

Tobias Fries, Jonas Karlsson

Dynamisk vägning – principer och metoder för checkweighers

Abstract

Dynamic weighing – principles and methods for checkweighers

Dynamic weighing is a process in which the weighed object and the weighing-machine are under relative motion. This is an area of great technical and economic importance. A checkweigher is used to control that required tolerances concerning weight are reached in production processes.

The force-measuring device produces a time-varying signal when the object to be weighed passes the weigh bridge. From this signal the object's mass can be estimated by different kinds of signal processing and optimization methods. To increase productivity in the process the speed of the conveyor needs to be increased. This increase makes it harder to perform accurate weighings.

In this report a mathematical model of a checkweigher and the weighing process is developed. The model is based on an ODE with time-varying coefficients. From this model an algorithm is presented that calculates an estimate of the mass being weighed, using the dynamics of the system.

The algorithm is based on a non-linear optimization method. The results are satisfactory when the algorithm is tested in the digital model environment and also in the physical reality. As a measure of accuracy we are using the relative error. When performing experiments with conveyor speeds up to 60 m/min and masses up to 2 kg, this error is typically found to be no larger than 10^{-3} , which fulfils the international recommendation OIML R 51 for checkweighers of accuracy class X(x). A slightly better accuracy is achieved in the digital model environment in which the algorithm is stable with conveyor speeds up to 120 m/min.

Key words: mathematical modelling, checkweigher, nonlinear regression, parameter estimation, optimization, dynamic weighing

**SP Sveriges Provnings- och
Forskningsinstitut**
SP Rapport 2003:32
ISBN 91-7848-970-9
ISSN 0284-5172
Borås 2003

**SP Swedish National Testing and
Research Institute**
SP Report 2003:32

Postal address:
Box 857,
SE-501 15 BORÅS, Sweden
Telephone: +46 33 16 50 00
Telefax: +46 33 13 55 02
E-mail: info@sp.se
Internet: <http://www.sp.se>

Innehållsförteckning

Abstract	2
Innehållsförteckning	3
Förord	4
Sammanfattning	5
Figurförteckning	6
Variabelförteckning	7
1 Introduktion	9
1.1 Tidigare angreppssätt	9
2 Modell av en checkweigher	11
3 Mätsignalen	14
3.1 Filtrering	14
3.2 Modell	15
3.2.1 Integration	16
3.2.2 Icke-linjär optimering	17
4 Mätdata	19
4.1 Förstärkarbrus	20
4.2 Störningskällor	20
4.3 Omvandling spänning – massa	22
5 Resultat	23
5.1 Modellkörning	23
5.2 Empiriska resultat	24
6 Slutsats	25
6.1 Vidare utveckling	25
7 Källförteckning	27
Appendix	28
Bilaga 1 – Empiriska mätresultat	28
Bilaga 2 – Matlabkod	31

Förord

Arbetet har utförts vid riksmätplatsen för massa på enheten för Mätteknik vid SP Sveriges Provnings- och Forskningsinstitut.

Arbetet ger 20 högskolepoäng vilket ska motsvara 20 veckors heltidsstudier. Examensarbetet utgör den avslutande delen på Industrimatematikprogrammet (160p) som ges på institutionen för matematik vid Göteborgs Universitet.

Vi vill tacka våra handledare på SP, Henrik Marcusson och Jan Hjelmgren, samt vår examinator, Jöran Bergh CTH/GU, för stöd och visat intresse.

Vi vill även tacka Måns Ackerholm, SP, och Kristoffer Hermansson på Keithley Instruments för era försök att hjälpa oss när vi stötte på problem med mätutrustningen.

Ett slutligt tack går till Teltek för utlålandet av en checkweigher, utan vilken denna rapport varit omöjlig.

Sammanfattning

Dynamisk vägning, det vill säga vägning där mätobjekt och våg befinner sig i relativ rörelse är ett område av stor teknisk och ekonomisk betydelse. En checkweigher används för att kontrollera att önskade toleranser med avseende på vikt upprätthålls i produktionsprocesser.

Då objekten passerar över vågbryggan ger den kraftmätande enheten en tidsvarierande signal ur vilken objektens massa kan uppskattas med hjälp av signalbehandling samt optimering. För att öka produktiviteten i produktionsprocessen önskas en allt högre transporthastighet vilket gör en noggrann bestämning av objektens massa allt svårare.

I denna rapport utvecklas en modellbeskrivning av en checkweigher samt vägningsförloppet. Kraftsituationen modelleras av en ordinär differentialekvation med tidsberoende koefficienter. Från denna modell utarbetas en algoritm som med information om systemets dynamik ger en massauppskattning.

Algoritmen är baserad på en icke-linjär optimeringsmetod. Den har testats i både modellmiljö samt vid verkliga försök med goda resultat. Som ett mått på algoritmens noggrannhet används det relativa felet. I verkliga försök, för hastigheter upp till 60 m/min samt massor upp till 2 kg, kan man förvänta sig att detta fel ligger i intervallet 10^{-3} till 10^{-4} , vilket uppfyller den internationella rekommendationen OIML R 51 för X(x)-klassade checkweighers. I modellmiljö uppnås en något bättre noggrannhet och algoritmen är även stabil för bandhastigheter upp till 120 m/min.

Figurförteckning

1.1	Principskiss av ett objekts rörelse över en checkweigher	9
2.1	Teltek 800 – principskiss	11
2.2	Fysikalisk modell av en checkweigher	11
2.3	Den modellerade massafunktionen	12
3.1.1	En samplad samt en filtrerad signal	14
3.2.1	En modellerad utsignal	15
3.2.2	En simulerad mätning	16
3.2.2.1	Modellsignal samt anpassning	18
4.1	En samplad signal	19
4.1.1	Förstärkarbrus	20
4.1.2	Spektrum förstärkarbrus	20
4.2.1	Ett spektrum över motor- och bandbrus, $v = 30$ m/min	21
4.2.2	Ett spektrum över motor- och bandbrus, $v = 60$ m/min	21
4.3.1	Samband spänning-massa	22
4.3.2	Avvikelse från linjäritet	22
5.1.1	Modellkörning	24

Variabelförteckning

M	Vågbryggans massa (kg)
m	Det vägda objektets massa (kg)
$m(t)$	Den massa som belastar vågbryggan (kg)
g	Lokal tyngdacceleration (m/s^2)
$x(t)$	Det vägda objektets vertikala rörelse (m)
k	Systemets fjäderkonstant (kg/s^2)
c	Systemets dämpning (kg/s)
f_{system}	Systemets egensvängning (1/s)
f_s	Samplingsfrekvens (1/s)
f_v	Vibrationernas frekvens (1/s)
$u(t)$	Lastcellens spänningssignal (V)
err	Det relativa felet

1 Introduktion

Människan har länge haft behov av att väga saker. Det man associerar vägning till är att lägga objekt på en våg och vänta på ett resultat. Detta kan vara alldeles för tidskrävande i en industriell miljö där man vill hålla en hög produktivitet. På grund av detta finns det även något kallat dynamisk vägning, där man väger föremål under rörelse. Man kan till exempel väga fordons totalvikt utan att dessa behöver stanna eller tågagnar för att kontrollera, till exempel, hur mycket malm de fraktar. I en produktionslinje vill man ofta kontrollväga varor för att se att dessa uppfyller satta toleranser med avseende på vikt. Ett exempel på detta är e-märkta¹ förpackningar. En färdigförpackad vara, som exporteras från ett land till ett annat, kontrolleras vanligen av det importerande landet för att bedöma om dess krav på rätt nettomängd är uppfyllda. Under vissa förutsättningar får varorna märkas med ett speciellt e-märke, vilket underlättar färdigförpackningarnas så kallade "fria rörlighet" inom EU. Detta innebär att av ett parti färdigförpackningar får bara en liten andel (högst 2.5 procent av antalet förpackningar) ligga under en viss toleransgräns. Hur många som helst får ligga över gränsen. Om nu någon färdigförpackning ligger under toleransgränsen, så får den i alla fall inte ligga under dubbla toleransen. Dessa toleransnivåer återfinns i OIML R87². I en sådan situation kan dynamisk vägning spara både tid och pengar.

I denna rapport har vi inriktat oss på dynamisk vägning med en så kallad checkweigher. En checkweigher är en del i ett transportbandssystem. Den består av ett påmatningsband, en vågbrygga med tillhörande lastcell³, i vårt fall en töjningsbaserad givare, samt ett utmatningsband. När ett objekt vägs så går det till på följande sätt (se figur 1.1)

- Objektet kommer in på inmatningsbandet. (1)
- Alldeles innan objektet åker in på vågbryggan bryter den en fotocell. (2)
- Med hjälp fotocellen kan man bestämma när objektet befinner sig helt på vågbryggan. (3)
- När objektet befinner sig på vågbryggan ger detta utslag i lastcellen. (4)
- Utifrån denna signal bestäms objektets vikt. (4)
- Objektet lämnar vågbryggan. (4)



Figur 1.1 -- Principskiss av ett objekts rörelse över en checkweigher

Det övergripande målet med denna rapport är att hitta en effektiv algoritm för massberäkningen. Med effektiv avses att behålla nuvarande noggrannhet även vid höga hastigheter utan att beräkningstiden blir för lång. Vi är även intresserade av att bygga upp en matematisk modell av en checkweigher för att kunna öka förståelsen av vägningsförloppet samt skapa en datorbaserad simuleringsmiljö för vidareutveckling.

1.1 Tidigare angreppssätt

Det har bedrivits en hel del tidigare forskning om dynamisk vägning, dock har det mesta varit koncentrerat på fordonsvägning och inte checkweighers. När det gäller checkweighers är en populär angreppsvinkel att modellera förloppet med hjälp av Newtons andra lag, $F = ma$. Den kraft som objektet belastar vågbryggan med, vilken är tidsberoende, modelleras som en fyrkantsvåg eller en något modifierad sådan. På denna lägger man brus av olika sorter för att efterlikna den sanna mätsignalen.

Det allra vanligaste sättet att komma förbi bruset och få en snäll funktion att jobba med är filtrering (lågpass- eller bandpassfiltrering). W.J. Shi, N. White och J.E. Brignell⁴ beskriver konstruktionen av ett adaptivt filter för att slippa egensvängningar i lastcellens utsignal. Man fouriertransformerar lastcellens utsignal för att sedan finna dess z-transform. Utifrån denna konstrueras ett adaptivt filter som ändras beroende på den applicerade massan. Ytterligare ett filter utvecklas för att kunna eliminera störningar, även detta adaptivt då det ger betydligt bättre prestanda.

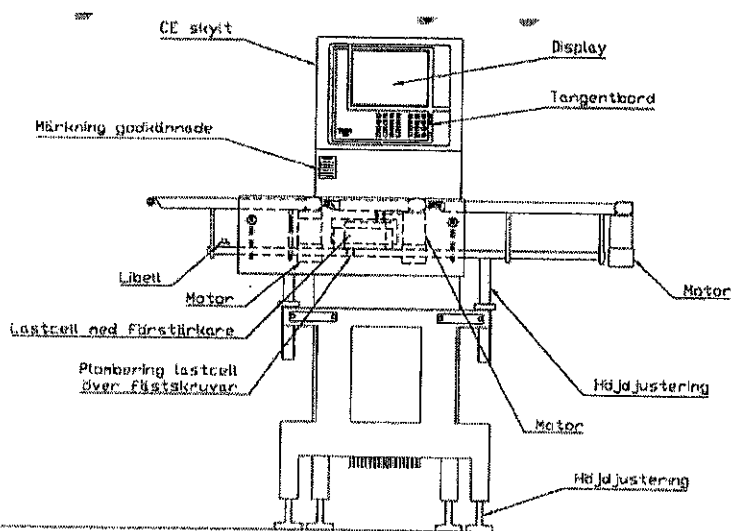
Värt att notera är att detta⁴ är en modell av enbart en lastcell och den tar därför inte hänsyn till hur lastcellen beter sig i ett system, med fler störningskällor. Detta görs av T. Yamazaki, Y. Sakurai m.fl.⁵, där man applicerar ett FIR-filter (Finite Impulse Response) på mätdata för att slippa egensvängningar. Filterkoefficienterna bestäms via Remez algoritmen med hänsyn till systemets dynamik. För att slippa brus från digitala processorer kaskadkopplas detta ihop med ett lågpassfilter av första ordningen. Med denna konfiguration uppnås ett fel som understiger 0.7%.

Ett annat mer regler tekniskt sätt att angripa problemet beskrivs av M. Tariq, W. Balachandran samt S. Song⁶. Här är tanken att en övergripande modell av en checkweigher inte är tillräckligt bra på att beskriva och identifiera orsaker till störningar i systemet. Man vill därför bryta ner det dynamiska systemet i logiska undersystem som förenklar analysen. Dessa undersystem är modeller av objekten som vägs, den lastmätande enheten och motor/transportbandssystemet. För att sedan kunna göra simuleringar av hur enheterna påverkar varandra använder man en mjukvara för reglersystem, till exempel Simulink.

Fördelen med detta angreppssätt är att det är mycket lätt att förändra och utvidga modellen utifrån vilken typ av checkweigher man studerar. Om man byter lastcellstyp eller väger viskösa föremål istället för solida behöver man bara göra ändringar i den aktuella modulen istället för att behöva omarbete hela den övergripande modellen från grunden. Även förbättrade algoritmer kan implementeras som liknande undermodeller. Som nackdel kan ses att det krävs betydande datorkraft för att få rimliga beräkningstider samt ett mycket stort arbete i initialskedet av modellbyggandet. Arbetssättet har därför sin största styrka i den klara förståelse den ger av de olika systemdelarnas inbördes samverkan.

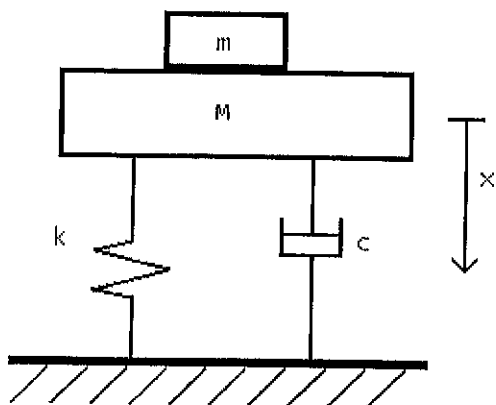
En hel del av den litteratur som vi läst betraktar problemet att bestämma ett objekts vikt medan det är i svängning. När ett objekt placeras på en vågplattform kan objektet och vågen vibrera en längre period. Att kunna ge en korrekt massa uppskattning utan att objektet har fått svänga in är därför av stort intresse. I en artikel av Shu, Wei-Qun⁷ beskrivs en sådan algoritm. Denna algoritm baseras på en modell av ett fjädersystem med dämpning. En lågpassfiltrering sker och brytfrekvensen sätts något högre än den högsta egenfrekvensen. Med hjälp av en rekursiv minsta kvadratmetod görs en parameter uppskattning utifrån vilken massan kan uppskattas. Med denna algoritm behövs endast en femtedel av den tiden som det tar att väga ett objekt statiskt och uppnå en noggrannhet på 1 %.

2 Modell av en checkweigher



Figur 2.1 – En skiss av en Teltek 800, den checkweigher som användes i denna rapport.

Målet med denna rapport är att utveckla en matematisk beskrivning av en checkweigher (figur 2.1), där det lätt går att identifiera vad som är relevant i problemet. Vi utgick från $F = ma$ och hade två kopplade ekvationer, en för objekt – vågbrygga och en för vågbrygga – lastcell⁵. Denna visade sig ha numeriska svårigheter och gav inga vidare resultat. Dessutom visade det sig senare att den checkweigher som skulle användas (figur 2.1) inte beskrevs bra av denna modell. Vi övergick istället till att ha endast en ekvation för systemet (figur 2.2).



Figur 2.2 – Den fysikaliska modellen av en lastcell med belastning

Efter lite mer eftertanke inses att Newtons andra lag i denna form inte är den lämpligaste i detta fall. Det krävs att massan är konstant, vilket den inte är. Objektet kommer att, under påmatningsfasen, belasta lastcellen mer och mer ända tills all massa ligger på vågbryggan. Samma sak gäller när objektet ska lämna vågbryggan. Låt oss istället utgå ifrån följande ekvationer.

$$F = \frac{d}{dt} P \quad (2.1)$$

$$F(t) = (M + m(t))g - kx - c\dot{x} \quad (2.2)$$

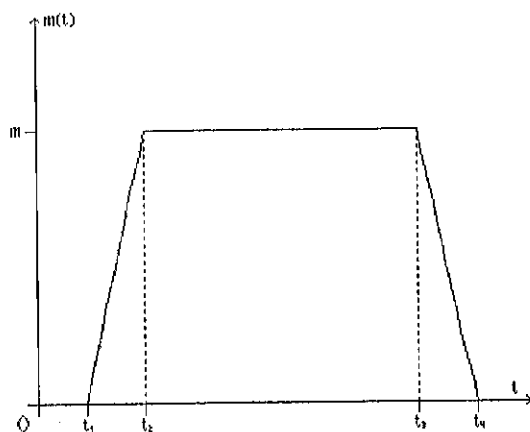
$$P(t) = (M + m(t))\dot{x}(t) \quad (2.3)$$

$$\ddot{x} + \frac{\dot{m} + c}{m + M} \dot{x} + \frac{k}{m + M} x = g \quad (2.4)$$

P är rörelsemängden, M är vågbryggans massa, m är objektets massa, g den lokala tyngdaccelerationen, k är systemets fjäderkonstant, c dess dämpning och x den vertikala rörelsen. Detta är en endimensionell modell och man får följaktligen bara med första ordningens egenfrekvens. I och med detta bortser vi från horisontella krafter som kan uppstå vid, till exempel, objektets passage över skarvar i transportbandssystemet. Vidare har vi antagit att det råder viskös dämpning, vilket är en approximation av de verkliga förhållandena.

$$m(t) = \begin{cases} 0 & t < t_1 \\ m \frac{t - t_1}{t_2 - t_1} & t_1 \leq t < t_2 \\ m & t_2 \leq t < t_3 \\ m \frac{t_4 - t}{t_4 - t_3} & t_3 \leq t < t_4 \\ 0 & t_4 \leq t \end{cases} \quad (2.5)$$

Ekvation 2.4 ger oss vår modell och 2.5 beskriver objektets passage över vågbryggan. Enligt 2.5 har massan ett tidsberoende, något som är en viktig faktor i problemet.



Figur 2.3 – Den modellerade massfunktionen $m(t)$.
 t_2, t_3 och t_4 kan härledas från t_1 , bandhastigheten och objektets längd.

Följden av att använda denna massafunktion (figur 2.3) blir att man bara kan betrakta solida föremål utan intern dynamik.

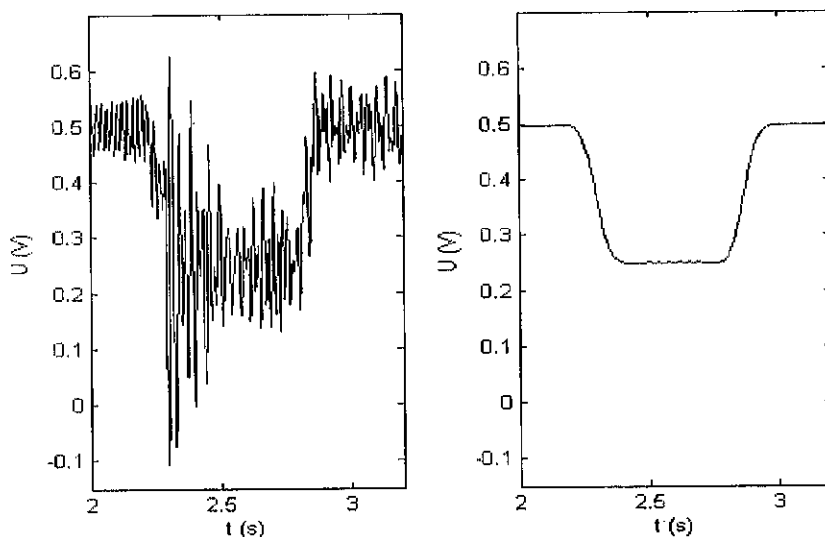
Ekvation 2.4 har en lösning $x(t)$ som beskriver lastcellens deformation. Vad som egentligen är intressant är massan som ger upphov till denna deformation. En sådan massa kan fås med hjälp av Hookes lag, $F = kx$. Vi omvandlar varje punkt, $x(t)$, till den skenbara massa, $F = m_{\text{skenbar}}g$, som skulle ge upphov till en sådan deformation. Den skenbara massan skulle konvergera till den sanna massan om systemet fick svänga in. Denna transformation förändrar inte problemet.

3 Mätsignalen

Lastcellen ger en elektrisk utsignal vilken beskriver hur stor deformation som sker. Den checkweigher som vi arbetat med har följande karakteristik. Mellan lastcell och mätpunkt finns en förstärkare och grundspänningen från ett obelastat system ligger på ungefär 0.5V. När man ökar belastningen minskar spänningen. Signalen antas vara direkt kopplad till massan med ett linjärt samband, detta utreds i kapitel 4.3. Dock tillkommer det mycket störningar. Det finns till exempel motorbrus, förstärkarbrus och egensvängningar i systemet. Hur egensvängningarna beter sig kan man få en bra uppfattning om från modellen som beskrivs av ekvation 2.4.

3.1 Filtrering

Det absolut vanligaste angreppssättet i de rapporter vi läst var filtrering. Vad man gör är att lågpasfiltrera mätsignalen för att ta bort både störningar och egenfrekvenser. Det man uppnår med detta är att den tidigare väldigt brusiga signalen nu består av en "snäll" svängning (se figur 3.1.1), och det blir relativt enkelt att bestämma en uppskattning av massan.



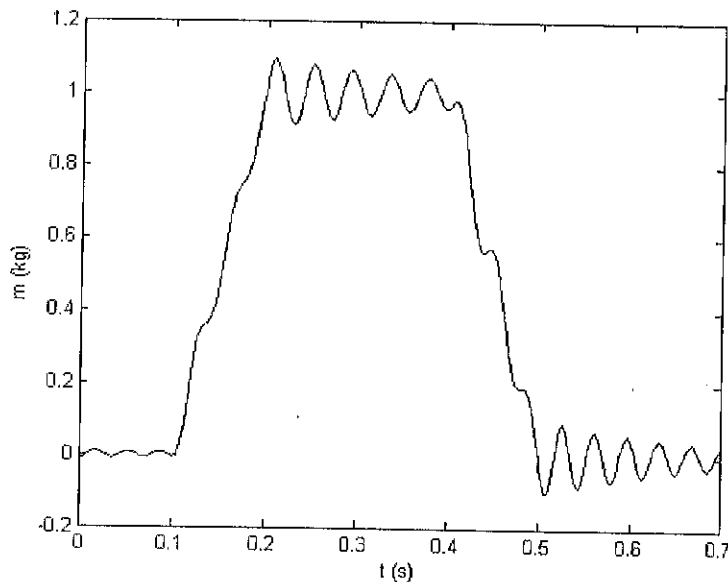
Figur 3.1.1 – Samplad samt filtrerad signal. $m = 0.413$ kg, $v = 30$ m/min

Vi kommer inte att använda oss av filtrering för att istället kunna använda all information som finns i mätsignalen. Detta kan man inte göra med en kraftigt lågpasfiltrerad signal då egenfrekvenser och annan information har förstörts. Om man istället försöker att bara filtrera bort brus och spara egenfrekvenserna stöter man också på problem. Relevant och irrelevant information kan ligga tätt i frekvensplanet, och man skulle då behöva ett filter av hög ordning för att inte klippa bort egenfrekvenserna. Detta låter knappast optimalt för ett problem som ska lösas i realtid⁸.

Det ska dock nämnas att en viss filtrering ändå måste ske. När man samplar mätdata sker detta med en samplingsfrekvens, f_s . Enligt Nyqvists sats⁹ kan man endast lita på frekvensinnehåll upp till $\frac{1}{2}f_s$. Därför måste det appliceras ett lågpasfilter för att den samplade signalen ska få någon mening.

3.2 Modell

Vi har nu en färdig modell (ekvation 2.4) för hur lastcellen och vågbryggan beter sig under störningsfri vägning av solida föremål (figur 3.2.1). Nästa steg är att utveckla en metod som utifrån en mätning kan bestämma massan på det vägda objektet. Låt oss titta på den tidsperiod då objektet belastar vågbryggan. Varje objekt får en stigtid, där hela massan inte ligger på vågen. Denna kommer att bli formberoende, och att ta hänsyn till detta i algoritmen medför komplikationer. Samma problem uppstår då objektet lämnar vågbryggan. Detta kan ske i olika faser av svängningen och beroende på detta kommer fjädersystemet att ge över- respektive underslag, och det kommer att bli svårigheter med en korrekt massauppskattning.

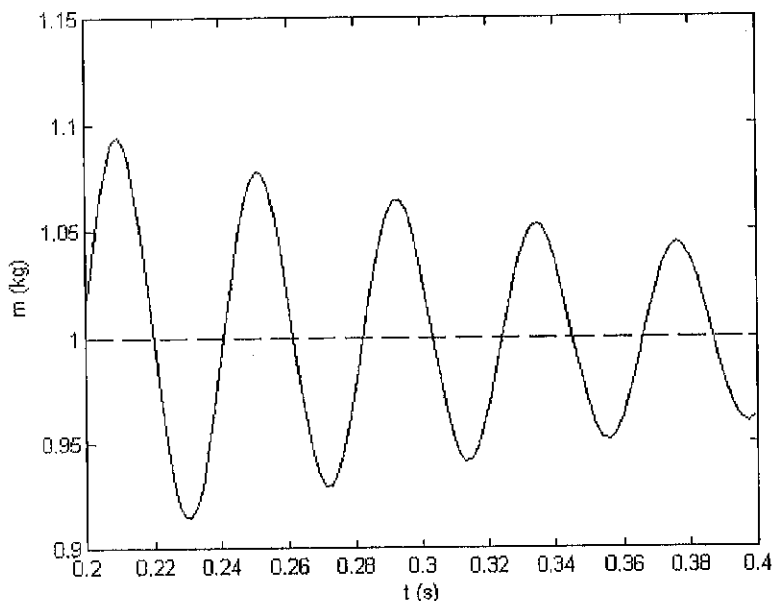


Figur 3.2.1 – En bild av en modellerad svängning, $v = 60$ m/min, $m = 1.0$ kg

Låt oss istället titta på den tidsperiod då hela objektet ligger på vågbryggan, ett angreppssätt som även är praktiskt utförbart vilket utreds närmare i kapitel 4. Vi får då en förenkling av problemet. Dels försvinner allt formberoende och man kan betrakta alla objekt som homogena rätblock, dels sker en förenkling av differentialekvation 2.4. Låt oss betrakta ekvation 2.5. Tidsintervallet är då begränsat till när $m(t) = m$. Ekvation 2.4 kan då skrivas

$$\ddot{x} + \frac{c}{m+M} \dot{x} + \frac{k}{m+M} x = g \quad (3.2.1)$$

Detta är en ordinär differentialekvation med konstanta koefficienter, och kan därför lösas explicit. Detta kommer vi att dra nytta av senare.



Figur 3.2.2 – En simulerad mätning. Den streckade linjen är den sanna massan. $v = 60$ m/min, $m = 1.0$ kg

Att bara titta på tidsintervall då $m(t) = m$ (se figur 3.2.2) får nu säkert en del att undra varför det gjordes en tidsberoende modell av en checkweigher i kapitel 2. I vår digitala simuleringsmiljö kan vi låta en serie objekt passera vågbryggan. Istället för att bestämma startvärden för n mätningar löser vi en ekvation för hela den tidsperiod som krävs för att n objekt ska ha passerat vågbryggan. Med kännedom om bandhastighet samt objektlängd kan sedan varje mätning extraheras.

Problemet med att bestämma startvärdet för en mätning ligger i att det kan ha passerat ett objekt innan. Om så är fallet befinner sig vågbryggan kanske i en egensvängning som är helt bestämd av föregående objekt. Föreställ dig nu att det har passerat n objekt, hur många av dessa kommer påverka ditt startvärde för objekt $n+1$? Vi anser detta problem vara avsevärt svårare jämfört med den tidigare differentialekvationen med tidsberoende massa.

3.2.1 Integration

Ett första försök att uppskatta objektets massa skulle kunna göras med integrering. Låt oss betrakta utsignalen när ett objekt befinner sig helt på vågbryggan. Ett lämpligt antagande är att signalen kommer att svänga kring ett jämviktsläge, och att detta kommer att ge en bra uppskattning av objektets massa. Vi vill alltså bestämma signalens jämviktsläge, eller rättare sagt, få ett medelvärde på mätningens (se figur 3.2.2) svängning. Medelvärdet ges av

$$x_{med} = \frac{1}{b-a} \int_a^b x(t) dt \quad (3.2.1.1)$$

där a och b är mätningens start respektive slutpunkter.

Denna algoritm har en given fördel, den är enkel och snabb. Eftersom mätsignalen består av en samling diskreta punkter kan integrationen utföras i realtid med till exempel trapetsmetoden. Detta leder till att oberoende av vilken hastighet man kör i, kan algoritmen alltid ge ett svar då objektet börjar lämna vågbryggan.

Hur bra det här än låter finns det vissa problem. För det första finns det inget som säger oss att bruset kommer vara normal- eller likformigt fördelat kring noll. Detta skulle innebära att bruset skulle bli en betydande faktor i vårt svar. Säg att bruset kommer in med en frekvens som kommer i fas med egensvängningarna. Detta skulle ha stor effekt på resultatet.

En annan svaghet som metoden har är att den behöver ett jämt antal perioder eller en lång mätperiod för att få ett bra medelvärde. Detta strider mot vårt mål att höja hastigheten så mycket som möjligt, vilket kommer att ge kortare mätperioder och färre svängningar. Det finns heller inga villkor som lovar att man börjar och slutar på samma punkt i en period.

3.2.2 Icke-linjär optimering

Som vi just sett är integrering inte den bästa angreppsmetoden. Betrakta istället ekvation 3.2.1. Denna ekvation beskriver en mätning, det vill säga när $m(t)=m$. Tidigare i kapitlet nämndes att (3.2.1) har en explicit lösning. Detta utnyttjar vi för att hitta en bas till Lösningssrummet. Vi antar följande Lösningsform

$$x(t) = Ae^{-\lambda_1 t} + Be^{-\lambda_2 t} + C \quad (3.2.2.2)$$

$$\lambda_{1,2} = \frac{c}{2(m+M)} \pm \frac{1}{2(m+M)} \sqrt{c^2 - 4(m+M)k} \quad (3.2.2.3)$$

$$C = x_p(t) = \frac{(m+M)g}{k} \quad (3.2.2.4)$$

Vågbryggans massa, M , anses vara konstant. Det finns då tre fria variabler, c , k och m . Vid en analys av partikulärlösningen (3.2.2.4) finner man att C medför en bra uppskattning av objektets massa. k är en mer lättbestämd variabel än c . Följaktligen kommer C att vara ett säkrare val än λ som innehåller två uppskattade variabler. Vi slipper även undan de numeriska svårigheter vi annars skulle ha haft.

Det har ännu inte tagits hänsyn till brus i modellen. Låt oss säga att det finns nätbrus i signalen, på formen $\alpha \sin(\beta t + \chi)$. Detta kan modelleras genom att omformulera (3.2.2.2) på följande sätt.

$$\begin{aligned} x(t) &= Ae^{-\lambda_1 t} + Be^{-\lambda_2 t} + C + D \sin(\lambda_3 t) + E \cos(\lambda_3 t) \\ \lambda_1, \lambda_2 &\in \mathbf{C} \\ \lambda_3 &\in \mathbf{R} \end{aligned} \quad (3.2.2.5)$$

Vad som händer är att vår partikulärlösning får in ett cosinus- sinuspar med en fri variabel, frekvensen λ_3 . Detta angreppssätt kan lätt modifieras till att ta med flera störningsfrekvenser från till exempel vibrationer. Ekvation (3.2.2.5) blir, för två störningsfrekvenser, vilket vi även använder i vår modell

$$x(t) = Ae^{-\lambda_1 t} + Be^{-\lambda_2 t} + C + D \sin(\lambda_3 t) + E \cos(\lambda_3 t) + F \sin(\lambda_4 t) + G \cos(\lambda_4 t) \quad (3.2.2.6)$$

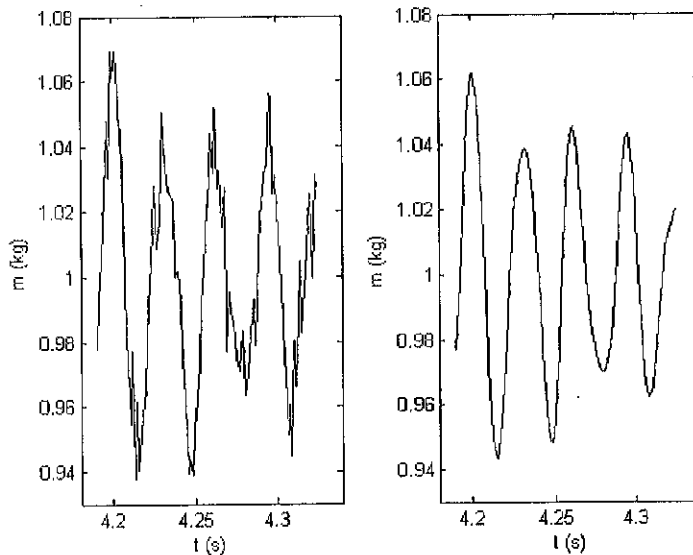
Vår idé är att anpassa en funktion på denna form till de sampelvärden som vi får från vår modell eller från verkliga mätningar. Ett sätt att göra detta är att använda en icke-linjär regressionsmodell¹⁰, till exempel minsta kvadratmetoden. Om y är mätningens sampelvärden kan en målfunktion för (3.2.2.6) definieras enligt följande

$$z = \min_{\psi, \lambda} \sqrt{\sum_i |x(t_i) - y_i|^2}$$

$$\psi = \{A, B, C, D, E, F, G\} \quad (3.2.2.7)$$

$$\lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$$

Det finns flera olika metoder för att lösa icke-linjära optimeringsproblem. Då vi arbetat i Matlab föll valet på rutinen `fminsearch` som är en implementation av Nelder-Meads metod¹¹. I figuren nedan kan man se en modellerad signal samt hur optimeringsmetoden anpassar sin lösning till denna.



Figur 3.2.2.1 – Till vänster är en modellsignal med vitt brus samt nätbrus, till höger dess anpassning.
 $v = 60$ m/min, $m = 1.0$ kg

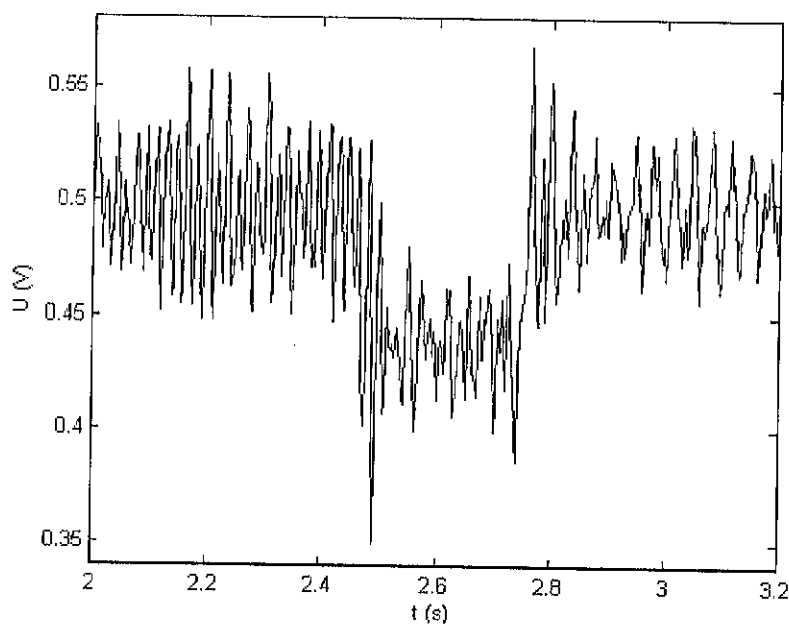
En viktig faktor i optimering är startvärden. En dålig startapproximation kan göra problemet så illakonditionerat att det inte konvergerar. Det visar sig även att det finns mycket att vinna på att ha bra startvärden i fråga om konvergenstakt och svarsnoggrannhet. I vårt fall, då man gör beräkningar i realtid, är det speciellt kritiskt att få korta beräkningstider.

Approximationer till fjäderkonstant och dämpning, k och c , kan bestämmas ur checkweigherns konstruktion medan massa och frekvens är betydligt svårare. Hur ska man kunna bestämma en skattning på den variabel man söker? Vad som behövs är en simpel men hyfsat noggrann metod för att uppskatta m . Här kommer integreringen (kapitel 3.1) åter in i bilden. Denna algoritim är snabb och ger i de flesta fall en god uppskattning av massan.

Hur ska man skatta frekvenserna? Detta får göras med hjälp av erfarenhet och spektralanalys. Det är ytterst troligt att motorerna kommer ge ett visst bidrag i form av nätbrus och detta ligger kring 50 Hz. Man skulle även kunna tänka sig att vibrationer är en väsentlig störningskälla, mer om detta i kapitel 4.

4 Mätdata

För att kunna undersöka vår modells effektivitet i verkliga situationer har vi fått låna en checkweigher från Teltek. Från denna kunde det uppmätas en förstärkt signal på lastcellens utspänning, $u(t)$. För att få en någorlunda vibrationsfri påmatning användes ett extra påmatningsband, av längd cirka 4 m, som drevs av en likspänningsmotor. Denna påverkade dock inte våra mätvärden då den inte stod i direkt kontakt med checkweighern. Mätningarna gjordes med en multimeter av modell Keithley 2700. Samplingsfrekvensen låg på ungefär 350 Hz. Utdata gavs med minst fem decimalers upplösning. Mätinstrumentet kopplades till en dator som lagrade samplingarna i Excel. För att frekvensinnehållet i mätningarna skall gå att lita på görs en lågpasfiltrering vid Nyquistfrekvensen på multimeters ingång. Ett exempel på utdata ges i figur 4.1.



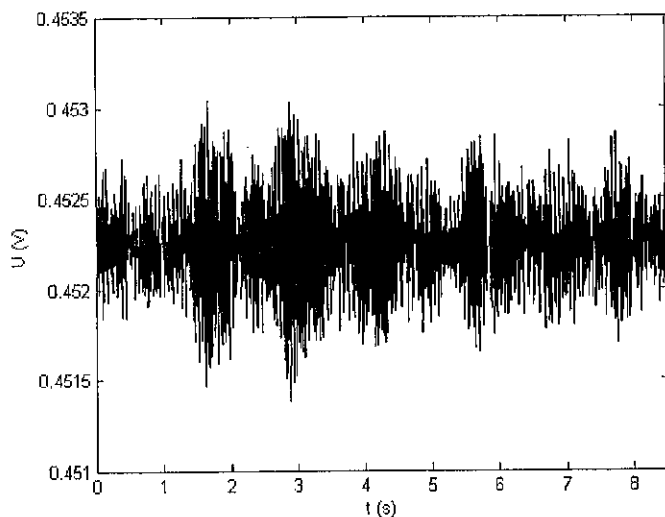
Figur 4.1 – Samplad signal, $v = 60$ m/min, $m = 0.1$ kg

I den första mätomgången samlades data in under kontinuerliga tidsperioder på 10 sekunder vardera. Vikterna placerades ut med ett sådant avstånd så att två föremål aldrig befann sig på vågbryggan samtidigt. Vi varierade främst två variabler, föremålens vikt samt bandhastighet.

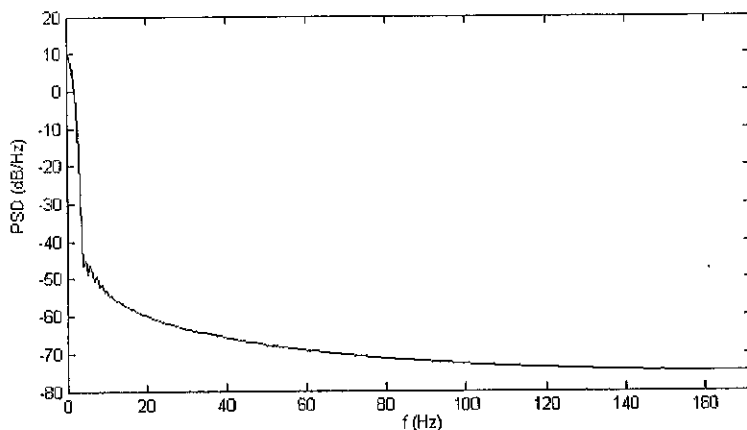
Modellens algoritmberäkningar sker enbart då $m(t) = m$. För att översätta denna angreppsvinkel till verkligheten måste mätningarna triggas. På checkweighern, alldeles före vågbryggan, sitter som bekant en fotocell (se figur 1.1). Denna ger ut en signal när den bryts. Det går då med kännedom om bandhastighet och föremålens längd att avgöra exakt när en mätning ska ske. Följaktligen har man all information som behövs för att kunna ta ut en mätning. I mätomgång två var tanken att denna strategi skulle användas. På grund av en begränsning i multimeters programvara var detta inte möjligt, och mätningarna fick istället plockas ut för hand. Den enda skillnaden från den första mätomgången är att vi använt oss av vikter vars massa är spårbar ner till fem värdesiffror.

4.1 Förstärkarbrus

Det nämndes tidigare att vi fick ut en förstärkt signal. Är förstärkaren en väsentlig brus-källa som man måste ta hänsyn till? För att undersöka detta gjordes en mätserie då lastcellen inte hade någon extra belastning och motorena var avstängda.



Figur 4.1.1 – Förstärkarbrus.



Figur 4.1.2 – Spektrum av förstärkarbrus.

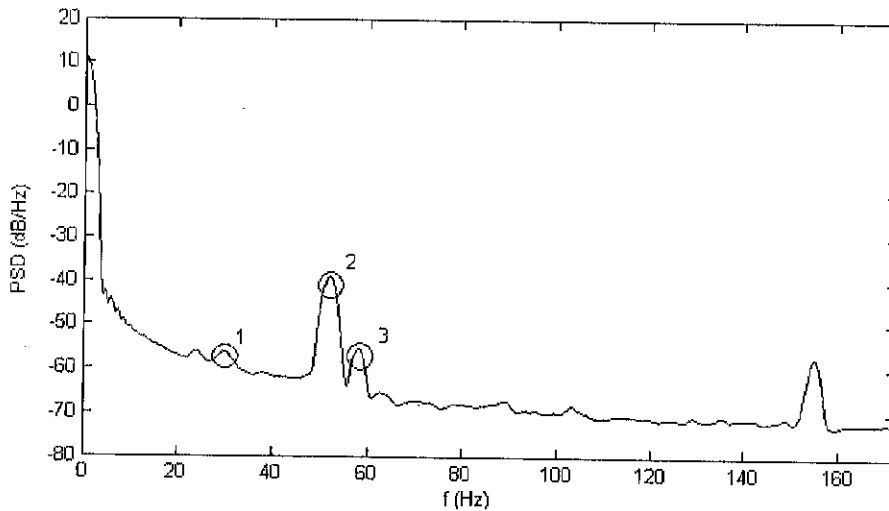
Som man ser i figur 4.1.1 är förstärkarbrusets amplitud mycket låg, mindre än 0.001 V, som motsvarar ungefär 0.001 kg. Vidare ser vi i figur 4.1.2 att bruset är relativt jämnt fördelat över hela spektrat. Vi tar därför inte hänsyn till förstärkarbruset utan anser detta vara vitt brus.

4.2 Störningskällor

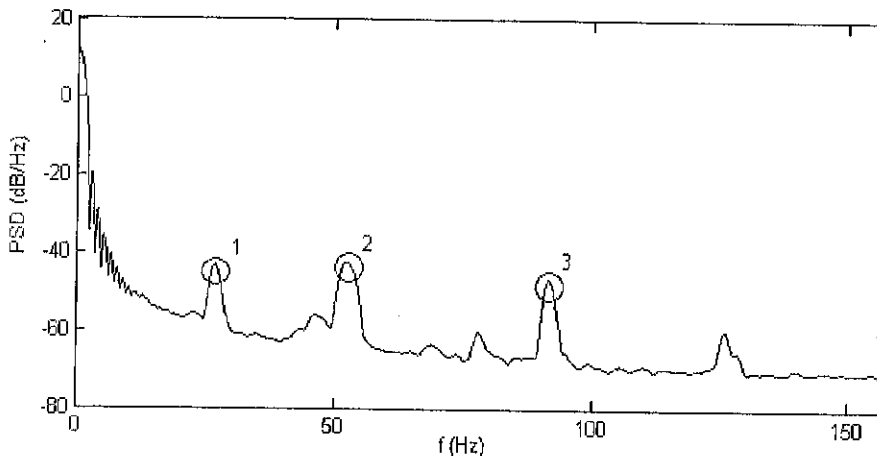
Övriga störningskällor antas, i enlighet med kapitel 3.2.2, att bestå av nätbrus samt vibrationer. Om den mekaniska konstruktionen vibrerar med en frekvens f_v , så kan man tänka sig att detta påverkar lastcellen och ger upphov till störningar. Efter att ha studerat spektra för en obelastad vågbrygga med bandhastigheterna 30 respektive 60 m/min kan det konstateras att frekvensen för nätbrus ligger fixt medan vibrationernas frekvens är

hastighetsberoende. Amplituden för vibrationerna växer med hastigheten, medan nätbrusets storlek är relativt fix.

Det svenska elnätet har som bekant en frekvens på 50 Hz. Eftersom motorerna på checkweighern drevs av nedtransformerad växelspanning, kommer detta in som en störkälla. Detta syns vid en spektralanalys av en mätsignal gjord under samma villkor som ovan.



Figur 4.2.1 - Spektrum av motor- och bandbrus. $v = 30$ m/min.



Figur 4.2.2 - Spektrum av motor- och bandbrus. $v = 60$ m/min.

Spektrumet i figur 4.2.2 har tre distinkta toppar. Topp 1 kommer från systemets egensvängning. Denna kan härledas från ekvationen¹²

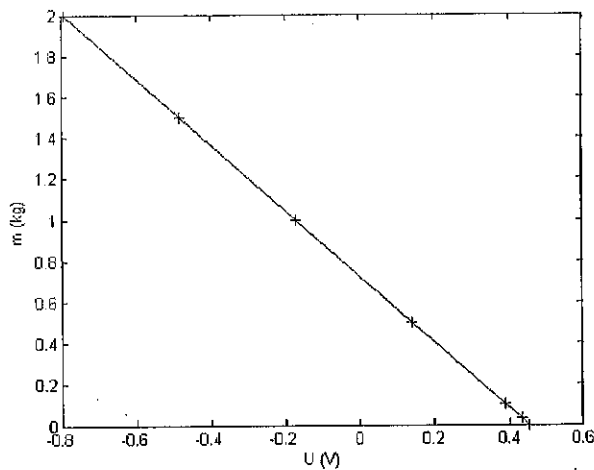
$$f_{system} = \frac{1}{2\pi} \sqrt{\frac{k}{M}} \quad (4.2.1)$$

där k är systemets fjäderkonstant och M är vågbryggans massa. I vårt system var k ungefär 80000 kg/s^2 , medan M var dryga 2,5 kg. Detta motsvarar ungefär 30 Hz för ett obelastat system. Topp 2, som ligger runt 50 Hz, antas komma från nätbrus och topp 3 är vad vi kallar vibrationer. Dessa vibrationer kan härledas till bandens drivmekanism.

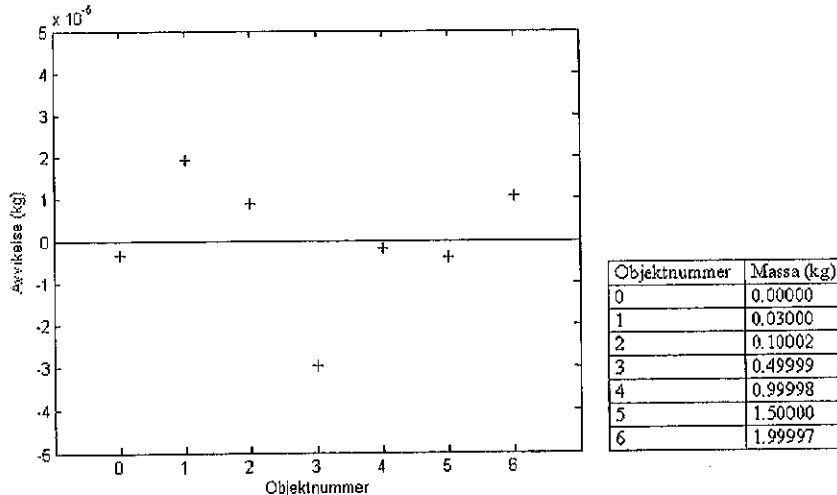
I figur 4.2.1 kan man även se en fjärde topp. Vi har inte tagit hänsyn till denna då dess amplitud är såpass mycket mindre än den i problemet dominerande frekvensens. I figur 4.2.1 har störningen en PSD (Power Spectral Density) som är en faktor 20 mindre än topp 2, och så även i figur 4.2.2. Då målet är att kunna köra på en så hög bandhastighet som möjligt får denna störningskälla en mindre betydande roll i problemet. Detta kan inte anmärkas på topp 1 eller 3 då dessa blir betydligt viktigare vid högre bandhastigheter. När man betraktar ovanstående figurer ska man komma ihåg att decibel är en logaritmisk skala.

4.3 Omvandling spänning – massa

Som tidigare nämndes i kapitel 3 får man ut en spänningssignal från lastcellen vilken sedan förstärks. Eftersom vi är intresserade av objektens massa skall signalen omvandlas. Utifrån sju mätpar (massa, spänning), vilka täcker hela mätintervallet, interpoleras en funktion på formen $y = ax + b$. Massorna vägdes på en våg med fem siffrors noggrannhet. Spänningen uppmättes statistiskt, det vill säga utan att motorerna har varit påslagna. Objektet placerades på vågbryggan och spänningvärden samplades under 10 s. Från dessa samplings bildades ett medelvärde och detta värde användes i interpoleringen. Som bilderna nedan visar är överensstämmelsen god, och en linjär regression ger korrelationsfaktorn -1.0000.



Figur 4.3.1 – Samband spänning-massa.



Figur 4.3.2 – Avvikelse mellan uppmätta och beräknade massor utifrån det linjära sambandet

5 Resultat

Vi har i princip inriktat oss på två mål. Dels att få en god modell som fungerar bra i datasimuleringar, dels att få en stabil algoritm för snabba och korrekta massauppskattningar i verkliga körningar. Som felmått har vi använt oss av det relativa felet som definieras enligt följande

$$err = \left| \frac{m_{sann} - m_{uppskattad}}{m_{sann}} \right| \quad (5.1)$$

5.1 Modellkörning

När det gäller datormodellen så har vi implementerat en fullständig rutin för hur lastcellen svarar på en serie rätblock. Denna serie av objekt kommer att ha tre gemensamma egenskaper: massa, längd samt inbördes avstånd. Det finns även möjlighet att styra vågens egenskaper i form av fjäderkonstant samt dämpning för eventuell utprovning av andra konstruktioner.

En egenskap hos problemet som bör uppmärksammas är att vid korta mätningar, runt 50 sampelpunkter, kommer de optimerade variablerna, λ_1 till λ_4 , att få varierande noggrannhet. Konstanten C kommer dock att vara välbestämd. Vid långa mätningar, fler än 100 samplings, kommer λ_1 till λ_4 att uppskattas väl medan C :s noggrannhet minskar. Denna dualitet kommer att skapa problem då den variabel som används för massauppskattningen blir beroende av mätsignalens längd. Detta beror på implementationen av finsearch, vilken optimerar över mängden $\{\lambda_1 \dots \lambda_4\}$ och bestämmer koefficienterna $\{A \dots F\}$ genom att lösa ett överbestämt linjärt system. Detta medför att $\{\lambda_1 \dots \lambda_4\}$ kommer att uppskattas bättre för stora datamängder. För att komma förbi detta har vi bortsett ifrån information som kommer efter n sampelpunkter. För att kunna få med så mycket information som möjligt utan att tappa noggrannhet har vi utifrån en mätning i 10 m/min experimenterat oss fram till ett bra värde. Det visar sig att $n = 100$ är en bra övre gräns för n . En typisk mätning, med en bandhastighet på 60 m/min, har ungefär 60 sampelvärden och i detta fall kommer vi använda all information. På detta sätt kommer C alltid vara en bra uppskattning av massan, och i vissa fall kommer beräkningarna att kunna påbörjas redan innan objektet lämnat vågbryggan. För att kunna använda λ_1 till λ_4 för massauppskattningen vid korta mätningar hade vi varit tvungna att sampla mycket tätare, något som kan bli alldeles för dyrbart. En annan nackdel är att problemets beräkningskomplexitet skulle öka då vi får fler sampelpunkter att betrakta.

Algoritmens stabilitet och flexibilitet har testats i olika datorkörningar. Om man betraktar fall med bra uppskattningar på k och c så har massan kunnat varieras över hela vårt tillåtna intervall (0-2 kg) samt hastigheten mellan 6 och 120 m/min. Beräkningstiderna varierar från 0.1 till 0.3 s. Detta är dock i Matlabkod och kan förbättras avsevärt om rutinen kompileras före körning. Det relativa felet ligger i allmänhet runt 10^{-4} (se figur 5.1.1). Det går även att öka bandhastigheten ytterligare utan någon avsevärd försämring av noggrannheten.

```

>> run
Solving ODE
Solving routine started
n = 1
Sann massa      (kg): 1.0000000
Beräknad massa (kg): 1.0003969
Relativt fel    : 3.97e-004
Beräkningstid  (s): 0.172
n = 2
Sann massa      (kg): 1.0000000
Beräknad massa (kg): 0.9996102
Relativt fel    : 3.90e-004
Beräkningstid  (s): 0.218
n = 3
Sann massa      (kg): 1.0000000
Beräknad massa (kg): 1.0023309
Relativt fel    : 2.33e-003
Beräkningstid  (s): 0.187
Complete

```

Figur 5.1.1 – En typisk modellkörning med störningar från nät, vibrationer samt vitt brus. $m=1.0$ kg, $v=30$ m/min

5.2 Empiriska resultat

Tanken var att mätomgång två skulle ske med hjälp av trigging. Så blev det dock inte på grund av en begränsning i programvaran för mätinstrumentet. Mätningarna har därför fått plockats ut för hand vilket inför ytterligare en felkälla. Massauppskattningsalgoritmen är en modifikation av modellalgoritmen. Vi samlar data under en period på 10 s då vi låter fyra objekt av ungefär samma massa passera över vågbryggan. Objekten färdas först på ett 5 m långt påmatningsband, som går med samma hastighet som banden på checkweighern, allt för att minska störningar vid objektpåmatningen. Mätvärden återfinns i bilaga 1.

Det uppstod ytterligare ett problem vid mätningarna. Transportbandet på vågbryggan var av ett neoprenliknande material. När stora massor skulle vägas, $m = 1$ kg och större, stannade bandet upp ett kort ögonblick på grund av alltför hög friktion mot underlaget. Effekten blev märkbar för låga hastigheter, 30 m/min och lägre. Att väga tyngre föremål i hastigheten 10 m/min blev därför omöjligt, då bandet inte orkade förflytta objektet.

Vi börjar med att betrakta fallet då $v = 60$ m/min. Det ska nämnas att massan 0.03 kg inte kunde mätas då dess amplitud inte kunde urskiljas gentemot övrigt brus, något vi hade klarat om mätningen triggats. Resultatet blev i allmänhet ett relativt fel på 10^{-3} och algoritmen var som stabilast vid massor runt 1 kg. Detta är väntat då avståndet till lastcellens maxbelastning (2 kg + vågbrygga) är stort, samtidigt som objektets utslag är väsentligt större än brusets. Då $v = 30$ m/min uppnådde vi en liknande noggrannhet. Vi ser dock en trend att för större massor får vi säkrare värden. Detta kan vara ett resultat av noggrannheten hos multimetern, då vi endast kan lita på fem decimaler.

Det gjordes även två mätrundor på 10 m/min för att se om algoritmen var stabil även här. Som tidigare nämdes omöjliggjordes en utförligare tabell av bandegenskaper. Resultaten blev mycket goda, precis som man förväntar sig. Det relativa felet sjönk över lag med en faktor 10.

Vi testade även algoritmens robusthet genom att köra på diverse olika villkor. Optimeringsmetoden konvergerar alltid då man försäkrat sig att objektet ligger helt på vågen. Det har även testats att starta en beräkning på inledning samt avslutning av en mätning. Vi har iakttagit att den är aningen känslig i början på en mätning. Detta på grund av det mellanrum som finns mellan påmatningsband och vågbrygga, vilket ger upphov till spikar i signalen. Följaktligen bör en beräkning påbörjas en bit in i en mätning då objektet fått en viss insvängningstid.

6 Slutsats

Vi har skapat en datormodell av en checkweigher. Från denna får man ut en simulerad mätsignal. Överensstämmelsen mellan modell och verklighet är god, dock kan det göras förfiningar i brussimuleringen.

Vi har även skapat en algoritm för att utifrån en mätsignal uppskatta massan av det vägda objektet. Algoritmen är snabb, en uppskattning tar oftast inte mer än 0.2 s att utföra. Detta är mindre än den tid det tar för ett objekt att passera vågbryggan. Lägg även märke till att dessa beräkningstider är från Matlab, som interpreterar rad för rad, och inte är färdigkompilerad kod. Om rutinerna, till exempel, skrivits i Fortran eller C/C++ kan man förvänta sig en avsevärd prestandaförbättring.

Vidare ger algoritmen mycket god noggrannhet i modellfall utan brus, vanligen 10^{-5} till 10^{-7} i relativt fel. Noggrannheten bibehålls även då man lägger på periodiska brusceller, men beräkningstiderna ökar något. Om man applicerar vitt brus med verklighetsliknande amplitud till den simulerade signalen sjunker noggrannheten något, mestadels ligger den mellan 10^{-3} och 10^{-5} .

Algoritmen fungerar även tillfredsställande i empiriska försök. I de undersökta hastigheterna, 60 m/min och lägre, kan man förvänta sig ett relativt fel i storleksordningen 10^{-3} . För lägre hastigheter kan man dock nå så lågt som 10^{-5} . Dessa resultat uppfyller den internationella rekommendationen OIML R 51 för X(x)-klassade checkweighers². Resultaten är även fullt jämförbara med dem som uppnåtts i övriga artiklar, t.ex. Yamazaki, T. m. fl.⁵.

Algoritmen är även robust gällande mätningens start- och slutpunkt, fast bäst prestanda fås om man ger objektet en viss insvängningstid.

6.1 Vidare utveckling

Även om vi har fått fram en slutlig ”produkt” så finns det saker kvar att tänka över och förbättra. Nedan följer en lista med sådana problem.

- Oberoende massor i modellen

Det kan vara intressant att se hur vågens egensvängningar beter sig då man, till exempel, först låter ett tungt objekt passera för att sedan låta ett lättare objekt vägas. Ger detta upphov till problem? Dämpas vågbryggans och lastcellens egensvängningar tillräckligt snabbt?

- Olika avstånd mellan objekten

I verkliga körningar är det väldigt sällan (aldrig) så att objekt kommer med ett fixt avstånd. Det hade därför varit intressant att implementera varierande avstånd mellan objekten.

- Val av optimeringsmetod

Användandet av Nelder-Meads metod motiveras av att det var den som fanns implementerad i Matlab, samt att den ger godtagbara resultat även på icke-konvexa problem. Dock finns det vissa problem med till exempel konvergens¹¹. Man skulle kunna utreda andra metoders egenskaper/effektivitet för denna problemställning.

- Överföringen av startvärden

Det finns en inbyggd rutin i modellen för att överföra uppdaterade startvärden mellan två massauppskattningar, för att på detta sätt nå snabbare konvergens. Vad som skulle behöva implementeras är ett skydd mot att dåliga värden förs vidare.

7 Källförteckning

1. <http://www.sp.se/metrology/pack/sv/forpackn.htm>
2. <http://www.oiml.org/>
3. King, Anne, "An introduction to weighing systems", *Electronic Technology* Vol21 February 1987, 1987, p 25-29
4. Shi, W.J., White, N.M., Brignell, J.E., "Adaptive filters in load cell response correction", *Sensors and Actuators A*, 37-38, 1993, p 280-285
5. Yamazaki, T., Sakurai, Y., Kurosu, S., Ohnishi, H. and Kobayashi, M., "Continuous mass measurement in checkweighers and conveyor belt scales", *VDI-BERICHT* nr.1685, p 295-301
6. Tariq, M., Balachandran, W., Song, S., "Checkweigher modeling using dynamical subsystems", *IEEE*, 1995, ISBN 0-7803-3008-0, p 1715-1722
7. Shu, Wei-Qun, "Dynamic weighing under nonzero initial conditions", *IEEE transactions on Instrumentation and Measurement* Vol42 no.4 August 1993, 1993, p 806-811
8. Chen, Chi-Tsong, "Digital signal processing – spectral computation and filter design", Oxford University Press, 2001, ISBN 0-19-513638-1
9. Bracewell, Ronald N, "The Fourier transform and its applications – 3rd edition", McGraw-Hill, 2000, ISBN 0-07-303938-1
10. Danaci, Mustafa, "Platform parameters identification by nonlinear regression method", *Simulation and Modelling. Enablers for a Better Quality of Life. 14th European Simulation. Multiconference 2000*, 2000, p 811-818
11. Lagarias, Jeffrey C., "Convergence Properties Of The Nelder-Mead Simplex Method In Low Dimensions", *Siam J. Optim* Vol9, no.1 December 1998, 1998, p 112-147
12. Hjelmgren, Jan, "Dynamic Measurement of Pressure – A Literature Survey", SP Swedish National Testing and Research Institute, SP Report 2002:34
13. Pärt-Enander, Eva och Sjöberg, Anders, "Användarhandledning för Matlab 6", Uppsala Universitet, 2001, ISBN 91-506-1473-8

Appendix

Bilaga 1 – Empiriska mätresultat

$v = 60$ m/min

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.10002	0.0998308	1.89e-003	0.156
2	0.10003	0.1000656	3.56e-004	0.187
3	0.10003	0.0995783	4.52e-003	0.172
4	0.10004	0.1000186	2.14e-004	0.188
Medelvärde			1.75e-003	

$v = 60$ m/min

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.49999	0.4971821	5.62e-003	0.187
2	0.50000	0.5010818	2.16e-003	0.188
3	0.50001	0.4967240	6.57e-003	0.141
4	0.50002	0.4982845	3.47e-003	0.172
Medelvärde			4.46e-003	

$v = 60$ m/min

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.99998	1.0036015	3.62e-003	0.212
2	1.00002	1.0001616	1.42e-004	0.203
3	0.99998	0.9964849	3.50e-003	0.172
4	0.99999	0.9957624	4.23e-003	0.234
Medelvärde			2.87e-003	

$v = 60$ m/min

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	1.50000	1.5041511	2.77e-003	0.189
2	1.49997	1.5056309	3.77e-003	0.218
3	1.50002	1.5052359	3.48e-003	0.193
4	1.50004	1.4993855	4.36e-004	0.201
Medelvärde			2.61e-003	

$v = 60$ m/min

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	1.99997	1.9955549	2.21e-003	0.225
2	1.99994	2.0022654	1.16e-003	0.216
3	2.00000	2.0044111	2.20e-003	0.212
4	2.00002	2.0072523	3.62e-003	0.197
Medelvärde			2.30e-003	

$\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.03000	0.0300701	2.34e-003	0.241
2	0.03004	0.0299417	3.27e-003	0.258
3	0.03001	0.0303523	1.14e-002	0.263
4	0.03002	0.0302092	6.30e-003	0.281
Medelvärde			5.75e-003	

 $\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.10002	0.1008143	7.94e-003	0.214
2	0.10003	0.1000708	4.08e-004	0.245
3	0.10003	0.1004429	4.13e-003	0.161
4	0.10004	0.0998980	1.42e-003	0.204
Medelvärde			3.47e-003	

 $\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.49999	0.5002932	6.06e-004	0.157
2	0.50000	0.5019115	3.82e-003	0.187
3	0.50001	0.5013368	2.65e-003	0.187
4	0.50002	0.5006340	1.23e-003	0.188
Medelvärde			2.08e-003	

 $\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.99998	1.0022110	2.23e-003	0.198
2	1.00002	0.9983797	1.82e-003	0.201
3	0.99998	1.0013903	1.41e-003	0.281
4	0.99999	1.0001552	1.65e-004	0.187
Medelvärde			1.16e-003	

 $\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	1.50000	1.5006188	4.13e-004	0.245
2	1.49997	1.4986128	9.05e-004	0.178
3	1.50002	1.5063278	4.24e-003	0.231
4	1.50004	1.5018998	1.24e-003	0.211
Medelvärde			1.70e-003	

 $\nu = 30 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	1.99997	2.0063225	3.18e-003	0.263
2	1.99994	2.0031824	1.62e-003	0.275
3	2.00000	2.0001182	5.91e-005	0.214
4	2.00002	1.9956422	2.19e-003	0.228
Medelvärde			1.76e-003	

$\nu = 10 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.10002	0.1000220	2.02e-005	0.231
2	0.10003	0.1001230	9.29e-004	0.205
3	0.10003	0.1000204	9.62e-005	0.243
4	0.10004	0.1000784	3.84e-004	0.253
Medelvärde			3.57e-004	

$\nu = 10 \text{ m/min}$

Objektnummer	Sann vikt (kg)	Uppskattad vikt (kg)	Relativt fel	Beräkningstid (s)
1	0.49999	0.4997001	5.80e-004	0.260
2	0.50000	0.4997272	5.46e-004	0.219
3	0.50001	0.4998891	2.42e-004	0.217
4	0.50002	0.5002061	3.72e-004	0.221
Medelvärde			4.35e-004	

Beräkningarna är utförda på följande system.

Pentium 4 2.4GHz
768Mb RAM
Matlab 6.5

Bilaga 2 – Matlabkod

- run.m

Vår huvudrutin. Här initieras alla variabler. För ingående beskrivning, se respektive avsnitt. En del rader har blivit isärskrivna med ... på grund av typsnittet. Denna är för modellkörning, men iden kan lätt göras om för verkliga körningar.

```

global kf                %% Minsta kvadratkoefficienterna

M = 2.5;                %% Upphängningens massa (kg)
g = 9.82;               %% Gravitationskonstanten (m/s^2)
c = 30.0;               %% Upphängningens dämpningskoefficient (kg/s)
k = 79149.1;           %% Upphängningens totala fjäderkonstant (kg/s^2)
m = .05;                %% Objektets sanna massa (kg)
v = 3;                  %% Bandets hastighet (m/s)
s = 0.4;                %% Mätsträckans längd (m)
L = 1;                  %% Avståndet mellan två vikter (m)
objekt = 0.1;           %% Objektets längd (m)
shift = 0.1;            %% Den tid det tar innan första vikten åker på
                        %% bandet (s)

delta_x = objekt/s;     %% Den procentuella lutningen på massfunktionen (%)
T = 5;                  %% Mätningens totala tid (s)
ampl1 = 0.1+0.1*v;      %% Nätbrusets amplitud
amp2 = 0.1+0.1*v;      %% Vibrationernas amplitud
fas1 = rand*2*pi;       %% Nätbrusets fasvinkel
fas2 = rand*2*pi;       %% Vibrationernas fasvinkel
f1 = 50+2*randn;        %% Nätbrusets frekvens
f2 = v*100+2*randn;     %% Vibrationernas frekvens (hastighetsberoende)

%% Differentialekvationen
disp('Solving ODE');
[t,y]=ode23('model2',[0 T],[1.9*M*6.5e-5 0],[], ...
M,g,c,k,m,v,s,L,shift,delta_x,ampl1,fas1,f1,amp2,fas2,f2);

%% Konvertering från töjning till massa
y(:,1)=k*y(:,1)/g-M;

%% Modellering av fotocell
[y(:,1),n,t_index]=plockaut(y(:,1),t,v,s,L,shift,delta_x);

%% Integrationsalgoritmen
area=abs(trapz(t(t_index(1):t_index(2)),y(t_index(1):t_index(2),1)));
vikt=area/(t(t_index(2))-t(t_index(1)));
sann_vikt=m;

%% Sätter upp begynnelsevärden
disp('Solving routine started');
a(1)=vikt;              %% Massa
a(2)=28;                %% Dämpningskoefficient
a(3)=80000;             %% Fjäderkonstant
a(4)=50;                %% Nätfrekvens
a(5)=v*100;             %% Vibrationernas frekvens
resultat=[];           %% Här sparas svaren efter en körning
warning off;
options = optimset('TolX',0.1);
nn=100;                 %% Max antal sampelvärden

```

```

%% Uppskättningsmetoden
for i=1:n
    temp1=[];
    temp2=[];
    disp(['n = ',num2str(i)]);
    t1=t_index(2*i-1);
    t2=t_index(2*i);
    temp1=t(t1:t2);
    temp2=y(t1:t2,1);
    temp2=temp2+0.01*randn(length(temp2),1);

%% Nedsampling vid låga hastigheter
    if length(temp1) > nn
        temp1=temp1(1:nn);
        temp2=temp2(1:nn);
    end

%% Plottar för debugsyfte. För att använda dessa, byt rad
%% 'estimated_lambda = ...'. Se även func.m
    %figure(3);
    %subplot(1,2,1);plot(temp1,temp2);
    %subplot(1,2,2);h=plot(temp1,temp2);

    start = a;

    tic;
    %estimated_lambda = fminsearch('func',start,options,temp1,temp2,M,h);
    estimated_lambda = fminsearch('func',start,options,temp1,temp2,M);
    tid=toc;
    minsta_kvadrat_vikt=norm(kf(3));
    relative_error=abs(sann_vikt-minsta_kvadrat_vikt)/sann_vikt;
    resultat=[resultat [minsta_kvadrat_vikt relative_error tid]];

%% Åskådliggörande av resultat
    str1=sprintf('Sann massa      (kg): %1.7f',sann_vikt);
    str2=sprintf('Beräknad massa (kg): %1.7f',minsta_kvadrat_vikt);
    str3=sprintf('Relativt fel      : %1.2e',relative_error);
    str4=sprintf('Beräkningstid   (s): %1.3f',tid);
    disp(str1);
    disp(str2);
    disp(str3);
    disp(str4);

%% Uppdatering av startvärden
    a(1)=minsta_kvadrat_vikt;
    a(2)=estimated_lambda(2);
    a(3)=estimated_lambda(3);
    a(4)=estimated_lambda(4);
end
disp('Complete');

```


- **model2.m**

Den inlinefunktion som används i ode23 för att beskriva vår differentialekvation. En del rader har blivit isärskrivna med ... på grund av typsnittet.

```
function ydot =
model2(t,y,options,M,g,c,k,m,v,s,L,shift,delta_x,amp1,fas1,f1,amp2,fas2,f
2)

ydot = zeros(2,1);

ydot(1) = y(2);
ydot(2) = g+(amp1*sin(2*pi*f1*t+fas1)+amp2*sin(2*pi*f2*t+fas2))/ ...
(massa(t,m,v,s,L,shift,delta_x)+M)-{(mdot(t,v,s,L,shift,delta_x)+c)/ ...
(massa(t,m,v,s,L,shift,delta_x)+M)}*y(2) - ...
(k/(massa(t,m,v,s,L,shift,delta_x)+M))*y(1);
```

- **massa.m**

Returnerar massafunktionens värde i tidssteg t.

```
function M = massa(t,m,v,s,L,shift,delta_x)

n = floor((v*t)/L);
if t < shift+n*L/v
    M = 0;
elseif t < n*L/v+s*delta_x/v+shift
    M = m*(v*t-n*L-v*shift)/(delta_x*s);
elseif t < n*L/v+s*(1-delta_x)/v+shift
    M = m;
elseif t < n*L/v+s/v+shift
    M = m*(s-v*t+n*L+v*shift)/(s-(1-delta_x)*s);
else
    M = 0;
end
```

- **mdot.m**

Returnerar massavågens derivata i tidssteg t.

```
function mdot = mdot(t,v,s,L,shift,delta_x)

n = floor((v*t)/L);

if t < shift + n*L/v
    mdot = 0;
elseif t < n*L/v+s*delta_x/v+shift
    mdot = 1/delta_x;
elseif t < n*L/v+s*(1-delta_x)/v+shift
    mdot = 0;
elseif t < n*L/v+s/v+shift
    mdot = -1/delta_x;
else
    mdot = 0;
end
```

- **plockaut.m**

Från indata bestäms hur många objekt som passerat samt när de passerar. Utdata y är en 'filtrerad' signal där stigtider och dylikt tagits bort. Kvar är bara rena mätningar ($m(t) = m$). te är en lista av index när olika mätningar börjar resp. slutar. n är antal objekt som passerat.

```
function [y,n,te]=plockaut(y,t,v,s,L,shift,delta_x)

j=length(y);

%% Hur många objekt har passerat under tiden t?
T=t(end);
T=T-shift-s/v;
n=0;
while T>=0
    n=n+1;
    T=T-L/v;
end

%% Kapa alla icketoppar
t1=1;
temp=find(t<shift);
t2=max(temp);
te=[];
y(t1:t2)=0;

for k=1:n
    t1=t2;
    temp=find(t<shift+k*L/v);
    t2=max(temp);
    temp2=[];
    for i=t1:t2
        if t(i)<shift+delta_x*s/v+(k-1)*L/v
            y(i)=0;
        elseif t(i)>shift+(1-delta_x)*s/v+(k-1)*L/v
            y(i)=0;
        else
            temp2 = [temp2 i];
        end
    end
    te = [te temp2(1) temp2(end)];
end

%% Kvarvarande delar kan inte vara en hel topp
y(t2:end)=0;
```

- **func.m**

Inlinefunktion till fminsearch. Här definieras vår målfunktion i optimeringsproblemet.

```
function err = func(lambda,t,y,M,handle)

global kf

A = zeros(length(t),7);
const = 1/(2*(lambda(1)+M));
const2 = const*sqrt(lambda(2)^2-4*lambda(3)*(lambda(1)+M));

A(:,1) = exp((-lambda(2)*const+const2)*t);
A(:,2) = exp((-lambda(2)*const-const2)*t);
A(:,3) = ones(length(t),1);
A(:,4) = sin(lambda(4)*2*pi*t);
A(:,5) = cos(lambda(4)*2*pi*t);
A(:,6) = sin(lambda(5)*2*pi*t);
A(:,7) = cos(lambda(5)*2*pi*t);

kf = A\y;
z = A*kf;
err = norm(z-y);

%%Dessa rader behövs om man vill göra debugploten i run.m
%set(gcf,'DoubleBuffer','on');
%set(handle,'ydata',z)
%drawnow
%pause(.04)
```

