

Sampling och tidsdiskreta signaler Labbmoment

I denna laboration ska vi undersöka samplingens påverkan på signaler (kap. 4).

OBS! För att kunna se frekvensinnehåll i de samplade signalerna användes FFT (Fast Fourier Transform) i MATLAB. Spektrum och frekvensanalys med hjälp av FFT kommer vi närmare undersöka i kapitel 8. I detta labbmoment används FFT enbart för att kunna ta fram frekvensinnehåll i den samplade signalen.

1. Sampling av sinusoidala signaler

I denna uppgift ska vi studera vad som händer med sinusoidala signaler vid sampling. Kap. 4-1&4-2 i kursboken har den relevanta teorin. Betrakta MATLAB skriptet Sampling.m som genererar en samplad signal $s[n]$ (kodrad 13). Denna har ett antal parametrar så som amplitud, frekvens, fas och samplingsfrekvensen. I figur 1 kan referens sinussignalen $r(t)$ i röd färg betraktas. I samma figur visas även den samplad signal $s[n]$ med blåa staplar samt den ideellt rekonstruerad sinussignal $rec(t)$ i svart färg. I figur 2 och 3 plottas frekvenssvaret för den samplad signal $s[n]$. Eftersom signalen innehåller enbart reella värden plottas bara den positiva frekvensaxeln mellan 0 och π radianer. I figur 2 normaliseras frekvensaxeln till f/f_s och därför blir x-axeln mellan 0 och 0.5. I figur 3 anges frekvens i Hertz mellan 0 och $f_s/2$.

I detta experiment undersöker vi om Shannons samplingsteorem håller. Ändrar parametern f (kodrad 10) eller f_s (kodrad 11). Vad händer när f är mindre, lika eller större än $f_s/2$? Testa även att välja f och f_s så att signalens frekvens viks mer än en gång. Vid val av f och f_s då vikning uppstår, överensstämmer den teoretiskt beräknat vikta frekvensen med den frekvens vi får i frekvensplotten?

OBS! Om du zoomar in i figuren 3 och toppen inte riktigt sammanfaller med den angivna frekvensen har det att göra med att denna frekvens ej faller exakt på sampel i frekvensdomän (mer om detta senare i kursen, kapitel 8).

2. Diskreta Fourier Serier

I de fall då vi har tillgång till det matematiska uttrycket för funktionen $x(t)$ som är en periodisk signal används integralen i ekv. 3.27 (s.95 i kursboken) för att beräkna a_k koefficienterna. I verkligheten är tyvärr detta inte ofta fallet. Många gånger har man tillgång till endast en samplad signal $x[n]$. Om denna innehåller en periodisk signal kan a_k koefficienterna skattas med hjälp av den Diskreta Fourier Serier Analysen som i detta fall kommer vara en ändlig summa enligt

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{N} kn}, k = 0, 1, \dots, N-1$$

I skriptet CoeffConvergence.m demonstreras hur skattningen för den valda a_k koefficienten påverkas av valt antal sampel N över en hel period för en fyrkantsvåg. Man kan se att då N ökar, konvergerar estimatet till den teoretiskt beräknad värdet.

Din uppgift blir att först undersöka den givna exempel och sedan utifrån denna modifiera koden så att exemplet anpassas till koefficienterna för triangelformad signal samt minst en till valfri form. Antar att här bör du välja samma val du gjort i labbmomentet för Fourier serier. Observera att exempel estimerar bara koefficient a_1 . Det är lämpligt att du börjar testa olika värden på k i exemplet.

Hur stort N behövs vara för att du ska anse att estimatet har konvergerat till dess teoretiska värde? Blir det samma värde på N för de olika a_k koefficienter i fyrkantstvågen när du anser att de har konvergerat? Finns det något mönster på värdet N när k ökar för samma vågform? Blir det samma värde på N för samma a_k koefficient fast för olika vågformer? Ge en liten sammanfattning utifrån dina svar på föregående frågor.

3. Inspelning av den egna rösten

Mac dator:

Tyvärr stöds inte Data Acquisition Toolbox, så inspelning av ljud måste ske via extern mjukvara. Använd sedan `audioread()` för att läsa in data in i MATLAB från audiofilen. Alternativt samarbeta med någon som har en PC dator.

PC dator:

MATLAB har möjlighet att samla in mätdata via Data Acquisition Toolbox. I detta experiment ska MATLAB sampla ljudet från omgivningen via datorns mikrofoner. För att detta ska fungera måste ett add-on installeras. Följ anvisningarna längre ner i dokumentet.

Skriptet `AudioSignal.m` spelar upp ljudet, plottar tidssignalen och gör frekvensanalysen efter att all data blev inspelad. För att veta när man ska börja prata skrivs ut *start* i Command Window så man vet när insamling av data har startat samt skrivs ut *slut* i Command Window när inspelningen avslutats.

Innan ni kör skriptet, får ni kolla att `dev = d(2)` (kodrad 15) har den index där mikrofonerna detekterades av MATLAB. Med `s.Rate=8000` (kodrad 20) bestäms samplingsfrekvensen (Hz) och `s.DurationInSeconds = 4;` (kodrad 21) väljer ni längden på hur länge ljudet samplas in (sec).

OBS! För Mac, ni kan byta ut första delen av skriptet (t.o.m. kodrad 30) med egen kod där ni läser in data från ljudfilen med `audioread()` och tilldelar variablerna `audio`, `fs` och `t` korrekt data.

Undersök vad är den lägsta och högsta samplingsfrekvensen ni kan åstadkomma på datorn. MATLAB kommer att ge ett felmeddelande om ni underskridit eller överskridit den möjliga samplingshastigheten där den lägsta/högsta möjliga specificeras. Väljer ni samplingsfrekvens som finns mellan min och max kommer MATLAB ge ett varningsmeddelande att den närmaste möjliga samplingsfrekvensen har valts om ni inte angivit den stödda hastigheten. På min nuvarande dator är minimum 8000Hz och maximum 192000Hz. Vilka gränser får ni fram?

Välj en lämplig längd på inspelningstiden gärna något tiotal sekunder för att kunna fånga tillräckligt med ljuddata då ni säger en mening och/eller spelar upp musik. Detta gör ni med den lägsta och högsta (om minnet tillåter, annars gå ner något i samplingen) samplingshastigheten. Kan ni höra skillnaden i uppspelningen? Hur ser frekvensinnehållet ut?

Ett annat experiment kan vara att ni testar vissla med samma eller varierande ton (gå från låg till hög eller från hög till låg ton). Hur ser spektrat ut för en sådan signal? (Här gäller det att man inte har allt för hög samplingshastighet.)

Vad händer om `fs` ändras i `soundsc(audio, fs)` (kodrad 39). Hur låter det om `fs` minskas eller ökas från dess ursprungliga värde?

4. Redovisning

Labbmomentet redovisas genom att du tillhandhåller alla modifierade skriptfilerna. Observera att dessa ska vara fullt körbara utan några fel. Dessutom skal frågorna besvaras och sammanställas i en mindre labbrapport. Studenter med Mac datorer kan ha lite problem med att besvara alla frågor i den sista uppgiften. Ni gör de ni kan om ni väljer att lösa uppgiften på Mac-dator.

Extra info om sampling (och rekonstruktion) i verkligheten

Nedan kan ni se ett typiskt blockschema för hela kedjan där både sampling av insignalen och rekonstruktion av utsignal förekommer. Antivikningsfilter tar bort frekvensinnehåll över $f_s/2$ och rekonstruktions filter jämnar ut signalen. Gällande uppgift 3 behöver ni därför inte bry er om vikningseffekten i den samplade signalen. Sampling hanteras av ljudkortet som ser till att frekvenserna över $f_s/2$ filtreras bort. Dock bör ni vara medvetna om att vid val av samplingsfrekvensen så begränsar ni även frekvensinnehållet i signalen till maximalt $f_s/2$.

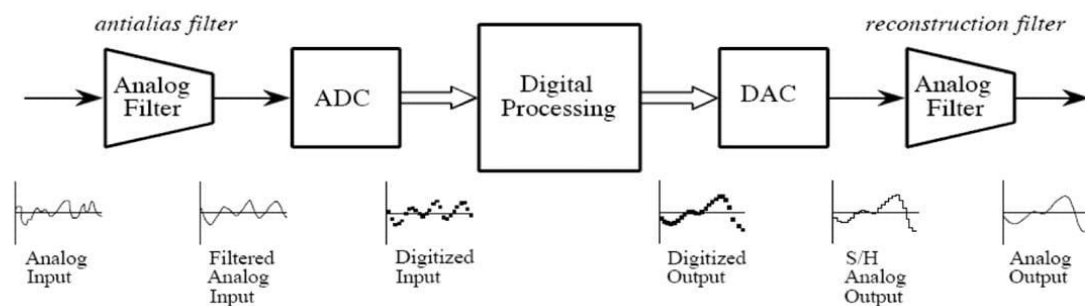
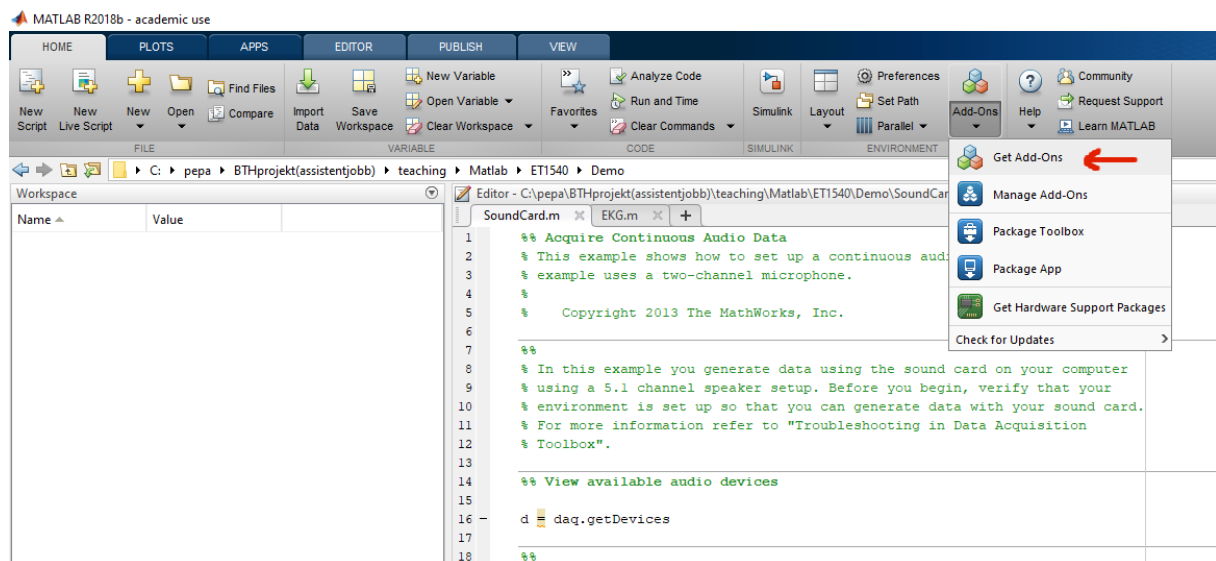


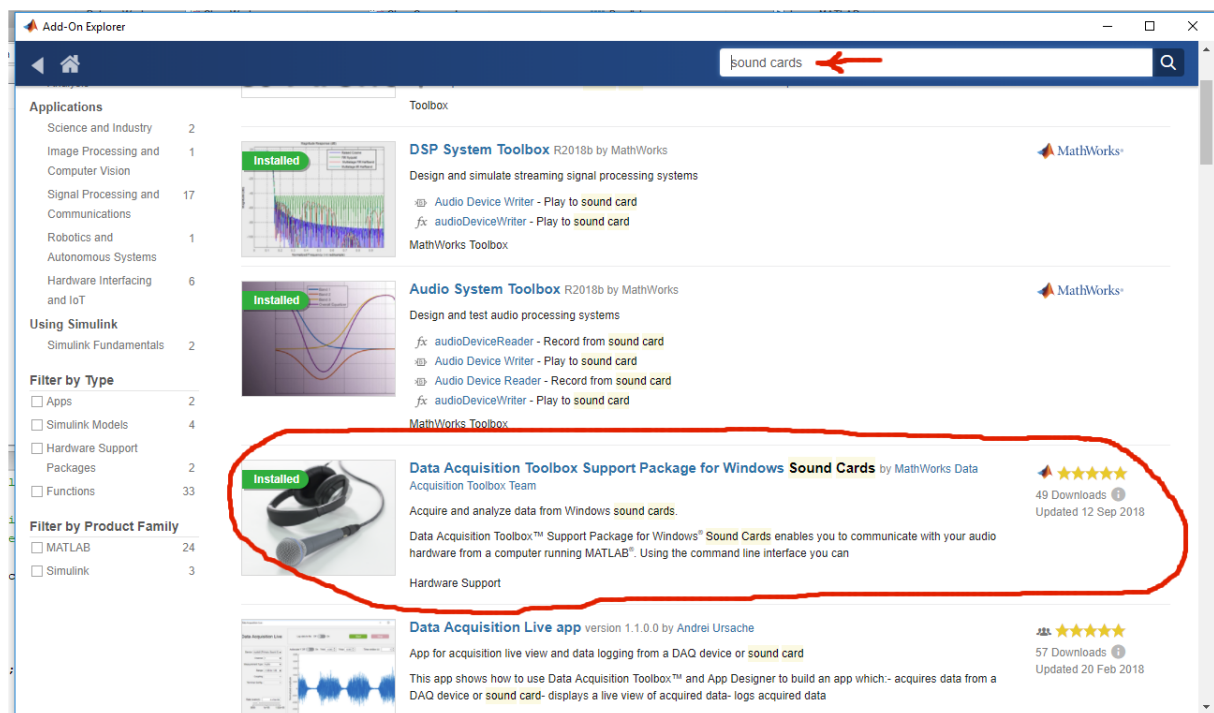
Figure 1: DSP Block Diagram

Add-on installation:

Under Home, tryck på Add-ons och välj Get Add-Ons



Sök på *sound cards* och välj *Data Acquisition Toolbox Support Package for Windows Sound Cards*



Om korrekt installerat bör MATLAB detektera ljudkortet med `daq.getDevices`

```

Command Window

>> daq.getDevices

ans =

Data acquisition devices:

index  Vendor  Device ID  Description
-----
1      directsound  Audio0    DirectSound Primary Sound Capture Driver
2      directsound  Audio1    DirectSound Internal Microphone (Conexant ISST Audio)
3      directsound  Audio2    DirectSound Primary Sound Driver
4      directsound  Audio3    DirectSound Speakers 2nd output (Conexant ISST Audio)

fx >>

```