

Detecting Perforations in a Continuous Converted Plastic Film Line



Per Fryking
Marcus Lindqvist

Dept. of Industrial Electrical Engineering and Automation
Lund University

Detecting perforations in a continuous converted plastic film line

Master Thesis by:

Per Fryking

Marcus Lindqvist

For

FAS Converting Machinery AB

Industriell Elektroteknik och Automation

Lund 6 January 2009

Förord

Vi vill rikta ett stort tack till FAS Converting Machinery AB, och alla anställda som har hjälpt oss i synnerhet Zoltan Darvas och Göran Persson. Anton Tornberg som var delaktig under första delen av projektet.

Vi vill även rikta ett stort tack till avdelningen för Industriell Elektroteknik och Automation och i synnerhet våra handledare Gunnar Lindstedt och Bengt Simonsson men även Getachew Darge som alltid varit hjälpsamma och ställt upp.

Per Fryking

Marcus Lindqvist

Lund den 16 januari 2009

Abstract

The project was performed for FAS Converting Machinery AB (FAS) in Ystad with supervision from Industriell Elektroteknik och Automation at LTH.

FAS is a company that makes machines for producing plastic bags on a roll. The problem given was to develop a sensor for detecting the perforations in the bags to ensure that the correct amount of bags was rolled up and that the tearing was in the right place. If the machine tries to tear off where there is no perforation the fine mechanics is subjected to great stress. This was solved by detecting light penetrating the bag. The spectrum of the light is infrared due to the fact that it penetrates plastic better than visible light. To ensure that a perforation is detected and not a deformation in the material two things were considered. The first thing was to eliminate as many wrinkles as possible this was done by letting the plastic run over a spherical surface. The second thing was to use redundant sensors and make sure both of them triggered at the same time. What makes it difficult is that the plastic is of various density, thickness and length and a manual calibration is not applicable. Therefore an automatic calibration algorithm had to be designed. The idea is to reduce 95% of the measured values to zero and therefore get less noise in the data. If the two sensors found a deviation at roughly the same time a signal was sent to the PLC which kept count of the produced bags.

Other problems are that the plastic might be folded up to 16 times and therefore gives a weak signal and the perforations are not perfectly overlapping each other. This was solved by designing the sensor holders like a "U" i.e. two tongues one with the IR-LED and one with the sensor. The tongue with the sensor is inserted in the fold and therefore reduces the amount of plastic to sense through.

The method that was developed did manage to count the perforations in the converted plastic films. The drawback was that this method could not be implemented to fully work with all the different types of material, folding and speeds. In order to use this method, a calibration needs to be done on every single material or design change of the product produced. The automatic calibration developed to get rid of the manual calibration was not robust enough to handle all possible situation without interaction with the operator.

Innehållsförteckning

FÖRORD	II
ABSTRACT	III
INNEHÅLLSFÖRTECKNING	IV
BILD OCH TABELL-FÖRTECKNING	VI
1 INLEDNING	1
1.1 BAKGRUND	1
1.2 SYFTE.....	1
1.3 MÅLSÄTTNING.....	1
1.4 BEHOVSSPECIFIKATION	2
1.5 AVGRÄNSNINGAR.....	2
1.6 RAPPORTÖVERSIKT	2
2 METOD	3
3 MASKINERNA	4
3.1 EXTRUDERAD PLAST	4
3.2 PERFORERING & SVETSNING.....	5
3.3 STANSNING.....	5
3.4 UPPRULLNING	6
3.5 VIKNING.....	7
4 KONCEPTGENERERING	8
4.1 KONCEPTFÖRSLAG ENLIGT XYZ - METODEN	8
4.2 URVAL AV KONCEPT	8
4.3 KONCEPTUTVECKLING	9
4.4 PRINCIPLÖSNINGAR	11
4.5 VAL AV KONCEPT.....	17
5 UNDERSÖKNING AV FRAMTAGNA KONCEPT	20
5.1 VIPPA MED TTG	20
5.2 TOTALREFLEKTION.....	21
5.3 FORTSATT ARBETE	22
6 EN FÖRSTA PROTOTYP	23
6.1 TESTER AV INTENSITETSKONCEPT.....	23
6.2 VIDAREUTVECKLING AV INTENSITETSKONCEPTET	24
7 VIDAREUTVECKLING AV PROTOTYP	25
7.1 RIGG.....	25
7.2 KORT	25
8 DELRESULTAT	26
9 SENSOR OCH VIDARE UTVECKLING AV KONCEPTET	27
9.1 FÖRSTUDIE AV LJUSKÄLLA.....	27
9.2 VAL AV LJUSRECEPTOR.....	28
10 ELEKTRONISK HÅRDVARA	29

10.1	FÖRSTUDIE FÖR PROCESSOR VAL.....	29
10.2	VAL AV PROCESSOR.....	29
10.3	VAL AV MICROCONTROLLER.....	30
10.4	KRETSKORTET.....	31
11	KONSTRUKTION AV SENSORHÅLLARE	34
11.1	SENSORHÅLLARE	34
11.2	TESTRIGG	34
11.3	MASKIN	34
12	SIGNALBEHANDLING	35
12.1	TESTFÖRUTSÄTTNINGAR	35
12.2	KURVDATA.....	35
12.3	AVKODNINGSMETODER	37
12.4	MEDELVÄRDESBILDNING.....	38
12.5	TILLÄMPNING.....	39
13	MJUKVARA.....	40
13.1	KODOPTIMERING	40
13.2	INITIERINGAR.....	41
14	RESULTAT, SVÅRIGHETER OCH LÄRDOMAR.....	42
14.1	TEST PÅ RIGG.....	42
14.2	TEST PÅ MASKIN.....	43
15	SLUTSATS	45
16	LITTERATURFÖRTECKNING	46
APPENDIX A. SCHEMA		47
APPENDIX B. KOD		48

Bild och tabell-förteckning

Figur 1 Exempel på en extruder	4
Figur 2 Exempel på en perforerare/svetsare (FAS Converting)	5
Figur 3 Exempel på en stansare. (FAS Converting)	6
Figur 4 Exempel på en upprullare (FAS Converting)	6
Figur 5 Exempel på vikbock (FAS Converting)	7
Figur 6 Systematisk genomgång av detektering	9
Figur 7 Principskiss för vals med TTG	11
Figur 8 Principskiss för vipa med TTG och mekanisk brytare	12
Figur 9 Principskiss för optisk markör	14
Figur 10 Totalreflektionens egenskaper	15
Figur 11 Principskiss för ultraljudsdetektor	16
Figur 12 Uppställning av TTG lab	20
Figur 13 Principiell funktion för Totalreflektion.....	21
Figur 14 Princip för Intensitetsskillnad med laser.....	23
Figur 15 Testtrigg för räkningen	25
Figur 16 Kretskortslayout.....	32
Figur 17 Ritning av sensorhållare.....	34
Figur 18 Graf över 4 påslängder för färgad sopsäck. spikarna nedåt är där påsarna är fogade.....	35
Figur 19 Graf över 5 skarvar på plast med tryck.	36
Figur 20 Graf över 4 skarvar på genomskinlig plast. Som synes går intensiteten ner vid en skarv.	36
Figur 21 Ett exempel på den multiplicerade signalen. Den bygger på samma rådata som Figur 17 ...	37
Tabell 1 Falltid för mekanisk vipa	13
Tabell 2 List of Metrics	17
Tabell 3 Utvalsmatris	18
Tabell 4 Viktat urval	19
Tabell 5 Reslutat från tester med olika optiska medium	22
Tabell 6 Lysdioder våglängd och material.....	27
Tabell 7 Viktad urvalstabell processorval	30
Tabell 8 Viktad urvalstabell för val av microcontroller där 3 är högst och 1 är lägsta betyg	31
Tabell 9 Exekvieringstid för olika operationer (Microchip, 2008).....	40

1 Inledning

Examensarbetet har utförts på avdelningen för Industriell Elektroteknik och Automation vid Lunds Tekniska Högskola för FAS Converting Machinery AB i Ystad. Arbetet började som ett projektarbete under hösten 2007 och våren 2008, då konceptgenerering och en första prototyp gjordes. Företaget gillade prototypen som gjordes och de ville utföra vidare utveckling och tester vilket blev examensarbetet som utfördes hösten 2008.

1.1 Bakgrund

FAS Converting Machinery AB (FAS) är ett företag som tillverkar maskiner för att producera konverterad plast¹. Idag tillverkar de moduler som perforerar/svetsar, viker och slutligen rullar upp plasten. Modulerna är fristående och kommunicerar inte med varandra utan har flytvalsar som fungerar som magasin mellan enheterna, detta för att öka flexibiliteten och göra ett enkelt system för operatören. Detta medför dock ett problem vid upprullaren som måste veta exakt när en skarv kommer för att kunna räkna rätt antal och riva av på rätt ställe. Om sensorn som detekterar skarvar skickar en felaktig signal gör det i bästa fall att det blir för få påsar på rullen men om den ska försöka riva av på ett ställe där det inte finns någon skarv utsätts maskinen för stora påfrestningar som skadar finmekaniken och ger rullar som inte går att sälja. Systemet som idag används för att detektera skarven bygger på en uppladdad stav som laddar ut mot en jordad vals då det blir en luftspalt istället för den isolerande plasten. Gnistsensorn har problem vid högre hastigheter samt att den sprider en elektromagnetisk störning i alla frekvenser då den laddar ur. Urladdningen kan vara skadlig för människor samt medför att det inte går att CE märka maskinen (en vidare beskrivning av maskinen finns i på sida 4). Denna sensor är inte önskvärd och därför ska nya principer undersökas.

Bakgrund till examensarbete

En prototyp togs fram i första projektet vilken kunde räkna skarvar med en acceptabel felmarginal. Arbetet har därför fortgått med att vidareutveckla den prototypen.

1.2 Syfte

Syftet med projektet är att undersöka möjligheterna att byta ut den befintliga sensorn. Den nuvarande sensorn är inte önskvärd då problemen kring den inte vägs upp av dess funktion.

Syftet med den första delen av projektet var att gå igenom alla fysikaliska principer som kunde tänkas och bland dessa sålla ut ett vinnande koncept som kan testas i ett tidigt prototypstadium.

Syftet med examensarbetet var att vidareutveckla prototypen från del 1 samt att installera den på en maskin för mer verklighetstroga tester. Flera olika plastmaterial testas för att undersöka dynamiken i sensorn.

1.3 Målsättning

Målsättningen för hela projektet var att utveckla en ny sensortyp för detektering av perforeringar i en löpande konverterad plast. Detta skall ske med hög noggrannhet och inte störa process eller operatör.

¹ Med konverterad plast menas att plastpåsar eller liknande har producerats ifrån extruderad plast.

Målsättningen med examensarbetet var att undersöka om det är lönsamt för företaget att gå vidare med den tidigare producerade prototypen för tester hos kund. Sensorn ska vara en "svart låda" som ska fungera utan kalibrering av operatören och utan förhandsinformation av påslängder och hastighet. Att det ska vara en "svart låda" beror på att inlärningstiden för en operatör inte bör ökas då det hos vissa kunder är stor omsättning på dessa. En annan anledning är att det finns olika modeller av upprullaren som har olika utrustning vilket gör att återkopplingar av olika slag blir svår att standardisera.

1.4 Behovsspecifikation

Behoven som är uppsatta är att produkten ska vara lättmonterad och inte vara allt för skrymmande då den väl är på plats. Den måste vara säker och varken kunna skada annan utrustning eller personal. Den måste också klara vissa slag och stötar det vill säga att den ska klara av att hanteras på ett relativt vårdslöst sätt. En av de viktigare aspekterna är att användarvänlighet och driftsäkerhet skall tas i beaktning.

1.5 Avgränsningar

Projekt 1 är färdigt då en prototyp fungerar tillfredsställande inom den satta tidsramen. Detta sker på en förenklad modell av maskinen.

FAS Converting vill att nya sensorn ska genomgå tre olika stadier. *Första prototyp, andra prototyp och prototyp hos kund.* Detta för att få med hela dynamiken som sensorn måste klara av. Avgränsningen för examensarbetet är att ta prototypen genom steg två och undersöka om det var möjligt att ta den vidare till steg tre.

1.6 Rapportöversikt

Här följer en genomgång av varje rubriks innehåll, kapitel 4 till 7 tillhör det tidigare utförda arbetet.

- Under kapitel 2 *Metod* förklaras hur utvecklingsprocessen har fortskridit.
- Under kapitel 3 *Maskinerna* förklaras hur FAS maskiner fungerar.
- Under kapitel 4 *Konceptgenerering* benas alla möjliga principer ut.
- Under kapitel 5 *Undersökning av framtagna koncept* förklarar hur utsällningen gick till.
- Under kapitel 6 *En första prototyp* förklaras tester av konceptet.
- Under kapitel 7 *Vidareutveckling av prototyp* konstrueras den första prototypen.
- Under kapitel 8 *Delresultat* utvärderas den första prototypen.
- Under kapitel 9 *Sensor och vidare utveckling av konceptet* undersöks vilka ljuskällor och receptorer som fungerar best.
- Under kapitel 10 *Elektronisk hårdvara* här utvärderas hur hårdvaran ska vara uppbyggd och vilka komponenter som ska vara med.
- Under kapitel 11 *Konstruktion av sensorhållare* här förklaras hur riggen är uppbyggd.
- Under kapitel 12 *Signalbehandling* här går de olika metoderna för att skarvar igenom.
- Under kapitel 13 *Mjukvara* här förklaras de algoritmer och initieringar som används.
- Under kapitel 14 *Resultat, svårigheter och lärdomar* här förklaras alla testers resultat.
- Under kapitel 15 *Slutsats* finns de tankar och idéer som framkommit under projektet.

2 Metod

Projektet genomgick fyra faser:

- Konceptgenerering
- Vidare tester av de vinnande förslagen
- En första prototyp
- Vidareutveckling av prototyp

Där de tre första ingick i en projektkurs under hösten 2007 och våren 2008.

Konceptgenerering.

Under konceptgenereringen samlas alla olika fysikaliska principer för att hitta en skarv, vilka senare viktas. Denna del av projektet utfördes under hösten 2007.

Vidare tester av de vinnande förslagen

De förslag som gick vidare efter konceptgenereringen testades och utvärderades för att välja vilken metod som skulle arbetas vidare med. Denna fas avhandlades under första kvartalet 2008.

En första prototyp

Den metod som övertygade mest i de inledande testerna utvecklades till en prototyp och testades under enklare former. Detta gjordes under andra kvartalet 2008.

Vidareutveckling av prototyp

Då företaget var intresserat av en vidareutveckling av den första prototypen genomfördes detta i formen av detta examensarbete under hösten 2008. Den första prototypen var inte optimerad för uppdraget. Därför inleddes arbetet med att från grunden förbättra varje enskild detalj. Områdena som undersöktes var:

- Ljuskälla
- Fotodiod
- Processor
- Signalbehandling

3 Maskinerna

FAS Converting säljer ett antal olika fristående moduler som tillsammans i en produktionsbana konverterar plast till olika formers påsar. Dessa moduler kan kombineras på det sätt som önskas men en upprullare och perforerare måste finnas med. Modulerna ställs sedan upp på en lång rad med mellanrum emellan. Plasten som tillverkas går igenom alla moduler och rullas till sist upp på rulle i den sista modulen, upprullaren.

Produkterna som produceras är alla på rulle men kan variera i övrigt. T.ex. finns smala och breda plaster, plastpåsar, plastpåsar med utstansade hål, plastpåsar med tryck o.s.v.

Nedan förklaras kort ett par av de modulerna som säljs av FAS.

3.1 Extruderad plast

Det första steget för att ta fram plastpåsar är att extrudera råmaterial till långa tuber av plast med tunna väggar. Dessa tuber av plast är varma direkt när de kommer ut från extrudern och därför måste de kylas. För att kyla plasten låter man tuberna löpa ett par meter upp i luften, hur långt tuberna måste löpa upp från extruderaren är beroende på dess hastighet. För närvarande säljer FAS inga maskiner för extrudering utan en sådan maskin måste köpas in från ett annat företag.



Figur 1 Exempel på en extruder

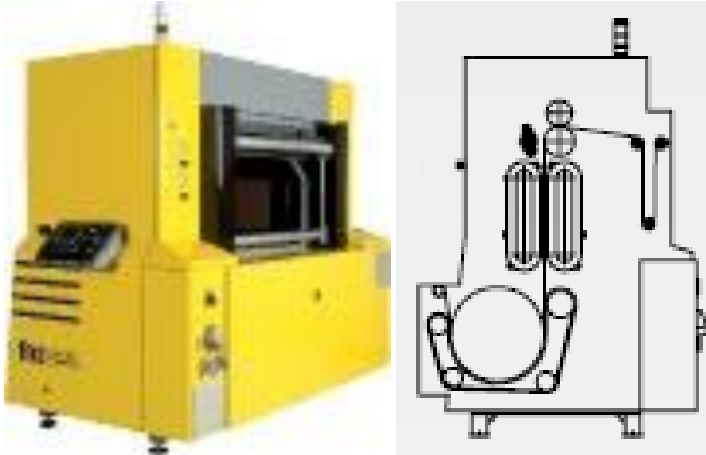
Plasten som matas in till FAS produktionsbana kan antingen matas in offline eller inline.

Med offline menas att produktionslinjen är kopplad till färdigextruderat material på rulle. Fördelen med offline är att produktionsbanan kan köras mycket snabbare än då den är kopplad direkt till en extruder. Nackdelen med offline är att dessa rullar måste bytas, vilket innebär jobb för operatörerna.

Inline kallas det när produktionsbanan är direkt kopplad till en extruder. Fördelen med att köra direkt mot en extruder är att momentet med rullbyte försvinner. Nackdelar är att extrudern fortsätter att producera plast vid ett avbrott i produktionsbanan. Eftersom extrudern kan producera material i upp till 200m per minut innebär det att operatören måste fixa avbrottet snabbt för att inte generera alltför mycket oanvändbart material.

3.2 Perforering & Svetsning

Nästa steg för att producera plastpåsar är att perforera och i de allra flesta fall svetsa plasten. Detta görs genom att materialet från extrudern åker in i en speciell modul från FAS som utför båda perforering och svetsning på det sätt som önskas.



Figur 2 Exempel på en perforerare/svetsare (FAS Converting)

Perforering

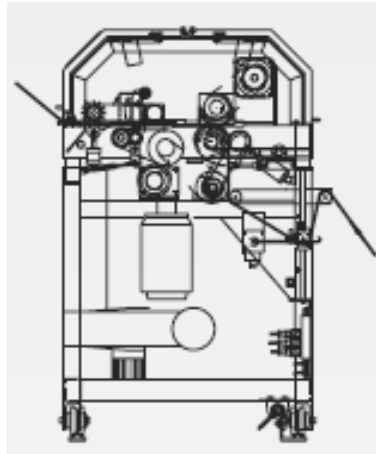
För att kunna riva av påsar ifrån de färdiga rullarna måste plasten perforeras. Detta görs genom att en stor kniv med många tänder slår till mot plasten. Plasten får då ett par millimeter långa hål längs med hela plastpåsen med ett par millimeter mellan varje hål. Perforeringen gör att banan med plast blir svagare och med en alltför generös perforering kan banan gå av vid höga spänningar.

Svetsning

I de fallen då en plastpåse önskas måste plasten svetsas i direkt anslutning till perforeringen, antingen innan eller efter. För att åstadkomma detta finns det ett par backar i maskinen. Dessa backar är beklädda med teflon och under teflonen finns en tunn glödtråd som löper längs med backen. När två backar går emot varandra med plasten emellan och med varma glödtrådar smälts de två väggarna ihop på plasttuben.

3.3 Stansning

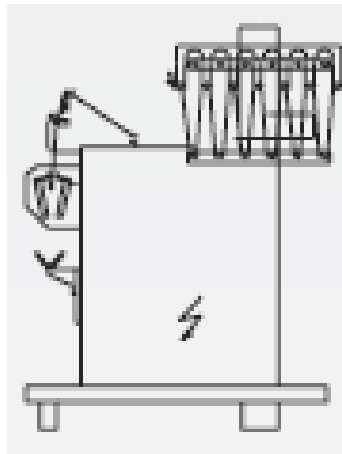
Ibland vill plasttillverkaren stansa ut olika saker i plasten. T.ex. kan det på plastpåsar stansas ut ett parti efter perforeringen så att handtag bildas i utvecklat tillstånd. Ett exempel på detta är matkassen som kan köpas i mataffären, där är ett hål utstansats för att bilda handtag.



Figur 3 Exempel på en stansare. (FAS Converting)

3.4 Upprullning

Det sista steget i plastpåproduktion är att rulla upp plastpåsar på rulle. Detta steg används inte i all plastpåstillverkning, men i alla FAS produktionslinor avslutas tillverkningen med att påsarna rullas upp på rullar.



Figur 4 Exempel på en upprullare (FAS Converting)

Upprullaren

För att separera de olika rullarna åt rivs perforeringen mellan den sista påsen på ena rullen och den första påsen på den andra rullen av. För att få till en avrivning accelereras upprullningen och en högre spänning på banan gör att perforeringen rivs av. Under rullbytet hålls plasten fast i banan med hjälp av två backar. När upprullningsarmarna åter har börjat rulla plast måste dessa rulla snabbare än banhastigheten. Detta för att komma ikapp den tid som förlorats under rullbytet.

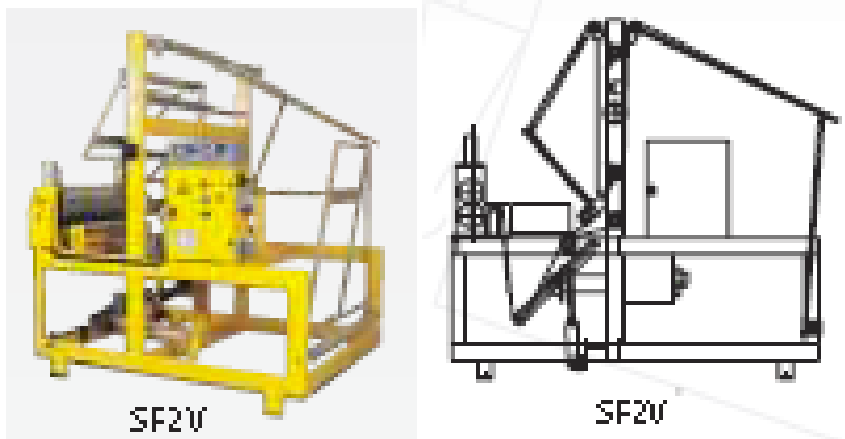
Flytvals

Som nämns ovan står plasten närmast upprullarmen stilla under ett rullbyte, medans resten av banan fortsätter att arbeta i samma hastighet. För att ta hand om allt material som kommer in i upprullarmodulen under tiden som ett rullbyte sker används en flytvals. Flytvalsen är uppbyggd av två rader av valsar. Den övre raden av valsar är fix och den undre raden kan höjas och sänkas. På denna finns en vikt som hjälper till att hålla banan lagom spänd. Den extra plasten ackumuleras

genom att flytvalsen rör sig nedåt och därmed tar hand om det extra material som kommer in i maskinen under ett rullbyte.

3.5 Vikning

I många produktionslinor viks plasten på ett eller annat sätt. Vikning kan ske både innan och efter svetsning och perforering. Ibland viks plasten för att få produkten mindre och ibland för att få till en något stjärnformad botten (exemplet med matkassen kan tillämpas även här) och i en del fall viks plasten för att uppnå båda två



Figur 5 Exempel på vikbock (FAS Converting)

4 Konceptgenerering

Innan några mer specifika lösningar tas fram analyseras de olika fysikaliska principer som kan tänkas kunna tillämpas för att lösa uppgiften igenom. Dessa bryts sedan ner till flera olika grova lösningar och utifrån dessa lösningar skapas sedan de olika principlösningarna.

4.1 Konceptförslag enligt XYZ - metoden

För att generera principlösningar till projektet används XYZ - metoden. Denna metod bygger på att X personer samlas kring ett bord och tar fram Y principlösningar. Dessa lösningar skickas sedan runt Z gånger mellan personerna. Efter detta kommer förhoppningsvis ett stort antal principlösningar vara framtagna. (Ulrich & Eppinger, 2004)

4.2 Urval av koncept

Urvalet av koncept kommer att göras med hjälp av en "List of Metrics" som de olika specifikationerna sätts in i för att kunna göra bedömningar om vilka lösningar som är intressanta (Ulrich & Eppinger, 2004)

De specifikationer som har beskrivits under dialog med FAS representant, Zoltán Darvas, är som följer.

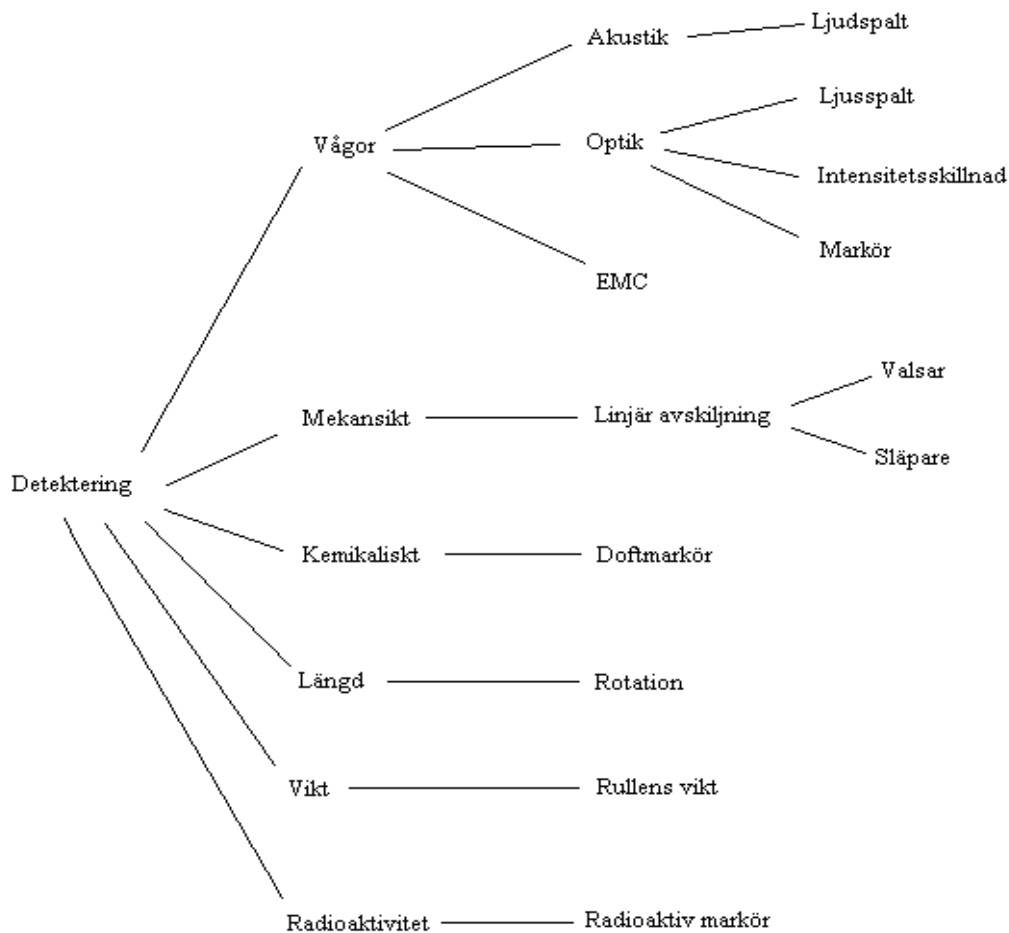
Specifikationer/Behov

- God felmarginal på räkningen (1‰)
- Lågt pris (max 7000kr)
- Lättmonterad
- Lättinstallerad (försumbart i totala monterings tiden)
- Användarvänlig
- Rimliga dimensioner (i förhållande till maskinens dimensioner)
- Säker
- Driftsäker
- Stöttålig
- Värmetålig (50°C)
- Tål väta

4.3 Konzeptutveckling

Här staplas alla fysikaliska principer upp och sållas därefter utifrån hur rimliga de är att genomföra.

Systematisk angripning av problemet



Figur 6 Systematisk genomgång av detektering

Akustiskt

Eftersom det är ett utrymme mellan varje påse efter perforering så borde det vara möjligt att få ljudvågor att passera igenom dessa. Genom att välja en frekvens som ligger utanför de områden som finns i omgivningen borde störningar kunna minimeras. Det är dock så att vissa påsar som tillverkas viks ett antal gånger innan de rullas upp. Eftersom påsen vikts så kommer det inte vara möjligt att få vare sig ljus eller ljud att passera igenom.

Med hjälp av kontinuerligt ultraljud så kan ojämnheter upptäckas i olika ytor genom att mäta intensitetsskillnaden i det reflekterande ljudet. Det kommer troligtvis att behövas ett kontinuerligt ljud för att inte bli begränsade av ljudets hastighet. Mätning med hjälp av kontinuerligt ultraljud borde vara fysikaliskt möjligt, samt uppfylla kraven för hastighet vilket gör det till en intressant lösningsprincip.

Mekaniskt

De mekaniska lösningarna brukar ha en högre driftsäkerhet men kan bli svåra att anpassa till olika storlekar och utförande av påsar.

Genom att placera en vals på vardera sida om plastfilmen, med den ena eller båda två löst monterade, kan en mätning göras av rörelsen vertikalt utifrån plastfilmen.

Släppprincipen är tänkt att fungera genom att små armar ligger släpandes längs plastfilmens ovan- eller undersida och kommer att röra sig vertikalt vid en skarv. Detta för att en svets är tjockare än resten av filmen, tillika är en perforering tunnare.

Dessa lösningar borde uppfylla kraven för hastighet och är därför intressanta för vidare undersökning.

Optiskt

Ljus skulle kunna användas på ett liknande sätt som ljud. Genom att få ljus att passera genom de utrymmen som uppstår emellan påsarna så kan skarvar detekteras. Dock är det samma problem med ljus som med ljud eftersom påsarna kan vara vikta.

Mätning av intensitet skulle kunna fungera genom att istället för att använda sig av helt digitala signaler ljus eller inte ljus använda sig av skillnaden i ljus som släpps igenom. Detta skulle få problem då det är svårt att ställa in de rätta nivåerna. Detta blir också ett problem då påsarna är vikta.

Genom att använda totalreflektion skulle det vara möjligt att upptäcka perforeringar. Totalreflektion uppstår när ljus, med en viss vinkel, färdas från ett tätare till ett tunnare medium. Om påsarna kan färdas längs en prisma så skulle totalreflektion uppstå då en skarv kommer, och ljus färdas då från ett tätare medium till ett tunnare. Detta är en intressant fysikalisk princip som kommer att undersökas närmare.

Kemiskt

Kemiska sensorer kan avkänna koncentrationer i gaser och vätskor av olika typer av kemiska ämnen. Genom att placera en markör när svetsningen sker skulle man kunna avläsa denna vid räkningen. De sensorer som finns idag har dock en uppdateringstid på ungefär tio sekunder vilket inte ens är i närheten av behovet. Detta gör att vidare undersökning inte är aktuell.

Längd

Eftersom längden på påsarna är känd så är det också känt hur många varv som krävs för att upprullningen skall vara klar. Detta är dock inte möjligt enligt FAS Converting eftersom packningsgraden i rullarna kommer att variera. Lösningen kommer då inte att uppfylla de krav på precision som ställs vid avrivning. Ett annat problem är att det inte går att ställa in exakt rätt längd och därför kommer felen att ackumuleras.

Vikt

Genom att väga det som är upprullat så kan man vid den aktuella vikten riva av. Detta kommer förmodligen att vara svårt då vägning av en förspänd rulle är svårt och tjockleken kan variera något.

Värme

Efter svetsningen kommer temperaturen på skarven mellan påsarna att vara högre än i resten av plasten. Den temperaturskillnaden som uppstår kommer dock att försvinna när plastfilmen färdas vidare i processen. Temperatursensorer uppfyller inte heller de hastighetskrav som är uppställda.

4.4 Principlösningar

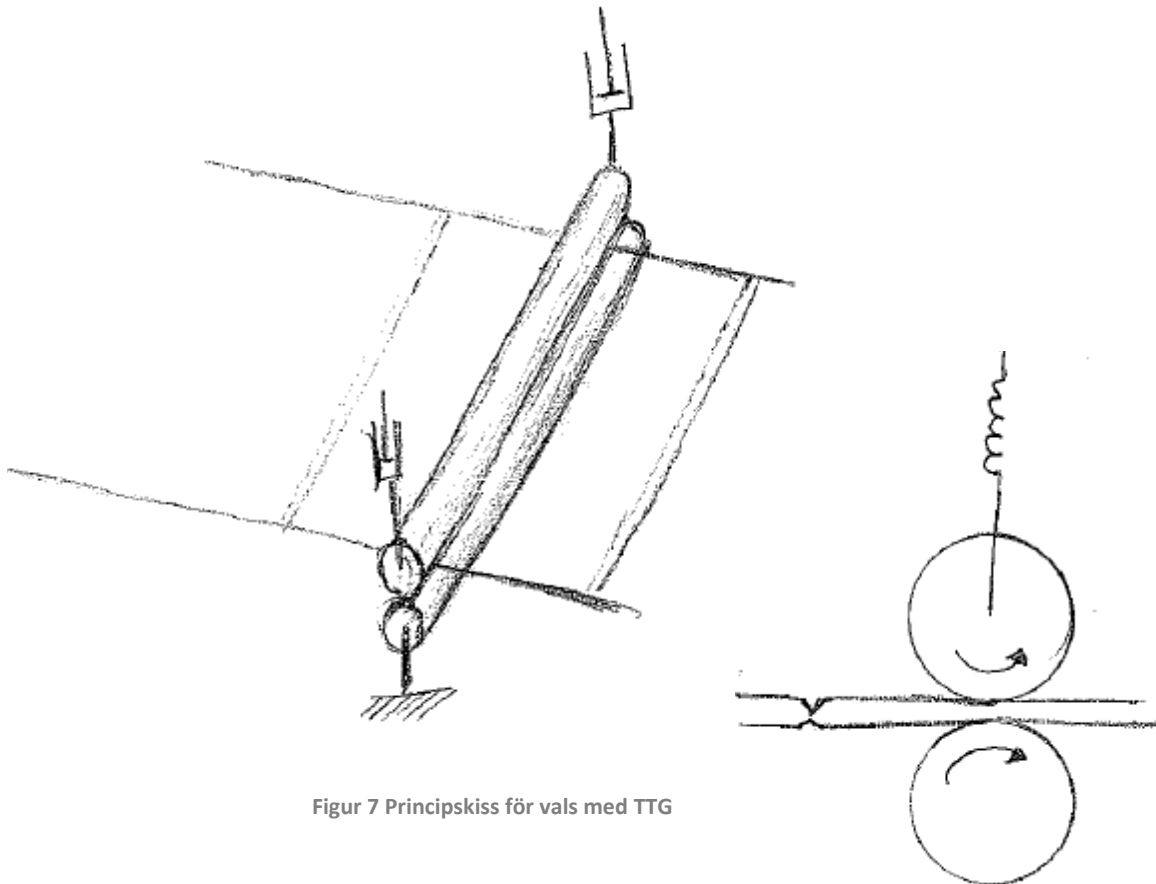
När specifikationen nu är uppställd och de fysikaliska principerna nedbrutna i grövre lösningar kan lite mer detaljerade principlösningar tas fram genom den nämnda XYZ metoden. Dessa lösningar är inte att ses som exakta utan är produkter av mer eller mindre rimliga idéer.

Vals med TTG

En vals som ligger an mot plastpåsarna är förspänd med två fjädrar. När en skarv passerar under valsen kommer fjädern att sträckas. Genom att placera en trådtöjningsgivare på stängeln under fjädern kan en sträckning mätas. Denna metod är väldigt enkel rent tekniskt, men ett frågetecken är om den är känslig nog. Ifall valsens egentyngd plus förspänningen av fjädrarna gör att påsens skarv pressas ihop så pass mycket att en längdändring i fjädern blir nära noll är metoden dålig.

Frågetecken

- Är valsens egentyngd tillsammans med fjäderns förspänning så stor att skarven inte går att upptäcka?
- Svets går bra men fungerar en perforering?



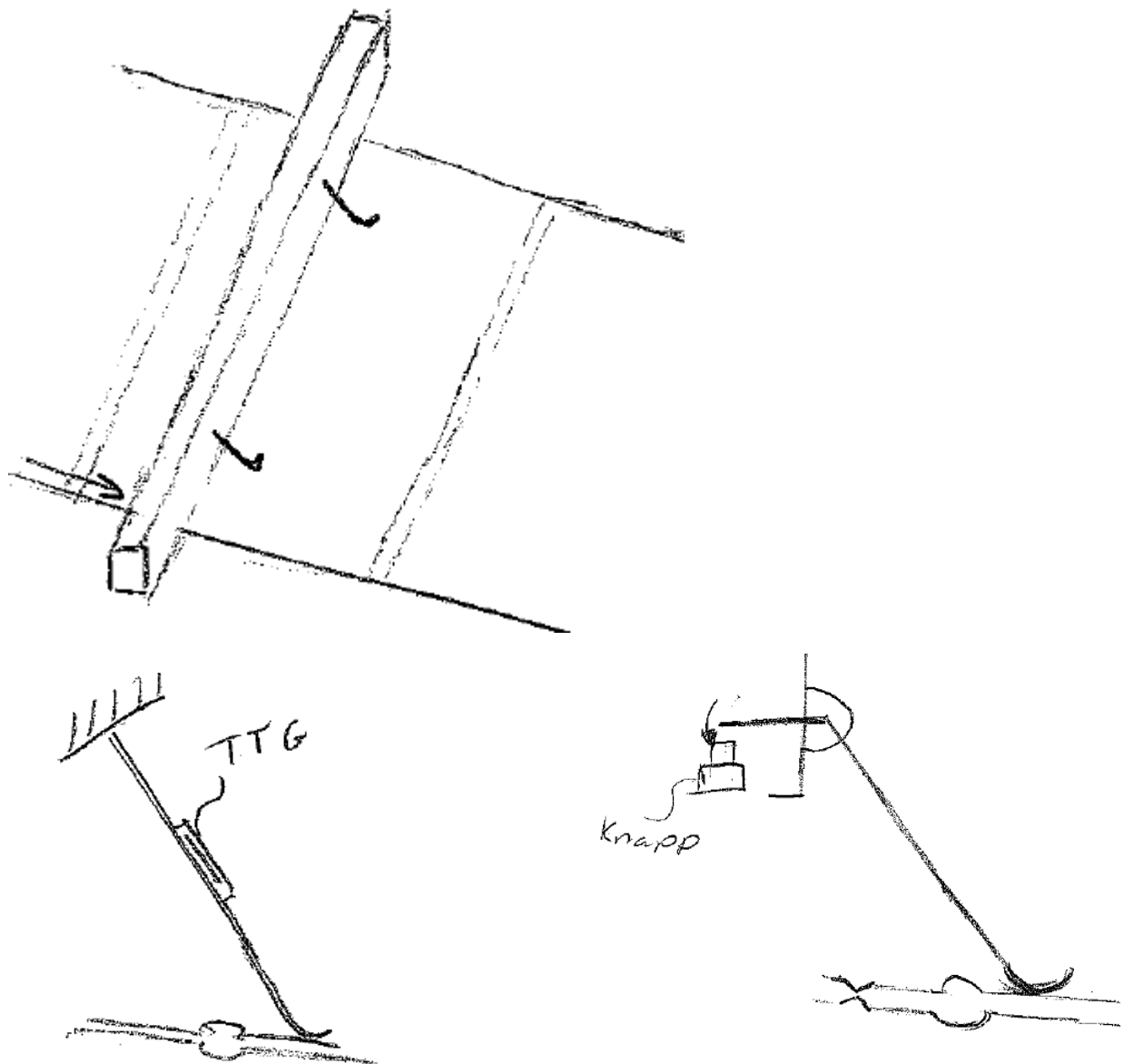
Figur 7 Principskiss för vals med TTG

Vippa med TTG

Istället för att ha en vals som ligger an mot plastpåsarna med förspända fjädrar kan en fast monterad vippa ligga an mot plastpåsarna. Ifall vippan är tillräckligt tunt dimensionerad kommer en liten böjning att ske varje gång en skarv eller en perforering passerar under den. Med hjälp av en fastmonterad TTG på vippan kan oerhört små böjningar detekteras. Med denna metod bör få god känslighet för små variationer.

Frågetecken:

- Kommer skarven eller perforeringen att vara stor nog för att generera en spänning i TTG?
- Kommer vippan att vara alltför känslig för industriella miljö den kommer utsättas för?



Figur 8 Principskiss för vippa med TTG och mekanisk brytare

Mekanisk vippa

Genom att ha en mekanisk vippa som slår emot svetsfogarna skapas en vertikal rörelse. Denna vertikala rörelse kan användas på olika sätt. Genom att ha en brytare i ett visst läge kan en elektrisk impuls skapas som är beroende av svetsfogen. För att detta ska funka måste vippan hinna ner innan nästa skarv kommer. Genom vippans vikt och frekvensen av skarvar som passerar kan vippan dimensioneras. Ifall vi approximerar och utgår ifrån att vippan kommer till sitt övre läge relativt snabbt kan tiden det tar för vippan att genom sin egenvikt och tyngdaccelerationen komma tillbaka till ursprungsläget räknas ut.

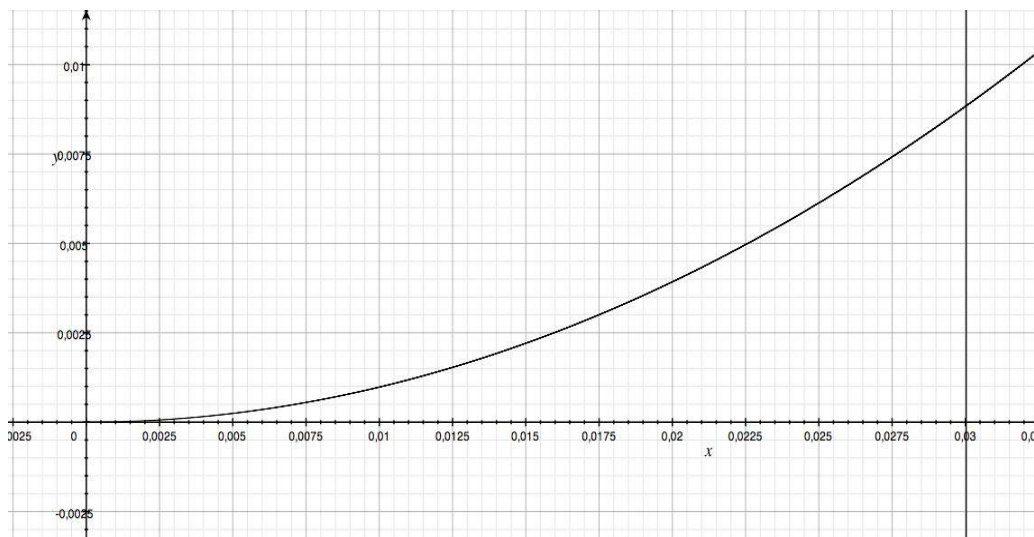
Falltid

Accelerationen är lika med sträckan deriverad två gånger med avseende på tiden. (Meriam, 2003)

$$a = \ddot{s} \Rightarrow s = \iint ad^2t \Rightarrow s = s_0 + v_0t + at^2$$

$$\begin{cases} s = s_0 + v_0t + at^2 \\ a = g \\ s_0 = 0 \\ v_0 = 0 \end{cases} \Rightarrow t = \sqrt{\frac{s}{a}}$$

För att slippa räkna ut den exakta höjden som vippan kommer att puttats upp av skarven tas ett diagram fram för att jämföra tiden (x) med en viss fallhöjd (y). Den vertikala kurvan $x=0,030$ motsvarar tiden mellan två skarvar i värsta fall, baserat på en påslängd på 10cm och en matningshastighet på 3.33m/s



Tabell 1 Falltid för mekanisk vippa

Av diagrammet kan vi se att ifall vi vill ha en marginal på 50% måste vippan dimensioneras för att max stiga 4mm.

Frågetecken

- Kan vippan dimensioneras för att inte stiga mer än 4mm?
- Kommer vippan att stiga över huvud taget?

Optisk markör

En lösning är att i närheten av varje svets lägga en markör som avger eller absorberar ljus av en viss våglängd vilket sedan med en enkel detektor går att avläsa vid upprullningsmaskinen. Att applicera markören skulle kunna lösas på flera olika sätt bland annat via något som kan liknas vid en bläckstråleskrivare, det vill säga att en piezokristall trycker ut en droppe bläck då en elektrisk puls ges. Detta har fördelen att det är en relativt billig och väl beprövad teknik men har nackdelen som alla andra markörer också har, att markören tar slut och måste fyllas på. Ett annat sätt att applicera markören är via en mekanisk arm som trycker ner en penna efter varje svets. Alternativt kan markören med ett högt tryck sprutas ut ur ett munstycke.

Om markören som används är fluorescerande så finns det på marknaden billiga detektorer som med tillräcklig hastighet klarar läsa av märkena. Detta system skulle vara billigt att installera och ha en hög noggrannhet men nackdelen är att det krävs underhåll som till exempel att fylla på markören. Ytterligare en kostnad skulle tillkomma per tillverkad påse, även om den kostnaden är liten så tillverkas så pass många påsar att det skulle märkas. Ett annat problem är att det tillkommer en kemikalie som kan vara skadlig, exempelvis om påsen ska användas inom vården.

Frågetecken

- Blir det problem med att applicera markören på den glatta plasten?
- Kommer färgen att sitta kvar efter kontakten med de många valsarna?
- Blir det för dyrt att applicera färg på varje plastpåse?

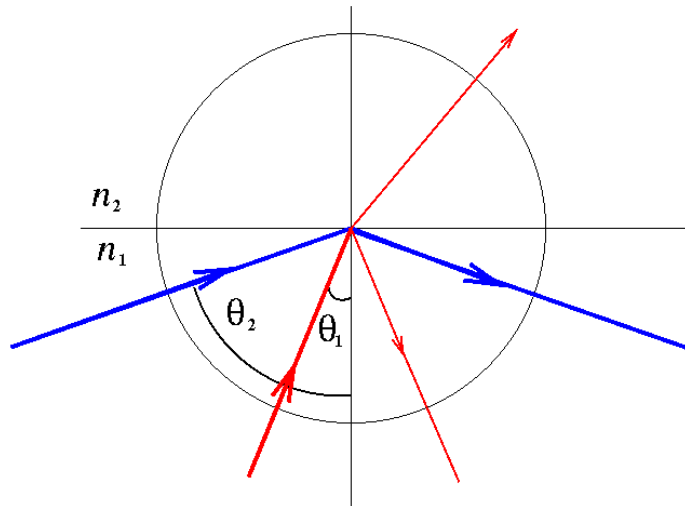


Figur 9 Principskiss för optisk markör

Totalreflektion

Ifall en ljusstråle går från ett tätare till ett tunnare material uppstår reflektion. Ifall infallsvinkeln mellan de två materialen är tillräckligt liten kommer en totalreflektion att inträffa. Genom att utnyttja luftspalterna som finns i perforeringen och eventuellt runt omkring svetsfogen, kan en totalreflektion uppstå.

Ekvation för totalreflektion



Figur 10 Totalreflektionens egenskaper

Vid den kritiska vinkeln θ_c kommer totalreflektion att uppstå ifall ljuset går ifrån ett optiskt tätare material till ett optiskt tunnare material. Den kritiska vinkeln fås genom nedanstående formel där n_2 är brytningsindex för tunnare materialet. (Jönsson, 2002)

$$\theta_c = \arcsin\left(\frac{n_2}{n_1}\right)$$

Tillämpning

Ifall ytan under x-axeln representeras av glas och ytan över plasten/luften kommer totalreflektion att uppstå vid en skarv eller perforering som skapar luftspalter. Vid luftspalterna kommer skillnaden mellan n_1 och n_2 att öka och därmed får vi en mindre kritisk vinkel och en totalreflektion kommer att uppstå ifall en bra infallsvinkel har valts på mätljuset. Ifall brytningsindex för det glasliknande materialet som påsarna löper över uppskattas till 1,5 fås:

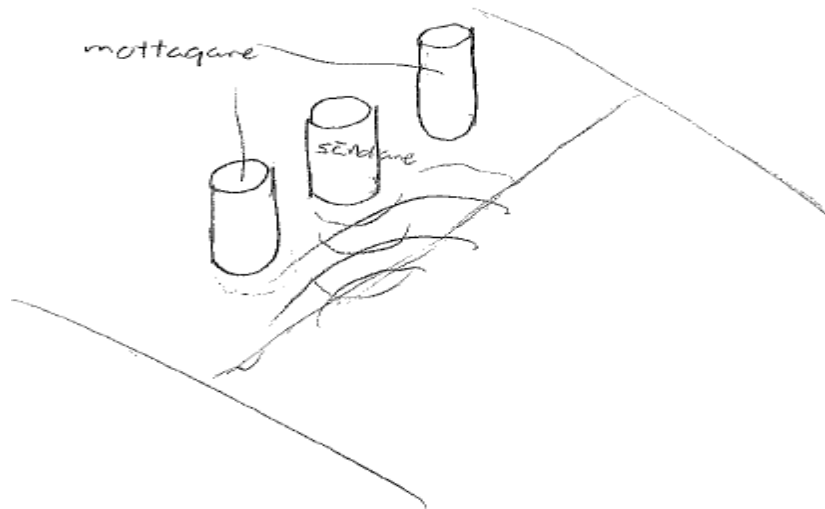
$$\begin{cases} n_1 = 1,5 \\ n_2 = 1 \end{cases} \Rightarrow \theta_c = 41^\circ$$

Om infallsvinkeln θ väljs så att $\theta > 41^\circ$ kommer totalreflektion att ske vid luftspalter men inte där plastpåsen ligger an.

Ultraljudsdetektor

En princip för detektering är att använda sig av ultraljud där en sändare kontinuerligt skickar ut en ultraljudsignal, ljudet stutsar mot plasten som är uppspänd av en vals och den reflekterande signalen läses av genom två mottagare. Eftersom plasten löper över en vals innebär det att det avlästa området blir ett rakt streck istället för en cirkulär yta. Detta ger ett tydligare utslag då en perforering kommer.

Principen som används är att då det inte finns någon skarv blir det en utsläckning och när höjden ändras sig kommer detta att ge ett tydligt utslag.



Figur 11 Principskiss för ultraljudsdetektor

4.5 Val av koncept

Urvalsprocess

För att kunna jämföra de olika specifikationerna med varandra förs de in i en "List of Metrics". (Ulrich & Eppinger, 2004) I matrisen viktas de olika specifikationerna för att kunna göra en objektiv bedömning av de olika koncepten som genereras, värden som används är angivna av Zoltan Darvas och de värden som inte angivits har uppskattats. Den prioritetsordning som är angiven av FAS Converting är; *felmarginal, pris och tid för slutförande*. Eftersom den tid som är uppsatt för arbetet begränsas av den tid som kursen pågår tas inte tid för arbete i urvalet, då alla valda lösningar ligger inom ett rimligt tidsintervall. Viktningen för felmarginal och pris kommer att viktas tyngst eftersom de är viktiga för FAS Converting.

Specifikation	Viktning	Enheter	Min Värde
Felmarginal på räkningen	5	%	0.1
Pris	5	SEK	7000
Lättmonterad	1	h	2
Användarvänlig*	2	Subj.	Subj.
Storlek	1	m ³	0,5
Säker	4	Subj.	Subj.
Driftsäker	3	%	0.1
Stöttålig*	2	N	100
Värmetålig	2	°C	50
Tål väta*	2	%	100
Stör omgivningen	4	Subj.	Subj.
Hastighet	5	Hz	35

**Uppskattade värden*

Tabell 2 List of Metrics

Första urvalet

Innan urval kan påbörjas med en viktad matris så kommer ett första urval genomföras med en enklare matris för att kunna se vilka lösningar som med störst sannolikhet är möjliga.

Specifikation	Vals med TTG	Vippa med TTG	Ultraljudsdetektor (Referens)	Mekanisk vippa	Optiskmarkör	Totalreflektion
Felmargnal på räkningen	0	0	0	-	0	0
Pris	+	+	0	+	-	0
Lättmonterad	-	-	0	-	-	0
Användarvänlig	0	-	0	-	-	0
Storlek	-	0	0	0	-	0
Säker	-	0	0	0	0	0
Driftsäker	0	-	0	-	-	0
Stöttålig	-	-	0	-	0	-
Värmetålig	-	0	0	0	0	+
Tål väta	0	0	0	0	-	-
Stör omgivningen	+	+	0	+	0	+
Hastighet	-	-	0	-	0	0
Summa +	2	2	0	2	0	2
Summa -	6	5	0	6	6	2
Summa 0	4	5	13	4	6	8
Poäng	-4	-3	0	-4	-6	0
Gå vidare?	Nej	Ja	Nej	Nej	Nej	Ja

Tabell 3 Utvalsmatris

De tre koncepten som visar sig vara mest troliga att fungera är ultraljudsdetektorn, totalreflektionen och vippa med TTG. Dessa koncepters funktionalitet är relativt svåra att särskilja utan vidare verkliga test. För att ultraljudet skall fungera så krävs speciella riktade högtalare. Några sådana komponenter finns ej färdiga på marknaden idag. Ett av FAS Convertings huvudkrav var att projektet skulle leda till en färdig och fungerande prototyp inom rimlig tid. Utveckling av sådana ultraljudskomponenter är väldigt resurskrävande, vilket gör att ultraljudsdetektorn härmed väljs bort.

Innan det går att verkligen avgöra vilket, eller vilka koncept som kommer att fungera på ett tillfredställande sätt så bör båda två tas med vidare till nästa fas. I den fasen kommer test med enklare prototyper att genomföras.

Specifikationsviktning

För att få en bild av hur de återstående koncepten förhåller sig inbördes så sätts även de in i den "List of Metrics", som tidigare upprättades. Koncepten rangordnas efter hur bra de är i jämförelse med varandra, dvs. den bästa får en 2:a och den sämsta en 1:a i betyg. Betygen viktas sedan efter respektive specifikations relevans. Detta kommer dock inte att leda till någon uteslutning av något koncept.

Specifikation	Viktning	Vippa med TTG	Totalreflektion
Felmarginal på räkningen	5	1	2
Pris	5	2	1
Lättmonterad	1	1	2
Användarvänlig*	2	1	2
Storlek	1	2	2
Säker	4	1	2
Driftsäker	3	1	2
Stöttålig*	2	2	1
Värmetålig	2	2	2
Tål väta*	3	2	1
Stör omgivningen	4	2	1
Hastighet	5	1	2
Viktade betyg	-	54	60

Tabell 4 Viktat urval

Enligt tabellen ovan så skulle konceptet med totalreflektion vara lite bättre det andra. Betygen är satta under förutsättningarna att båda koncepten fungerar optimalt utifrån sina förutsättningar. Vissa av betygen är satta på grunder som "att ljus är snabbare än gravitation". Detta innebär att för att kunna skilja på dem så har egenskaper dragits till sin spets, långt ovan de satta kraven. Det bör än en gång nämnas att det ej är säkerställt om dessa principer fungerar rent praktiskt.

5 Undersökning av framtagna koncept

Här följer en del praktiska försök med de framtagna koncepten.

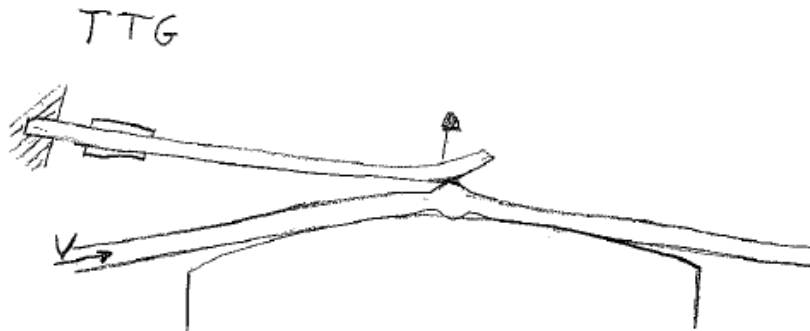
5.1 Vippa med TTG

En mekanisk lösning på problemet är att använda sig av en vippa med monterad trådtöjningsgivare. Denna testades under förenklade former.

Material

- Trådtöjningsgivare (TTG) fastmonterade på linjal
- TTG fastmonterade på balk
- Skruvstäd
- Oscilloskop
- Plastpåsar från FAS Converting

Uppställning



Figur 12 Uppställning av TTG lab

Utförande

Genom att fixera linjalen med TTG mot en plan yta med hjälp av skruvstället åstadkoms en konstant spänning i linjalen. Därefter fördes plasten mellan linjal och ytan så att resistansskillnaden i TTG:n kunde mätas upp i ett oscilloskop.

Resultat

Utfallet av experimentet var att skillnaden i resistans var så pass liten så att detektorn skulle ha svårt att urskilja en skarv från vanligt mätbrus. Detta skulle kräva avancerade och svårkalibrerade installationer. En annan nackdel är att då en påse går av under drift måste den nya träs i under former av hög tidspress och då skulle utrustningen lätt komma till skada.

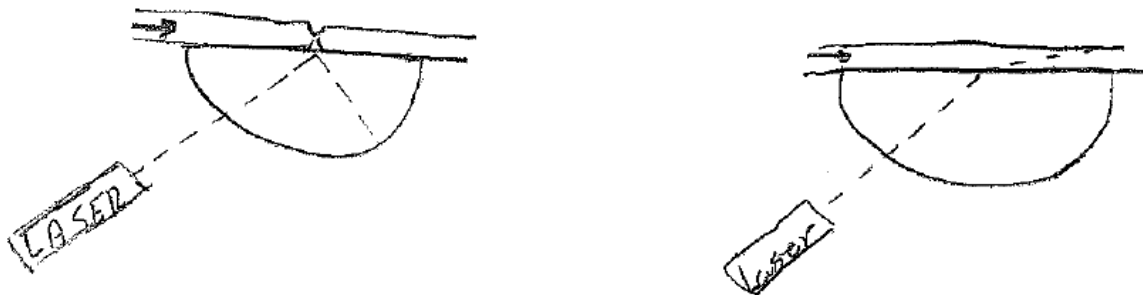
5.2 Totalreflektion

En metod är att med hjälp av ljusets optiska egenskaper detektera perforeringen. Denna byggde på principen om totalreflektion och testades under förenklade former.

Material

- Plexiglascylinder (Ø50, tjocklek 20)
- Laserpekare
- Ställning med klämma
- Skruvstäd
- Vatten
- Såpa
- CRC 5-56
- Kokplatta
- Plastpåsar från FAS Converting

Uppställning



Figur 13 Principiell funktion för Totalreflektion

Utförande

För att få en riktig infallsvinkel så att totalreflektion möjliggörs kapades plexiglascylindern på mitten. Detta medför också att plasten får en plan yta att löpa över.

Den halva plexiglascylindern monterades med den plana ytan uppåt i skruvstället, detta för att kunna föra plast över plexiglascylindern utan att föra den ur position. Genom att montera laserpekaren med ställningens klämma fixeras laserpekaren samtidigt som vinkeln mot plexiglascylindern är justerbar. Vinkeln för totalreflektion ställdes in med hjälp av ett papper ovanför den plana ytan så att man tydligt såg när vinkeln för totalreflektion uppstod. När inget ljus penetrerar den plana ytan är totalreflektion uppnådd.

För att undersöka om plasten upphäver totalreflektionen drogs plast, med skarv, över den plana ytan. Detta fungerade inte därför att det alltid blev ett visst luftglapp mellan plasten och cylindern. Vidare tester gjordes för att få bort luftspalten. Testerna bestod av olika typer av kontakt mellan den plana ytan och plasten; luft, vatten, såplösning, vattenånga och CRC 5-56 användes.

Resultat

Testerna visade i att totalreflektion inte är en rimlig lösning. Utan något optiskt kontaktmedel mellan plasten och den plana ytan kunde inte totalreflektionen upphävas. Med hjälp av såplösning, vattenånga eller CRC 5-56 var det fullt möjligt.

Problemet med vatten var den ytspänning som uppstod. Detta ledde till att vattnet inte fördelades jämnt över ytan utan bildade mindre ihopsamlingar. Såplösningen, CRC 5-56 och vattenången gav visserligen den önskade effekten men det är inte önskvärt att tillföra något medium på plastpåsen precis innan upprullningen. Detta ledde till att fortsatt utveckling av konceptet totalreflektion inte kommer utföras.

Optiskt kontaktmedel	Luft	Vatten	Såplösning	CRC 5-56	Vattenånga
Effekt	Ej möjligt	Oregelbundet	God effekt	Mycket god effekt	Mycket god effekt

Tabell 5 Resultat från tester med olika optiska medium

5.3 Fortsatt arbete

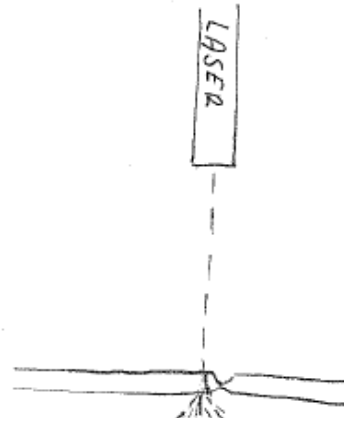
Under försöken upptäcktes att samtliga plastpåsar var möjliga att lysa igenom med en laserpekare. Detta medförde att ett koncept vilket tidigare setts som mindre möjligt togs fram. Konceptet innebär att genom mätning av intensitetsskillnaden i penetrerat ljus upptäcka en skarv.

6 En första prototyp

Då samtliga plastpåsar släpper igenom ljus kommer mängden ljus att bero på plastens täthet och tjocklek. Genom att lysa från ena sida och sedan detektera mängden penetrerat ljus på den andra kan en skarv upptäckas.

Våglängden på ljuset kan vara av stor betydelse eftersom materialets täthet kan absorbera vissa våglängder. På grund av detta kommer tester med röd laser samt IR-dioder att utföras.

6.1 Tester av Intensitetskoncept



Figur 14 Princip för Intensitetsskillnad med laser

Material

- IR-diod
- Plastpåsar från FAS Converting
- Laserpekare
- Valsar
- Fototransistor
- Spänningskub
- Digital multimeter

Uppställning

En fototransistor monteras på ett prototypbräde kopplad till en 5 volts spänningskub. Mellan fototransistorn och spänningskuben sitter en voltmeter parallellt över ett motstånd. Detta fungerar på samma sätt som AD omvandlingen i Appendix A.

Ljuskällan i testet är antingen en IR-diod kopplad till en spänningskälla med ett motstånd i serie så att strömmen i kretsen blev 30mA, eller en laserpekare av billigare sort kopplad till en 5 volts spänningskub.

Mellan ljuskällan och fototransistorn löper plastpåsar över två valsar. Valsarna hålls upp av en ställning med två bockar och ett enklare fackverk.

Genomförande

Voltmetern gav utslag i millivolt eller volt-området beroende på intensiteten av det infallande ljuset. Genom att belysa plastens ovansida med en fast fixerad ljuskälla kan olika intensitetskillnader utläsas på den digitala multimetern beroende på vilken sorts plast som användes. Genom att dra plasten över valsarna och mellan fototransistorer och ljuskällan, undersöktes skillnaden i intensitet vid svetsen och perforeringen.

Resultat

Både testet med laser och IR-diod gav fullgoda resultat. Därför kan ingen av metoderna uteslutas på grund av några tekniska begränsningar, men än har bara ett begränsat antal påsar testats så båda metoderna får testas vidare innan det bestäms vilken som ska användas.

6.2 Vidareutveckling av intensitetskonceptet

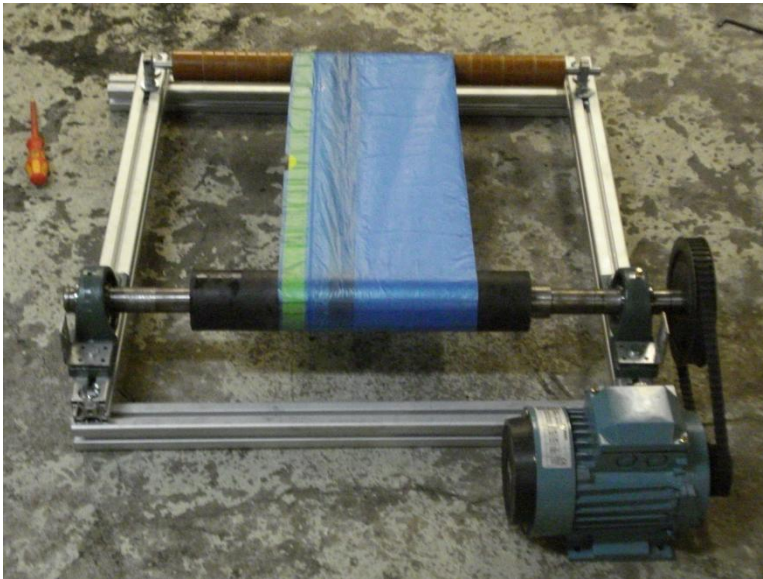
Testet av intensitetskillnadskonceptet gav goda resultat. De frågetecken som finns är bland annat hur signalhanteringen ska lösas på bästa sätt och ifall alla olika sorters plaster kommer kunna användas i detta koncept.

7 Vidareutveckling av prototyp

Under våren 2008 tillverkades en första prototyp för att vidare testa principen i mer verkliga förhållanden. En rigg och ett första styrkort togs fram och utvärderades.

7.1 Rigg

Riggen bestod av två valsar varav den ena driven av en asynkronmotor för att kunna driva plasten runt i 300m/min. På riggen monterades sedan sensorhållare vilka lyfte upp plasten lokalt för att ta bort veck och ojämnheter.



Figur 15 Testrigg för räkningen

7.2 Kort

På styrkortet fanns tre processorer, två avkodade sensorvärdena och skickade en signal till den tredje som kollade om de triggade samtidigt. Den tredje var master och hade även hand om kommunikationen med operatören. Anledningen att det blev tre var att AD omvandlingen blev för långsam för att köra båda på samma kärna och att utskrifter på display och inläsning från knappsatsen var långsam så viss data kunde missas.

8 Delresultat

Den prototyp som togs fram visade att det gick att räkna plastpåsar med principen ljusintensitet. Styrkortet räknade den plastpåse som åkte runt i riggen med god precision dvs. den räknade endast över faktiska skarvar och missade ca 1 på 1000. Styrkortet var tillverkat av standardkomponenter vilket medför låga tillverkningskostnader och liten risk att komponenter utgår. Principen är ofarlig till skillnad från den nuvarande vilket var ett krav.

Signalen från sensorn såg inte ut på samma sätt för två olika sorters påsar utan var därför tvungen att kalibreras varje gång man bytte sort. Testunderlaget som skapades byggde på få typer av plast och en testtrigg som inte fullt ut motsvarade verkligheten dvs. samma skarv åkte runt under hela testet.

Fördelar

- God felmarginal
- Billig, standard komponenter
- Ofarlig princip

Nackdelar

- Ej självkalibrerande
- Begränsat testunderlag
- Testtrigg motsvarar inte riktiga maskinen fullt ut

9 Sensor och vidare utveckling av konceptet

Nu fanns det en metod som fungerade under enklare former. Fortsatta arbetet gick därför ut på att undersöka varje del för sig och se om det fanns eventuella förbättringar att göra. Detta för att sedan montera på en riktig maskin och undersöka om ljusprincipen klarade de ställda kraven i verkligheten.

Sensorerna som ska läsa av intensitetsskillnaden i plasten består av en ljuskälla och en ljusreceptor. Ljuskällan avger ljus som penetrerar plasten. Intensiteten på det penetrerade ljuset mäts sedan av en ljusreceptor.

Kravet på ljuskällan är att den är tillräckligt stark för att penetrera alla typer av plaster, vikta som ovikta. Kravet på ljusreceptorn är att den ska vara tillräckligt snabb för att hinna med den snabba förändringen som en passerande skarv medför.

9.1 Förstudie av ljuskälla

En liten förstudie på olika ljuskällor genomfördes för att få en något bättre bild av de olika alternativen som fanns.

LED

Light Emitting Diodes, förkortat LED är små halvledare. När likspänning strömmar från anoden till katoden genom en LED kommer elektroner att förlora energi och som följd av det utstråla ljus. Beroende på vilken typ av halvledarmaterial som används kommer spänningsfallet över LED:en att variera och ljus av olika våglängd att utstrålas.

Tabell 6 Lysdioder våglängd och material

Färg	Våglängd	Spänningsfall	Ex. Halvledarmaterial
Infraröd	$\lambda > 760$	$V < 1,63$	AlGaAs
Röd	$610 < \lambda < 760$	$1,63 < V < 2,03$	AllnGaP
Gul	$570 < \lambda < 590$	$2,10 < V < 2.18$	GaAsP
Grön	$500 < \lambda < 570$	$2,18 < V < 2,48$	InGaN
Blå	$450 < \lambda < 500$	$2,48 < V < 2,76$	InGaN
Ultraviolett	$\lambda < 400$	$V > 3,1$	AlGaN

(Avagotech.com, 2009)

Det ljus som utstrålas ifrån en LED håller en väldigt stabil intensitet ifall strömmatningen är stabil runt ett värde. En normal LED trivs bäst med en strömmatning på runt 20-60 mA. Ljuset från en LED håller sig även inom ett väldigt snävt spektrum på våglängden vilket gör den effektiv som ljuskälla.

Lysrör

Att använda lysrör som ljuskälla har fördelen att de är billiga och lätta att montera och det behövs bara en. Att använda ett lysrör som sträcker sig över hela påslängden har fördelen att placeringen av fototransistorerna blir enklare. Lysrör finns i olika våglängder bland annat vitt (450 – 760nm) och UV (400nm). Lysrör har nackdelen att intensiteten förändras varje gång det elektriska fältet ändras vilket i praktiken innebär att de blinkar i 100Hz.

Infravärmare

En annan sorts ljuskälla är att använda infravärmare som sedan avläses med en IR-sensor. Den har samma fördelar som ett lysrör vilket innebär att fototransistorn kan placeras var som helst då ljuskällan går längs hela påsbredden. En nackdel kan vara att effekten till värmaren endast omvandlas till värme och inget synligt ljus. Därför kan plasten bli väldigt varm om banan går långsamt. En dimmer skulle nog behöva monteras också.

Val av ljuskälla

Efter förstudien valdes LED som ljuskälla för sensorerna. Både lysrör och infravärme hade så pass stora nackdelar att de inte kunde vara något seriöst alternativ som ljuskälla.

Ett antal olika typer av lysdioder med varierande våglängd och intensitet införskaffades. Tester av enklare slag genomfördes med resultatet att en IR LED med en strömmatning på 100mA används som ljuskälla till sensorn.

9.2 Val av ljusreceptor

Ingen djupare förstudie utfördes vid val av ljusreceptor. Eftersom en ljuskälla i form av en LED används finns det två bra val av ljusreceptor, fotodiod och fototransistor. Fototransistorn är långsammare än fotodioden och därför används en fotodiod som ljusreceptor. (Avagotech.com, 2009)

10 Elektronisk hårdvara

I detta kapitel förklaras i grova drag hur kretskortet är uppbyggt.

10.1 Förstudie för processor val

I tidigare prototyp har en styrning genomförts med hjälp av en microcontroller. Alternativ till microcontrollers finns och då i synnerhet FPGA. Därför undersöks alternativet att använda en FPGA istället för en microcontroller.

FPGA

FPGA går att göra i princip hur snabb som helst eftersom den kan utföra flera operationer parallellt men har nackdelen att den är omständlig att programmera, svår att felsöka och långsam på att utföra multiplikation. FPGA:s processorkärna går att dela upp så att en del sköter behandlingen av indatan en annan jämför datan från de olika sensorerna och skickar ut signalen till PLC:n, detta gör att behandlingen blir väldigt snabb.

Microcontroller

Microcontrollern, även kallad MCU (Micro Controller Unit) eller μC består av en microprocessor med integrerat minne och minst ett par I/O interface pinnar. Minnesmängden varierar mellan olika tillverkare och processorfamiljer men består för det mesta av en del ROM och en del RAM. ROM minnet används för att lagra programkoden i och är i storleksordningen 512 bytes till 4096 bytes, RAM minnet används för att logga data av olika former och varierar stort i storlek mellan olika tillverkare och familjer. (Minns, 2007)

Microcontrollern kan vara baserad på en 8, 16, 32 eller 64 bitars arkitektur. Antal bitar indikerar hur många bitar varje registerpost består av och hur mycket data som behandlas under varje klockcykel. Fördelen att använda en 16 bitars arkitektur mot en 8 bitars är t.ex. att en 16 bitars integer får plats i en minnesplats, till skillnad från 8 bitars arkitekturen där samma integer måste lagras i två bytes.

I en microcontroller finns det ofta en ADC (Analog Digital Converter).

10.2 Val av processor

En viktning mot olika parametrar genomfördes och microcontrollern blev det överlägset bästa alternativet utifrån användningsområdet. En FPGA blir dyr både i utveckling och i styckepris och detta kompenseras inte upp tillräckligt av den extra snabbhet man får. Därför kommer en microcontroller att användas.

Kriterier	Vikt	FPGA	Microcontroller
Pris	2	-	+
Utvecklingstid	5	-	+
Teoretisk samplingsid	3	+	-
Aritmetisk förmåga	4	-	+
Driftsäkerhet	5	+	+
Felsökningsförmåga	4	-	+
	Summa	8	20

Tabell 7 Viktad urvalstabell processorval

10.3 Val av microcontroller

Det finns två typer av microcontroller som uppfyller de krav som har ställts PIC och ATMEL vilka beskrivs nedan.

PIC

PIC (Peripheral Interface Controller) är en familj av olika microcontrollers som utvecklats av Arizona Microchip Technology. Alla Microchips Controllers bygger på en RISC arkitektur, med separerad data- och instruktionsbus. Instruktionsbussen är bredare än databussen och möjliggör för snabba instruktioner. PICen implementerar även pipelining som laddar en ny instruktion under tiden en annan instruktion bearbetas. Vilket ytterligare ökar prestandan.

PIC familjerna finns i 8, 16 eller 32 bitars arkitektur. Det finns även en variant med en integrerad DSP (Digital signal processor). PIC16 och PIC18 familjerna bygger på en 8 bitars arkitektur medans PIC24 och dsPIC30 bygger på en 16 bitars arkitektur. (Microchip produktfamiljer, 2009)

Stacken i PICen är hårdvarubaserad och kan inte kommas åt av mjukvaran. Detta kan skapa problem vid viss hög-nivå programmering och utesluter rekursiva programmeringstekniker då stacken enbart kan innehålla maximalt ett tiotal hopp. De senare dsPIC familjerna implementerar en mjukvarustack som teoretiskt tillåter lika många hopp som det finns tillgängligt minne.

Priset på en PIC ligger mellan 30kr och 150kr om man beställer enstaka kretsar.

ATMEL

ATMEL AVR är en 8 bitars RISC processor. ATMEL har en fördel att instruktionerna sker "pipelined" det vill säga att nästa instruktion införs under tiden som föregående exekveras, detta medför att de flesta instruktioner tar en eller två klockcykler. ATMELs instruktionsminne och ramminne delar adress vilket gör att antalet nästlade programhopp bara begränsas av ramminnets storlek. En nackdel med ATMEL är att många av I/O pinnarna är dedikerade till antingen ingångar eller utgångar. En fördel med ATMEL är att i programmeraren finns JTAG som möjliggör "In circuit" debugging

Kriterier	Vikt	PIC16/18	PIC24	ATMEL AVR
Pris	2	3	3	3
Utvecklingskostnad	2	3	1	2
Utvecklingstid	5	3	2	1
Teoretisk samplingstid	3	1	3	2
Aritmetisk förmåga	4	2	3	2
Driftsäkerhet	5	2	3	2
Felsökningsförmåga	4	1	3	3
	Summa	52	66	54

Tabell 8 Viktad urvalstabell för val av microcontroller där 3 är högst och 1 är lägsta betyg

10.4 Kretskortet

Under examensarbetet togs ett nytt styrkort för att behandla signalerna fram. Huvudanledningen till att ta fram ett nytt kort var för att ersätta den gamla arkitekturen som byggde på 3 separata microcontrollers. Den microcontroller som användes i första delen av projektet klarade nämligen enbart att sampla tillräckligt snabbt för en sensor. Då 2 sensorer fanns krävdes 1 microcontroller per sensor plus ytterligare en microcontroller för att samordna de två andra. På det nya kortet används enbart en microcontroller.

Fördelen med att använda PIC24 familjen är att den är så pass snabb att det räcker med en microcontroller för att hantera båda kanalerna. Detta är bra eftersom det minskar konstruktionstiden av kretskortet samt att kommunikation mellan olika microcontrollers går långsamt och är en stor källa till fel. De olika processorerna som har gåtts igenom kostar ungefär lika mycket och därför är det även en kostnadsbesparing att bara använda en microcontroller.

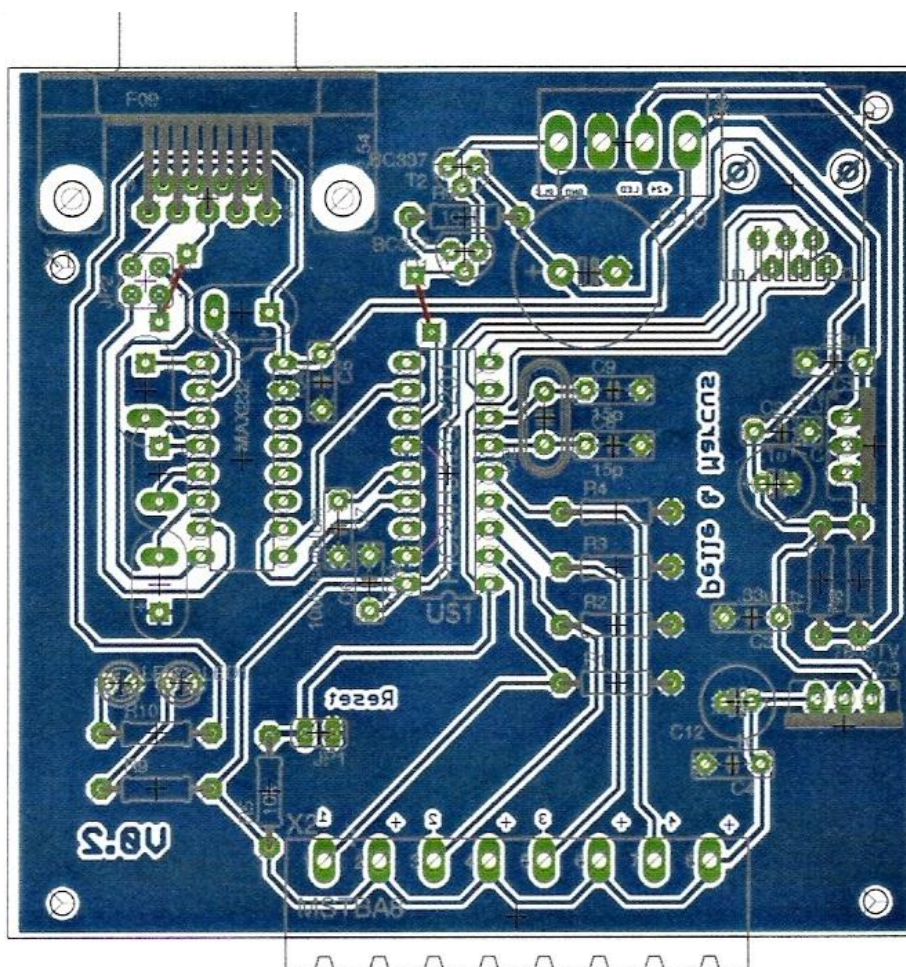
PIC24 familjen från Microchip valdes och specifikt microcontrollern *PIC24HJ12GP201* till styrenhet på styrkortet. Till det valdes diverse spänningsregulatorer, transistorer, kondensatorer samt en RS232 drivkrets. Skruvplintar av vanlig sort användes för att ta emot signalen från sensorerna. Nedan ses en illustrativ bild där komponenternas placering framgår. Kopplingschema finns i Appendix A.

Komponenter

- PIC24HJ12GP201(Microcontroller)
- MAX232(Drivkrets seriell port)
- TS7805(24-5V)
- LF33ABV(5-3,3V)
- 8000SKY (Oscillator)

Prestanda

- 40 MIPS (million instructions per second)
- 4 A/D kanaler
- 250 kHz samplingshastighet. per A/D kanal
- RS232 via DSUB9 i 115200 bit/s



Figur 16 Kretskortslayout

Funktionalitet

Styrkortet fungerar som så att signalen från sensorn tas emot på någon av de fyra ingångarna längst ner på kortet. Dessa är sedan kopplade till varsin AD-ingång på microcontrollern (AN0 till AN3). När microcontrollern anser att en skarv är detekterad går en utgång (RB9) hög i exakt 100ms. Den porten är sedan kopplad genom en serie transistorer för att från 3.3V förstärkas till 24V. Denna 24V signal går sedan till en utgång på styrkortet för att sedan kopplas till en PLC. Det är denna 24V signal som signalerar skarvar i plastbanan.

Styrkortet matas från maskinen med en spänning på 24V. På kortet sitter kretsar och komponenter som kräver antingen 5V eller 3.3V matningsspänning. För att sänka spänningen från 24V till 5V och 3.3V användes spänningsregulatorer i serie med varandra. Först skalades 24V till 5V och från 5V till 3.3V. På kortet finns mycket olika störningar att ta hänsyn till och olika avkopplingskondensatorer användes. En större elektrolytkondensator ansluts vid varje spänningskälla. En mellan jord och 24V, en mellan jord och 5V samt en mellan jord och 3.3V. Utöver det användes keramiska avstörningskondensatorer mellan jord och matning till varje IC-krets som användes.

RS232

På kortet lades även en RS232 länk in i form av en DSUB9 hona för att enklare kunna interagera med sensorn under drift, felsöka och för att logga data. PIC24H har bra stöd för RS232 men kommunicerar med logiska nivåer på endast 0 till 3.3V. (Microchip, 2008) En vanlig PC kommunicerar med logiska nivåer som ligger högre och därför krävs en krets mellan dator och microcontroller.

För att interagera med styrkortet användes en vanlig PC med ett terminalprogram. Genom terminalprogrammet sändes kommandon av användaren i form av bokstäver, svar returnerades och presenterades av terminalfönstret. Exempelvis kunde sensorvärdena och diverse signalbehandlingsvariabler hämtas via kommandon på detta sätt.

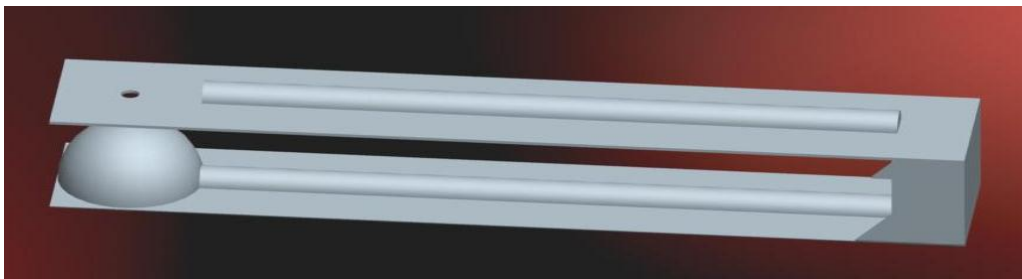
11 Konstruktion av sensorhållare

Riggen för att hålla upp sensorerna är gjorda med en tungprincip som kan stickas in i veck i plasten. Detta gör att läsbarheten ökar för annars nästintill omöjliga plaster. Tungan sticks in genom enbart ett lager plast vilket innebär att perforeringen alltid syns även om plasten absorberar ljus så att inget går igenom till sensorn. En annan faktor som ökar läsbarheten är att plasten lyfts upp lokalt av en halvsfär som har egenskaperna att antalet veck minskar och att perforeringen dras isär.

En nackdel med de starka ljuskällorna som valts är att om plasten är i princip genomskinlig så bottenas fotodioderna och den högre intensiteten som blir vid en perforering inte kan märkas. Därför installerades en potentiometer som begränsar strömmen till IR-dioderna dvs. minskar intensiteten på det ljus som skickas ut. En intressant upptäckt som gjordes är att då den skickade intensiteten minskade linjärt minskade den avlästa också linjärt om det inte var någon plast mellan men om det var plast minskade det avlästa exponentiellt. Detta kan utnyttjas genom att om strömmen optimeras fås en större spik vid perforering än vad som annars skulle ske vilket i sin tur förenklar detekteringen.

11.1 Sensorhållare

Efter tester framkom att om perforeringen lyftes upp väldigt lokalt över exempelvis en sfär drogs perforeringen isär något vilket ökade intensitetsskillnaden och risken för veck och andra ojämnheter eliminerades. Hållarna tillverkades för hand av vanlig stålplåt. Hållaren var tvungen att vara böjstyv för att klara av krafterna som uppstår vid acceleration och retardation, om tungan böjs ner ökar avståndet till IR-dioden och intensiteten minskar vilket stör kalibreringen och kan eventuellt leda till felräkning av skarvarna. För att bli tillräckligt böjstyv men fortfarande tillräckligt tunn för att kunna träs in i ett veck skapades långa försänkningar.



Figur 17 Ritning av sensorhållare

11.2 Testrigg

På testriggen som även användes under första delen av projektet fungerade de nya sensorhållarna utmärkt. Dessa tester var trots allt bara en förenkling av de förutsättningar som fanns på de riktiga upprullningsmaskinerna. Detta för att på testriggen var det samma skarv som åkte förbi gång på gång och krafterna var konstanta.

11.3 Maskin

För att få reda på om de nya sensorhållarna klarade krafterna monterades de in i en riktig maskin. Det visade sig att krafterna var mycket större än de som tidigare hade antagits och sensorn böjde ner för mycket vid acceleration. Detta löstes under testerna genom att fixera sensorhållaren mot en balk. Eftersom hållarna var gjorda för hand och utan att räkna på böjstyvheten går detta med all sannolikhet att konstruera bort om en liknande princip kommer att användas vid ett senare tillfälle.

12 Signalbehandling

För att kunna göra bättre algoritmer för skarvigenkänning och autokalibrering samlades data från samplingarna in och behandlades i MATLAB. Detta för att kunna laborera med samma data för olika algoritmer och ta fram den bästa.

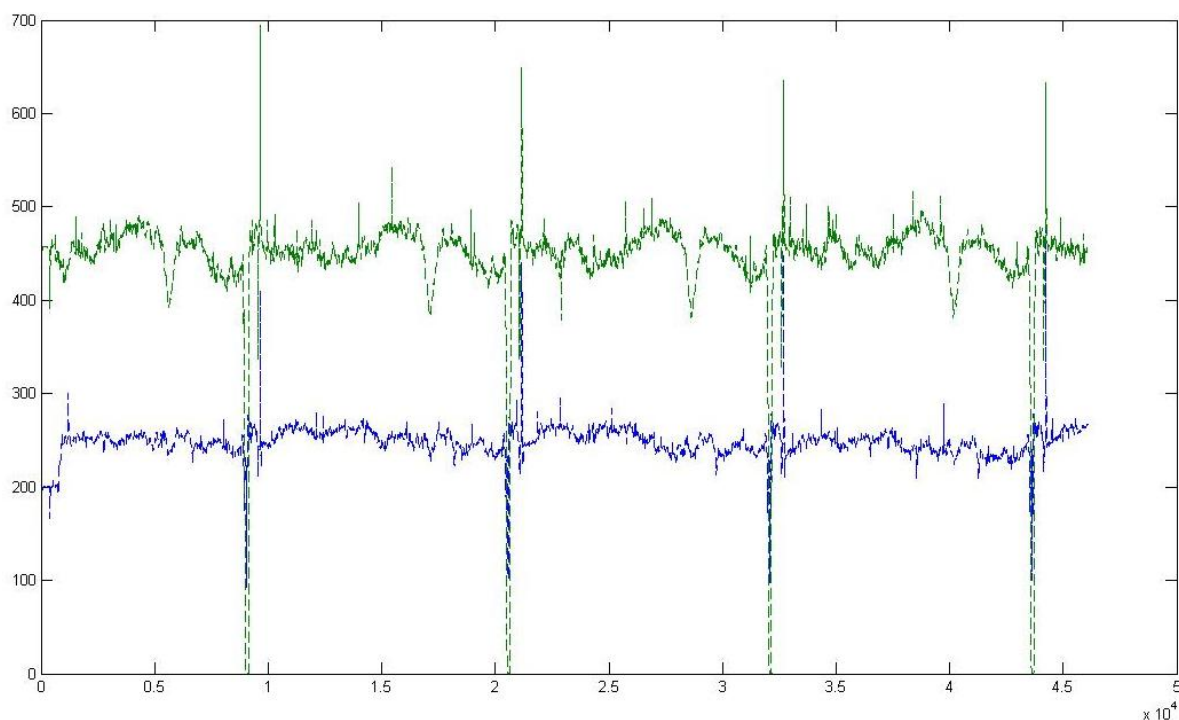
12.1 Testförutsättningar

Kurvdata samlades in via kortet och skickades till MATLAB via RS232. Hastigheten på riggen var liten för att få med så mycket data som möjligt. RS232 är ett långsamt protokoll med datahastigheter på 115200 b/s. Två kanaler samplades parallellt och varje kanal var på 5 bytes och sen för att skilja dem skickades även ett TAB och ett Enter tecken, vilket tillsammans ger 13 bytes per sampling. Detta ger en samplingsfrekvens på $\frac{115200}{8 \cdot 13} = 1,1 \text{ kSamplingar/sekund}$.

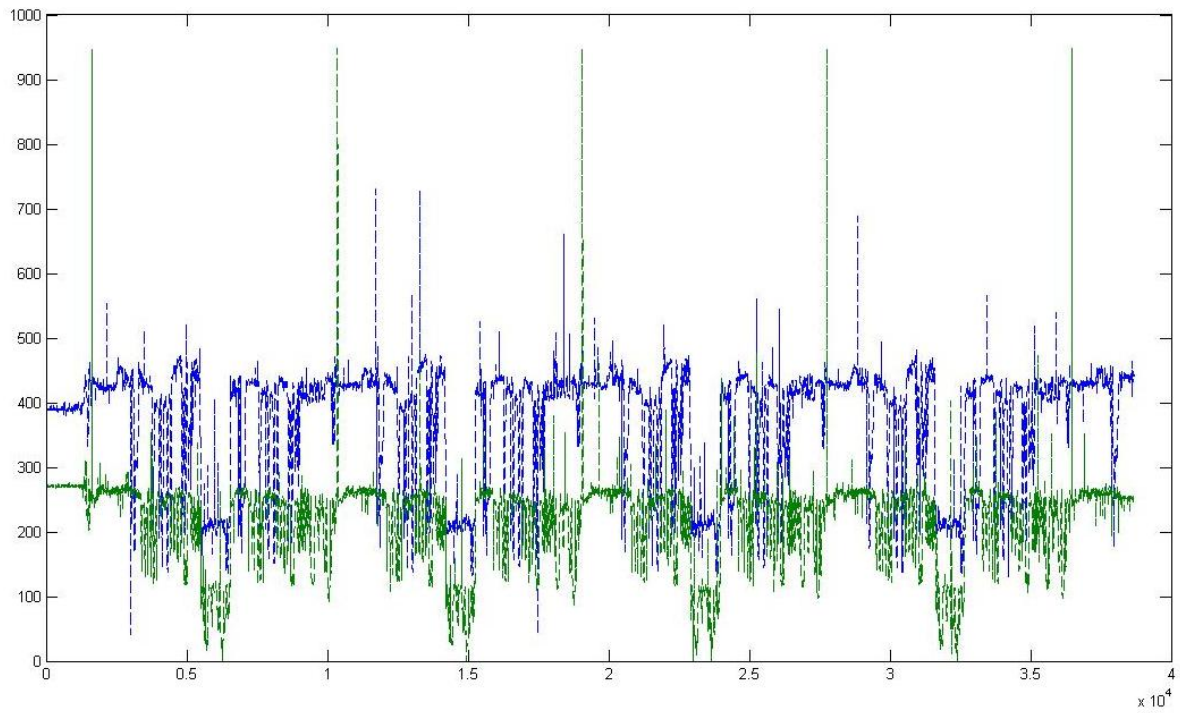
Y-axeln motsvarar det samplade AD-värdet som ligger mellan 0 och 1024. X-axeln motsvarar antalet samplingar.

12.2 Kurvdata

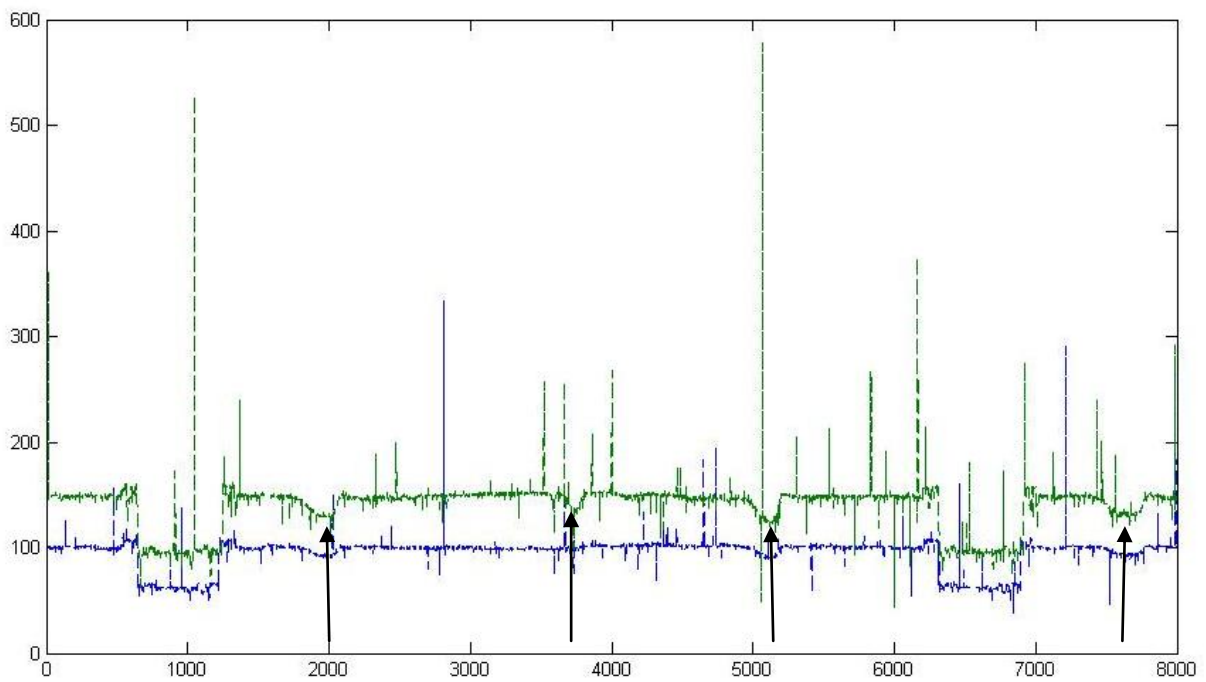
En typisk graf från sensorn ligger på en relativt jämn nivå och ökar då en skarv kommer, detta gäller då påsen inte är helt genomskinlig och inte har tryck. Om påsen har tryck så minskar intensiteten över trycket eftersom det är ljuset som har passerat påsen som avläses. Om påsen är nästintill genomskinlig så minskar mängden genomsläppt ljus över en perforering, vilket kan bli problem då en autokalibrering ska implementeras.



Figur 18 Graf över 4 påslängder för färgad sopsäck. spikarna nedåt är där påsarna är fogade.



Figur 19 Graf över 5 skarvar på plast med tryck.



Figur 20 Graf över 4 skarvar på genomskinlig plast. Som synes går intensiteten ner vid en skarv.

12.3 Avkodningsmetoder

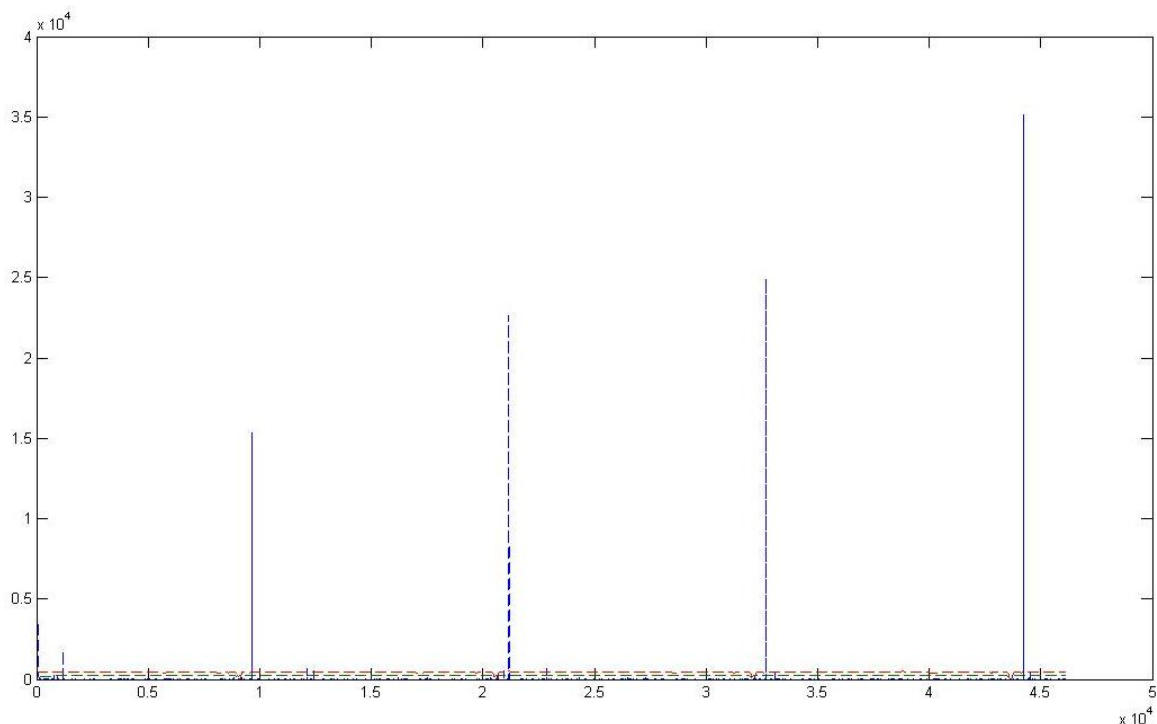
Grundprincipen för att hitta skarven går ut på att hitta medelvärdet av kurvan och dra bort det från det inlästa värdet. Detta resulterar i att de flesta värdena blir 0 detta gör att avvikelserna blir lättare att urskilja. Då en genomskinlig påse ska detekteras inverteras signalen så samma algoritm kan användas.

Singelsensor

Den enklaste metoden för att hitta en skarv är att bara använda sig av en sensor. Det har fördelen att koden blir mer lättöverskådlig och att det går att sampla dubbelt så fort. En nackdel däremot är att sensorn blir mer känslig för störningar och kan därför trigga för mycket. Metoden går ut på att man tar fram ett värde som ca 90 % av samplingarna ligger under. Detta värde dras sen bort från det inlästa värdet och blir 0 om det skulle blivit negativt. Därefter kvadreras resten och jämförs mot ett triggvärde. Triggvärdet sätts som en faktor av maxvärdet från föregående skarv.

Multiplikation

Ett sätt att öka säkerheten mot att hitta för många skarvar är att använda två sensorer och utnyttja samma algoritm som singelsensor använder sig av men med skillnaden att sensorvärde 1 multipliceras med sensorvärde 2. Detta resulterar i att störningar och avläsningsfel elimineras, men nackdelen är att sensorerna måste kalibreras så att de ligger helt parallellt med perforeringarna då perforeringarna är runt 0,1 mm breda.



Figur 21 Ett exempel på den multiplicerade signalen. Den bygger på samma rådata som Figur 18

Fördröjning

Ett sätt att komma förbi problemet med att sensorerna måste vara helt parallella är att låta sensorerna verka individuellt likt singensensoralgoritmen och att senare undersöka om båda triggas ungefär samtidigt. För att kolla tidsskillnaden mellan de båda sensorerna räknas antalet samplingar

som den första är hög och om den andra triggas inom ett rimligt antal samplings i förhållande till hur länge den första var hög så skickas en puls till PLC:n som då räknar upp.

12.4 Medelvärdesbildning

I alla metoder för skarvigenkänning behövs ett medelvärde. För att få fram ett medelvärde som uppdateras inom rätt tidsperiod utvecklades en del olika teorier. Uppdateringstiden måste vara tillräckligt långsam för att inte påverka skarvalgoritmen, men ändå tillräckligt snabb för att ändra sig då ljusförhållandena i rummet och tjockleken på plasten ändras.

Moving average

En enkel princip som först testades var att konstruera ett digitalt lågpasfilter vilket är lite omskrivet för att passa den begränsade aritmetiska förmågan och arbetsminnet som processorn har.

$$\hat{y} = (y_{(k)} * a + y_{(k-1)} * b + y_{(k-2)} * c + y_{(k-3)} * d) / (a + b + c + d) \text{ (Olsson \& Rosen, 2005)}$$

Där $\{a + b + c + d\} = 2^n$ för att kunna bitskifta istället för att använda vanlig division. Viktningen väljs utifrån hur långsamt filter man vill ha. Fördelen med denna algoritm är att den inte kräver så mycket processortid och att den är utspridd jämnt över varje sampling. Nackdelen är att filtret blir för snabbt då samplingshastigheten ligger på ca 200kHz. Den bör inte vara snabbare än en tiondelssekund för att inte störas för mycket av påsar med tryck.

Medelvärde av fler samplings

För att få in fler samplings utvecklades en princip med en [2,16] matris. I ruta[1,1] sätts medelvärdet av de 16 första samplingsarna, i ruta [1,2] de nästföljande 16 och så vidare till hela kolumnen är full. Då skapas ett medelvärde av de 16 medelvärdena och läggs in i ruta [2,1] och så vidare till hela matrisen är full bildas ett medelvärde från totalt 4096 samplings. Denna operation går att utöka för att få fler samplings. Nackdelen är att då matrisen blir full tar det mycket processortid att komma ur den så att det går att börja om igen. Den ger även ett medelvärde av allt vilket gör att den blir något sämre för påsar med mycket tryck.

Medianvärde

Istället för att ta ett medelvärde kan man undersöka vilket värde som förekommer mest frekvent och på så sätt få fram något som kan liknas vid ett medianvärde. Detta programmerades på följande sätt, en vektor med 1024 platser skapades och för varje ny sampling adderas den platsen som motsvarar värdet på samplings med ett. Då en plats i vektorn överstiger ett angivet värde används detta värdet som median. Se exempel nedan:

```
Median1[AD_Värde1]++;  
If(Median1[AD_Värde1] > x) medianSensor1 = AD_Värde1;
```

Denna metod fungerade utmärkt i MatLab där arbetsminnet var i princip obegränsat men på PIC:en visade det sig att arbetsminnet bara var 2kB. Eftersom en integer är 2 Bytes och, vektorerna är 1024 djupa och en vektor för varje sensor behövdes innebar detta att den erforderade mängden minne som behövdes var dubbelt den som fanns att tillgå lades denna metod på is.

Procent kvantil

Då de andra metoderna hade sina brister skapades en ny metod som först gjorde en medelvärdesbildning av ett mindre antal samplingar. Sen jämfördes den nya samplingen mot medelvärdet och om den var över räknades en klickare upp. När sen ett större antal samplingar var gjorda kontrollerades hur många procent som var över respektive under och utifrån detta höjdes respektive sänktes medelvärdet så att i snitt 95 % av alla samplingar var under. Detta var ett bra sätt att ta bort mycket av bruset på signalen. Nackdelen med denna är att regleringen är svår att få till då det är svårt att ställa in bra parametrar som fungerar för alla plaster och banhastigheter. Om till exempel banan går av eller glider av sensorn kommer medelvärdet att öka till max och sen när banan kommer tillbaka tar det tid innan regulatorn har sänkt medelvärdet.

12.5 Tillämpning

Det som bör tas i beaktning i signalbehandlingen är att det finns många sorters störningar. Ett exempel på långsam störning är ljusförhållandena mellan dag och natt, en något snabbare är om belysningen i lokalen släcks. Sen tillkommer problem från banan, det kan vara veck eller håligheter i plastmaterialet vilket inte får tolkas som skarvar. En annan utmaning är att optimera parametrarna för att passa till just den hastighet som banan körs i då denna kan variera från 0 till 200 m/min.

13 Mjukvara

För att behandla de sensorvärden som plockades in till microcontrollern fodras en hel del programmering. Koden tar även stor hänsyn till de snäva tidskrav som finns tack vare den snabba A/D-omvandlingen.

13.1 Kodoptimering

Det speciella med att programmera en microcontroller är att det finns olika skillnader mellan de olika familjerna av microcontroller. T.ex. får beaktning tas till hur lång tid aritmetiska operationer tar på just den microcontrollern som används. Andra saker att beakta är t.ex. hur lång tid ett avbrott tar (göra en s.k. "context save" mm.).

Anledningen till att dessa överväganden bör göras i detta projekt är att det finns hårda tidskrav. Tidskravet som finns är att det mellan två A/D-samplingar inte får existera så pass mycket kod att värsta exekveringstiden för denna kod är längre än samplingstiden för A/D-omvandlingen. Självklart är det kritiska inte att alltid ligga på en värstatid under samplingshastigheten för A/D-omvandlaren utan att ha detta i åtanke. Faran ligger i att inte beakta detta och hamna på värstatider som ligger långt över samplingshastigheten på A/D-omvandlaren.

Exekveringstider

På den specifika microcontrollern som används i detta projekt, Microchip PIC24H12GP201, finns goda förutsättningar för en effektiv kod. Nedan följer lite operationer och dess exekveringstid mätt i hur många instruktioner de tar att utföra.

Funktion	Tid i instruktioner
Multiplikation	1
Division	1 instruktion per bit i täljaren
Avbrott	4
Bitskiftning	1

Tabell 9 Exekveringstid för olika operationer (Microchip, 2008)

Som man kan utläsa ur tabellen är multiplikation gjort på endast en instruktion och är därför väldigt snabbt. Detta gör att man inte behöver ta någon större hänsyn till multiplikationer om talen håller sig till heltal (16 bitar). Division är däremot kostsamt. För att dividera ett tal med ett annat krävs det lika många instruktioner som antal bitar i nämnaren.

Problem med division

Om man t.ex. dividerar talet 100 med 10 krävs det 4 instruktioner. Talet 10 på binär form är 0000 0000 0000 1010, den mest signifikanta bit ligger på plats 4 och alltså tar divisionen 4 cykler. Om man istället delar 100 med 10 000 som i binär form skrivs 0010 0111 0001 0000 tar det 12 instruktioner (MSB ligger på plats 12).

För att komma till rätta med division och slippa ödsla instruktioner i onödan kan man vid statiska divisioner använda sig av bitskiftningar. Ifall division enbart delar med 2, 4, 8, 16, 32 o.s.v. (2^n) kan bitskiftning användas istället för division. Som tabellen utläses tar bitskiftning av en integer en

instruktion i anspråk. Så istället för att som i exemplet ovan dela med talet 10 är det mer önskvärt om det går, att dela med säg 8 eller 16. Detta skulle innebära en bitskiftning av talet 100_3 eller 4 gånger ($100_{>>3}$ eller $100_{>>4}$).

När ett språk som C används istället för assembler vid programmering av en microcontroller får inte detta glömmas bort. Bara för att divisionen får plats i en C rad innebär det inte att den enbart tar en instruktion att utföra.

Varför kodoptimering till detta styrkort?

För att visa vikten av kodoptimering kan följande konfiguration beaktas. Hastigheten på oscillatoren är 8MHz som med hjälp av en "Phase locked Loop" (PLL) i microcontrollern har blivit 80Mhz. Microcontrollern tar 2 klockcykler att utföra en instruktion vilket i slutändan ger en hastighet på 40MIPS. MIPS står för Million Instructions Per Second. A/D-omvandlaren kan som mest jobba med en frekvens på 1 miljon samplingar per sekund. I detta fall finns 2st A/D kanaler (1 per sensor) inkopplade och en konfiguration som ger ett avbrott när båda kanalerna har blivit omvandlade. Detta ger ett avbrott som har en frekvens på 500kHz.

Alltså finns en hastighet på 40MIPS och en A/D-omvandling på 500kHz. För att inte störa A/D-omvandlingen krävs alltså att det mellan dessa inte tar för många cykler i anspråk. 40MIPS kan vid en första anblick låta som väldigt mycket men om vi räknar ut hur många instruktioner det ger mellan varje A/D-omvandling fås enbart 80 instruktioner. På dessa 80 instruktioner skall en stor del av signalbehandlingen utföras och därav vikten att optimera koden.

13.2 Initieringar

För att få en microcontroller att fungera tillfredställande krävs en rad konfigurationer av den specifika microcontrollern som används. Beroende på vad microcontrollern ska utföra krävs olika konfigurationer. Skall den kunna kommunicera via t.ex. RS232 eller I2C? Ska vissa portar vara in eller utgångar? Kommer den interna oscillatoren att användas eller ska en extern användas? Ifall en extern används, vilket hastighetsintervall ligger kristallen i? Förutom de egna algoritmerna krävs en del konfiguration av microcontrollern för att den ska fungera tillfredställande till den tillämpningen man har valt.

De initieringarna som görs och som finns beskrivna i detalj i Appendix B är:

- Initiering av klockan
- Initiering av alla avbrott
- Initiering av AD-omvandlaren
- Initiering av timern
- Initiering av alla I/O-portar
- Initiering av UART

14 Resultat, svårigheter och lärdomar

I detta kapitel avhandlas tester och dess resultat som genomfördes under projektets gång. Testerna har skett både på rigg i icke industriell miljö samt i FAS testhall på riktiga maskiner.

14.1 Test på rigg

Under styrkortets utveckling genomfördes kontinuerliga tester på den testrigg som konstruerats för ändamålet. Olika plaster testades i olika hastighet och styrkortet kopplades mot ett oscilloskop för att testa dess funktion.

Förutsättning

Testet utfördes på skolan i den egenutvecklade riggen. Förutsättningarna för testet var följande.

- Konstant banhastighet
- En plast som roterades runt
- Hastigheter från 1 – 300 m/min

Testerna utfördes alltså under optimala förhållanden. Ingen acceleration på banan, samma plast som roterade runt. Hastigheten på banan var däremot fullt verkliga mot de hastigheter som förekommer på en riktig maskin.

Plaster

Under testerna fanns tillgång till fem olika typer av plast från FAS. Dessa testades alla under olika hastigheter. Plasterna som fanns var följande.

- Vanlig blå plast
- Tjock svart plast
- Vit plast utan svets
- Tjock plast med mycket varierande tryck
- Genomskinlig plast (fryspåse)

Alla de fem plasttyperna var olika på ett eller annat sätt, men de representerade inte hela repertoaren av plasttyper som kan förekomma hos FAS kunder.

Genomförande

En tillräckligt lång bit av plasten som skulle testas skars ut. Denna fixerades sedan med hjälp av vanlig genomskinlig tejp genom att plastbiten veks runt båda valsarna och tejpades ihop. Efter detta spändes den ena valsen bakåt och på det sättet uppkom en spänning i plasten.

När plasten var monterad drevs den runt av en motor kopplad till ena valsen. Olika banhastigheter testades och utgången ifrån styrkortet som signalerar en skarv kopplades in till ett oscilloskop. På oscilloskopet registrerades alla spikar som motsvarade en plastpåse.

Under genomförandet dök olika problem i själva riggkonstruktionen upp. Plasten hade för vana att antingen slitas upp vid tejpfoggen eller spricka vid perforeringen. Mycket tid gick åt till att tejpa om plaster.

Resultat

Själva mätandet gick bra och på oscilloskopet fanns tydliga spikar i den frekvens som skarvarna passerat. Alla olika typer av plast räknades på ett tillfredställande sätt och en klar och tydlig 24V signal genererades av styrkortet vid varje skarv. Dessa resultat får anses mycket goda.

14.2 Test på maskin

Då resultaten från den egna testriggen var goda gick projektet över till den väsentliga fasen nämligen verkliga tester på maskinen som den var konstruerad för.

Förutsättning

Testerna utfördes på en riktig upprullningsmodul. Dels med enbart upprullningsmodulen där plast matades för hand och dels med både upprullningsmodul, perforerare och avrullare. Förutsättningarna för det testet med alla moduler var följande.

- Varierande banhastighet
- Varierande acceleration
- Varje skarv passerar sensorn endast en gång
- Hastigheter från 1 – 100 m/min

Detta motsvarar i princip verkliga förhållanden. Enda skillnaden är att banhastigheten är något begränsat på grund av att allting kördes offline, vilket gör att det går åt väldigt mycket plast om högre hastigheter skall hållas. Plasten som rullas upp på rullar går nämligen inte att återanvända.

Banhastigheten var aldrig uppe i 100m/min, däremot var plastens hastighet under sensorn ibland högre än banhastigheten. Efter ett rullbyte försöker upprullaren komma ikapp och ta sig igenom den ackumulerade plasten i flytvalsarna.

Den varierande accelerationen och banhastigheten uppstår på grund av rullbytena då upprullaren försöker komma ikapp.

Plaster

I testerna fanns de plaster som testats på riggen samt en ny att tillgå. Den nya plasten var genomskinlig och mycket hård i sin perforering.

Genomförande

Två typer av tester genomfördes. Ett mer grundläggande test med enbart upprullarmodulen och där plasten matades för hand. Plasten löpte genom flytvalsen vilket tyngde ner plasten och skapade en spänning i banan. Genom att dra en skarv förbi sensorn kunde man genom oscilloskopet kolla att allting fungerade som det bör.

När första steget var färdigt och räkningen fungerade för manuellt matad plast började det verkliga testet. Detta innebar att styrkortet kopplades in mot avrullningsmaskinens PLC och en hel produktionsbana var installerad och matade plast till upprullaren.

PLC:n använde styrkortets pulser för att räkna antalet påsar som hade löpt över sensorn. Med hjälp av den informationen producerade upprullaren påsrullar med ett visst antal påsar per rulle och i detta testet valdes 10 påsar per rulle.

Två olika plaster testades. Den ena var den nya typen som var genomskinlig och hård i perforeringarna. Den andra var den tidigare testade tjocka platen med tryck. Till den tjocka platen las extra vikter i flytvalsens för att kompensera styvheten.

Resultat

Den första platen som testades var den genomskinliga platen med hård perforering och manuell matning. Det första som kunde noteras var att signalnivåerna från en perforering inte var den tänkta. Signalnivån från en perforering var en avtagande kurva, alltså släpptes mindre ljus genom perforeringen än på den homogena delen av plastpåsen. Detta trots att sensorhållaren med sin sfäriska yta vek upp perforeringarna när dessa passerade igenom sensorn. För att tackla detta gjordes modifieringar i koden.

Med modifieringarna i koden testades sedan den genomskinliga plastpåsen i en hel bana med upprullare, svetsare och perforerare samt plastmatare. Med PLCn inställd på 10 påsar per rulle producerades en mängd rullar med rätt antal påsar. Det som kunde noteras trots goda resultat var att vid acceleration på platen i banan böjde sig sensorhållaren på grund av de stora krafterna.

Ytterligare en till plast testades och detta var den tidigare testade tjocka plastpåsen med tryck. Återigen fick koden modifieras för att fungera mot signalnivåer som ökar vid perforeringar. Räkningen för denna plast fungerade bra under samma förutsättningar som vid testriggen. Med de extra vikterna i flytvalsens ökade krafterna vid accelerationer över sensorhållaren alltför mycket. Detta bidrog till att sensorhållaren böjdes och intensiteten ändrades med böjningen.

För att komma åt detta problem stabiliserades sensorn mot en underliggande balk. Problemet löstes, men frågetecknen dök upp då tungprincipen inte kan utnyttjas med denna balk under sensorhållaren. Vid just denna plast gick det dock att lysa igenom då den var ovikt.

15 Slutsats

Då den metod som har framarbetats inte fungerar tillfredställande med avseende på brister i självkalibreringen kan konstateras att en ny sensortyp måste tas fram, denna måste ge tydligare och mer likvärdiga utslag än ljuset som avslutningsvis har behandlats i denna rapport. Kretskortet som tagits fram har stora möjligheter att klara av att avkoda signaler i de hastigheter som behövs om bara en annan sensortyp kan konstrueras.

Det fundamentala problemet med principen var att ljusgenomsläppet ibland ökade och ibland minskade då en perforering åkte över. Detta gjorde självkalibreringen mycket svårare och nästintill omöjlig utan att återkoppla hastigheter eller påslängder.

Under sluttesterna på maskin då den genomskinliga plasten med hård perforering testades framkom just det att signalnivån minskade vid en perforering. Detta ställer till stora problem då ingen distinkt skillnad kan ses mellan en nedåtgående signalnivå på grund av tryck eller perforering, utan nedgången har samma karakteristik och kan således inte mjukvarumässigt skiljas från varandra.

Förutsättningen som fanns för att en smidig automatisk kalibrering utan inblandning från en operatör skulle fungera behövs en distinkt och likvärdig signal som är likadan för alla plasters perforering. Detta troddes ha uppnåtts genom att lokalt höja upp plasten så att en perforering alltid släpper igenom mer ljus än någon annan del på plasten. Ifall detta hade varit fallet hade kriterierna för den tänkta automatiska kalibreringen varit uppfyllda.

Ett annat problem under testerna var att då sensorhållaren var konstruerad för kunna stickas in under vikt plast blev böjstyvheten mindre bra. Detta resulterade i att sensorn böjde sig för mycket under accelerationer och detta främst vid styva plaster då extra vikt används i flytvalsarna. Böjningen av sensorhållarna gör att intensiteten ändras vid dessa accelerationer och ger sämre förutsättningar för att avkoda signalen på ett korrekt sätt.

Denna brist tillsammans med andra som har uppkommit under projektets gång har stora möjligheter att elimineras. Men vid upptäckten att alla plastsorter inte gav positivt utslag vid perforering försvann möjligheten för en helt självkalibrerande sensor. Detta i sin tur gjorde att projektet lades ner och ingen vidare utveckling av de andra mindre problemen utfördes.

Det kan vara lönt att gå tillbaka till början och omvärdera de förslag som har förkastats. Om man går tillbaka och tittar på principlösningarna en gång till, samtidigt som de icke fungerande förslagen utesluts finns ett alternativ kvar, nämligen optisk markör. Denna metod uteslöts därför att den bedömdes som dyr då en rörlig kostnad tillkom. Men utan andra alternativ kanske denna bör undersökas närmare.

16 Litteraturförteckning

Avagotech.com. (den 08 03 2009). Hämtat från

http://www.avagotech.com/pages/en/leds/avago_led_through_hole_lamps/

FAS Converting. (u.d.). Hämtat från FAS Converting: www.fasconverting.se den 09 01 2009

Jönsson, G. (2002). *Våglära och Optik*. Lund: Teach Support.

Meriam, J. L. (2003). *Engineering Mechanics Dynamics*. NewYork: Wiley.

Microchip. (den 27 6 2008). *PIC24HJ12GP201*. Hämtat från Microchip:

<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en520472> den 10 01 2009

Microchip produktfamiljer. (den 11 01 2009). Hämtat från

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2551

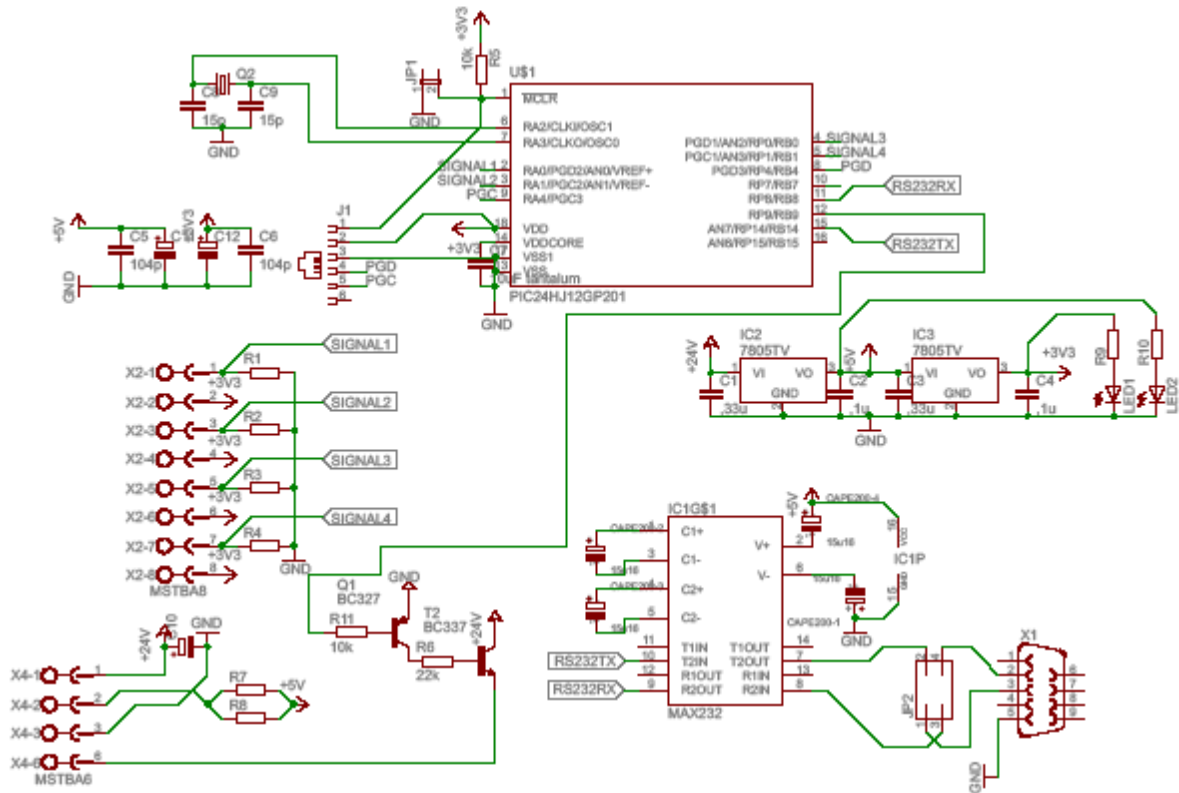
Minns, P. (den 14 juni 2007). *Courses at the University of Bolton*. Hämtat från University of Bolton:

http://www.ami.ac.uk/courses/ami4655_micros/ den 1 September 2008

Olsson, G., & Rosen, C. (2005). *Industrial automation, aplcations ,structures and systems*. Lund: Department of Industrial Engineering and Automation.

Ulrich, K., & Eppinger, S. (2004). *Product Design and Development*. Boston: McGraw-Hill.

Appendix A. Schema



Appendix B. Kod

Counter_Median.c

```
#include <p24hxxxx.h>
#include "Interrupt.h"
#include "settings.h"
#include "UART.h"
#include "Init.h"

#define CALLIBRATE 0b00000001
#define COUNT 0b00000010
#define HOMOGEN 0b00000100
#define SET 0b10000000
#define MAX_INTEGER 0xffff

char UARTBufferEmpty = 1;
char countFlag1 = 0;
char countFlag2 = 0;
char nyttAD = 0;
char negSwitch = 0;

unsigned int AD1 = 0;
unsigned int AD2 = 0;
unsigned int value1 = 0;
unsigned int value2 = 0;
unsigned int homogen1 = 0;
unsigned int homogen2 = 0;
unsigned int trigValue1 = 16;
unsigned int trigValue2 = 16;
unsigned int countSample1 = 0;
unsigned int countSample2 = 0;
unsigned long over1 = 0;
unsigned long over2 = 0;
unsigned long under1 = 0;
unsigned long under2 = 0;
unsigned int temp1 = 0;
unsigned int temp2 = 0;

void __attribute__((__interrupt__)) _T1Interrupt(void);
void __attribute__((__interrupt__)) _ADC1Interrupt(void);
void __attribute__((__interrupt__)) _U1RXInterrupt(void);

/*
** Transmit funktion som sänder ifall det finns något att sända i bufferten
*/
void transmit() {
    // Om bufferten kan bli tom, sätt flagga
    if(UARTBufferHasNext() < 5)
    {
        UARTBufferEmpty = 1;
    }

    // Fyll UART buffert sålänge tecken finns att sända
    // och sålänge UART buffert INTE är full
    while((UARTBufferHasNext()) && (!U1STAbits.UTXBF))
    {
        U1TXREG = popUARTBuffer();
    }

    //Nolla UART Interrupt Flag
```

```

        IFS0bits.U1TXIF = 0;
    }

void main()
{
    int i;

    // Initsiera inställningarna för klockhastigheten och PLL
    OSCInit();

    // Initsiera alla I/O portar till µC
    IOPortInit();

    // Initsiera de självständiga timer modulerna
    timerInit();

    // Initsiera alla parametrar för RS232 länken
    UARTInit();

    // Initsiera alla parametrar för A/D-Convertern
    ADCInit();

    // Initsiera alla avbrott som används i µC
    interruptInit();

    // Stäng av interrupts
    disableInterrupts();

    // Samla initsiala AD-värden
    for(i = 0; i < 8; i++)
    {
        AD1CON1bits.SAMP = 1;
        IFS0bits.AD1IF = 0;
        while(!IFS0bits.AD1IF);
        AD1 = ADC1BUF0;
        AD2 = ADC1BUF1;
        homogen1 = homogen1 + AD1;
        homogen2 = homogen2 + AD2;
    }

    // Sätt igång interrupts
    enableInterrupts();

    if(negSwitch)
    {
        homogen1 = 1024 - (homogen1>>3);
        homogen2 = 1024 - (homogen2>>3);
    }
    else
    {
        homogen1 = homogen1>>3;
        homogen2 = homogen2>>3;
    }

    // Läs in nya värden på AD1
    AD1CON1bits.SAMP = 1;

    while(1)
    {

        // Funktion som kollar uppdaterar homogenvärden efter förändringar

```



```

if(nyttAD){
    //nytt
    disableInterrupts();
    if(value1>0) over1++;
    else under1 ++;
    if(value2>0) over2++;
    else under2 ++;
    if((long)over1 + under1 >200000)
    {
        if(over1 > 10000)
        {
            homogen1 = homogen1 + 2;
            if(trigValue1 > 10)
            {
                trigValue1 = trigValue1 - 2;
            }
        }
        else if(over1 < 10000)
        {
            homogen1 --;
            trigValue1 ++;
        }
        over1 = 0;
        under1 = 0;
        if(over2 > 10000)
        {
            homogen2 = homogen2 + 2;
            if(trigValue2 > 10)
            {
                trigValue2 = trigValue2 - 2;
            }
        }
        else if(over2 < 10000)
        {
            homogen2 --;
            trigValue2 ++;
        }
        over2 = 0;
        under2 = 0;
    }

    nyttAD = 0;
    enableInterrupts();
}

if(IFS0bits.U1TXIF)
{
    transmit();
    IFS0bits.U1TXIF = 0;
}

/*
** Kontrollera om det finns någonting att sända och
** ifall UART TX är i idle
**/
if(UARTBufferEmpty && UARTBufferHasNext())
{
    IFS0bits.U1TXIF = 1;
}
}
}

```

```

/*
** Interruptrutin för Timer 1
*/
void __attribute__((__interrupt__)) _T1Interrupt(void)
{
    LATBbits.LATB9 = 0;
    T1CONbits.TON = 0;
    TMR1 = 0;
    IFS0bits.T1IF = 0;           // Clear Timer1 interrupt flag
}

/*
** Kolla ifall multiply nivån ligger i nivå för en skarv.
** Om den gör det, sätt countflaggan hög samt räkna en påse.
** Notera hysteresen i while villkoret.
*/
void __attribute__((__interrupt__)) _ADC1Interrupt(void)
{
    AD1 = ADC1BUF0;
    AD2 = ADC1BUF1;

    if(negSwitch)
    {
        AD1 = 1024 - AD1;
        AD2 = 1024 - AD2;
    }

    // Läs in nya värden på AD1
    AD1CON1bits.SAMP = 1;
    if (AD1 > homogen1) value1 = AD1 - homogen1;
    else value1 = 0;
    if (AD2 > homogen2) value2 = AD2 - homogen2;
    else value2 = 0;

    // Kolla efter toppar till AD1
    if( value1 > (trigValue1 - (trigValue1>>1)*countFlag1) )
    {

        // Räkna antal sample för topp
        countSample1++;

        if(!countFlag1)
        {
            countFlag1 = 1;
        }

        // Justera homogena nivå efter toppar
        if(value1 > temp1) temp1 = value1;
    }
    else
    {
        if(!countSample1)
        {
            countFlag1 = 0;
            temp1 = temp1>>2;
            // Justera homogena nivå efter toppar
            if( temp1 > trigValue1) trigValue1 = temp1;
            else if (temp1 * 3 < trigValue1 && temp1 > 5)
                trigValue1 = temp1 * 3;
            temp1=0;
        }
        else countSample1--;
    }
}

```

```

}

// Kolla efter toppar till AD2
if( value2 > (trigValue2 - (trigValue2>>1)*countFlag2) )
{
    int temp = value2>>2;

    // Räkna antal sample för topp
    countSample2++;

    if(!countFlag2)
    {
        countFlag2 = 1;
    }

    // Justera homogena nivån efter toppar
    if(value2 > temp2) temp2 = value2;
}
else
{
    if(!countSample2)
    {
        countFlag2 = 0;
        temp2 = temp2>>2;
        if( temp2 > trigValue2) trigValue2 = temp2;
        else if (temp2 * 3 < trigValue2 && temp2 > 5)
        {
            trigValue2 = temp2 * 3;
        }
    }
    else countSample2--;
}

if(countFlag1 && countFlag2) {
    if(!LATBbits.LATB9)
    {
        LATBbits.LATB9 = 1;
        T1CONbits.TON = 1;
    }
}
nyttAD =1; //nytt
IFS0bits.AD1IF = 0;
}

/*
** Interruptrutin för UART Recieve
*/
void __attribute__((__interrupt__)) _U1RXInterrupt(void)
{
    // Läs in char
    char receivedChar = U1RXREG;

    switch(receivedChar) {

        // Sänd homogenvärde för varje sensor
        case 'h':    UARTSerialSend('h');
                    UARTSerialSend('1');
                    UARTSerialSend(0x9);
                    UARTSerialSend('h');
                    UARTSerialSend('2');
                    UARTSerialSendEnter();

```

```

        UARTSerialSendInt (homogen1);
        UARTSerialSend (0x9);
        UARTSerialSendInt (homogen2);
        UARTSerialSendEnter ();
        break;

// Sänd AD värde för varje sensor
case 'a':    UARTSerialSend ('A');
             UARTSerialSend ('D');
             UARTSerialSend ('1');
             UARTSerialSend (0x9);
             UARTSerialSend ('A');
             UARTSerialSend ('D');
             UARTSerialSend ('2');
             UARTSerialSendEnter ();
             UARTSerialSendInt (AD1);
             UARTSerialSend (0x9);
             UARTSerialSendInt (AD2);
             UARTSerialSendEnter ();
             break;

// Sänd korrigerat värde för varje sensor
case 'v':    UARTSerialSend ('V');
             UARTSerialSend ('1');
             UARTSerialSend (0x9);
             UARTSerialSend ('V');
             UARTSerialSend ('2');
             UARTSerialSendEnter ();
             UARTSerialSendInt (value1);
             UARTSerialSend (0x9);
             UARTSerialSendInt (value2);
             UARTSerialSendEnter ();
             break;

// Sänd trigg värde för varje sensor
case 't':    UARTSerialSend ('T');
             UARTSerialSend ('V');
             UARTSerialSend ('1');
             UARTSerialSend (0x9);
             UARTSerialSend ('T');
             UARTSerialSend ('V');
             UARTSerialSend ('2');
             UARTSerialSendEnter ();
             UARTSerialSendInt (trigValue1);
             UARTSerialSend (0x9);
             UARTSerialSendInt (trigValue2);
             UARTSerialSendEnter ();
             break;
    }

IFS0bits.U1RXIF = 0;
}

```

Init.c

```
#include <p24hxxxx.h>
#include "settings.h"
#define BRGVAL ((FCY/BAUDRATE)/16)-1
#define ADCSVAL 4

void ADCInit()
{
    AD1CON1bits.AD12B = 0; // Set 10-bit conversion
    AD1CON2bits.CHPS = 1; // DUAL-Channel conversion
    AD1CON2bits.VCFG = 0; // Set voltage reference to AVdd and AVss
    AD1CON1bits.FORM = 0; // Data is presented as an Unsigned Integer
    AD1CON1bits.SIMSAM = 0; // Sequential sampling
    AD1CON2bits.SMPI = 1; // Generate interrupt after two conversion
    AD1CON1bits.ASAM = 0; // Manual sample
    AD1CON1bits.SSRC = 7; // Automatic conversion
    AD1CON3bits.SAMC = 2; // Auto sample time 2xTAD
    AD1CON3bits.ADCS = ADCSVAL; // Set tad
    AD1CON3bits.ADRC = 0; // AD clock derived from system clock
    AD1CHS0bits.CH0SA = 1; // Select AN1 as positive input
    AD1CHS0bits.CH0NA = 0; // Select Vref- as negative input
    AD1CHS123bits.CH123SA = 0; // Select AN0, AN1, AN2 input for CH1,CH2, CH3
    AD1CHS123bits.CH123NA = 0; // Select Vref- as negative input
    AD1CON1bits.ADON = 1; // Activate ADC module
}

void interruptInit()
{
    INTCON1bits.NSTDIS = 0; // Enable nested interrupts
    IFS0bits.U1RXIF = 0; // Reset UART Receive interrupt flag
    IEC0bits.U1RXIE = 1; // Enable UART Receive interrupt
    IFS0bits.U1TXIF = 0; // Reset UART Transmit interrupt flag
    IEC0bits.U1TXIE = 1; // Enable UART Transmit interrupt
    IFS0bits.T1IF = 0; // Reset TIMER1 interrupt flag
    IEC0bits.T1IE = 1; // Enable TIMER1 interrupt
    IFS0bits.AD1IF = 0; // Reset AD1 interrupt flag
    IEC0bits.AD1IE = 1; // Enable AD1 interrupt
}

void UARTInit()
{
    U1MODEbits.STSEL = 0; // 1-stop bit
    U1MODEbits.PDSEL = 0; // No parity, 8-data bits
    U1MODEbits.ABAUD = 0; // Auto-baud disabled
    U1MODEbits.BRGH = 0; // Low speed mode

    U1BRG = 21; // Set BAUD rate
    U1STAbits.URXISEL = 0; // Interrupt after one RX char
    U1STAbits.UTXISEL1 = 1; // Interrupt when buffer is
empty

    U1MODEbits.UARTEN = 1; // Enable Uart
    U1STAbits.UTXEN = 1; // Enable transmit
}

void IOPortInit()
{
    RPINR18bits.U1RXR = 8; // Set UART Recieve "U1RX" to RP8
    RPOR7bits.RP14R = 3; // Set UART Transmit "U1TX" to RP14
    TRISA |= 3; // Set port RA0 and RA1 to input
    TRISB |= 3; // Set port RB0 and RB1 to input
}
```

```

        TRISB &= 0b1111110111111111;    // Set port RB9 to output
    }

    /* Init the timer to interrupt once ever second */
    void timerInit()
    {
        T1CONbits.TON = 0;                // Disable Timer
        T1CONbits.TCS = 0;                // Select Fcy
        T1CONbits.TGATE = 0;              // Disable gated timer mode
        T1CONbits.TCKPS = 0b11;           // Prescaler 1:256
        TMR1 = 0x00;                       // Reset Timer
        PR1 = 31250;                        // overflow after 15625 T1clk
    }

    /* Configure Oscillator to operate the device at 40 MIPS
       Fosc = Fin*M/(N1*N2), Fcy = Fosc/2
       Fosc = 8M*40(2*2) = 80 Mhz for 8Mhz input
    */
    void OSCInit()
    {
        PLLFBD = 38;                       // M = 40Mhz
        CLKDIVbits.PLLPOST = 0;            // N1 = 2
        CLKDIVbits.PLLPRE = 0;            // N2 = 2
        while(!OSCCONbits.LOCK);          // Wait for PLL to lock
    }

```

Interrupt.c

```

#include <p24Hxxxx.h>
#include "UART.h"

void __attribute__((__interrupt__)) _U1TXInterrupt(void)
{
    char ticker = 0;
    while(UARTBufferHasNext() && ticker < 4)
    {
        U1TXREG = popUARTBuffer();
        ticker++;
    }
    IFS0bits.U1TXIF = 0;    // Clear UART transmit flag
}

void enableInterrupts()
{
    /* Set CPU IPL to 0, enable level 1-7 interrupts */
    /* No restoring of previous CPU IPL state performed here */
    SRbits.IPL = 0;
}

void disableInterrupts()
{
    /* Set CPU IPL to 7, disable level 1-7 interrupts */
    /* No saving of current CPU IPL setting performed here */
    SRbits.IPL = 7;
}

```

UART.c

```

#include <p24hxxxx.h>
#include "Interrupt.h"

```

```

unsigned char TXBuffer[60];
char counterLow = 0;
char counterHigh = 0;
char numElement = 0;
char bufferEmptyFlag = 1;

void setBufferEmptyFlag(char a)
{
    bufferEmptyFlag = a;
}

char checkBufferEmptyFlag()
{
    return bufferEmptyFlag;
}

char UARTBufferHasNext()
{
    return numElement;
}

unsigned char popUARTBuffer()
{
    unsigned char temp = 0;
    // Check if the buffer is empty
    if(numElement > 0)
    {
        temp = TXBuffer[counterLow];
        if(counterLow < 59)
        {
            counterLow++;
        }
        else
        {
            counterLow = 0;
        }
        numElement--;
    }
    return temp;
}

void UARTSerialSend(unsigned char c)
{
    if(numElement < 60)
    {
        TXBuffer[counterHigh] = c;
        if(counterHigh < 59)
        {
            counterHigh++;
        }
        else
        {
            counterHigh = 0;
        }
        numElement++;
    }
}

void UARTSerialSendEnter()
{
    UARTSerialSend(0x0D);
    UARTSerialSend(0x0A);
}

```

```

}

void UARTSerialReceive(char *p)
{
    if(U1STAbits.FERR == 1)
    {
        UARTSerialSend('f');
    }

    if(U1STAbits.OERR == 1)
    {
        U1STAbits.OERR = 0;
        UARTSerialSend('o');
    }

    if(U1STAbits.URXDA == 1)
    {
        *p = U1RXREG;
    }
}

void UARTSerialSendInt(unsigned int d)
{
    UARTSerialSend(0x30 | d/10000);
    d = d%10000;
    UARTSerialSend(0x30 | d/1000);
    d = d%1000;
    UARTSerialSend(0x30 | d/100);
    d = d%100;
    UARTSerialSend(0x30 | d/10);
    d = d%10;
    UARTSerialSend(0x30 | d);
}

```

UART.h

```

void UARTSerialSend(unsigned char);
void UARTSerialSendInt(unsigned int);
void UARTSerialSendEnter();
void UARTSerialReceive(char *);
char UARTbufferHasNext();
void setBufferEmptyFlag(char);
char checkBufferEmptyFlag();

```


Init.h

```
void IOPortInit();
void UARTInit();
void timerInit();
void interruptInit();
void ADCInit();
```

Interrupt.h

```
void __attribute__((__interrupt__, __shadow__)) _T1Interrupt(void);
void enableInterrupts(void);
void disableInterrupts(void);
```

Settings.h

```
/* USER SET PARAMETERS */
#define FOC      80000000           // The clock speed in Mhz
#define BAUDRATE 115200           // Serial RS232 speed in kBit/s
#define TAD      100              // Desired AD conversion clock in ns

/* CALCULATED PARAMETERS */
#define FCY      FOC/2             // The processor speed in Mhz
#define TCY      (1000000000)/FCY // Length of one instruction in ns
```