

En jämförelse av *Short-Time Higher-Order Chirp-Fouriertransformen* och *Computed Order Trackingmetoden*

Referat

Det finns ett konstant behov av förbättringar inom fältet för vibrationsanalys av roterande maskineri. En del tekniker har funnits redan länge och är således mycket beprövade. Dessa metoder kommer ibland något till korta vid hanteringen av icke-stationära signaler. Denna typ av signal förekommer i nästan all slags maskineri, för att inte tala om annorstädes, i dagens läge, vilket innebär att behovet av en korrekt hantering av ovannämnda signaler är stort.

I detta arbete kommer en jämförelse mellan en gammal och beprövad metod – *Computed Order Tracking* – och en nyligen utvecklad teknik – *Short-Time Higher-Order Chirp-Fouriertransformen* att ses.

Teknikernas amplituduppskattningsförmågor kommer att mätas med ett stort antal olika simulationer och jämförelsemetoder. Till exempel kommer uppskattningen att mätas för konstanta och varierande amplituder och tre olika variationer av den momentana frekvensen. Dessutom kommer signalerna att utsättas för olika grader av störning.

Parametermöjligheterna för båda teknikerna kommer att gås igenom iterativt, uppskattningen kommer att mätas och sparas. Då alla parameterar undersökts kommer de bästa att väljas för vardera teknik, varefter utförandena jämförs. Detta torde möjliggöra ett undvikande av partiskhet i väljandet av parametrar. Ett flertal uppskattningsfel kommer också att beräknas, för att försöka undvika att resultaten blir partiska, vilket kunde ske ifall endast en felberäkning användes.

Innehållsförteckning (ungefärlig, ändras kanske)

1. Introduktion

2. Alternativ i dagsläget

- 2.1 Fouriertransformen
- 2.2 Short-Time Fouriertransformen
- 2.3 Wigner-distributionsfunktionen
- 2.4 Gabortransformen
- 2.5 Wavelettransformen
- 2.6 Order Tracking
- 2.7 Snabb fouriertransformsbaserad digital omsampling
- 2.8 Kalmanbaserad Order Tracking
- 2.9 Tidsvarianta diskreta fouriertransformen
- 2.10 Hybrid Order Tracking

3. Teoretisk analys

- 3.1 Chirpsignaler
- 3.2 Fouriertransformen
- 3.3 Short-Time Fouriertransformen
- 3.4 Order Tracking
- 3.5 Frekvensspektrum vs. orderspektrum
- 3.6 Synkron sampling
- 3.7 Syntetiserad synkron sampling
- 3.8 Short-Time Higher-Order Chirp-Fouriertransformen
 - 3.8.1 Chirp-Fouriertransformen
 - 3.8.2 Short-Time Higher-Order Chirp-Fouriertransformen

4. Simulering, resultatinsamling och felkalkyl

- 4.1 Simulering
- 4.2 Resultatinsamling
- 4.3 Felkalkyl

5. Resultat och jämförelse

- 5.1 Resultat

6. Slutledning

7. Referenser

Förkortningar

COT	-	Computed Order Tracking	
CFT	-	Chirp-Fouriertransform	
DCFT	-	Diskret Chirp-Fouriertransform	
DFT	-	Diskret Fouriertransform	
FFT	-	Snabb Fouriertransform	
FT	-	Fouriertransform	
IFFT	-	Invers Fouriertransform	
Kalman	-	Kalmanbaserad Order tracking	
MAE	-	Mean Average Error	- medelfel
MRE	-	Mean Relative Error	- relativt medelfel
OT	-	Order Tracking	
RMSE	-	Root Mean Square Error	- roten av kvadrerat medelfel
STCFT	-	Short-Time Higher-Order Chirp-Fouriertransform	
STFT	-	Short-Time Fouriertransform	
TF	-	Time-frequency	
TFT	-	Time-frequencytekniker	
TVDFFT	-	Tidsvariant Diskret Fouriertransform	
WDF	-	Wignerdistributionsfunktion	

Figurer

1. Andra ordningens chirp och dess linjära frekvensvariation
2. Tredje ordningens chirp och dess kvadratiska frekvensvariation
3. Fjärde ordningens chirp och dess kubiska frekvensvariation
4. Frekvensspektrum med tydlig topp
5. Orderspektrum med klarare förhållande mellan axelhastighet och amplitud
6. Order track av en tidigare identifierad 12:e order

Tabeller

1. Andra ordningens chirp, konstant amplitud, ingen störning
2. Tredje ordningens chirp, konstant amplitud, ingen störning
3. Fjärde ordningens chirp, konstant amplitud, ingen störning

1. Introduktion

Analysen av vibrationer från roterande maskineri är en viktig del av prediktiva underhållsprogram inom industrin. Den används för att övervaka slitage och för att upptäcka defekter i rörliga delar, vilket möjliggör reparation av maskineriet före det havererar och minskar således drifts- och underhållskostnader.

En viktig aspekt av vibrationsanalys är amplituduppskattning. Två olika tekniker, för att uppskatta amplituder för icke-stationära signaler med variabel frekvens, kommer att undersökas: *Computed Order Tracking (COT)*, som är väletablerad inom branschen, och *Short-Time Higher-Order Chirp-Fouriertransformen (STCFT)*, som relativt nyligen utvecklades och är ganska okänd. Den äldre tekniken, COT, har ett antal problem, fall där den nyare tekniken potentiellt kunde fungera bättre. Deras utföranden kommer därför att jämföras och bokföras för ett flertal olika fall. För att få en så grundlig jämförelse som möjligt och för att undvika partiskhet i väljandet av parametrar, kommer så många olika fall som möjligt att beaktas. Denna process kommer att förklaras mer detaljerat längre ner i texten.

Till att börja med kommer dagslägets alternativa metoder att gås igenom, följt av teorin som jämförelsen och teknikerna baseras på. Dessa kapitel – *Alternativ i dagsläget* och *teoretisk analys* – behandlar koncept som; grunderna i *Order Tracking*, *syntetiserad synkron sampling*, samt STCFT-teori. Därpå följer ett kapitel över simuleringsproceduren, resultatinsamlingen, samt även felkalkyleringsformlerna. Därefter finns ett slutlednings- och slutsatskapitel.

Eftersom ingen publicerat en liknande undersökning eller jämförelse kan detta arbete definitivt sägas vara bland de första i sitt slag. Det enda publicerade arbete som gjorts längs samma linjer och som överhuvudtaget tangerar denna jämförelse är några undersökningar där metoder av samma slag jämförs med metoder som i grund och botten är varianter av samma teknik. Till exempel föreslog och jämförde *X.Xia* den *diskreta Chirp-Fouriertransformen (DCFT)* med sig själv för diverse olika fall [6]. Denna transform är inte helt samma som STCFT:n som behandlas i detta arbete, men den är tillräckligt nära besläktad för att förtjäna ett omnämnande här. Likaledes jämförde *Bossley, Mckendrick, Harris* och *Mercer* i sin undersökning endast olika COT-metoder [8]. Detta bara understryker påståendet om att denna jämförelse är den första i sitt slag.

2. Alternativ i dagsläget

I dagens läge finns det ett ganska stort antal *time-frequencytekniker (TFT)* att välja mellan. De har alla sina tillämpningsområden och valet av teknik är således, alltsom oftast, beroende av vad man vill åstadkomma och vilken komplexitetsnivå man strävar efter. I det här fallet kommer en signal vars karakteristika är tidsvarianta att beaktas. Signalen är, med andra ord, icke-stationär.

2.1 *Fouriertransformen*

Den icke-stationära egenskapen innebär i praktiken att, till exempel, den traditionella *fouriertransformen (FT)* inte kan användas, då den endast kan behandla signaler som är stationära eller ungefärligt stationära korrekt. Detta kommer sig av att transformen har en statisk kärna, vilket leder till att den inte kan anpassa sig till förändringar i frekvens eller amplitud. [9]

2.2 Short-Time Fouriertransformen

Short-Time Fouriertransformen (STFT) är aningen bättre på den här fronten. Den har också en statisk kärna, men för att kompensera använder den sig av en fönsterfunktion för att dela upp signalen i mindre bitar. På det här sättet kan transformen till en viss mån runt begränsningen den statiska kärnan förorsakar. Fönstringen innebär dock att man blir tvungen att prioritera mellan att ha en bra *tids-* eller *frekvensresolution*. Ett mindre fönster gör det möjligt att vara mer exakt gällande den exakta *tidpunkten* då en förändring sker i *frekvensinnehållet*, medan ett större fönster gör det möjligt att mer exakt identifiera frekvensen i fråga. Problemet med den här tekniken är dock att transformen, som sagt, använder sig av samma kärna som FT:n och kan därför inte följa förändringar inom fönstret, vilket gör den olämplig i detta arbete. [9]

2.3 Wigner Distributionfunktionen

Ytterligare ett exempel är **Wigner Distributionfunktionen (WDF)**, som ursprungligen föreslogs av Eugene Wigner på 1930-talet. Denna TFT är vanligen mycket enkel att tolka och analysera, men den har en väldigt hög beräkningskomplexitet och använder sig av en autokorrelationsfunktion, som, speciellt för flerkomponentssignaler, ger upphov till ett stort antal **cross-termer**. Funktionens kärna har också svårigheter att följa icke-linjära variationer i den momentana frekvensen. Även om WDF producerar mycket bra resultat och har ett antal bra egenskaper, gör cross-termproblemet, den höga beräkningskomplexiteten och kärnproblemen den till ett dåligt val här.

2.4 Gabortransformen

En teknik som, sett till egenskaperna, är så gott som WDF:s motsats är *Gabortransformen* (GT). Den liknar STFT såtillvida att den spjälkar signalen med hjälp av en fönsterfunktion varefter den utför en FT på innehållet i fönstret. GT:n har en låg beräkningskomplexitet och saknar **cross-termer**. Dess problem ligger i de svårtydda och något oklara resultaten den producerar. På grund av detta har GT:n i praktiken mestadels använts – i industrin och av övriga signalbehandlare – till att filtrera ut **cross-termerna** från WDF:en. Denna hybridtransform kallas *Gabor-Wigner* och kombinerar de bästa egenskaperna från vardera transform, men innefattar fortfarande beräkningskomplexiteten från WDF:en och är således olämplig.

2.5 Wavelettransformen

Wavelettransformen spjälkar även den upp signalen i mindre bitar. Eftersom den kan göra detta med ett fönster vars bredd i tidsplanet är beroende av frekvensen, vilket i praktiken innebär att den kan följa **time-frequencyvariationer (TF)** mycket bättre än STFT, men producerar dock märkbara fel i amplituduppskattningen. Detta har visats för, till exempel, morletwaveleten. Ytterligare ett problem här, är att man måste veta något om signalen och dess egenskaper för att kunna välja den bästa waveleten, då de alla har olika karakteristika. Detta, tillsammans med amplituduppskattningsfelen leder till att *wavelettransformen* inte är lämplig att tillämpas inom vibrationsanalys. [9]

2.6 Order Tracking

De ovannämnda exemplen har inte direkt haft något att göra med **Order Trackingmetoder** (OT) eller deras avkommor. Om genomgången av denna familj av tekniker börjas från den äldsta medlemmen finnes *FT-baserad OT*. Då OT är mycket nära besläktad till COT-tekniken som används i denna jämförelse, kommer den att beskrivas mer detaljerat i den *teoretiska analysen*. Tills vidare räcker det dock att nämna att OT använder sig av FT:ns, eller snabba FT:ns (FFT), kärna och har således alla problem relaterade till en statisk kärna och därför

också olämplig. [1]

2.7 *Snabb fouriertransformsbaserad digital omsampling*

Ett annat alternativ skulle kunna vara en digital omsamplingsteknik med en FT- eller FFT-bas. En dylik teknik skulle först samla data med konstant hastighet (tidsmässigt) och samplar sedan om med konstant vinkelhastighet. Det här tillvägagångssättet är definitivt en förbättring, men gör, förutom att den låter FFT:n följa icke-stationära frekvenser, inget för att lösa problemet med en statisk kärna. Icke-stationära amplituder förorsakar alltså fortfarande problem. Teorin runt den här metoden kommer att behandlas mer i detalj i den *teoretiska analysen*. [1]

2.8 *Kalmanbaserad Order Tracking*

Kalmanbaserad OT (Kalman) kunde ha använts som andra teknik. Den har ganska långt överkommit de ovan nämnda begränsningarna och klarar alltså av både icke-stationära frekvenser och amplituder. Dock är den onödigt komplex och innehåller ett antal parametrar vars värden kräver en hel del erfarenhet för att väljas rätt. Om parametrarna inte är optimala lider slutresultatet. På grund av att det fanns begränsat med tid kunde denna erfarenhet inte skaffas, vilket ledde till att denna metod valdes bort. Det samma gäller *Vold-Kalman OT* och i förlängningen allt baserat på *Kalman*. [1]

2.9 *Tidsvariant diskret fouriertransform*

Ytterligare finns även den *tidsvarianta diskreta fouriertransformen* (TVDFFT). Den är ett specialfall av *chirp-z transformen*, vars kärna är tidsvariant i fråga om *frekvens* och *dämpning*. TVDFFT:n använder sig av samma kärna, men håller *dämpningen* tidsinvariant. Denna teknik använder sig av mycket av det som är bra med *omsamplingstekniker* medan den också lättar på *beräkningskomplexiteten* avsevärt. Dock antar denna teknik att amplituden är konstant, eller i alla fall varierar mycket långsamt, och är på grund av det olämplig för vissa av testerna i denna jämförelse. [1]

2.10 *Hybrid Order Tracking*

Utöver detta finns ännu en teknik kallad *Hybrid Order Tracking*. Syftet med denna har främst varit att förbättra upptäckandet av varvmätarpulser vilka är kritiska för *OT-omsamplingstekniker*. På sätt och vis är målet med den här tekniken att åstadkomma en pulsmedvetenhet som kan tävla med den traditionella OT-teknikens. Referensarbetet [3] nämner att man använt en STFT-bas för att göra sin OT, vilket innebär att ganska långt påminner om den teknik som använts här. Den specialbyggda hårdvaran som behövs för att producera den förbättrade pulsmedvetenheten var dock inte tillgänglig. [3] *Hybrid OT:n* skulle ha varit ett bra val, men avsaknaden av en del av de behövliga verktygen innebar att den i slutändan inte var möjlig att använda.

För den som intresserar sig för dylikt är *referens [1]* en väldigt bra samling grundläggande teori över de ovan listade OT-teknikerna. Om, som i fallet *Hybrid OT*, ingen annan specifik referens använts.

3. Teoretisk analys

3.1 Chirpsignaler

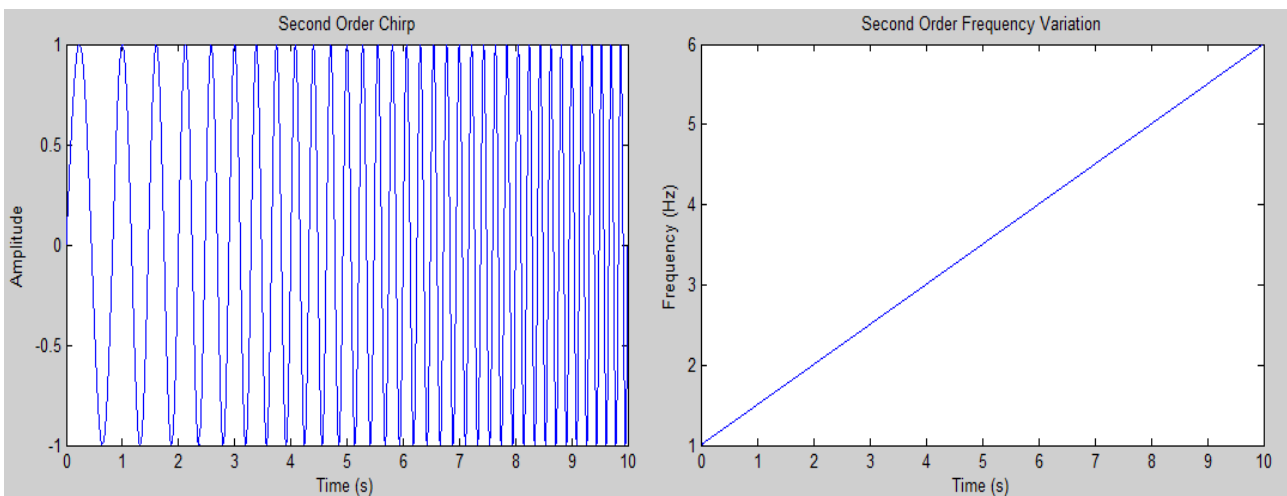
En *chirp* är en sinusoid vars frekvens varierar med tiden. Frekvensen är, med andra ord, *icke-stationär* till skillnad från, till exempel, en *sinusvåg* vars frekvens är *stationär*. *Chirparna* som använts i detta arbete kommer att presenteras i en *exponentiell form*. Detta inverkar inte på deras egenskaper, men kommer att göra det lättare att se hur de samverkar med de olika *transformkärnorna*.

Tekniskt sett finns det två olika typer av *chirpar*. Den ena klassen har en *linjärt varierande* momentan frekvens och kallas helt enkelt *chirpar*. Den andra klassen har en *icke-linjärt* varierande momentan frekvens och kallas *chirpar av högre grad*. Då termen *chirp* används från och med nu, avses alla slags typer av *chirpsignaler*. Den är således en samlingsterm och om en specifik typ av *chirp* behandlas kommer minst graden av variation att nämnas.

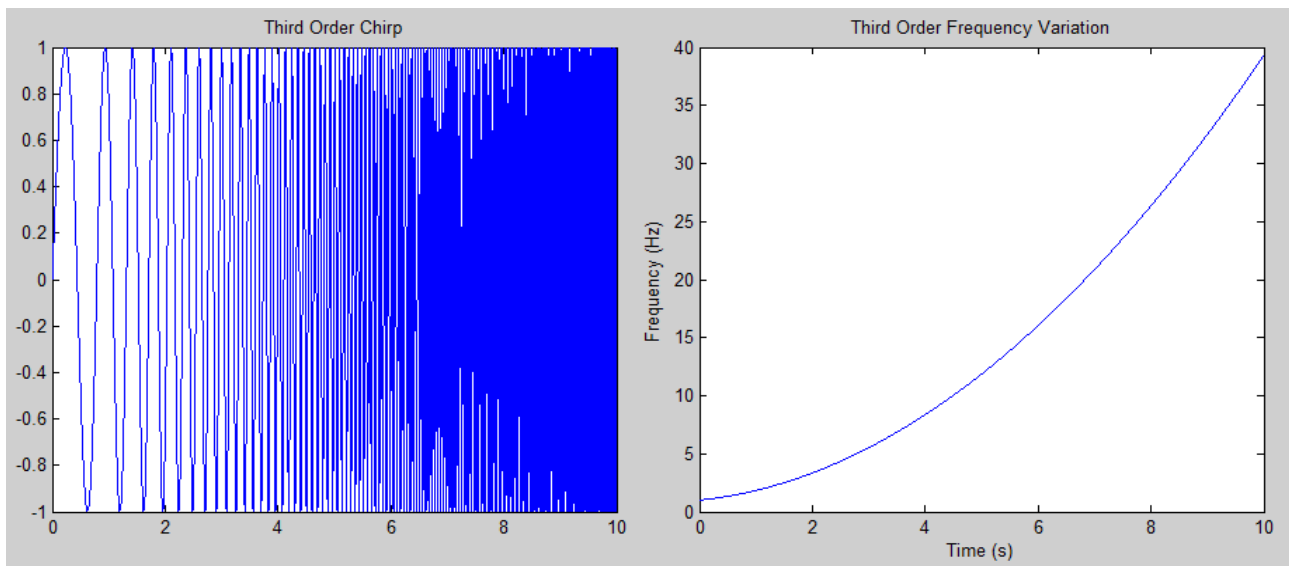
Alla *chirpar* som används är av formen

$$x(t) = Ae^{j2\pi(f_0 t + V(t))} \quad (1)$$

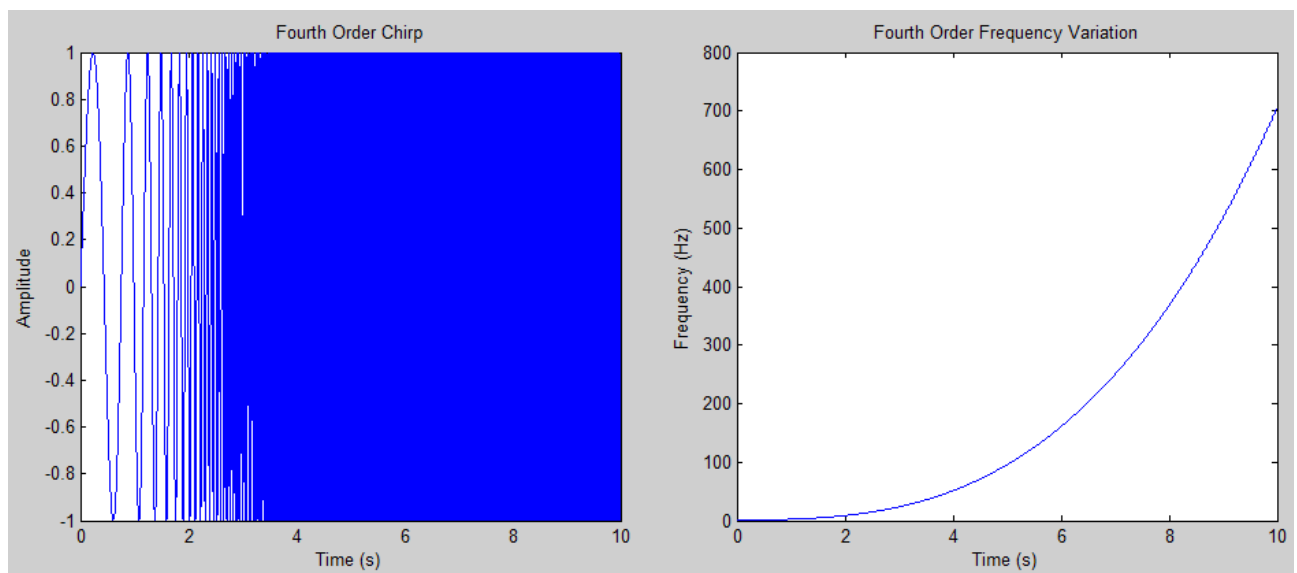
där f_0 är initialfrekvensen och den enda konstanta delen av *frekvensinnehållet*, $V(t)$ kommer att bestämma hur den *momentana frekvensen* varierar med tiden. Antalet parametrar i $V(t)$ anger vilken grad av variation signalen har, men detta kommer att behandlas mer i detalj i delen om **Short-Time Higher-Order Chirp-Fouriertransformen (STCFT)**, eftersom denna typ av chirpdefinition är nära besläktad med STCFT:ns kärna. [9] Klassifikationerna här tar inte *delvisa chirpar* i beaktande, då de består av en eller flera av de två huvudklasserna.



Figur 1 – Andra ordningens chirp och dess linjära frekvensvariation



Figur 2 – Tredje ordningens chirp och dess kvadratiska frekvensvariation



Figur 3 – Fjärde ordningens chirp och dess kubiska frekvensvariation

3.2 Fouriertransformen

Transformen som gjort detta arbete möjligt är, så klart, *fouriertransformen*. Den har använts sen början av 1800-talet och föreslogs av manne som är dess namne. Han utvecklade ursprungligen transformen för att beskriva värmeavledning, men den visade sig vara väldigt användbar för att analysera alla slags vågfunktioner. Den formella definitionen av FT:n är en oändlig integral av formen:

$$F(\nu) = \int_{-\infty}^{\infty} f(t)(\cos(\omega t) - i \sin(\omega t))dt, \quad (2)$$

där $\omega = 2\pi f$, $f(t)$ är funktionen vi vill transformera, $F(\nu)$ är resultatet och *sinus* och *cosinusdifferensen* är kärnan. Denna kärna är endast skriven i en annan form och är således ekvivalent med övriga FT-kärnor presenterade här tack vare sambandet mellan exponentiella funktioner och komplexa tal. [10]

3.3 *Short-Time fouriertransformen*

STFT:n är, kort sagt, en *delvis utförd FT*. Om den utförs med kontinuerlig tid måste signalen multipliceras med en *fönsterfunktion* – d.v.s. en funktion som endast är större än noll över en kort tidsperiod – och en FT utförs alltså *fönstret* glider längs *tidsaxeln*.

$$X(\tau, \omega) = \int_{-\infty}^{\infty} w(t-\tau)x(t)e^{-j\omega t} dt, \quad \omega = 2\pi f \quad (3)$$

där $w(t)$ är fönsterfunktionen, $x(t)$ är signalen som ska transformeras och exponentialfunktionen är kärnan.

Då tiden i det här fallet kommer att vara diskret kommer även signalen att vara diskret, vilket betyder att man helt enkelt kan välja fönsterstorlek, hur mycket fönstren ska överlappa och därefter utföra, till exempel, en *FFT* för vart och ett av de resulterande segmenten. Detta ger en mycket enklare procedur. Det formella uttrycket är så gott som identiskt till det i *ekvation (3)* ovan, förutom att *integralen* blivit utbytt till en *summering* och att alla värden så klart är diskreta. STFT:ns utförande ger ett mycket bättre resultat än FFT:ns vad gäller icke-stationära signaler, men saknar inte sina egna fel och brister. Fönsterfunktionen introducerar ett avvägningsproblem. Å ena sidan kan ett relativt stort fönster väljas, vilket ger en bra *frekvensresolution* och gör det lättare att identifiera specifika *frekvenskomponenter*. Å andra sidan kan ett relativt litet fönster väljas, vilket ger en bra *tidsresolution* och gör det lättare att identifiera tidpunkten då frekvenserna förändras mer precist. Tråkigt nog kan ingetdera åstadkommas utan att vara på den andras bekostnad. Dock är målsättningen här inte att bokföra exakt när *frekvensinnehållet* förändras, utan *amplitud-uppskattningen* är det enda intressanta och således kan det fönster som ger den bästa *uppskattningen* gladeligen väljas. [9]

3.4 *Order Tracking*

En metod för vibrationsanalys kallas *order tracking*. Den här tekniken utvecklades för att möjliggöra isoleringen och analysen av specifika vibrationskomponenter i förhållande till rotationsfrekvens. Denna analysmetod använder sig av *multipler av systemets hastighet*, så kallade *orders*, istället för *absolut frekvens (Hz)*. En *order* är en *fasvektor* som varierar med tiden och roterar med en momentan frekvens vars storlek är i förhållande till referensaxelns rotationsfrekvens, som i *ekvation (4)*. [1] Man kan se att den roterande fasvektorn kommer att innehålla en frekvens som varierar som rotationsperioden (R.P.M) varierar:

$$X(t) = A(k, t) \sin(2\pi i(k/p)t + \phi_k) \quad (4)$$

där $A(k, t)$ är en amplitud av *order k* som en funktion av tiden, ϕ_k är *fasvinkeln* för *ordern k*, p är *primärorderns period i sekunder*, t är tid och k är *ordern* vi vill följa, där $k=0$ (DC offset) och $k<0$ (negativa frekvenser). [1] Rotationsfrekvensen bestäms vanligen med en *tachometer* (varvräknare). En tachometerpulsant signalerar att en viss vinkelposition har nåtts och rotationshastigheten kan därefter beräknas genom att beräkna antalet klockcykler mellan två på varandra följande pulskanter. Genom att sedan fortsätta fånga in pulskanter kan man avgöra huruvida rotationshastigheten ökar eller inte. [2]

På sätt och vis kan *orders* liknas vid harmonik, i och med att de exciteras vid specifika RPM-värden. Till skillnad från i harmoniken tenderar ett flertal intressanta *orders* dyka upp som icke-heltalsmultipler av den första *ordern*. Motorer i en del fordon, till exempel, har *orderkomponenter* som är icke-heltalsmultipler av högre *order*. [2]

3.5 *Frekvensspektrum vs. Orderspektrum*

I frekvensplanet dyker signaler som är periodiska i tidsplanet upp som toppar. Det är dock svårt att relatera dessa toppar till rotationshastighet, även om toppen är klar och lätt att urskönja. Däremot dyker signaler som är periodiska i rotationsplanet upp som toppar i *orderplanet*. Till exempel kommer en vibrationstopp som inträffar två gånger per revolution

vid samma axelpositioner att ta form som en topp av *andra ordern* i ett *orderspektrum*. Figurerna nedan illustrerar dessa skillnader. Båda spektren kommer från samma teststart av en motor. Y-axeln representerar amplitud i båda figurerna. X-axeln representerar *frekvens* i *frekvensspektret*, medan samma axel representerar *rotationsorders* i *orderspektret*. [2]

Frequency spectrum map of 6 cylinder engine run-up

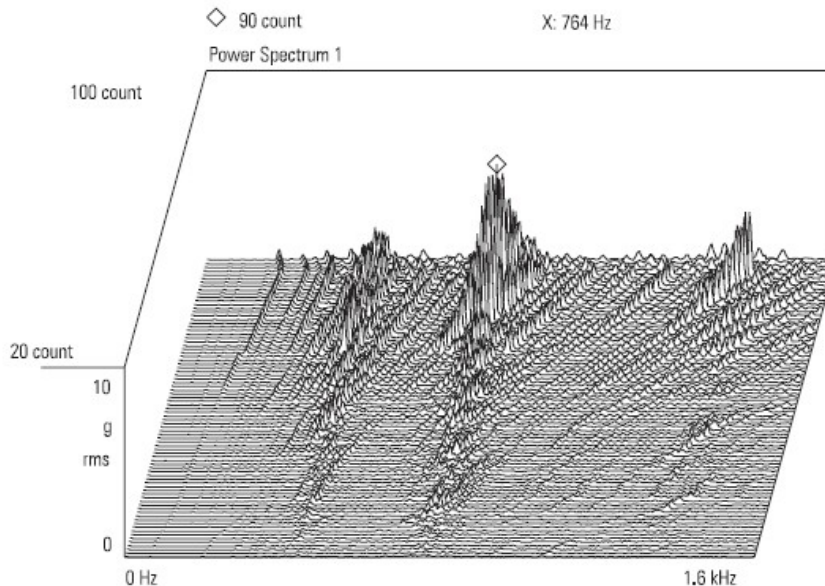


Figure 2:
A frequency map reveals peaks but it is difficult to relate them to shaft speed.

Figur 4 – Frekvensspektrum med tydlig topp [2]

Den tydligaste skillnaden mellan de två spektren är topparnas placering. Där *frekvensspektret* visar topparna i förhållande till frekvens och således gör förhållandet mellan axelhastighet och amplitud otydligt, uppvisar *orderspektret* ett mycket klarare förhållande. I *figur (5)* nedan kan snabbt ses att den största toppen inträffar vid 3815 RPM och 12:e *ordern*. Detta antyder att en komponent exciteras 12 gånger per revolution. Om denna *order* sedan isoleras och följs skulle det ge upphov till en så kallad *order track*. Denna *order track* ger sedan ett väldigt lättläst förhållande mellan motorhastighet och vibrationer, vilket kan ses från figuren, lånad från samma källa som de två övriga, ovan. [2]

Figureerna ovan torde, relativt entydigt, illustrera hurpass viktig *order tracking* är då det kommer till separationen av specifika vibrationskomponenter från signalen och dess inverkan överlag. Då det kommer till maskintillståndövervakning är *order tracking* således användbar så den lätt kan identifiera hastighetsrelaterade vibrationer som axeldefekter och lagerslitage. Metoden har dock problem, som kommer att förklaras mer i detalj senare, att hänga med om R.P.M-värdena ändras mycket snabbt.

3.4 *Synkron sampling*

För att använda *order tracking analysis* måste vibrationssignalen samplas med en konstant ökning av rotationsvinkeln, eller proportionellt mot axelhastigheten om man så vill. Det klassiska sättet att åstadkomma detta har varit att anpassa samplingen enligt en tachometersignal. Denna procedur kräver en specifik utrustning, nämligen en *ratio synthesiser* och ett kantutjämnande spårfilter (*anti-aliasing tracking filter*). *Ratio synthesiser* genererar en signal som är proportionell mot axelhastigheten. Denna används sedan för att

kontrollera **samplingshastigheten** och det analoga spårfiltrets **cut-off frekvens**. Så fort ett datablock har samplats med konstanta vinkelintervall utförs, vanligtvis, en FFT, vilket resulterar i ett orderspektrum. Detta tillvägagångssätt är dock både kostsamt och komplext att implementera. Det har även problem att följa snabba förändringar i axelhastigheten. [4]

Order spectrum map of 6 cylinder engine run-up

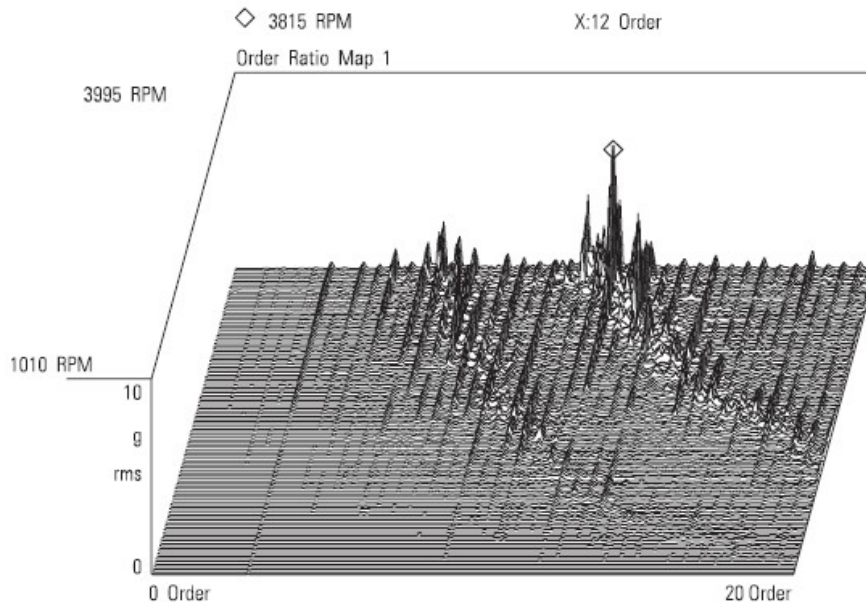


Figure 3:
An order map clarifies the relationships between shaft speed and vibration amplitude.

Figur 5 – Orderspektrum med klarare förhållande mellan axelhastighet och amplitud [2]

Den synkrona samplingstekniken för **order tracking**, nämnd ovan, använder en FFT för att flytta över signalen till ett orderspektrum. FFT:n igen är baserad på den diskreta fouriertransformen (DFT), som har en statisk kärna och är därför till sin natur dålig på att följa icke-stationära signaler. Att en FFT används innebär också att variationer inne i datablocken, t.ex. variationer i amplitud, inte upptäcks. [1] Detta, kombinerat med faktum att hårdvaran i den traditionella metoden är skräddarsydd, låst i sin ena uppgift och inte kan modifieras utan att man går miste om data då systemet körs leder, som man kunde anta, till en oförmögenhet att följa axelhastigheter, som varierar kraftigt inom datablocken, utan att förorsaka estimeringsfel. [4]

FFT-baserad **order tracking** är dock både lättast att implementera och den mest använda tekniken. Den antar dock, förutom allt som nämnts ovan, att frekvensen och amplituden är konstant, vilket kan leda till ytterligare fel. Då kärnan dessutom är statisk kan den inte följa en frekvens som varierar med tiden. [1]

Order track of 12th order, 6 cylinder engine run-up

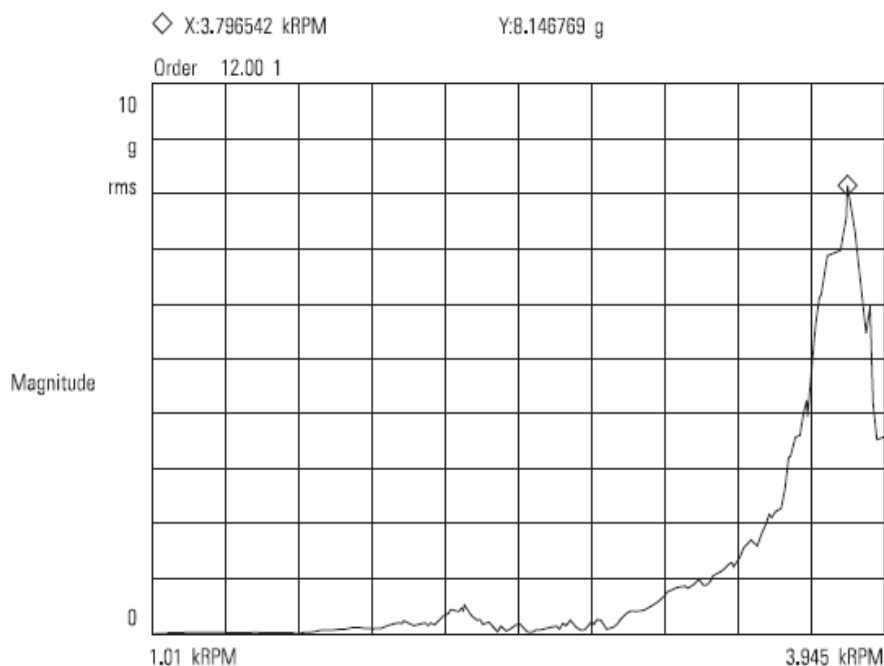


Figure 4:
Order tracking
clarifies the
relationship of a
particular shaft
speed to amplitude
vs. rpm.

Figur 6 – Order track av en tidigare identifierad 12:e order [2]

3.7 Syntetiserad synkron sampling

Ett annat tillvägagångssätt är att sampla signalen med konstant tidsintervall och därefter sampla om med konstant vinkelintervall med hjälp av en huvudfasvektor. *Huvudfasvektorn* är vanligtvis en händelse som inträffar en gång per revolution och används för att mäta axelhastighet och som referens för att mäta vibrationens fasvinkel. Till skillnad från den traditionella metoden är denna metod, även kallad **Computed Order Tracking (COT)**, så gott som helt digital. Vibrationssignalen skickas genom ett lågpasfilter med fixerad frekvens och samplas med ett konstant Δt . Så här långt fungerar COT mer som de traditionella metoderna för frekvensanalys än **order tracking**. Signalen kommer dock, efter att den samplats en gång, att omsamplas med hjälp av huvudfasvektorn för att göra det möjligt att extrahera signalamplituder med konstant $\Delta \theta$. [2, 4]

Under omsamplingen genomförs två distinkta procedurer. Den första handlar om att korrekt placera ut de omsamplade datapunkterna på tidsaxeln. Systemet åstadkommer detta genom att registrera vilka tider pulskanter tas emot från tachometern. En utlösningssnivå bestäms för att bestämma när en pulskant har anlänt. Som man kan tänka sig, påverkas resolutionen av den asynkrona samplingshastigheten, d.v.s. hastigheten vibrations- och tachometersignalerna samplas med. Teoretiskt sett borde tachometersignalen samplas med högsta möjliga hastighet, för att möjliggöra en så noggrann lokalisering av pulskanten som möjligt. I praktiken är samplingshastigheten ofta begränsad och både vibrations- och tachometersignalen blir således samplade med samma hastighet. Resolutionen kan förbättras genom uppsampling, vilket, teoretiskt sett, resulterar i en förbättrad noggrannhet i tachometersignalens tidsbestämning. Konventionell bandbegränsad (*band-limited*) signalrekonstruktion används för att interpolera mellan tachometersignalens sampel. En FFT med N sampel genomförs, varefter M nollor införs mellan varje frekvenskomponent. Då en invers FFT (IFFT) sedan utförs fås den ursprungliga signalen uppsamplad med faktor M .

Huruvida detta är möjligt eller inte beror på om tachometersignalen har blivit lågpasfiltrerad. [2, 4]

Att precis upptäcka och registrera tiden då pulskanter anländer är ett grundläggande krav, då dess genomförande kommer att inverka på interpolationsprocessens möjlighet att producera konsekvent precisa resultat. Detta demonstrerades i en publikation av Fyfe och Munck. [4] Utan precision i det här stadiet har interpolationen ingen chans att uppnå en konsekvent noggrannhet.

Den andra proceduren, den ovannämnda interpolationen, placerar de omsamplade datapunkterna på amplitudaxeln. För att bestämma omsamplingspunkterna antas att axeln genomgår en konstant vinkelacceleration. På denna grund kan då axelvinkeln, θ , beskrivas med en kvadratisk ekvation av följande form: [4]

$$\theta(t) = b_0 + b_1 t + b_2 t^2 \quad (5)$$

De okända koefficienterna b_0, b_1, b_2 kan finnas genom att passa in tre på varandra följande tidpunkter, t_1, t_2, t_3 , då huvudfasvinkelns pulskant anländer, vilket sker i lika stora vinkelinkrement, $\Delta\Phi$. Genom att anta att det endast finns en puls per revolution fås då; $\Delta\Phi = 2\pi$ radianer, vilket i sin tur ger följande villkor: [4]

$$\begin{aligned} \theta(t_1) &= 0 \\ \theta(t_2) &= \Delta\Phi \\ \theta(t_3) &= 2\Delta\Phi \end{aligned} \quad (6)$$

Ankomsttiderna, t_i , ges, som nämndes tidigare, av huvudfasvinkeln. Då villkoren i (6) substitueras in i ekvation (5) och arrangeras i matrisformat resulterar det i följande: [4]

$$\begin{pmatrix} 0 \\ \Delta\Phi \\ 2\Delta\Phi \end{pmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \quad (7)$$

Denna ekvation löses sedan å avseende av de okända komponenterna, b_i . Då dessa värden beräknats kan sedan ekvation (5) lösas å avseende av t med rotformeln. Då fås:

$$t = \frac{1}{2b_2} \left[\sqrt{4b_2(k\Delta\theta - b_0) + b_1^2} - b_1 \right] \quad (12)$$

Eftersom de omsamplade punkterna, tidsmässigt, kan ligga mellan två godtyckliga tidssampel från den ursprungliga samplingsprocessen måste en interpolation utföras för att klargöra det egentliga amplitudvärdet för punkten i fråga. Ingen undersökning har tills vidare påvisat att en viss interpoleringsmetod inte bör användas. Bossley, Mckendrick, Harris och Mercer [8] visade dock i en av sina publikationer att vissa metoder kan fungera bättre.

Då amplitudvärdena för de omsamplade punkterna har bestämts överförs data till **orderdomänen** genom en FFT eller SFTF [4].

Orderfunktioner kan genereras av en godtycklig roterande input från i brukvarande maskineri och kan variera med tiden vad gäller både frekvens och amplitud. Egenskaper som föranleder variationer i amplitud kommer att resultera i fel i vilken slags **order trackinganalys** som helst, då de alla antar att en orders amplitud är uppskattningsvis konstant över analysperioden. Detta antagande kan leda till omfattande analysfel.

3.8 **Short-Time Higher-Order Chirp-Fouriertransformen**

STCFT:n har relativt nyligen föreslagits av Cranfield University. Den är ett alternativ till CFT:n föreslagen och undersökt av X. Xia och P. Fan. [6, 7]

Då CFT:n i första hand utvecklats för att behandla linjära variationer av den momentana frekvensen togs STCFT:n fram för att göra det möjligt att även behandla *olinjära*

(högre ordningens) variationer. [5]

Vad de båda har gemensamt är att de utvecklats för att, som nämndes i *kapitel 2*, de traditionellt använda teknikerna, *DFT:n* och *FFT:n*, är beklagligt dåliga på att behandla *icke-stationära frekvenser* och *amplituder* på grund av den *statiska kärnan*. Dessa fall är mycket viktiga, eftersom diverse roterande maskineri uppvisar dessa egenskaper under start- och stoppfaser.

De skiljer sig då det kommer till hurpass bra frekvensvariationer kan följas. Där *CFT:n* endast klarar linjära variationer av den momentana frekvensen utan att producera fel, kan *STCFT:n* följa både linjära och olinjära variationer av den momentana frekvensen. Den största skillnaden mellan dessa två är dock att *CFT:n* är en *frekvenstransform*, medan *STCFT:n* är en *time-frequencytransform*. [5, 6]

En *frekvensktransform* ger endast frekvensinnehållet för en signal, medan en *time-frequencytransform* ger frekvensinnehållet i förhållande till tid. [9]

3.8.1 Chirp-Fouriertransformen

För att försöka undvika förvirring vad gäller *CF-* och *STCF-transformer* kommer skillnaderna att förklaras kort. *CFT:n* är, som nämndes ovan, inte speciellt bra på att behandla *högre ordningens variationer* av den momentana frekvensen. Dessa problem kommer sig av att dess kärna endast befattar sig med parametrar upp till *chirp raten* och för att den alltid tar hela signalen i beaktande. [6]

Xia definierar sin *N-punkts DCFT*, eller *CFT* om man vill, som:

$$X_c(k, l) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) W_n^{ln^2+kn}, \quad 0 \leq k, l \leq N-1, \quad (13)$$

där k är konstanta frekvenser, l är *chirp rates* och, som för en *DFT*, $W_N = e^{-j2\pi/N}$. *Xia* använder därefter en *FFT* för att beräkna *DCFT:n*. [6] Det skulle vara relativt enkelt att utveckla denna teknik för att även klara av *högre ordningens chirpar*, men då *Xia* inte gjort detta ledde det till utvecklingen av *STCFT:n*. Detta förändrar inte faktum att de två teknikerna är nära besläktade, men ändå såpass olika att man ska vara noggrann med att skilja på dem.

3.8.2 Short-Time Higher-Order Chirp-Fouriertransformen

STCFT:n är en generalisering av den klassiska *STFT:n* och *CFT:n*. [5] Kort sagt kombinerar den fönstringsegenskaperna från *STFT:n* med *CFT:ns* förmåga att behandla icke-stationära signaler och expanderar dessa för att även inkludera högre ordningens icke-stationära signaler. Transformen definieras som följer:

$$S(f, T, c_2, c_2, \dots, c_n) = \frac{1}{T} \int_{k-\infty}^{\infty} h_k(t-T) x(t) e^{-j2\pi(ft + c_2 \frac{t}{2} t^2 + c_3 \frac{t}{3} t^3 + \dots + c_n \frac{t}{n} t^n)} dt, \quad (14)$$

där $h_k(t)$ är ett tidsfönster, T_k är fönstrets varaktighet, $k=1,2,\dots,K$ är antalet delar i ett *piecewise time-frequency dependency* och T är fönstrets mittpunkt. Transformparametrarna, $c_2(t), \dots, c_n(t)$, är: [5]

$$\begin{aligned} c_2 &= \text{Variabel chirp rate} \quad [Hz/s] \\ c_3 &= \text{Frekvensacceleration} \quad [Hz/s^2] \\ c_4, \dots, c_n &= \text{Högre ordningens parametrarna} \end{aligned}$$

Då icke-stationära signaler med en olinjär variation av den momentana frekvensen bör kunna behandlas och transformeras, fås en signal av formen:

$$x(t) = A e^{j2\pi(f_0 t + V(t))}, \quad (15)$$

där A är amplitud, f_0 är ursprunglig frekvens och $V(t)$ är variationsfunktionen:

$$V(t) = \frac{c_{20}(t)}{2} t^2 + \frac{c_{30}(t)}{3} t^3 + \dots + \frac{c_{n0}(t)}{n} t^n, \quad (16)$$

där parametrarna c kontrollerar egenskaper motsvarande sina transformditen. Från *ekvation (15)* kan ses att transformens momentana frekvens är: [5]

$$f(t) = \frac{\partial(f_{0t} + V(t))}{\partial t} = f_0 + c_{20}t + c_{30}t^2 + c_{40}t^3 + \dots + c_{n0}t^{n-1}, \quad (17)$$

Denna tekniks prestanda beror, ganska långt, på att *anpassningsvillkoren* uppfylls. För att kunna uppfylla dessa villkor måste signalens *frekvensinnehåll* kännas till. *Frekvensinnehållet*, som kan fås från till exempel en *tachometer*, tillhandahåller parametrarna som behövs för anpassningen. I simuleringarna i *referens [11]* används signaler som producerats artificiellt. Frekvensinnehållet är alltså på grund av detta identifierat och en ideal anpassning kan garanteras. I praktiken måste anpassningsvillkoren konstant uppdateras för att försäkra deras giltighet.

Då en total anpassning uppnås kommer kärnans och signalens parametrar att neutralisera varandra. I *ekvation (14)* finns en signal multiplicerad med *STCFT-kärnan*. Om den plockas ut från integralen fås:

$$x(t) e^{-j2\pi(f_0 + \frac{c_2(t)}{2}t^2 + \frac{c_3(t)}{3}t^3 + \dots + \frac{c_n(t)}{n}t^n)} \quad (18)$$

Där $x(t)$ är samma som i *ekvation (15)* och $V(t)$ motsvarar $V(t)$ i *ekvation (16)*. Substituerar man in *ekvation (16)* i *ekvation (15)* och sedan detta in i *ekvation (18)* får man:

$$A e^{j2\pi((f_0 + \frac{c_{20}(t)}{2}t^2 + \frac{c_{30}(t)}{2}t^3 + \dots + \frac{c_{n0}(t)}{n}t^n) - (f_0 + \frac{c_2(t)}{2}t^2 + \frac{c_3(t)}{2}t^3 + \dots + \frac{c_n(t)}{n}t^n))} \quad (19)$$

Kärnan är alltså anpassad då dess parametrar är lika med signalens motsvarande parameterar. Med andra ord då:

$$\begin{aligned} c_2 &= c_{20} \\ c_3 &= c_{30} \end{aligned} \quad (20)$$

$$c_4, \dots, c_n = c_{40}, \dots, c_{n0}$$

Med dessa anpassningsvillkor uppfyllda tar parametrarna ut varandra och kvar blir då:

$$A e^{j2\pi(f_0 - ft)} \quad (21)$$

Om anpassningsvillkoren, av någon anledning, inte uppfylls fås inte en lika snygg och stadig exponent som i *ekvation (21)*. Istället fås något likt *ekvation (21)*, men med de kvarvarande resterna från parameterdifferensen i exponenten, vilket leder till ett felande slutresultat. [5]

4. Simulering, resultatinsamling och felkalkyl

I källmaterialet [11] gjordes ett ganska stort antal jämförelser och simuleringar. Eftersom de båda teknikerna inte fungerar optimalt under exakt samma förhållanden måste dessa förhållanden identifieras.

4.1 Simulering

På grund av detta producerades ett antal olika chirpsignaler, alla samplade med en frekvens på 65536 Hz, som utgångspunkt. De första och sista 65536 punkterna klipptes bort, eftersom de kunde lida av *Gibbs fenomen*. Fönstren överlappade varandra till 20% och hölls konstant då dess inverkan inte är att tala om. För *COT* användes linjär interpolation i alla fall.

Enligt MATLAB-koden kommer endast vågtopparna att uppskattas och ges som resultat. [11]

Källmaterialet behandlade både konstanta och variabla amplituder – varierar enligt $A = f(t)^2$ - för tre olika variationsgrader av den momentana frekvensen (*linjär, kvadratisk och kubisk*) och utsatte alla dessa för fyra olika störningsgrader (*ingen störning, 5dB, 10dB och 25dB*). Dessa behandlades sedan med fem olika samplingsfrekvenser (4096, 8192, 16384, 32768 och 65536 Hz). Även olika fönsterstorlekar undersöktes. Dessa gick i tvåpotens allt från 2^6 upp till den dåvarande samplingsfrekvensen.

Om man räknar ihop de ovanvarande fallen får man 1080 olika fall för de båda transformerna. Då sedan fyra olika felberäkningar användes för vart och ett av dessa fall blir omfattningen på en dylik undersökning alltför stor för att kunna dupliceras här. En titt på en del av de, i källmaterialet [11], insamlade resultaten kommer dock att tas.

4.2 Resultatinsamling

Alla resultat samlades i Excel och gavs en färgkod beroende på hur bra resultatet i fråga var. De jämfördes sedan på basen av ett antal olika kriterier. En bas var respektive transforms bästa resultat för en given samplingsfrekvens och störningsgrad, vilket även gav de bästa parametrarna för de givna fallen. En annan var att identifiera fönsterstorleken som presterade bäst för varje enskild samplingsfrekvens över alla störningsgrader, vilket i princip är en fönsterstorleksoptimering. Denna bas togs även ett steg vidare och då hölls fönsterstorleken konstant över de olika störningsgraderna **och** frekvensvariationerna. Även felintervallen över alla samplingsfrekvenser, störningsgrader och fönsterstorlekar för varje frekvensvariation jämfördes. Denna jämförelse torde påvisa hur väl transformerna presterar från de absolut sämsta upp till de absolut bästa resultaten. [11]

4.3 Felkalkyl

Lögn, förbannad lögn och statistik. Felkalkyl kan vara missvisande om man inte är försiktig. För att undvika detta användes fyra olika felkalkyler. Den första och enklaste är **Mean Absolute Error (MAE)**. [11]

$$MAE = \frac{1}{n} \sum_{i=0}^n |Estimate_i - Envelope_i| \quad (22)$$

där *Estimate* är det uppskattade amplitudvärdet och *Envelope* är det egentliga. *MAE* är helt enkelt skillnaden mellan dessa två. Då de uppskattade värdena kan infalla vid olika tidpunkter, beroende på transform, kommer de att jämföras med det egentliga värdet vid just den tidpunkt då de infaller. Detta skulle annars, om någondera transformen hade fått diktera tidpunkterna mot vilka uppskattningarna jämförs, ha kunnat förvränga resultaten. Absolutbeloppet beräknas eftersom amplituduppskattningarna kan oscillera runt den egentliga amplituden, vilket medför en risk att felen tar ut varandra och ter sig mycket mindre än de egentligen är. Detta gäller för alla felkalkyleringar. *MAE* är dock endast intuitivt och enkelt tolkad för fallen med *konstant amplitud*, då uppskattningsfelen för fallen med *variabel amplitud* kan te sig mer grava än de egentligen är.

Detta leder till följande felkalkyl; *Mean Relative Error (MRE)*: [11]

$$MRE = \frac{1}{n} \sum_{i=0}^n \frac{|Estimate_i - Envelope_i|}{|Envelope_i|} \quad (23)$$

Denna kalkyl är lik *MAE*, men differensen är här viktad mot det egentliga amplitudvärdet. Detta för att uppskattningsfelen för de variabla amplituderna ska bli mer lika felen för de konstanta amplituderna. Idén är att ett fel på, säg, 0,01 då det egentliga värdet är 1 är en hel del större än ett fel på 0,01 då det egentliga värdet är 1000, så genom att dividera differensen med det egentliga amplitudvärdet ger en indikation om hurpass allvarligt uppskattningsfelet egentligen är. Precis som ovan beräknas absolutbeloppet för att förhindra att oscillerande uppskattningar förvränger resultaten. Egentligen behöver inte absolutbeloppet beräknas för nämnaren, då detta värde alltid är positivt. Det görs mest för att illustrera och understryka att felet alltid är positivt.

MRE används främst för att ge en indikation på hurpass allvarligt ett *momentant uppskattningsfel* är jämfört med det motsvarande *momentana amplitudvärdet*. Som antydde tidigare kommer amplituderna, och följaktligen även felen, att vara en hel del större för *variabla amplituder* än motsvarande fel för *konstanta amplituder* och det skulle således vara svårt att mäta teknikernas prestanda utan något slags *felallvarlighetsmått*.

De ideala fönsterstorlekarna kommer antagligen inte att vara samma över alla *fönsterstorlekar*, *samplingsfrekvenser* och *störningsgrader*. Detta är inget problem vad gäller jämförelsen av enskilda *störningsgrader*, men gör det tvunget att beräkna ett medelfel av medelfelen i de fall då jämförelsen sker över flera olika fall.

Följande felberäkning på agendan är **Root Mean Squared Error (RMSE)**. Formeln för denna är: [11]

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n \frac{(Estimate_i - Envelope_i)^2}{Envelope_i^2}} \quad (24)$$

Den här kalkylen fungerar på ett sätt som liknar *MRE:s*, förutom att absolutbeloppen inte beräknas här och allting kvadreras istället. Då kvadratroten sedan beräknas fås ett värde som är i samma storleksordning som de övriga beräkningarna. Denna kalkyl fungerar också som ett slags *fel-allvarlighetsmått* och hjälper till att bekräfta resultaten från *MRE* och vice versa.

Den sista felkalkylen är egentligen inte en felkalkyl, utan en kvot. Denna kvot är: [11]

$$Gain = \frac{1}{n} \sum_{i=0}^n \frac{Estimate_i}{Envelope_i} \quad (25)$$

Denna kvot kommer att visa huruvida tekniken *över-* eller *underskattar* amplituden överlag. Om *Gain* är mindre än *1* har tekniken *underskattat* amplituden och om *Gain* är större än *1* har amplituden blivit *överskattad*.

Det är inte praktiskt att använda sig av någon av de tidigare formlerna (22, 23, 24) i detta syfte, då de innehåller en differens som kan introducera negativa värden, som därefter försvinner tack vare absolutbeloppet. Absolutbeloppsberäkningen kommer att göra det omöjligt att säga huruvida det uppskattade värdet låg ovanför eller under det egentliga amplitudvärdet. Nackdelen här är dock att oscillerande värden kan neutralisera varandra och ge upphov till ett *gainvärde* som är mindre än vad det egentligen borde vara. På grund av detta används *ekvation (25)* endast för att bedöma ifall tekniken i fråga tenderar till *över-* eller *underskattning* av amplituden.

Även om det inte direkt rör sig om en felkalkyl kommer även en kvot baserad på felberäkningarna att beräknas. Denna kvot resulterar i ett konkret jämförelsetal teknikerna emellan. Den fungerar i princip helt som *ekvation (25)*, såtillvida att ett resultat på *1* innebär att teknikerna har presterat lika väl, medan resultat längre från *1* väger någongendera tekniken till fördel. Formeln är: [11]

$$rError = \frac{\cot_{error}}{STCFT_{error}} \quad (26)$$

Enligt ovanstående kommer alltså ett *rError*-värde som är mindre än *1* att vara till *COT:ns* fördel, medan ett värde större än *1* är till *STCFT:ns* fördel.

Alla medeltal ovan är *aritmetiska* [11]

5. Resultat och jämförelse

Då en grundlig simulering, resultatinsamling och därpå följande analys blir lite för omfattande – mängden data var till och med aningen för omfattande för att allting skulle kunna analyseras i källmaterialet [11] – för detta arbete kommer en titt istället att tas på några av de i källmaterialet – referens [11] – uppvisade resultaten att tas. Fokus kommer att vara på de bästa resultaten för konstanta amplituder med linjära och kvadratiske frekvensvariationer utan störning.

5.1 Jämförelse

Eftersom det är lättast att jämföra resultaten i, så att säga, stigande ordning kommer de bästa *MAE* resultaten för en konstant amplituds chirpsignal med en linjär frekvensvariation och utan störning att undersökas först. I tabellen nedan kan ses de bästa resultaten för båda

transformerna, varje samplingsfrekvens (F_s), samt de olika felkalkylerna för den i frågavarande fönsterstorleken. Det tål påpekas att *Gain* har justerats med 1. Det vill säga de listade *gain*värdena är egentligen $Gain = 1 - Gain$ och negativa *gain*värden påvisar alltså en överskattning, medan positiva påvisar en underskattning. Fönsterstorlekarna är i antal sampelpunkter. En fönsterstorlek på 4096 antyder alltså att fönstret är 4096 sampel stort.

I tabellen nedan finns alltså resultaten för en chirpsignal med *linjär variation* av den momentana frekvensen, *konstant amplitud* och *ingen störning*. Börjandes från kolumnen längst till vänster (fönsterstorlek) kan ses att *STCFT:n* tenderar välja ett så stort fönster som möjligt, medan *COT:n* nöjer sig med ett mindre fönster. Då båda transformerna använder sig av en *STFT-bas* innebär det här att *COT* har en bättre *tidsresolution*. Det har dock i den här undersökningen ingen inverkan, eftersom tyngdpunkten ligger på *amplituduppskattning* och frekvensen varierar relativt långsamt, för att inte tala om att *STCFT:n* producerar mindre fel i vilket fall som helst. Faktum är att det enda värdet i tabellen som lutar till *COT:ns* fördel är *gain*värdet för $F_s = 4096 \text{ Hz}$. Detta kan, då det endast sker en gång, tillskrivas slumpen och

Transform	F_s	Win Size	MAE	MRE	RMSE	Gain	rMRE	rRMSE
STCFT	4096	4096	0,013985	0,013915	0,018416	-0,00804		1,40289
COT	4096	1024	0,02252	0,02255	0,025836	0,00739	1,6206	
STCFT	8192	8192	0,013933	0,013862	0,018509	-0,00826	1,5311	1,49723
COT	8192	4096	0,02135	0,021224	0,0277	0,00860		
STCFT	16384	16384	0,014007	0,013937	0,018613	-0,00848	1,9203	1,75037
COT	16384	4096	0,02684	0,026762	0,0326	0,01198		
STCFT	32768	32768	0,014053	0,013982	0,018636	-0,00853	2,2824	2,04135
COT	32768	8192	0,03202	0,031912	0,0380	0,01816		
STCFT	65536	65536	0,01555	0,015473	0,019751	-0,00984	2,7916	2,63733
COT	65536	16384	0,04329	0,043196	0,0521	0,02726		

Tabell 1: 2:a ordningens chirp, konstant amplitud, ingen störning [11]

*gain*formelns egenskaper istället för *COT:ns* prestanda. De övriga felvärdena i tabell (1), som är ungefär dubbelt så stora som motsvarande fel för *STCFT:n*, stöder denna slutsats. Detta understryker även varför *Gain* endast används för att se ifall vi har en *över-* eller *underskattning*.

De kvoterade felen längst till höger visar även att *STCFT:n* presterar ungefär dubbelt så väl som *COT:n*. Detta är något överraskande eftersom denna signal är den simplaste som producerats.

STCFT:n visar här en tendens att *överskatta* och *COT:n* en tendens att *underskatta* amplituden.

Transform	F_s	Win Size	MAE	MRE	RMSE	Gain	rMRE	rRMSE
STCFT	4096	4096	0,01128	0,011114	0,0178	0,00718	0,9226	0,80395
COT	4096	2048	0,01028	0,010254	0,0143	0,00322		
STCFT	8192	8192	0,01126	0,011098	0,0177	0,00733	1,2024	1,10036
COT	8192	1024	0,01338	0,013343	0,0195	0,00187		
STCFT	16384	16384	0,01123	0,011061	0,0177	0,0075	1,5934	1,30403
COT	16384	1024	0,0176	0,017625	0,0230	0,0013		
STCFT	32768	32768	0,01122	0,011053	0,0176	0,00755	2,3309	1,71966
COT	32768	2048	0,0258	0,025764	0,0304	0,00853		
STCFT	65536	65536	0,01209	0,011902	0,0187	0,00797	4,5698	3,52629
COT	65536	4096	0,05442	0,054391	0,0658	0,03473		

Tabell 2: 3:e ordningens chirp, konstant amplitud, ingen störning [11]

Här, i tabell (2), kan ses samma tendenser då det kommer till fönsterstorlek för *STCFT:n*, medan *COT:n* verkar välja fönsterstorlek på högt nu. I det tidigare fallet – tabell (1) – verkade

fönsterstorleken alltid vara *en fjärdedel* av *samplingsfrekvensen*, med undantag för $F_s=8192\text{ Hz}$. Nu, för den *kvadratiska variationen* finns det inget mönster att skönja.

COT:n verkar även prestera sämre ju högre *samplingsfrekvensen* är. Att prestandan skulle bli sämre ju närmare den använda signalen man kommer är milt sagt skumt. *COT:n* är ändå den som producerar det bästa resultatet här och det dessutom med den lägsta möjliga *samplingsfrekvensen*. Å andra sidan uppvisar *STCFT:n* en jämnare prestanda för de olika *samplingsfrekvenserna* och *fönsterstorlekarna* här. Utgående från *Gain* verkar båda teknikerna här underskatta amplituden något.

Bortsett från det första fallet presterar *STCFT:n* även här ungefär dubbelt så bra och förhållandevis bättre anefter som *COT:ns* prestanda blir sämre. Faktum kvarstår att *COT:n* producerade det bästa enskilda resultatet här, men skillnaden teknikerna emellan vid högre *samplingsfrekvenser* är enorm.

Transform	F_s	Win Size	MAE	MRE	RMSE	Gain	rMRE	rRMSE
STCFT	4096	4096	0,01057	0,010448	0,0175	0,0005	4,8909	3,54376
COT	4096	64	0,05168	0,0511	0,0620	0,04937		
STCFT	8192	4096	0,00898	0,008949	0,0116	0,00343	2,0468	2,40277
COT	8192	128	0,01847	0,018316	0,0280	0,01079		
STCFT	16384	8192	0,00826	0,008263	0,0111	0,00202	2,0627	2,20064
COT	16384	256	0,01706	0,017044	0,0244	0,00885		
STCFT	32768	16384	0,00877	0,00872	0,0116	0,00403	2,0657	1,85028
COT	32768	2048	0,01815	0,018013	0,0214	0,01518		
STCFT	65536	32768	0,00777	0,007755	0,0104	0,00187	4,2703	3,47360
COT	65536	32768	0,0334	0,033116	0,0362	0,03312		

Tabell 3: 4:e ordningens chirp, konstant amplitud, ingen störning [11]

Resultaten i tabell (3) är längs samma mönster som de tidigare. Intressant att se är dock att *STCFT:n* inte tenderar mot största möjliga *fönsterstorlek* här, utan håller sig ganska långt till en storlek som är hälften så stor som det största möjliga. Detta kan möjligen bero på ökningen i *frekvensvariation*, som kan kräva en noggrannare *tidsresolution*.

COT:n uppvisar igen tecken på en försämrad prestation vid högre *samplingsfrekvenser*. Den är också något inkonsekvent med fel allt från 0,017 upp till 0,05, vilket motsvarar en skillnad på 0,033 mellan största och minsta fel. *STCFT:n* å andra sidan visar upp felvärden från 0,0078 upp till 0,011, vilket med en skillnad på 0,0032 är ungefär *en tiondel* av vad *COT:n* uppvisar. Allt annat här verkar också favorisera *STCFT:n*.

Prestandagapet vidgas fortsättningsvis. Då det minsta kvoterade felet ligger på 1,85 och det största på 4,89 torde det vara ganska säkert att säga att *STCFT:n* presterar bättre i det här fallet.

Utgående från de ovanstående resultaten ser det ut som om *STCFT:n* presterar bättre. [11] Det är dock något svårt att avgöra då en fullskalig undersökning inte kan genomföras i detta arbete. Inte ens undersökningen genomförd i källmaterialet – referens [11] – ger en fullständig bild av läget då den endast beblandat sig med simulerade, eller konstgjorda om man så vill, signaler. Vad som kan konstateras är dock att, på basis av resultaten, det finns grund för vidare undersökning om jämförelse transformerna emellan. *COT:n* har varit något av en de facto standard inom industrin redan en längre tid, men då den ligger så pass långt efter i testerna här borde antyds ett möjligt tronskifte.[11]

Teoretiskt sett borde *COT:n* konsekvent producera hyfsat bra resultat. Då den samplar om signalen från en tidsbas till vinkelbas får man alltid samma antal sampel per period och de ligger alltid på samma plats. Med rätt antal sampel per period borde således alltid ett sampel ligga på toppen av kurvan och borde då vara ekvivalent med amplitudvärdet. Faktum att dessa värden måste interpoleras fram är dock något av en svaghet och kan leda till feluppskattning.

Detta blir mer uppenbart i de fall där amplituden varierar mycket och snabbt. Omsamplingen innebär också att ett extra steg tillförs i beräkningarna. Det extra steget har oftast inte så stor inverkan på hur lång den totala beräkningstiden är, men kan i vissa fall vara en faktor.

STCFT:n igen använder sig av en mer klassisk approach då den håller sig till sampel i tidsplanet. Detta kan möjligen leda till problem om signalen oscillerar för ofta inom ett fönster. Om det vill sig illa kan det väl hända att samplen råkar placeras på dåliga ställen och ge upphov till resultatdegradation. I testerna sågs dock inga tecken på detta. Beräkningsmässigt är *STCFT:n* egentligen, de skrämmande formlerna till trots, ganska enkel. Transformparametrarna anpassas till signalen enligt villkoren i *kapitel 3.8.2* och därefter utförs transformen.

6. *Slutledning*

Efter introduktionen konstaterades varför olika transformeringar som redan finns i användning inte kunde användas.

Teorin som ligger bakom de två jämförda teknikerna och därkringliggande viktiga koncept gick igenom förklarades.

En titt togs på hur en jämförelse av de två teknikerna utförts, hur resultaten samlats in och hur feluppskattningarna beräknats. Därefter diskuterades resultaten från jämförelsen och de båda teknikernas olika meriter nämndes.

7. *Referenser*

[1] Jason R. Blough, A Survey of DSP methods for rotating machinery analysis, what is needed, what is available, *Journal of Sound and Vibration* 262 (2003) 707-720

[2] Hewlett-Packard, Realtime Update, Fall 1996 – Winter 1997, available online at: http://www.home.agilent.com/upload/cmc_upload/All/6C06DATAACQ_ORDER.pdf (link alive on 11.08.2010)

[3] K.M. Bossley and R.J. Mckendrick, Hybrid Computed Order Tracking, *Mechanical Systems and Signal Processing* (1999) 13(4), 627-641, Article No. mssp.1999.1225

[4] K.R. Fyfe and E.D.S. Munck, Analysis of Computed Order Tracking, *Mechanical Systems and Signal Processing* (1997) 11(2), 187-205

[5] Gelman L., Ottley M. (2006). New processing technique for transient signals with nonlinear variation of the instantaneous frequency in time. *Mechanical Systems and Signal Processing*, 20(5): 1254–1262

[6] X. Xia, Discrete chirp-Fourier transform and its application to chirp rate estimation, *IEEE Transactions on Signal Processing* 48 (11) (2000) 3122–3133.

[7] P. Fan, A modified discrete chirp-Fourier transform scheme, *World Computer Congress 2000*, vol. 1, 2000, pp. 57–60.

[8] Bossley, K. M., Mckendrick, R. J., Harris, C. J. and Mercer, C. (1999) Hybrid Computed Order Tracking, *Mechanical Systems and Signal Processing* (1999) 13(4), 627-641, Article No. mssp.1999.1225, available online at <http://www.idealibrary.com>

[9] Digital Signal Processing module, Cranfield University, Academic Year 2009-2010

[10] Bracewell R.N., The Fourier Transform, Scientific American (June 1989), pages 62-69, online at: http://www.cs.bgu.ac.il/~ben-shahar/Teaching/Computational-Vision/Readings/1989-Bracewell-The_Fourier_Transform.pdf (link alive on 11.08.2010)

[11] Sjöblom, J (2010), Novel Comparison of the Short-Time Higher-Order Chirp-Fourier Transform and Computed Order Tracking Method, Master's Thesis, Cranfield University