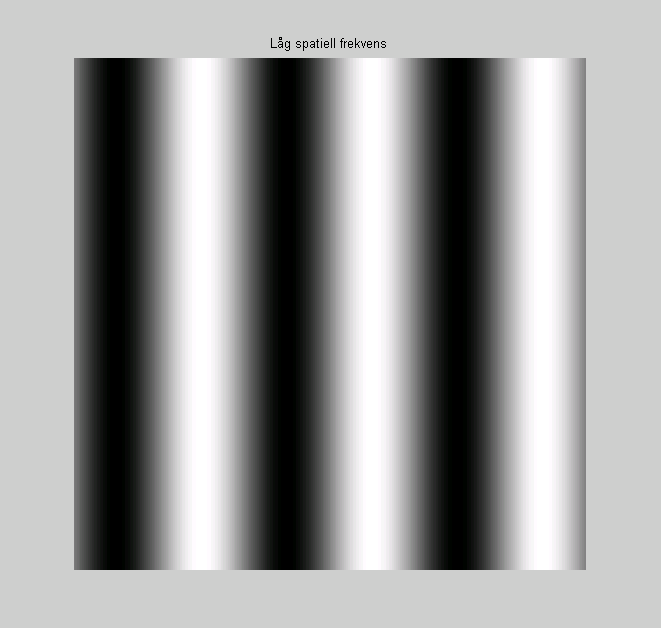
Som du säkert förstår är det inte bara ljudsignaler eller andra ”endimensionella” signaler som kan filtreras. Även en bild, som är tvådimensionell (höjd och bredd), kan filtreras. När man pratar om bildfiltrering så är begreppet frekvens lite annorlunda än begreppet ljudfrekvens. En ljudsignal med hög frekvens motsvarar en hög ton som du kan lyssna på.

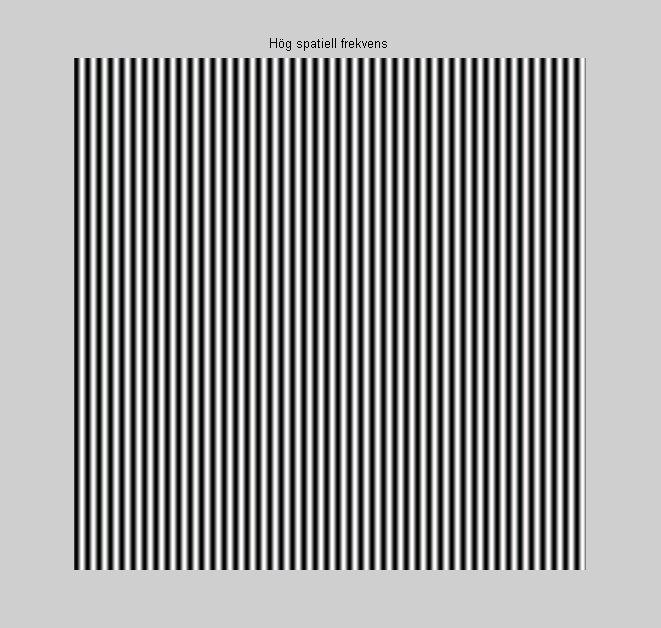
För bilder gäller att en bild kan ha olika *spatiala* frekvenser. Enheten för spatiala frekvenser är **perioder/meter** (perioder per meter) och är ett mått på hur ofta en sinusoidal (sinusformat mönster) upprepas i bilden. Det sinusformade mönstret tas fram ur Fouriertransformen för bilden.

Man kan också speciellt i bildsammanhang använda enheten **perioder/mm**, men också **linjepar/mm**.

En bild med låg spatial frekvens har få linjepar per distansenhet, en bild med hög spatial frekvens har många linjepar per distansenhet.

Titta på följande två bilder (figur 1). Den övre bilden har låg spatial frekvens (horisontellt) medan den nedre bilden har relativt hög spatial frekvens (horisontellt). Vertikalt har båda bilderna samma spatiala frekvens, dvs så vitt vi kan se är den noll.





Figur 1: Spatiala (rumsliga) frekvensen i horisontell riktning

När du filtrerar ljudsignaler har du lärt dig att det kan man se som en faltning mellan insignalens sampel x[n] och filterkoefficienterna (impulssvarets sampelvärden) h[n].

Formelmässigt ser det ut som:



När det gäller bilder så sker filtrering på liknande sätt. Skillnaden är att ”impulssvaret” för bildfiltersystemet nu är tvådimensionellt  och att insignalen, dvs bilden, är tvådimensionell , där varje bildpunkt (pixel) har ett värde. Bilden byggs upp av rader (n-värden) och kolumner (m-värden). Om bilden har 256 rader och 512 kolumner så

gäller att  och att .

Filterkoefficienterna för  är vanligtvis betydligt färre än antalet bildpunkter.

För svart/vita (gråskalebilder) är det vanligt att bildpunkterna har värden mellan 0-255 när de representeras i en dator (digitala bilder). T.ex. kan en bildpunkt på 23:e raden och 59:e kolumnen vara helt svart, vilket betyder att . Om bildpunkten på 45:e raden och 7:e kolumnen är helt vit, betyder det att . Representationen för svart och vitt kan också vara tvärtom (0=svart, 255=vitt). Bildpunktsvärdena mellan 0 och 255 representerar olika nyanser av grått. På så vis kan rimligt många nyanser mellan helt vitt och helt svart, representeras med 8 databitar (.

För färgbilder använder man en lite annan representation eftersom färgbilder (enligt en metod) byggs upp av tre grundfärger (rött, grönt och blått, RGB-färger). En specifik färg fås då genom att mixa ihop tre olika matriser, en röd-matris, en grön-matris och en blå-matris.

Faltning av gråskalebilder skulle formelmässigt se ut som:

 (tvådimensionell faltning)

Tvådimensionell faltning utförs i MATLAB med funktionen *conv2( ).*

Nu är inte den här laborationen en laboration i bildbehandling i första hand utan tanken är bara att du ska få en liten inblick i att signalbehandlingsverktygen kan användas på områden som har med bilder att göra. Du kommer säkert att ha möjligheter att läsa mer om det här i en särskild kurs för bildbehandling.

**1.1 Några MATLAB övningar med bildfiltrering**

Du ska nu försöka filtrera två enkla bilder (bilderna i figur 1) med hjälp av ett lågpassfilter för bilder. Om du kommer ihåg ett av de första LP-filtren för ljud som presenterades i kursen så var det ett enkelt medelvärdesbildande filter med differensekvationen



Medelvärdet här är summan av fem stycken x-sampel, dividerat med antalet. Impulssvaret för LP-filtret är alltså koefficienterna framför varje x-sampelterm, i det här fallet .

När det gäller bilder så kan vi skapa ett LP-filter för bilder på motsvarande sätt. Skillnaden ligger i att ”bildimpulssvaret” bör vara 2-dimensionellt (men det är inte nödvändigt).

1. Du ska nu skapa och rita bilderna från figur 1 med följande kod:

**colormap('gray');**

**for r=1:512**

**for c=1:512**

**B1(r,c)=uint8(128-127\*sin(2\*pi\*3\*c/512));**

**B2(r,c)=uint8(128-127\*sin(2\*pi\*45\*c/512));**

**end**

**end**

**figure(1); imshow(B1); figure(2); imshow(B2);**

1b. Förklara hur formeln för skapandet av bildmatriserna B1 och B2 fungerar. Funktionen *uint( )* i programkoden, skapar ett 8-bitars heltal.

1c. Skapa ytterligare en bildmatris B3(r,c) med samma storlek men med dubbelt

så hög spatiell frekvens som B2. Rita ut denna i *figure(3).*

Redovisning:

2. Bilderna B1, B2 och B3 ska nu filtreras med både ett LP-filter och ett HP-filter. Impulssvaret (filtret) för LP-filtrering skapas med matrisen:

**h\_lp = (1/100)\*ones(10,10)**

Impulssvaret för HP-filtrering skapas med matrisen:

**h\_hp = [1 0 -1; 2 0 -2; 1 0 -1];**  (observera hak-parenteser här!)

Filtrera alla bilderna genom att utföra en 2-D faltning, med

nedanstående MATLAB kod (funktionen *single( )* omvandlar ett flyttal till

enkel precision, tar mindre plats i datorn). Ersätt matriselementet ”**h**” med respektive **h\_lp** och **h\_hp** samt ange de filtrerade bildmatriserna på motsvarande sätt.

**B1\_filt = uint8(conv2(single(h),single(B1)));**

**B2\_filt = uint8(conv2(single(h),single(B2)));**

**B3\_filt = uint8(conv2(single(h),single(B3)));**

2b. Rita alla filtrerade bilder (totalt 6 stycken, dvs LP/HP för tre olika bilder).

Redovisning:

3. Frekvensanalys av bilder kan göras på motsvarande sätt som frekvensanalys av ljud. Eftersom bilder är tvådimensionella använder du kodraden:

**B\_frek = fft2(B);**

där **B** = aktuell bildmatris och **B\_frek** = frekvensmatris för aktuell bildmatris.

För att rita ut ett 2-d frekvensspektrum ska du använda kodraden:

**mesh(abs(B\_frek(1:256, 1:256)));**

där absolutbeloppet ger magnituden av de komplexa talen i frekvensmatrisen. Observera att det bara är relevant att titta på hälften av varje dimension i frekvensmatrisen, därav konstruktionen 1:256 för just denna bildmatris.

Skapa frekvensspektrum för bilden **B2** och rotera figuren i MATLAB:s graffönster (via ”Rotate 3D”-knappen) så att du ser så mycket som möjligt av frekvensspektrum.

3b. Skapa även ett frekvensspektrum för den LP-filtrerade **B2** och jämför med ovanstående spektrum, som är för den ofiltrerade **B2**.

Redovisning:

4. Det är möjligt att på ett enkelt sätt framhäva vissa strukturer i en bild och

undertrycka andra. Bilden (figur 2) visar en fågel som sitter på en gren. Bildens

mest framträdande struktur horisontellt kan sägas vara den gren som fågeln sitter

på.

Men det finns också tydliga vertikala strukturer i bilden som utgörs av kortare

grenar, som pekar uppåt i bilden. Bilden finns på kurshemsidan ”*Bird\_lab3”*

Läs in bilden och visa den på skärmen:

**Bird = imread(’Bird\_lab3.jpg’);**

**imshow(Bird);**



Svart-vit i kompendiet, färg på bildskärmen.

Figur 2: Bild med tydliga horisontella och vertikala strukturer

4b. Du ska nu utföra två filtreringar av fågelbilden. Vid den ena filtreringen ska

horisontella strukturer synas tydligt. Vid den andra filtreringen ska vertikala

strukturer synas tydligt.

De filterkärnor (impulssvar) som behövs för filtreringen ska se ut som **h\_hp** matrisen i uppgift 2. Genom att falta med **h\_hp** gör du en högpassfiltrering i vertikal led. För att filtrera i horisontell led måste du skapa transponatet till **h\_hp**.

Innan filtreringen bör du förbereda fågelbilden genom att göra den till en svartvit bild. I MATLAB görs det med:

**Bird\_black = rgb2gray(Bird);**

**imshow(Bird\_black);**

4b forts …….

Redovisa dina filtreringar genom att klistra in de två filtrerade fågelbilderna och

kommentera hur du gjort filtreringen.

Användbara MATLAB-funktioner: *rgb2gray( ) , conv2( ) , uint8( ),*

*imshow( ).*

Redovisning:

**2. ÖVNINGAR MED ADAPTIVA FILTER OCH A.N.N.-STRUKTURER**

**2.1 Adaptiva filter för att eliminera störningar (välj detta avsnitt eller 2.2)**

Studera det avsnitt i kurskompendiet som handlar om att eliminera störningar ur en signal. Du bör också förstå figur 3 som är en kopia från kurskompendiet.







**+**





**-**

ADAPTIVT SYSTEM



Figur 3: Adaptivt system för att eliminera störningar.

Det väsentliga här är att det adaptiva systemet ”tränas” med en störsignal som liknar den riktiga störningen. Den har samma egenskaper som den riktiga störningen. Men i allmänhet känner man inte till exakt hur den riktiga störningen ser ut. Visste man det skulle det vara enklare att eliminera den direkt genom att helt enkelt göra en subtraktion:



I det adaptiva systemet söks istället en så bra uppskattning av störningen , som möjligt, för att använda den i en subtraktion:

 där 

I följande laborationsuppgift ska du undersöka hur effektivt ett adaptivt system som bygger på en FIR-filterstruktur, kan bli på att eliminera störningar.

5a. Studera programkoden i exempel 13.5 i kurskompendiet samt figur 3 (i denna handledning). Kopiera koden (finns på S: mappen) och klistra in den i

en MATLAB script. Variera antal epoker när du kör koden. Du kan t.ex. börja med att bara köra en epok och titta på resultatet. Öka därefter antal epoker och jämför ifall resultatet blir bättre.

OBSERVERA att det kan ta relativt lång tid att exekvera koden, beroende på datorkapacitet.

5b. Rita grafer på indata och utdata enligt förslaget på den här sidan och

kommentera resultatet.

Redovisning:

Insignal

Insignal + störning

Utsignal

6. Förändra förutsättningarna i den adaptiva inlärningen genom att skapa ett mindre nätverk. Ett mindre nätverk kan skapas genom att förändra följande två

kodrader till ett nätverk med 5 fördröjningselement:

**id = [0 1 2 3 4 5];** och raden **Pi = {0 0 0 0 0};**

Spara tidigare utdata på lämpligt sätt så att du kan jämföra den här inlärningen med tidigare inlärning. Provkör och rita resultat som tidigare.

Redovisning:

Insignal

Insignal + störning

Utsignal

för ett mindre

nätverk

7a. En annan parameter som kan vara avgörande för slutresultatet är den s.k.

”learning rate”-parametern (inlärningskoefficienten). Börja med att återställa antal inlärningsepoker till 30 stycken.

7b. Öka därefter ”learning rate” värdet stegvis från ursprungliga 0.01 till max. 0.30. Kör programmet med de nya förutsättningarna. Redovisa den bästa körningen.

Redovisning:

Insignal

Insignal + störning

Utsignal

8. För att kunna jämföra olika körningar (med olika parametrar) är det bra att ha ett kvalitetsmått, någon siffra som säger om körningen är bra eller dålig.

Hitta därför på ett kvalitetsmått så att du kan jämföra de olika försöken och se vilken körning som ger bäst resultat. Tips: vad menas med bättre/sämre körning? Kan du jämföra insignal med utsignal på ett bra sätt?

När du har tänkt ut ett bra kvalitetsmått ska du testa programmet med egna värden på antal epoker, inlärningsfaktorn eller annat som du anser kan vara relevant för adaptionen.

Redovisa ditt bästa försök, dvs rita grafer (eller bifoga grafer från MATLAB) och ange vilka värden du använt vid den bästa körningen.

Redovisning:

Insignal

Insignal + störning

Utsignal

(bästa körning)

**2.2 Adaptiva filter för att utföra funktionsapproximering**

**(välj detta avsnitt eller 2.1)**

I de följande övningarna får du prova en nätstruktur som mer liknar det som brukar förknippas med A.N.N.

Nätverket du ska använda är ett ”feedforward”-nätverk med olinjära processelement. Inlärningsalgoritmen blir ”backpropagation by error”, eller kortare ”backprop”-algoritmen.

9. Studera exempel 13.6 i kurskompendiet och kopiera in MATLAB-koden från

mappen S:

Kör programmet och rita ut (eller bifoga grafer för) både insignalen och utsignalen.

Redovisning:

Insignal

10. Undersök hur stabilt nätverket är genom att minska dess storlek. Gör detta

Utsignal

genom att ta bort två eller fler processelement i det gömda lagret. Justera nätstorleken via kodraden:

**net = feedforwardnet ( 10, 'trainlm');**

Siffran 10 anger antal processelement i det gömda lagret.

Provkör programmet med olika storlek och rita ut både insignalen och de olika utsignalerna (eller bifoga grafer från MATLAB).

Redovisning:

Insignal

Utsignaler

(rita flera i samma bild)

11. Återställ nätverket till 10 processelement.

Variera därefter antal epoker som nätverket tränas via kodraden:

**net.trainParam.epochs = 20;** (här 20 epoker)

Redovisa några olika resultat på utdata baserat på ett nätverk som tränats olika mycket.

Redovisning:

Insignal

Utsignal

12. Du ska slutligen försöka att skapa ett nätverk som gör en närmast perfekt

approximation. Experimentera med några av de parametrar som är möjliga att variera i MATLAB-koden nämligen *nätstorlek, antal tränade epoker* men också *slutliga felet*, det som anges av kodraden:

**net.trainParam.goal = 0.01;**

Redovisning:

Insignal och utsignal från det funktionsapproximerande nätverket (bästa körning)