

Automatisering av testsekvens för HLSC, Haldex Limited Slip Coupling



Oskar Lagerås
Anton Sjunnesson

Dept. of Industrial Electrical Engineering and Automation
Lund University

Förord

Vi vill rikta ett stort tack till Haldex Traction AB, och alla anställda som har tagit sig tid och ork att hjälpa oss. Förhoppningsvis har vi lyckats ge något i utbyte.

Särskilda tack riktas även till våra handledare, Jonas Jönsson och Nikolaos Ligouras på Haldex, samt Gunnar Lindstedt och Bengt Simonsson på IEA, LTH, för att ni engagerat er och uppmuntrat oss.

Oskar Lagerås

Anton Sjunnesson

Lund, den 4 februari 2008

Abstract

The degree project has been carried out at the department of Industrial Electrical Engineering and Automation, at Lund University, in cooperation with Haldex Traction AB.

Haldex manufactures four wheel drive couplings, HLSCⁱ, for the automotive industry. To make sure the couplings meets the high requirements and expectations from the customer, standard tests are constantly being carried out.

This project aims at automating some of these test procedures. Thereby reducing the workload, and reduce the influence of the human factor. A test rig, on site at Haldex in Landskrona, will be used. The rig has been inactive for almost a year, due to a cancelled project.

Focus on this thesis will be on the software development. To have a firm understanding of the problem, the hardware configuration will also be described. Further, the work process and some of the problems we encountered will be discussed.

The goal of the project has been achieved and a functional machine is in place at Haldex. Before the rig can be used to it's full extent, some small changes and additions has to be made, e.g. more coupling models has to be implemented.

Apart from the work described above, a user's guide on how to operate the machine has been produced.

Keywords: Wet clutch, LabView, four wheel drive, test, software

ⁱ Haldex Limited Slip Coupling

Innehållsförteckning

Förord.....	i
Abstract.....	ii
Innehållsförteckning	iii
Bildförteckning	v
1 Inledning	1
1.1 Bakgrund.....	1
1.2 Syfte	2
1.3 Målsättning	3
1.4 Metod	4
1.5 Avgränsningar.....	4
1.6 Resultat	5
2 Riggkonstruktion.....	6
2.1 Mekanisk konstruktion.....	6
2.1.1 Övergripande.....	6
2.1.2 Temperaturhuv	8
2.2 Hårdvara.....	14
2.2.1 Sensorer.....	14
2.2.2 Databasinsamling	15
2.2.3 CAN-Case	18
2.2.4 Temperaturskåp WT-180/70 -UTA-300	19
3 Mjukvaruutveckling.....	20
3.1 Programmeringsmiljö	20
3.1.1 LabView.....	20
3.2 Programstruktur	22
3.2.1 Programöversikt.....	24
3.3 Styrning av CAN-bussen	37
3.4 Rapportgenerering.....	39

4 Problem och svårigheter	42
4.1 Temperaturreglering	43
5. Förbättringar och fortsatt arbete.....	45
6. Resultat	48
7. Referenser och litteraturlista	50
Bilaga A	51
Instruktionsmanual.....	51

Bildförteckning

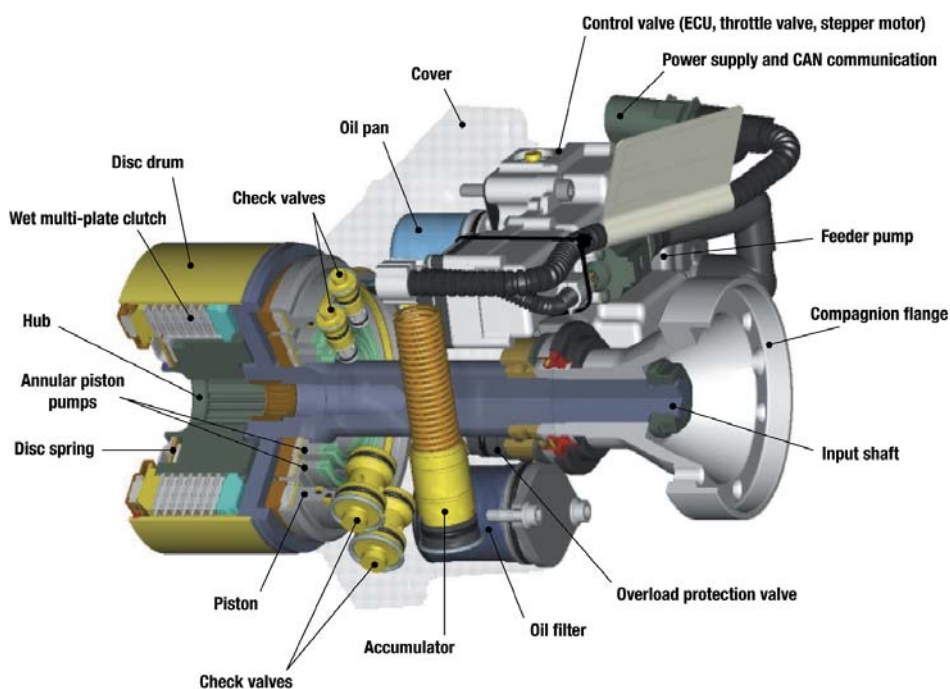
Figur 1, Haldex Limited Slip Coupling	1
Figur 2, Hela Rigguppställningen	6
Figur 3, Temperaturhuv	8
Figur 4, Smält frigolit	9
Figur 5, Temperaturhuv med ny isolering	11
Figur 6, Temperaturgraf, ner till -20°C.....	13
Figur 7, Temperaturgraf, upp till 100°C	13
Figur 8, Kopplingslåda SC-2345 med moduler	16
Figur 9, CANcaseXL	18
Figur 10, Flödesschema för programstrukturen.....	24
Figur 11, Receptvektor med testparametrar	25
Figur 12, Flödesschema av temperaturregleringen	26
Figur 13, Temperatursteg med prediktor	28
Figur 14, Stegsvär från 50°C till 80°C i steg om 10°C.....	29
Figur 15, Loggat moment, varvtal och temperatur från en testsekvens. Att temperaturen i kopplingen inte varit homogen kan ses på temperaturkurvan.	33
Figur 16, Receptvektor med borttagna element.....	35
Figur 17, CAN-databas och vlv-fil	37
Figur 18, Momentsteg.....	40
Figur 19, Rapport med tillhörande lvm-fil.....	41
Figur 20, Bild på urtagning i temperaturhuv	45

1 Inledning

Examensarbetet har utförts på institutionen för Industriell Elektronik och Automation vid Lunds Tekniska Högskola och i samarbete med Haldex Traction AB i Landskrona. Arbetet har genomförts under hösten 2007.

1.1 Bakgrund

Haldex tillverkar kopplingar för fyrhjulsdraft till olika företag inom bilindustrin. Kopplingen består i stort av två delar, ett momentöverförande system som i grova drag innefattar ett våtlamellpaket, en kolv, en ackumulator, och en ventil. Trycket byggs upp i ackumulatorn av pumpen varefter ventilen reglerar trycket till kolven som i sin tur verkar på lamellpaketet. Det andra systemet är ett mikrodatorbaserat styrsystem, som styr ventil och pump. Detta gör att man kan få kopplingen att arbeta olika vid olika körsituationer.



Figur 1, Haldex Limited Slip Coupling

För att kunna verifiera om kopplingarna uppfyller ställda krav, från exempelvis kunden, krävs det att kopplingarna testas på olika sätt. Haldex har utvecklat olika standardprov som till exempel kan vara att testa momentkapacitet, reaktionstider etc. Dessa tester skall sedan utföras vid olika temperaturer från -30°C till 120°C.

1.2 Syfte

För tillfället tempereras kopplingen i ett temperaturskåp varefter den monteras in i riggen, och ett test körs. När testet är klart sätts kopplingen in i temperaturskåpet för nästa temperatursteg osv.

Eftersom Haldex startar nya projekt samt får nya kunder, krävs det att tiden för olika provningsförfaranden minskar, eller att provningen effektiviseras.

Syftet med examensarbetet är att koppla samman temperaturskåp, provbänk och koppling till en enhet. Till denna skall sedan ett styr/mätprogram utvecklas, så att testsekvenser kan utföras automatiskt. Fördelarna med ett sådant förfarande är många, bland annat minskat antal arbetsmoment för varje provobjekt, och repeterbarheten mellan olika prov ökar.

Riggen som skall automatiseras har legat i malpåse i ett år och används ej. Detta eftersom kundprojektet har lagts på is.

1.3 Målsättning

Målsättningen med arbetet kan delas upp i två kategorier. De mål/delmål som är uppsatta av Haldex samt de mål vi själva har med arbetet.

Haldex mål, enl. examensjobbspecifikation

- Skapa en övergripande teststruktur
- Definiera vilka programvaror som skall användas
- Utveckla kommunikation HLSC/provningsrigg/temperaturskåp
- Skapa testsekvenser
- Genomförande/verifiering av testsekvenser
- Postprocessa mätdata
- Dokumentation
- Förslag till fortsättning

Mål som är av intresse för haldex

- En färdig maskin inom tidsramen för examensarbetet
- Användarvänlig och enkel miljö för operatören
- Mjukvaran ska vara anpassningsbar/flexibel och kunna användas i många driftsfall, en enhetlig uppbyggnad underlättar detta.
- Slutligen är det viktigt att maskinen är robust och pålitlig.

1.4 Metod

För att kunna styra upp arbetet har vi delat in problemet i olika delproblem. Detta görs för att kunna utföra arbetet på ett smidigt sätt, samt det faktum att vissa delar måste vara klara innan man kan gå vidare till nästa problem.

Troligtvis kommer vissa moment att innebära en del väntan, vilket medför att vissa uppgifter får lösas parallellt.

I inledningen av projektet kunde vi identifiera dessa delproblem:

- Modifiering och färdigställning av riggen mekaniskt
- Skapa kommunikation mellan dator och testbänk, exempelvis samla in data och styra motorer.
- Skapa kommunikation mellan temperaturskåp och dator
- Reglera temperaturen i temperaturhuvu med datorn
- Bygga upp en teststruktur
- Skapa kommunikation mellan HLSC och dator
- Ta fram information om de olika testförfarandena

1.5 Avgränsningar

Riggen kommer endast att anpassas för vissa specifika standardtest. Antalet kopplingstyper som kommer att implementeras blir också begränsat, då det inte är rimligt att utveckla stöd för hela produktserien. Examensarbetet innefattar inte heller att utveckla/ta fram ny hårdvara till riggen.

Då projektet är komplext och består av många delar, kommer tidsaspekten att vara en viktig faktor. Fokus i projektet ligger på att få en fungerande rigg, inte att optimera alla beståndsdelar.

1.6 Resultat

Huvudsyftet med examensarbetet uppnåddes, och en maskin som kan utföra automatiserade testcykler finns på plats på Haldex i Landskrona. Arbetet förlöpte väl utan några större incidenter, och tidsmässigt höll det sig inom ramarna.

Samtliga delmål uppsatta från Haldex uppfylldes. Våra egna delmål är inte lika lätta att göra en objektiv bedömning på. Trots det tycker vi att de uppfyllts till största del.

Maskinen och teststrukturen har ej tagits i bruk än. Troligtvis kommer vissa justeringar att vara nödvändiga.

Utöver jämförande tester och justering av till exempel PID-parametrar, kalibrering etc., bör stöd för ytterligare kopplingar implementeras (främst i form av CAN-kommunikation). En förstudie på detta område har utförts, och med smärre justeringar i huvudprogrammet skall det inte utgöra något större problem. Det är främst en fråga om att veta hur kommunikationen för respektive koppling fungerar.

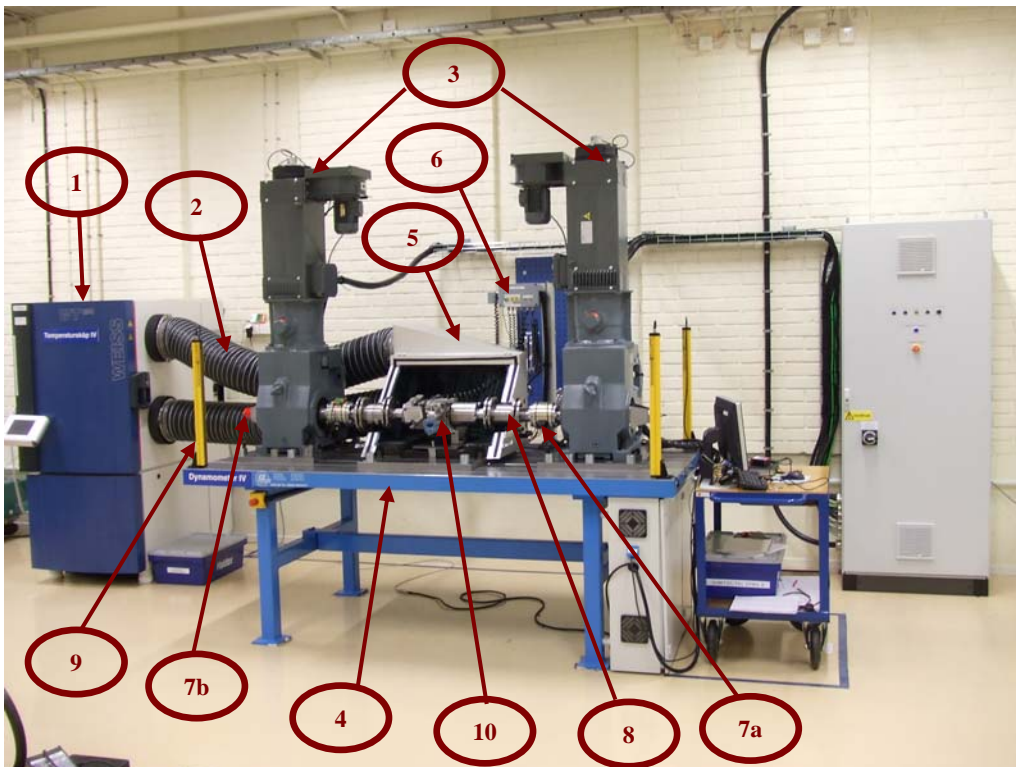
Slutsatsen är att en fungerande maskin finns, men att visst fortsatt arbete bör utföras för att den skall kunna tas i fullt bruk.

2 Riggkonstruktion

Trots att modifiering/utveckling av riggens hårdvara inte ingick i examensarbetet, ges här en kort översikt av denna. Detta för att de yttre fysiska förutsättningarna för examensarbetet ska klargöras.

2.1 Mekanisk konstruktion

2.1.1 Övergripande



Figur 2, Hela Rigguppställningen

- | | |
|--|--------------------------|
| 1) Temperaturskåp | 7) Givare |
| 2) Isolerade cirkulationsslanger | a) Momentgivare |
| 3) Drivpaket (fläkt, broms, motor, växel) | b) Pulsgivare |
| 4) Provbänk | 8) Torsionsstyv koppling |
| 5) Isolerad kåpa | 9) Ljusridå |
| 6) Givarbox, med ingång för temperatur och tryckgivare | 10) Testobjekt, HLSC |

På testbänken är två drivpaket (motor, växellåda, fläkt etc.) monterade. Dessa står på vars en fästplåt, vilken löper i T-spår. Fästplåten ligger på ett par kulrullar för att underlätta förflyttning i sidled. Mellan de två drivpaketen finns en fästplatta för Haldex-kopplingen. Även denna platta löper i T-spår och är utrustad med kulrullar.

I slutet av varje utgående axel från drivpaketen sitter en torsionsstyv koppling som kompenserar för vinkelfel och förskjutning av provobjektet i förhållande till axlarna. Denna torsionsstyva koppling har även den fördelen att den termiskt isolerar ingående och utgående axel (på så sätt behöver man inte oroa sig för att momentgivare och annan utrustning utsätts för kondens och andra ogynnsamma förhållanden). Axlarna mot Haldex-kopplingen är utrustade med kilspår så att de kan löpa i axiell led, vilket underlättar montering.

Uppe på testbänken har en isolerhuv monterats. Inne i denna huv monterar sedan provobjektet. Luften konditioneras med cirkulerande luft från temperaturskåpet. Utöver dessa delar utgörs provbänken av tillhörande motorstyrning, skyddsanordningar i form av ljusridåer och nödstopp, styrdator, IO-enheter och diverse givare.

2.1.2 Temperaturhuv

För att kunna kyla/värma kopplingen på plats i riggen har Haldex specialbeställt en temperaturhuv. Temperaturhuvens är uppbyggd av ett aluminiumskal och är isolerad med 50 mm tjock frigolit. Frigoliten är sedan täckt med epoxi för att göra den tålig mot slag och stötar. Huvens är utrustad med en lucka, kabelgenomföringar samt luftslanganslutningar och hål för axlarna.



Figur 3, Temperaturhuv

För att se vilka begränsningar temperaturhuvens har samt för att se hur lång tid det tar att kyla respektive värma kopplingen, gjordes några tester. Kopplingen utrustades med en temperaturgivare vid lamellpaketet, vidare placerades en temperaturgivare löst i huvens för att kunna mäta temperaturen i luften runt kopplingen.

Kyltestet började vid +20°C. När luften i temperaturhuvuven passerar -23°C kan man urskilja frost på axlarna som går in i huvuven, samt på vissa dåligt isolerade ytor. Testet avbröts efter tre timmar och femtio minuter då temperaturen i kopplingen var -30°C. Det vi kan se av det här testet är att det är möjligt att kyla kopplingen till den lägst förekommande testtemperaturen (se Figur 6).

Värmetestet började vid +30°C. Efter ca 50 minuter har temperaturen inne i kopplingen ökat till 50°C och temperaturen i lådan är då 80°C. När temperaturen i lådan når 110°C börjar det ryka från temperaturhuvuven och det hörs smällar när frigoliten lossnar från väggarna på huvuven. Vi hade fått intrycket av att man skulle kunna värma huvuven, detta visade sig vara felaktigt (se Figur 7).



Figur 4, Smält frigolit

De flesta cellplatser har en smältpunkt runt 80°C, så i efterhand var det inte speciellt genomtänkt att försöka värma kylhuvuven.

För att välja ett nytt isoleringsmaterial som skulle kunna fungera under rådande förhållanden, togs det fram en materialspecifikation på de egenskaper som materialet skulle ha.

- Ett temperaturspann från -55°C till 150°C
- Fukt och vattenavvisande
- Stöttåligt
- Partiklar får inte lossna från ytan
- Det får inte väga för mycket, huven ska vara hanterbar

Vi undersökte flera olika leverantörers utbud av isoleringsmaterial, nedan följer ett urval av möjliga alternativ.

Namn	PAROC Fire Slab 100	PAROC Fire Slab 100 AluCoat	Isover Fasadskiva 31	Foamglas S3 900/45
Material	Stenull	Stenull aluminiumbeklädd	Glasull	Skummat glas och kol
Värmeledning λ -värde (mW/m°C)	38 ⁱ	38 ⁱ	33	45
Densitet (kg/m ³)	100	100	---	135
Användnings- temperatur (°C)	max	700	80	260
	min	---	---	---
Stöttåligt	Ja	Nej	Ja	Nej
Vattenavvisande	Nej	Ja ⁱⁱ	Nej	Ja
Dammfritt	Nej	Ja	Nej	Ja

Tabell 1, utvärdering av isoleringsmaterial

Efter noga avvägning valde vi foamglas som isoleringsmaterial, främst för att det var det enda materialet som klarade både höga temperaturer och inte var känsligt för fukt eller vatten. Nackdelarna med materialet är det att det inte är stöttåligt. Stenullen och glasullen hade varit ett bättre alternativ om man ser till vikt och värmeledning.

ⁱ Vid 50°C

ⁱⁱ Vattenavvisande ytskikt

Isoleringsmaterialet kom i skivor om 1200x600x60 mm. Mallar i wellplast skapades utifrån temperaturhuvens mått, dessa användes sedan för att såga ut rätt konturer av foamglas. De utsågade bitarna sattes sedan fast med ett lufthärdande konstruktionslim. Detta lim är bra på det sättet att det fäster på alla typer av ytor och har en temperaturbeständighet som ligger mellan -40°C till 100°C . Utsatta skarvar tätades med silikontätning som klarar höga temperaturer. För att säkerställa att foamglasets ytskikt, vilket består av papp som är pålimmat med tjära, tål höga temperaturer, placerades en liten provbit i en ugn i 140°C . Ingenting hände förutom att tjäran som håller fast pappen blev lite mjukt, pappen satt dock kvar på foamglas.



Figur 5, Temperaturhuv med ny isolering

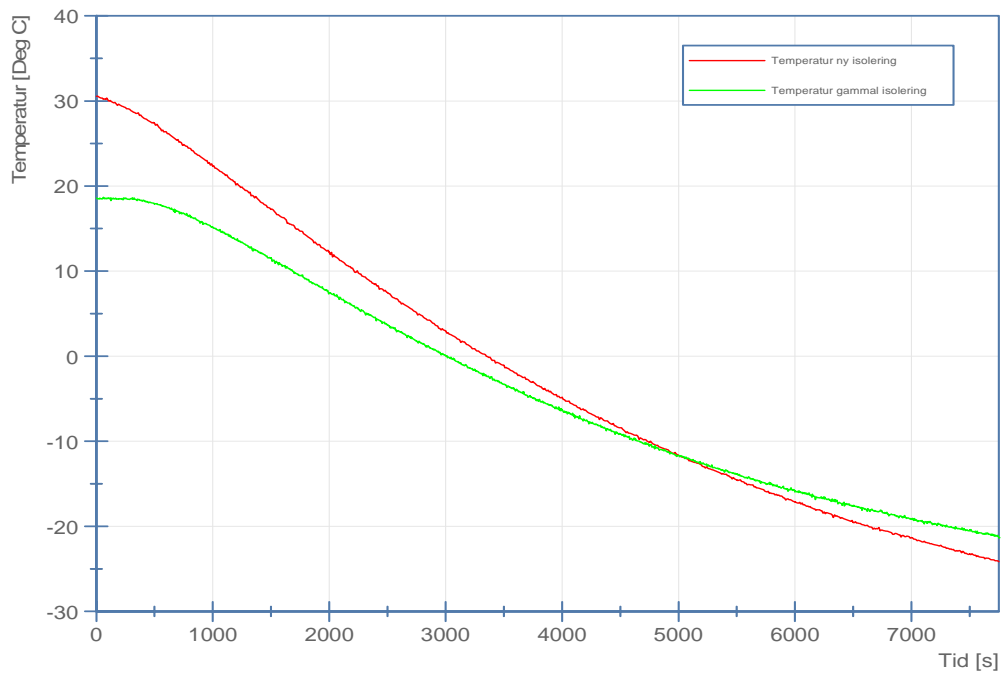
Nya temperaturtester gjordes för den omisolerade temperaturhuven.

Kyltestet började vid +30°C i kopplingen. Efter 2 timmar och 30 minuter avbröts testet. Då hade kopplingens temperatur sjunkit ner till -27°C, detta kan jämföras med den gamla isoleringen då kopplingen var nere på -30°C efter 3 timmar och 50 minuter (se Figur 6).

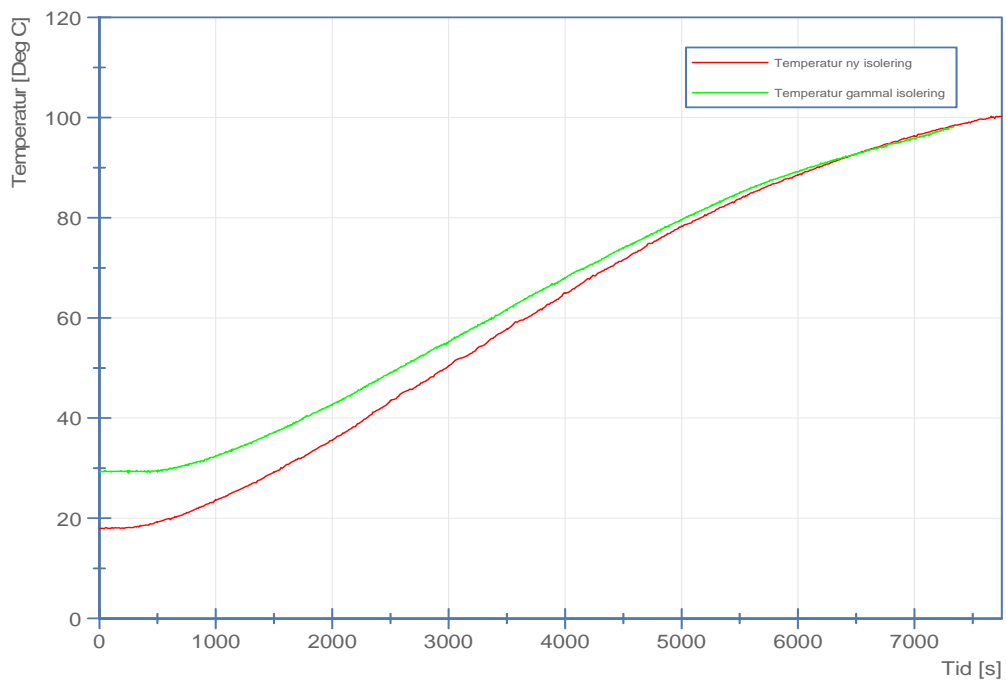
Värmetestet började vid 20°C och avslutades när temperaturen i kopplingen hade kommit upp till 110°C. Då hade temperaturen i huven varit 120°C under en längre tid. Det tog 2 timmar och 20 minuter för kopplingen att nå den önskade temperaturen (se Figur 7).

Testerna visar att temperaturhuven klarar de krav som Haldex ställer vid test av sina kopplingar.

Figur 6, Temperaturgraf, ner till -20°C



Figur 7, Temperaturgraf, upp till 100°C



2.2 Hårdvara

Nedan presenteras de olika komponenter som hade ett mer direkt inflytande på vårt arbete.

2.2.1 Sensorer

Momentgivare

De utgående axlarna från drivpaketen är utrustade med vars en momentgivare. Momentet mäts med trådtöjningsgivare monterade i fullbrygga.

Då axlarna roterar under drift, krävs ett telemetrisystem för att kunna läsa av momentet. Det som använd på denna rigg kommer från företaget Microstrain, och utgörs av en radiosändare monterad på axeln, samt en mottagare med usb-utgång.

Temperaturgivare

För att kunna mäta temperaturen på olika punkter i testobjektet är givarboxen försedd med 4 stycken ingångar för temperaturgivare.

Temperaturgivarna kommer att vara termoelement av typen K (nickelkrom-nickelaluminium).

Pulsgivare

Drivpaketens utgående axlar är även försedda med pulsgivare, för att möjliggöra mätning av ingreppsvinklar etc. Upplösningen är 1024 alt. 2048 pulser per varv.

För tillfället är dessa givare inte inkopplade, då riggen inte anses vara vridstyv nog för att ge pålitliga resultat (detta har vi dock ej kunnat kontrollera).

Tryckgivare

Givarboxen är utrustad med 4 stycken ingångar för tryckgivare.

2.2.2 Datainsamling

För att ta hand om in och ut signaler krävs det någon form av hårdvara. Riggen är utrustad med två olika DAQ:arⁱ. Den ena DAQ:en används för att samla in data från olika givare, samt för att styra riggen. Den andra DAQ:en är tänkt att användas för att räkna pulser från pulsgivarna som är placerade på riggens utgående axlar.

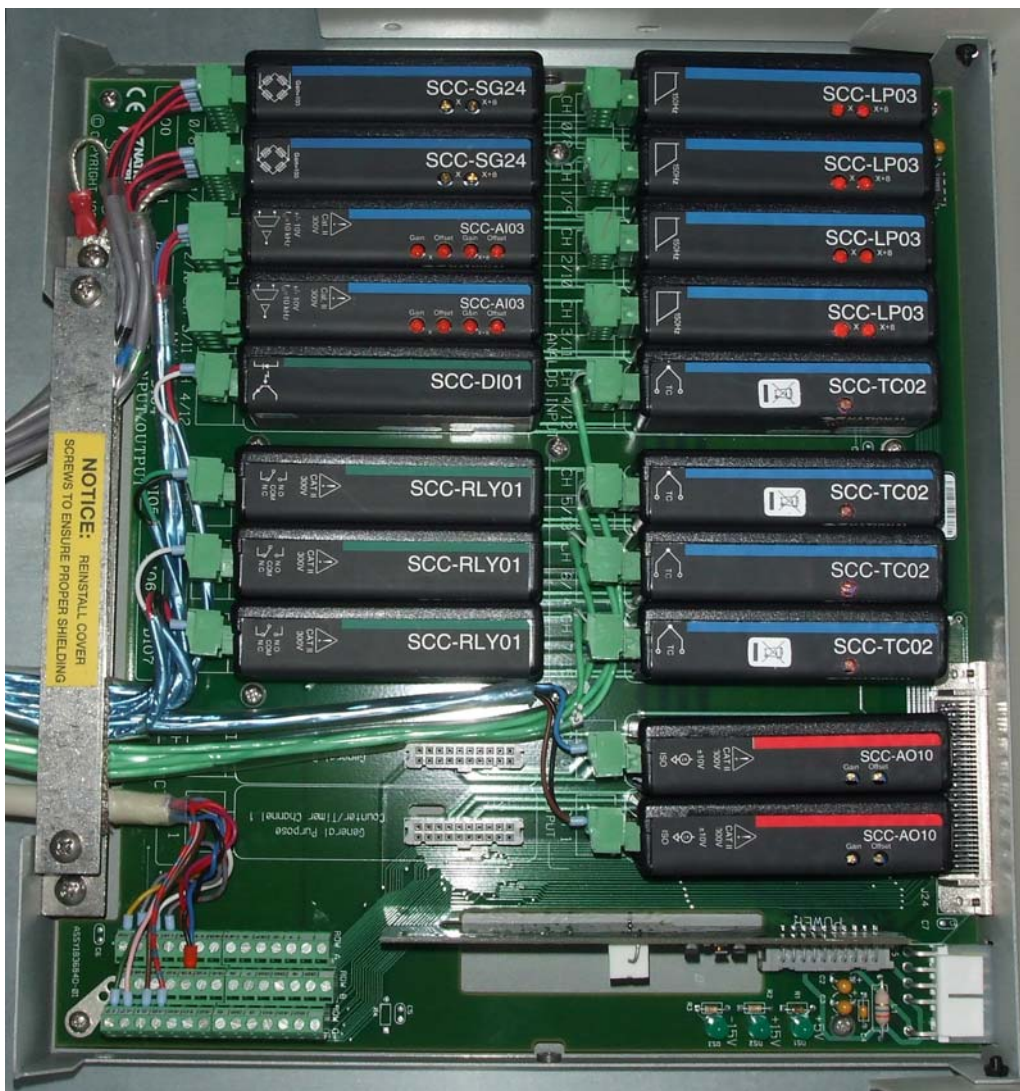
PCI-6221 – Flerfunktions DAQ

DAQ:en som används för att logga data och styra riggen är inköpt från National Instruments, och är ett PCI kort som monteras direkt i datorn. Den är försedd med 16 analoga inkanaler med 16 bitars upplösning. Maximal samplingshastighet är 250 kS/s.

DAQ:en är även försedd med 2 stycken analoga utkanaler som har ett spänningsspann på $\pm 10V$. Dessa kanaler har en upplösning på 16 bitar, samt en maximal samplingshastighet på 833 kS/s. Det finns även 24 digitala I/O kanaler samt två räknare/timers.

ⁱ DAQ, Data Acquisition

För att kunna koppla in de olika givarna på ett smidigt sätt har Haldex valt att använda sig av en SC-2345 kopplingslåda innehållande ett bakplan med plats för signalbehandlingsmoduler. Den kan utrustas med olika moduler, så som signalförstärkare och filter. Det finns även moduler för olika sorters givare, vilket gör den väldigt mångsidig. Detta underlättar även, programmeringen i Labview eftersom man då slipper att tänka på skalning och olinjäritet hos givarna. De olika modulerna som används i kopplingslådan samt deras funktion är listade nedan.



Figur 8, Kopplingslåda SC-2345 med moduler

- *SCC-AO10, utsignalsmodul, $\pm 10V$*
Dessa moduler används för att sätta varvtalet på motorstyrningarna.
- *SCC-TC02, signalbehandlingsmodul för temperaturgivare*
Används för att behandla signaler från temperaturgivare. Denna modul stödjer flera olika sorters termoelement bland annat J, K och T. Modulen har ett inbyggt lågpasfilter samt en förstärkare.
- *SCC-LP03, lågpas Butterworth-filter*
Filtrerar bort frekvenser över 150Hz. Modulen används för att filtrera de förstärkta signalerna från tryckgivare och motorstyrningsvarvtal. En modul kan filtrera två signaler samtidigt.
- *SCC-RLY01, relä*
Detta relä klarar av att isolera 5A vid 30V. Modulen används för att slå på/av bromsarna och motorstyrningen.
- *SCC-DI01, optiskt isolerad digital insignalsmodul*
Används för att läsa av motorstatussignalen. Den isolerar signaler upp till 30VDC.
- *SCC-AI03, analog insignalsmodul, $\pm 10V$*
Modulen används för att läsa av varvtalet från motorstyrningarna. Den är försedd med ett lågpasfilter och en förstärkare.
- *SCC-AG24, förstärkare*
Denna modul förstärker signalen från tryckgivarna 100 gånger.

PCI-6624 – Räknare/timer DAQ

För att kunna räkna pulser från pulsgivarna har en räknare/timer införskaffats. Eftersom pulsgivarna inte är inkopplade kommer vi inte att använda oss av denna.

2.2.3 CAN-Case

CAN-protokollet används inom bilindustrin för att kommunicera mellan de olika ECU:ernaⁱ i fordonet.

För att kommunicera med kopplingen via CAN-bussen krävs ett gränssnitt mellan dator/CAN-buss. Det hårdvarugränssnitt vi använder är en CanCaseXL från företaget Vector, vilken ansluts till datorns usb-port.



Figur 9, CANcaseXL

ⁱ Electronic Control Unit

2.2.4 Temperaturskåp WT-180/70 -UTA-300

För att temperera testobjektet används ett temperaturskåp med möjlighet för konditionering av externa testutrymmen. Temperaturskåpet och isolerhuven kopplas ihop med ca 2m långa luftcirkulationsslangar i ett slutet system. Temperaturspannet som kan uppnås i det externa testutrymmet är -55°C till $+150^{\circ}\text{C}$.ⁱ

Skåpet är försett med ett RS232-gränssnitt, vilket möjliggör styrning via dator. Även om denna standard är utvecklad för kontorsmiljö, och kan vara känslig för störningar i industrimiljö, har vi inte upplevt några sådana problem.

ⁱ Enligt temperaturskåpets manual kap 5-10

3 Mjukvaruutveckling

3.1 Programmeringsmiljö

När det gäller val av programmeringsmiljö, så var det ett önskemål från Haldex att vi skulle använda LabView. Dels för att det redan används på avdelningen, dels därför att riggen är utrustad med DAQ:ar från NI.

Tyvärr har vi ingen utbildning på LabView, men vi har använt det i ett tidigare projekt. Utifrån den erfarenheten tror även vi att LabView lämpar sig för detta projekt.

3.1.1 LabView

LabView, Laboratory Virtual Instrument Engineering Workbench, är en programmeringsmiljö och ett grafiskt programmeringsspråk utvecklat av National Instruments. Som namnet antyder utvecklades LabView ursprungligen för laboriemiljö, och är därför inriktat på insamling, bearbetning och presentation av data. Nuförtiden används dock LabView inte bara i laboratorier, utan i många andra delar av industrin.

En stor fördel med LabView är att det går relativt fort att utveckla applikationer. Då programmeringsspråket är grafiskt, behöver man inte lära sig någon textspråksyntax. LabView är även försett med en stor uppsättning bibliotek som t.ex. hanterar datainsamling, analysering, datalagring, olika sätt att presentera data etc. Eftersom LabView används frekvent inom industrin, finns det även många färdigskrivna drivrutiner till diverse instrument/utrustning, så att de lätt kan inkorporeras i det aktuella projektet.

Till skillnad från textbaserade programmeringsspråk, sker exekveringen inte sekventiellt, utan avgörs av dataflödet (dvs. hur de olika blocken är ihopkopplade rent grafiskt). Detta underlättar en del saker, som t.ex. parallell exekvering av kod, samtidigt som det blir lätt att följa strukturen i programmet. Att subfunktioner försetts med ikoner som intuitivt ger en idé om vad som utträttas i

subdiagrammen, bidrar även till en enkel grafisk representation av koden. Dock finns det vissa nackdelarna med grafiska programmeringsspråk, bl.a. kan det vara svårt att dokumentera koden på ett smidigt sätt, och är programmet väldigt stort, kan det bli oöverskådligt på skärmen.

I korthet kan ett LabView-program (kallas VI, virtual instrument), sägas bestå av två delar; ett blockdiagram, som innehåller koden i grafisk form, samt frontpanelen, vilket är det användaren ser. På samma sätt som det finns en stor uppsättning bibliotek som kan användas i blockdiagrammet, finns det en mängd olika kontroller och indikatorer för att snabbt och enkelt skapa ett användarvänligt GUI.

Då vårt projekt involverade en hel del in och utsignaler, samtidigt som tidsramen var rätt snäv, och ett enkelt och lättanvänt användargränssnitt var viktigt, lämpade sig LabView väl.

3.2 Programstruktur

Innan och under arbetet med att ta fram en lämplig programstruktur, formulerade vi vissa önskemål och principer för hur vi ville att programmet skulle fungera och se ut. Bland dessa kan enkelhet sägas vara den överordnade. Vi vill undvika en snårig och svårmanövrerad struktur, och hålla den så enkel som möjligt. Detta sker bäst genom en relativt linjär/sekventiell uppbyggnad av koden. Genom att bygga upp programmet på detta sätt, utnyttjar man också en av LabViews viktigaste principer, dataflödet. Vissa händelser kommer visserligen att bryta detta linjära mönster (t.ex. om det uppstår ett fel, eller testet avbryts), men den övergripande principen är ändå att allt ska ske i en sekvens. Man kan invända att detta sekventiella mönster gör programmet mindre flexibelt. Men som alltid måste en avvägning mellan enkelhet/robusthet och flexibilitet göras.

För att göra koden lättolkad och enkel, är det viktigt att de olika delarna i sekvensen har en tydlig funktion. De ska sköta sin uppgift och till så stor del som möjligt vara oberoende av andra kodsektioner. Fördelarna rent programmeringsmässigt är att man kan testa varje kodblock för sig, och på så sätt underlätta felsökningen.

Slutligen så slipper man på detta sätt strukturer för att sköta exekveringen, t.ex. tillstånd och övergångsvillkor. Dessa utgör väl inga problem i sig, men minskad overhead-kod gör det enklare att följa dataflödet, och minskar antalet felkällor.

Sammantaget kan man säga att det ovan nämnda syftar till att göra programmet så pass pålitligt som möjligt, vilket var ett av våra övergripande mål med projektet.

Ett annat mål var att programmet skulle vara enkelt att underhålla. Detta går lite hand i hand med principen om enkelhet. Ett lättöverskådligt program, där

dataflödet går från vänster till höger, utan inblandning av lokala och globala variabler, och där koden för varje del i sekvensen är oberoende, kommer att vara lättare att sätta sig in i och modifiera.

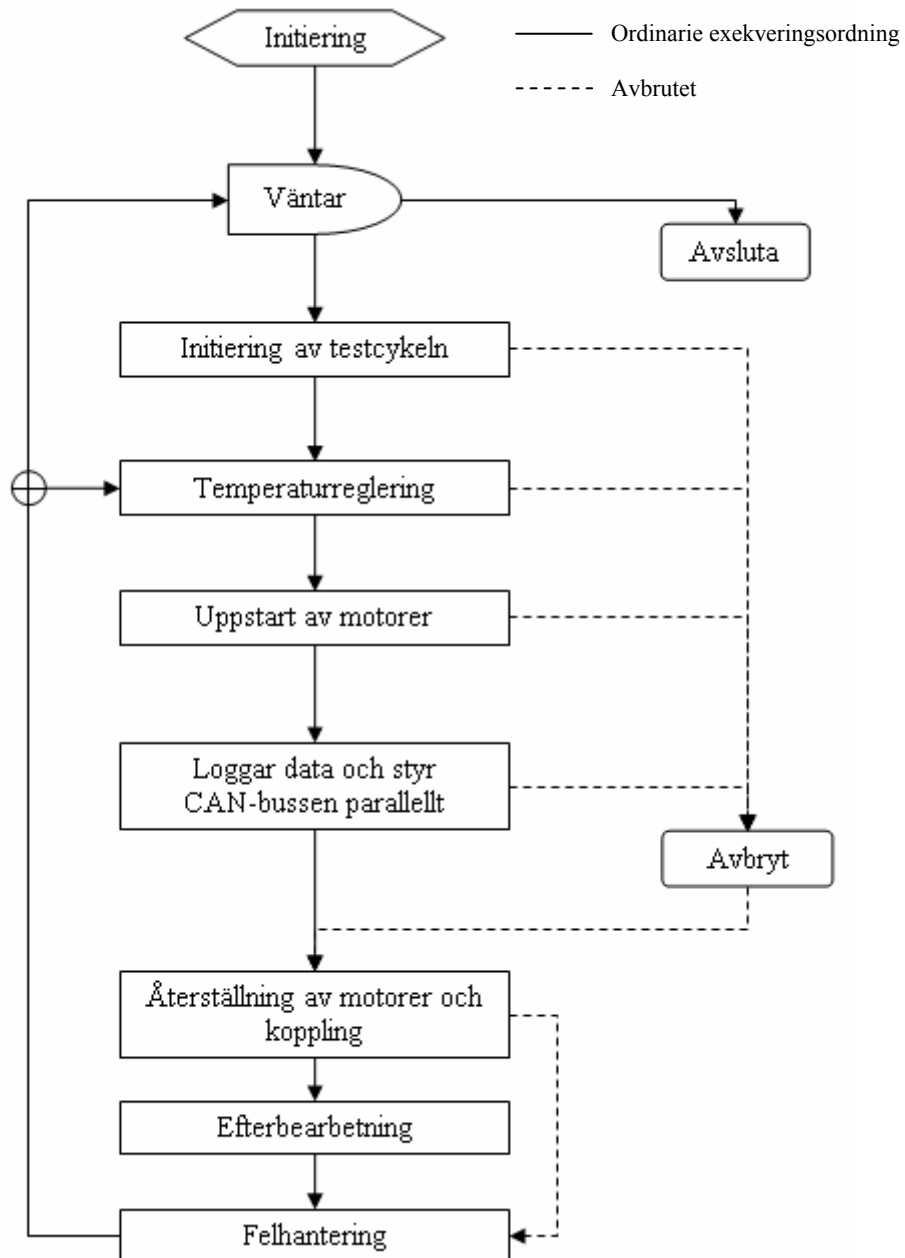
Förutom ovan nämnda mål, så har vi strävat efter att göra de flesta SubVI (kan liknas vid en subrutin i ett textbaserat programmeringsspråk) så pass generella att de ev. kan användas i kommande projekt för riggen.

Ett krav från användarens sida är att användargränssnittet ska vara enkelt. För att få bort så många parametrar från operatörmiljön som möjligt, använder vi oss av testrecept. Ett testrecept är en textfil som innehåller all information om testsekvensen (så som temperatur, varvtal, gränser för underkänt/godkänt etc.), på så sätt behöver operatören bara välja vilket recept som skall köras. Detta är en beprövad teknik som med framgång används på andra riggar.

När det gäller testreceptets utformning, har vi försökt göra det så modulärt som möjligt, dvs. man ska kunna köra olika sorters momentsteg i vilken kombination som helst (se kapitel 3.3, Styrning av CAN-bussen). Med testreceptsförfarandet har man förflyttat arbetsbördan från operatören till utformaren av receptet. Då de flesta testen är standardtest, kommer de skapade receptet att återanvändas många gånger, och den samlade arbetsbördan blir mindre. Därför kan man kosta på sig en aning mer komplicerat struktur på receptet. Dock har vi försökt göra även detta så enkelt och logiskt som möjligt. (se bilaga A, användarmanual).

3.2.1 Programöversikt

Nedan följer en detaljerad beskrivning av programmet. Flödesschemat nedan visar i stora drag hur programmet är uppbyggt.



Figur 10, Flödesschema för programstrukturen

Initiering

Det första som sker när programmet startas upp är att alla *task:ar*ⁱ som kommer att användas i programmet skapas och initialiseras. Även USB-portarna som momentgivarna är inkopplade på initieras. Kalibreringsvärdena för respektive kanal hämtas från en initialiseringsfil (ini-fil).

Väntar

I detta steg väntar programmet på indata från användaren. Bland annat skall operatörs-id, kopplings-id, recept och annan nödvändig information anges.

Initiering av testcykeln

När användaren trycker på kör och testcykeln startas, läses receptet in och sparas i en vektor.

Temperatur: -20	Stegnamn: "Moment1"	Stegnamn: "Moment2"	Temperatur: 0	Stegnamn: "Moment3"
Varvtal: 65	Börvärde: 2500	Börvärde: 800	Varvtal: 30	Börvärde: 1500
Stabiliseringstid: 2	Tid: 5	Tid: 2	Stabiliseringstid: 2	Tid: 3
...

Figur 11, Receptvektor med testparametrar

Vektorn innehåller inställningsparametrar för testcykeln, samt parametrarna för varje enskilt momentsteg i själva testet. Receptet kontrolleras även så att temperaturerna inte ligger utanför givna gränser, detta av säkerhetsskäl.

Utöver detta så initieras och startas momentgivarna.

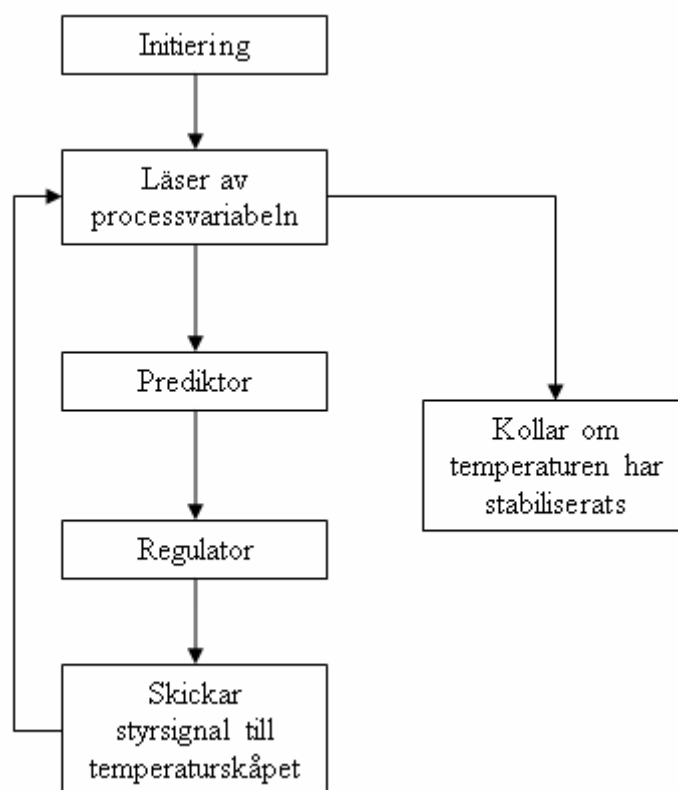
ⁱ En *task* i detta sammanhang är en samling av en eller flera fysiska kanaler innehållande information om samplingsfrekvens, skalor, vilken typ av givare man använder sig av etc.

Temperaturreglering

För att reglera temperaturen används en PID-regulator. Då processen är väldigt långsam och vi vill undvika överslängar (se kapitel 4.1, Temperaturreglering), använder vi oss av en prediktor.

När kopplingstemperaturen har uppnått rätt temperatur, stabiliseras den under en viss tid, specificerad av användaren. Detta för att försäkra sig om att rätt temperatur uppnåtts i alla delar av kopplingen.

Under tiden som temperaturen i kopplingen ändras kollar man också att batterierna till momentgivarna inte tar slut.



Figur 12, Flödesschema av temperaturregleringen

I stora drag går temperaturregleringen till enligt figuren ovan, förklaringen av flödesschemat finns listade nedan.

Initiering

Inne i temperaturregleringen använder vi oss av en prediktor. Denna nyttjar en vektor av gamla värden för att kunna bestämma temperaturen i framtiden. Vektorn initieras med den aktuella temperaturen.

Läser av processvariabeln

För att minimera inverkan från brus, läser man av temperaturen 1000 gånger under 10 sekunder. Genom att medelvärdesbilda över dessa punkter minskas inverkan från bruset.

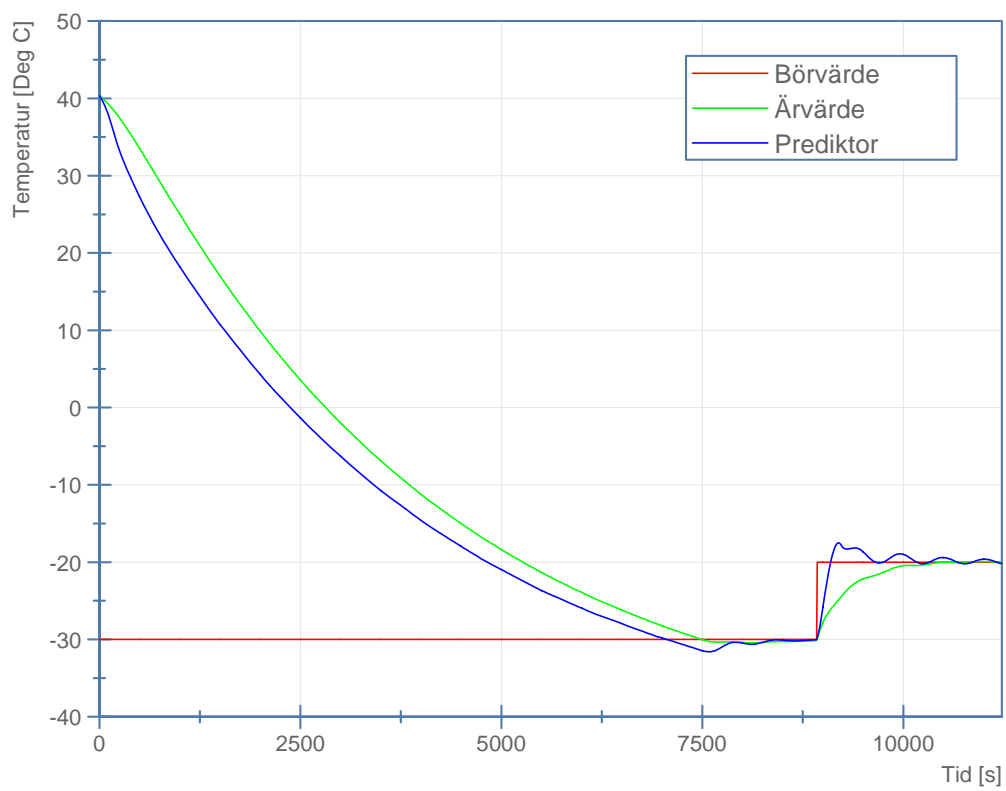
Medelvärdet sparas sedan i en vektor för att användas av prediktorn.

Prediktor

Prediktorn fungerar på så sätt att en rät linje anpassas efter de senaste 30 medelvärdena.

Derivatn av linjen räknas ut och multipliceras med en konstant. Detta värde adderas sedan till processvariabeln.

På detta sätt förskjuts temperaturkurvan ett visst antal minuter framåt i tiden.

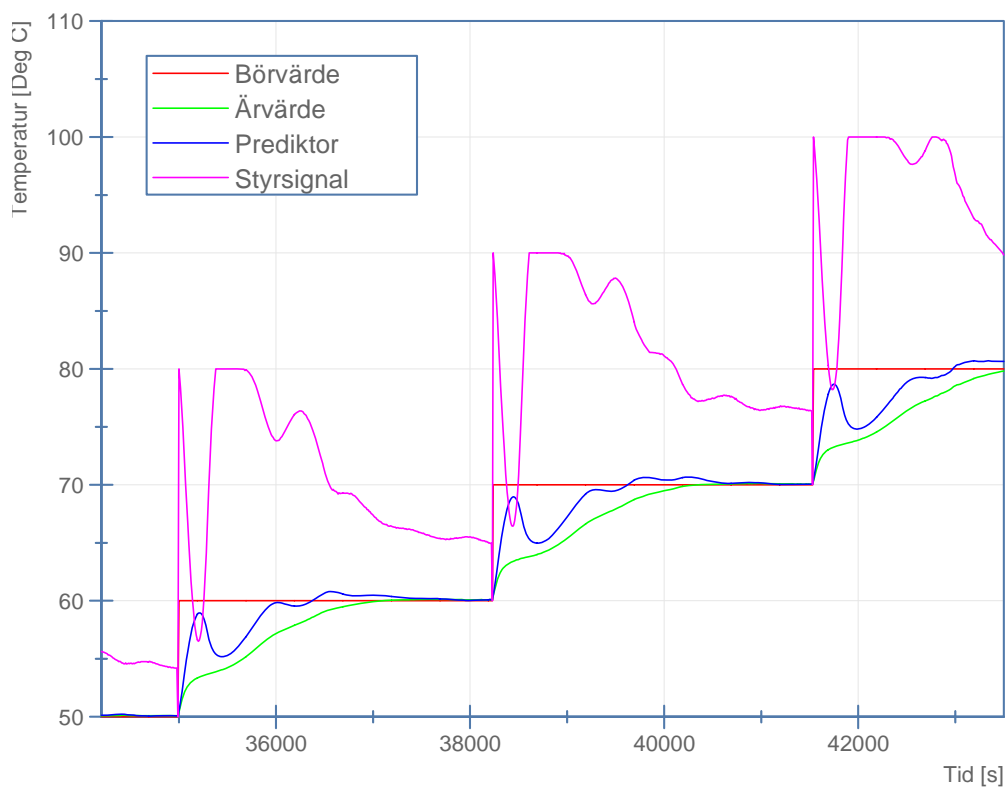


Figur 13, Temperatursteg med prediktor

Regulator

Regleringen sker med en PID-regulator, där processignalen kommer från prediktorn.

Regulatorn är utrustad med anti-windup-funktion, men då systemet är långsamt begränsas styrsignalen till börvärdet $\pm 20^\circ\text{C}$. Detta för att minimera risken för oscillation.



Figur 14, Stegsvär från 50°C till 80°C i steg om 10°C

Mellan temperaturstegen i figuren har en testsekvens körts, därav den stora derivatan i början av steget.

Skickar styrsignal till temperaturskåpet

När regulatorn är klar skrivs den nya styrsignalen till skåpet, varpå hela cykeln upprepas igen.

Kollar om temperaturen har stabiliserats

Tiden för hur länge temperaturen skall vara stabil är angivet av användaren i receptet.

Då processignalen befunnit sig inom ett intervall av $\pm 0,2^{\circ}\text{C}$ från den önskade temperaturen under angiven tid, avslutas temperaturregleringen och testsekvensen påbörjas.

Uppstart av motorer

Testbänken förbereds inför provet, detta sker i tre steg. Det första som händer är att motorstyrningen slås till, därefter sätts varvtalen på motorstyrningarna. Eftersom motorstyrningen rampar upp varvtalet (en säkerhetsåtgärd gällande ett annat projekt), kontrollerar man därefter att rätt varvtal uppnåtts innan programmet fortsätter sin exekvering.

Loggning av data och styrning av CAN-bussen, sker parallellt

Fem sekunder innan testsekvensen inleds slås spänningen till ECU:n på, detta för att pumpen ska hinna bygga upp ett tryck i ackumulatorn innan testsekvensen startar.

Samtidigt börjar loggningen. Det finns två anledningar till att man startar loggningen några sekunder innan testsekvensen inleds. Dels försäkrar man sig om att man loggar hela testsekvensen, dels behövs de extra inledande sekunderna för den automatiska rapportgenereringen.

Styrning av CAN-bussen

Testsekvensen består av en serie momentsteg efter varandra. Information om hur dessa steg ser ut är lagrat i receptvektorn.

För att styra kopplingen, och få den att överföra önskat moment, måste den kontinuerligt förses med viss kritisk information via CAN-bussen.

Vector har utvecklat färdiga drivrutiner för CanCase:en till LabView. För att sända signaler måste en sändbuffert laddas med signalerna och deras respektive värden, varefter bufferten är redo att sändas ut på CAN-bussen. Bufferten töms inte vid sändning, utan innehållet förblir oförändrat, varför det är enkelt att uppdatera bussen med samma signaler om och om igen med önskad frekvens.

Utöver sändning och uppdatering av buffert finns även drivrutiner för att läsa av signaler från CAN-bussen, kolla status etc.

Sekvensen inleds med att man hämtar information om första momentsteget ur receptvektorn.

De nödvändiga signalerna laddas in i CanCase:ens sändbuffert, och sänds ut på CAN-bussen. För att hålla reda på starttiden för respektive steg, sparas nu tiden för steget i en vektor. Då kopplingen måste ta emot signaler med 10 ms mellanrum, sänds signalerna ut om och om igen under hela momentsteget.

När steget är slut laddas sändbufferten med nästa momentsteg i vektorn och proceduren upprepas. Så sker tills det inte finns fler momentsteg kvar i testsekvensen.

Loggning av data

I de standardtest som är tänkta att köras i riggen, är den högsta samplingsfrekvensen som förekommer 200 Hz. Därför loggar vi samtliga test med denna frekvens.

De signaler som loggas är; temperatur, tryck, varvtal och moment. Signaler från CAN-bussen kan även loggas (i nuläget loggas endast ECU-temperaturen).

Då loggningen är en kritisk del av testningen, är det viktigt att man kan lita på resultatet. För att göra loggningen så resurssnål och stabil som möjligt, har en del åtgärder vidtagits.

DAQ:ens klocka används till timingen, den sköts alltså av hårdvaran och inte av mjukvaran. Vidare har användargränssnittet optimerats genom att resurskrävande grafik så som grafer etc. inte uppdateras under teststeget, utan visas när steget är klart. Detta skall inte vara något problem ur användarsynpunkt, då riggen antagligen kommer att vara obemannad under testcykeln.

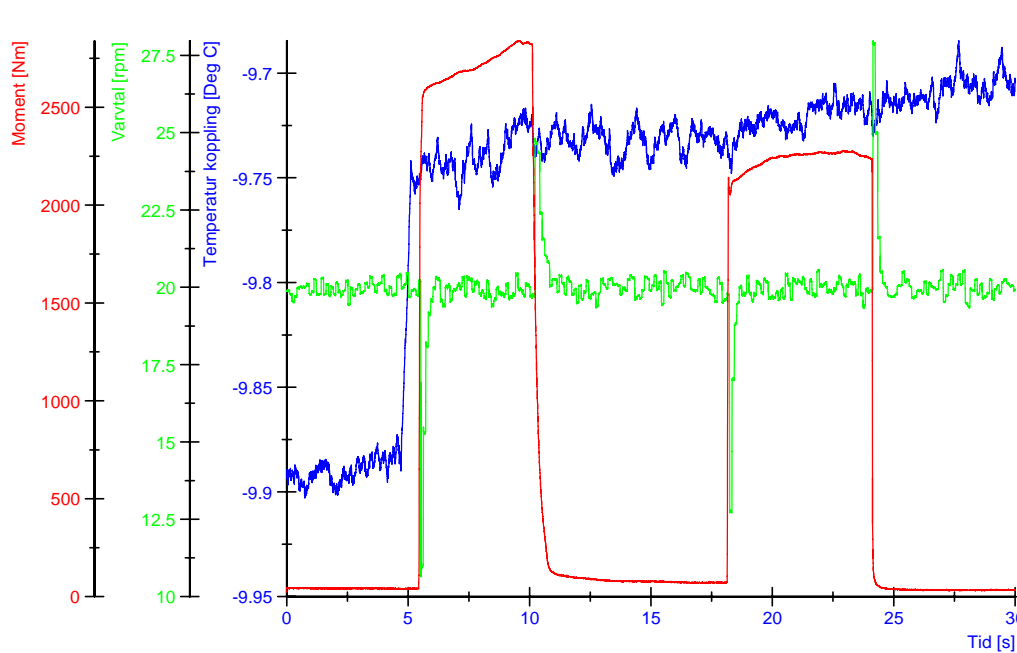
För att förhindra att loggningsdata fyller minnet, så förs dessa över till en binär fil med 0,5 s mellanrum.

Dessutom sker så lite datamanipulation som möjligt under själva loggningen (skalning och dyl. sker i efterbearbetningen).

För att loggningen av momentet ska ske med samma hastighet som övriga signaler, loggas det i samma loop, dvs. momentloggningen tidmärks även av DAQ:ens klocka.

Ett problem med loggningen var att de allra första samplingarna inte skedde med riktigt 5 ms mellanrum, utan de drog sig en del. Detta löstes genom att de första 0,5 sekunderna inte skrivs till filen, samt att t_0 , starttiden (en kritisk komponent i efterbearbetningen), justeras för att kompensera för detta.

Efter att testsekvensen är utförd, läses den binära filen av, och data skickas vidare till efterbearbetning.



Figur 15, Loggat moment, varvtal och temperatur från en testsekvens. Att temperaturen i kopplingen inte varit homogen kan ses på temperaturkurvan.

Återställning av motorer och koppling

När provsekvensen har körts klart slås spänningen till ECU:n av.

Då vi saknar insignaler för att i mjukvaran upptäcka om nödstoppet blivit utlöst, måste det ske på annat sätt. Ett av dem är att kolla om motorerna stannat under testsekvensens gång. Därför kollas varvtalet på motorerna i slutet av sekvensen.

Därefter stängs motorstyrningarna av och varvtalet på båda motorerna sätts till noll.

När motorerna står stilla skakas ventilen. Detta görs för att säkerställa att ackumulatorn är tom inför nästa temperatursteg, eller för det fall att det var den sista testsekvensen och kopplingen ska monteras bort från riggen.

Återställningen sker alltid, även om det har uppstått ett fel eller om användaren har avbrutit testet. Detta är för att säkerställa att riggen alltid befinner sig i utgångsläget.

Efterbearbetning

Tiderna för momentstegen är tagna med en referensklocka i LabView, och subtraheras från tiden för när loggningen startar. På så sätt är alla tider angivna relativt starttiden för loggningen.

Ett problem är att det finns risk för att klockan slår om under testsekvensen, vilket leder till att starttiderna för momentstegen kan bli negativa. Har detta inträffat adderas starttiden för momentsteget med referensklockans maximala värde.

Data som kommer från loggningen saknar information såsom kanalnamn, samplingsfrekvens, enheter etc. All sådan information är viktig när data ska sparas till loggfilen, och adderas därför till dessa data.

Momentgivarna ger ut ett oskalat moment, därför hämtas kalibreringsvärdena från ini-filen och momentkurvan skalas om.

Efter det att datamängden har kompletterats med ovanstående information sparas den i en lvm-filⁱ tillsammans med information om provsekvensen, såsom temperatur, varvtal på motorer och tiderna för de olika momentstegen. Detta görs för att det ska vara lätt att manuellt kunna använda data och göra andra uträkningar om så önskas.

Utöver mätfilen skapas en rapport som innehåller en graf med momentkurvan från testsekvensen. Den innehåller även information om kopplings-id, temperatur, aktuellt recept och skillnadsvarvtal för testsekvensen. I rapporten ingår även information om de olika momentstegen, t.ex. stig och falltider, maxmoment etc. (se kapitel 3.4, rapportgenerering).

ⁱ LabView Measurement File

Felhantering

Har ett fel uppstått blir användaren informerad om detta med en popup-ruta, där det framgår vad som är fel. Användaren får sedan tre alternativ att välja bland.

Vill användaren köra om den provsekvens som kördes när felet uppkom, hoppa till nästa temperatur, eller avbryta testcykeln helt och återgå till början av programmet?

I felhanteringen sätts temperaturskåpet på 20°C, så att skåpet inte står och går om användaren inte är närvarande när ett fel uppstår.

Uppstår inget fel tas den del av receptet som körts bort från receptvektorn, och nästa temperatur i testcykeln inleds. Är receptvektorn tom avslutas programmet och återvänder till vänteläget.

Temperatur: -20 Varvtal: 65 Stabiliseringstid: 2 ...	Stegnamn: "Moment1" Börvärde: 2500 Tid: 5 ...	Stegnamn: "Moment2" Börvärde: 800 Tid: 2 ...	Temperatur: 0 Varvtal: 30 Stabiliseringstid: 2 ...	Stegna Börvär Tid: ...
---	--	---	---	---------------------------------

Figur 16, Receptvektor med borttagna element

Avslutar

Om användaren väljer att avsluta programmet frigörs de upptagna COM-portarna och DAQ:arna återställs. Därefter stängs programmet ner.

Avbryt

Avbrytfunktionen går parallellt med programmet och väntar på att användaren ska trycka på avbryt. När så sker avbryts processen och felhanteringsstrukturen vidarebefordrar meddelandet genom programstrukturen. På så sätt krävs ingen extrakod för att ta hand om avbrytfunktionen.

3.3 Styrning av CAN-bussen

För att kommunicera med kopplingen via CAN-bussen krävs, förutom den tidigare nämnda hårdvaran, CanCaseXL, att man vet vilka signaler som är nödvändiga för kopplingens funktion. Detta är en definitionsmässig fråga som varierar mellan olika sorters kopplingar. Till viss del beror det även på vilken mjukvaruversion kopplingens ECU är laddad med.

Signalerna som förekommer på fordonets CAN-buss är definierade från respektive tillverkare, och finns fastställda i särskilda dokument. Baserat på dessa dokument finns CAN-databaser där de olika signalerna är specificerade.

Kommunikationen kommer att ske med hjälp av dessa CAN-databaser. För att kunna använda LabView drivrutiner till hårdvaran krävs att man länkar ihop signalerna i CAN-databasen med symboliska namn, vilka senare kommer att användas för att referera till signalerna i LabView. Denna, samt ytterligare information (t ex in/utsignal, aktuellt protokoll, kanal etc.) sparas i en *.vlv-fil, CAN-projekt filen.

Filename	Message	Signal	Alias	Direction	Channel	Use
PQmix.DBC	mLW_1	LW1_Lenkradwinkel	SWA_ABS	Out	CAN 1	active
PQmix.DBC	mLW_1	LW1_Lenkradwinkel_ID	SWA_CAL	Out	CAN 1	active
PQmix.DBC	mLW_1	LW1_Lenkradwinkel_Sign	SWA_SIGN	Out	CAN 1	active
PQmix.DBC	mLW_1	LW1_Lenkradwinkel_Status	SWA_STATUS	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Fahrerwunschmoment	DRIVER_TRQ	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Fahrpedalwert	ACC_PEDAL	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Inneres_Motor_Moment	DRIVER_TRQ_LOSSES	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Kupplungsschalter	COUPLING_PEDAL	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Motor_Verlust_Moment	DUMMY	Out	CAN 1	active
PQmix.DBC	mMotor_1	MO1_Motordrehzahl	ENGINE_SPEED	Out	CAN 1	active
PQmix.DBC	mMotor_2	MO2_Bremslichtschalter	BRAKE_ACTIVE	Out	CAN 1	active
PQmix.DBC	mBrems_2	BR2_Querbeschl	LAT_ACC	Out	CAN 1	active
PQmix.DBC	mBrems_2	BR2_Querbeschl_Gemessen	LAT_ACC_MFAS	Out	CAN 1	active
PQmix.DBC	mBrems_2					
PQmix.DBC	mBrems_2					
PQmix.DBC	mBrems_3					
PQmix.DBC	mBrems_3					
PQmix.DBC	mBrems_3					
PQmix.DBC	mBrems_3					
PQmix.DBC	mBrems_4					
PQmix.DBC	mBrems_5					
PQmix.DBC	mBrems_5					
PQmix.DBC	mKombi_1					
PQmix.DBC	mBrems_1					
PQmix.DBC	mGatewayKombi					
PQmix.DBC	mKombi_3					
PQmix.DBC	mMotor_5					

PQmix.DBC - Anteckningar				
Arkiv	Redigera	Format	Visa	Hjälp
BO_1088 mGetriebe_1: 7 STG_Getriebe				
SG_	GE1_Zielgang	: 8 401+	(1,0) [0 0]	"" STG_Halddex
SG_	GE1_Botschaftszahler	: 52 401+	(1,0) [0 15]	"" STG_Halddex
SG_	GE1_Ubertragungsfunk	: 16 801+	(1,0) [0 0]	"" STG_Halddex
BO_912 mGatewayKomfort_1: 8 STG_Gateway				
SG_	GKL_Ruckfahrlicht	: 28 101+	(1,0) [0 1]	"" STG_Halddex
BO_800 mKombi_1: 6 STG_Kombi				
SG_	K01_Handbremse_Angezogen	: 9 101+	(1,0) [0 0]	"" STG_Halddex
SG_	K01_Botschaftszahler	: 40 401+	(1,0) [0 15]	"" STG_Halddex
BO_704 mAllrad_1: 8 STG_Halddex				
SG_	AL1_Fehler_Kupplung	: 0 101+	(1,0) [0 1]	"" vector_XXX
SG_	AL1_Botschaftszahler	: 60 401+	(1,0) [0 15]	"" vector_XXX
SG_	AL1_Byte7	: 48 1201+	(1,0) [0 4095]	"" vector_XXX

Figur 17, CAN-databas och vlv-fil

Utöver de CAN-signaler som förekommer i fordonet, finns ytterligare CAN-databaser mer specifika för Haldex-kopplingarna. Med hjälp av dessa kan man styra kopplingen med avseende på ström till ventilen, begära ett visst moment samt styra andra saker, så som spänning till pumpen etc. Dock är det de två förstnämnda som är aktuella för vårt projekt.

Sammantaget kan man säga att det finns tre sätt att styra kopplingen:

Man simulerar signalerna i bilen. Kopplingen kommer då att uppföra sig så pass verklighetstroget som möjlig. Detta ger en möjlighet att testa olika funktioner, så som *fast open*ⁱ.

En andra möjlighet är att styra kopplingen på moment och en tredje att styra den med avseende på ström. De två sistnämnda ger en möjlighet att kontrollera t.ex. momentnoggrannhet och maxmoment.

Utöver att styra kopplingen, kan man även läsa av saker så som temperaturen i kopplingen, uppskattad temperatur i lamellpaketet, ventilström etc. Dock skall nämnas att detta inte gäller för alla kopplingar samt att det är beroende av vilken mjukvaruversion ECU:n är laddad med.

ⁱ Att så snabbt som möjligt öppna ventilen vid t.ex. ingrepp från anti-sladdsystemet

3.4 Rapportgenerering

Presentation av de data som loggats sker på två sätt. Det skapas en rapport som innehåller datan i bearbetad form. Där presenteras de viktigaste resultaten samt övergripande information om testsekvensen.

Utöver rapporten, genereras en mätfil, innehållande data i obearbetad form. I mätfilen framgår den information som är nödvändig för att manuellt kunna utvärdera och dra slutsatser av testet.

HTML-rapport och mätfil

Vi valde att skapa rapporten i HTML-format, då det är ett enkelt sätt att presentera bilder och tabeller på. Dessutom krävs det inga specialprogram för att titta på rapporten, det enda som behövs är en vanlig webbläsare.

En annan fördel är att rapporten blir bakåtkompatibel, dvs. rapporten kommer att vara läsbar även om några år, trots ny mjukvara.

De resultat som skall presenteras i rapporten är; medelvärdet av det uppnådda momentet, max och minvärden, skillnad mellan max och minvärden, samt stig alternativt falltider.

Då receptet är modulärt uppbyggt, ställer det vissa krav på rapportgenereringen. Hade testen alltid sett likadana ut, hade man kunnat använda en färdig mall till rapporten. Så är alltså inte fallet nu.

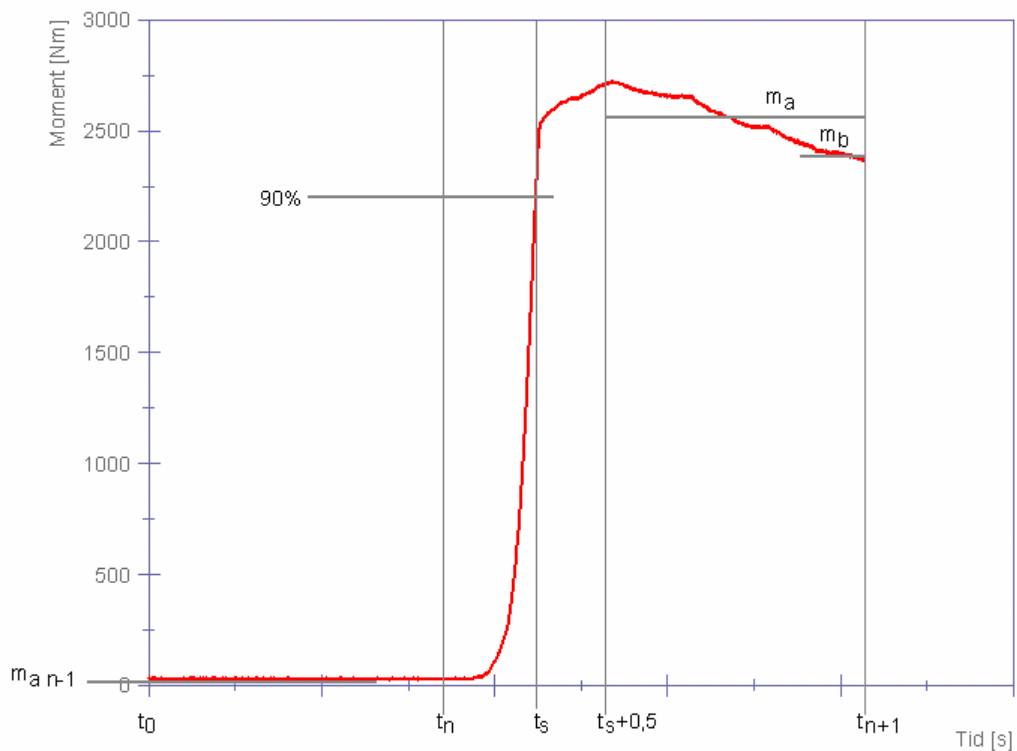
För att rapporten ska klara av alla möjliga testkombinationer, så sker en automatisk rapportgenerering, baserad på loggade data samt tiderna för varje steg.

Medelvärdesbildningen ska ske efter att momentet har stabiliserats. Detta är en definitionsfråga, och är svårt att utvärdera automatiskt. Baserat på vissa antaganden kan dock en uppskattning göras (se Figur 18 ovan).

Max respektive minvärdet är de max/minpunkter som finns inom medelvärdesbildningsintervallet.

Stigtiden har vi definierat som tiden det tar att nå 90 % av skillnaden mellan m_a och $m_{a,n-1}$. På motsvarande sätt definieras falltiden.

Figuren nedan ger en idé om hur resultaten räknas ut.

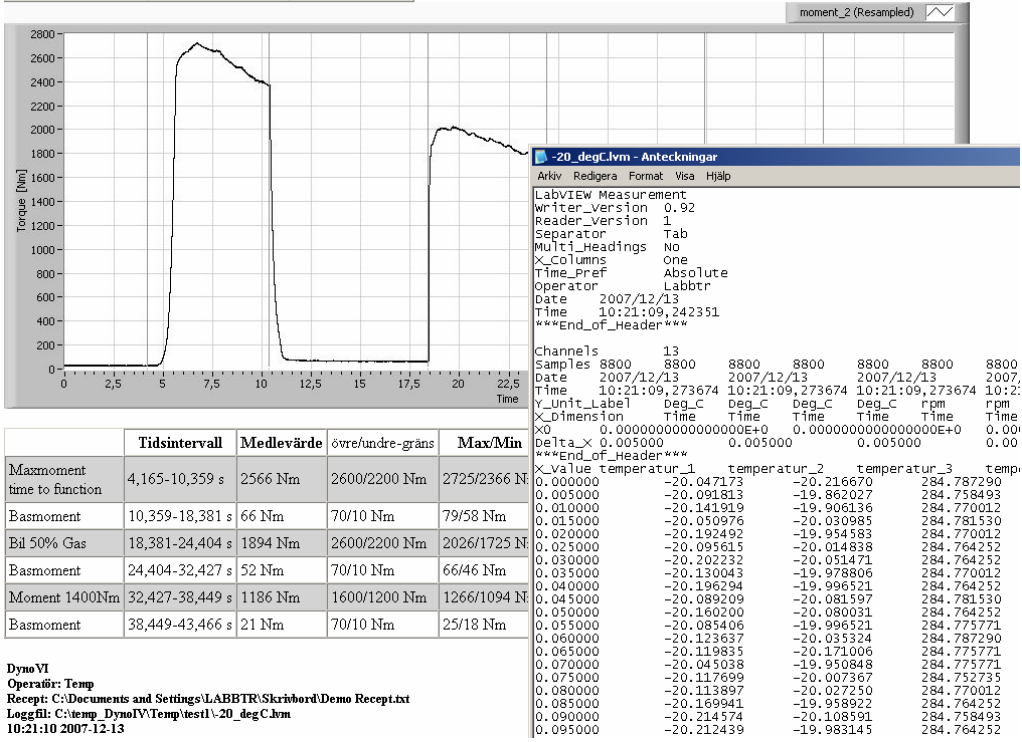


Figur 18, Momentsteg

- 1) Först medelvärdesbildas momentet över de sista 0,5 sekunderna, m_b . Detta bör vara en rimlig uppskattning av det riktiga medelvärdet.
- 2) Därefter räknas stigtiden ut med avseende på m_b .
- 3) Slutligen räknas ett nytt medelvärde, m_a , ut över tiden $t_s + 0,5s$ till t_{n+1} , det är detta medelvärde som presenteras i rapporten.

Slutligen fogas resultaten för alla momentstegen för aktuell temperatur ihop i rapporten och presenteras för användaren.

Kopplings-ID: test1 Temperatur: -20 °C rpm1=65, rpm2=0



Figur 19, Rapport med tillhörande lvm-fil

4 Problem och svårigheter

Självklart har inte projektet förflutit helt smärtfritt, utan vi har råkat ut för en del motgångar. Lyckligtvis har vi lyckats lösa de flesta av dem, även om det inneburit en del arbete.

Ett av de första problemen vi stötte på var monteringen av kopplingen i isolerhuven. Då flänsarna på axlarna interfererade med lådan, var en förlängningsadapter tvungen att tillverkas. I samband med monteringen av adapter var även andra justeringar tvungna att göras, bland annat var utgående axel från den torsionsstyva kopplingen presspassad, vilket inte var meningen. Detta försvårade monteringen av Haldex-kopplingen.

Att kylhuven råkade ut för en olycka har redan diskuterats i kapitel 2.1.2.

Momentgivarna har också varit ett stort problem. Det händer ibland att de hänger sig under loggningen (yttrar sig på så vis att de sänder ett konstant värde på momentet). Vidare så är de svåra att starta, då de går in i resetläge om man inte slår på och av strömbrytarna upprepade gånger när batteripacken är fränkopplad. Vad dessa problem beror på vet vi inte riktigt. Tyvärr kan man inte programmera om flash-minnet i noderna (sändarna), då de är av en äldre modell. För att uppgradera mjukvaran måste de skickas till tillverkaren.

Dessa problem är mest utpräglade för den vänstra momentgivaren. Den högra fungerar tillfredsställande, dock ej helt utan problem. Tills vidare använder vi oss därför enbart av den högra givaren.

4.1 Temperaturreglering

Temperaturregleringen visade sig vara svårare att få rätt på än vad vi först trodde. Det största problemet var att varje stegsvar tog väldigt lång tid, vilket i sin tur kraftigt begränsade antalet reglerparametrar som kunde testas under en arbetsdag.

Problemen som vi stötte på var bland annat att regleringen började oscillera väldigt mycket, vilket hade sin grund i flera olika saker. För det första så innehöll signalen som kom från temperaturgivaren inuti kopplingen väldigt mycket högfrekventa störningar. Eftersom vi använde oss av en derivatadel i regulatorn, blev den väldigt känslig för störningarna. Vi löste problemet genom att sampla ett stort antal gånger för att sedan medelvärdesbilda över dessa samplingsar. Medelvärdet utgjorde en relativt störfri processvariabel.

Ett annat problem som vi hade var lång dötid i systemet. Ändrades styrsignalen tog det mellan tio minuter och en kvart innan temperaturen i kopplingen påverkades.

För att komma runt de problem detta gav upphov till försökte vi först justera D-delen i regulatorn, utan framgång. På grund av tidsbrist beslutade vi oss därför att skapa vår egen prediktor. Baserat på temperaturkurvans lutning förutses vad temperaturen kommer att vara om cirka tio minuter, förutsatt att trenden på kurvan håller i sig.

Vi började med att räkna ut derivatan mellan de två senaste mätningarna för att sedan multiplicera derivatan med ett tal, detta fungerade men prediktorn ”rusade” iväg då stigningen på temperaturkurvan blev för brant.

Istället för att endast använda två mätpunkter, anpassas en rät linje längs de 30 senaste mätpunkterna.

Linjäriseringen sker med minstakvadratmetoden, vilken ofta används för att minimera felet i en funktion som ska anpassas efter observerade mätvärden. Derivatans multipliceras sedan med en konstant och summeras till den aktuella

temperaturen. Resultatet blev en kurva som låg förskjuten ca 10 minuter framåt i tiden (se Figur 13, sidan 28). Detta gjorde att temperaturen nästan inte fick någon översläng.

5. Förbättringar och fortsatt arbete

Under tiden som arbetet har pågått har vi funnit en hel del fel och brister med riggupställningen. Bristerna har försvårat arbetet och i vissa fall har vi behövt gå omvägar för att lösa annars enkla problem. Andra förbättringar är önskemål från vår sida för att riggen skall bli mer exakt i t.ex. temperaturregleringen eller mätning av varvtal. Nedan är de brister listade som vi har funnit medan vi har jobbet med riggen.

Temperaturskåp och temperaturhuv

- Ett rör som riktar om luftflödet mot kopplingen i temperaturhuvu bör installeras. Luften cirkulerar nu i huvudsak i bakkant av huvu.
- I botten av lådan finns det en urtagning där kopplingen skall stå, den skulle behöva isoleras. Fästplattan som temperaturhuvu står på är gjord i stål och fungerar därför som en kyl/värme-fläns.



Figur 20, Bild på urtagning i temperaturhuv

- Temperaturskåpet behöver temperaturgivare för att kunna reglera temperaturen inne i temperaturhuven. Dessa ligger i nuläget på golvet i lådan. Det skulle vara önskvärt att göra någon form av hållare för dessa. Anledningen är att skydda dem från att fördärvas samt att placeringen har betydelse för vilken temperatur de läser av.
- Ytan på foamglaset är sprött. För att göra ytan mer stöttåligt borde den förses med någon form av ytbehandling.

Riggen

- Adaptern som sitter mellan kopplingen och den ena axeln är otymplig och försvårar inmonteringen av kopplingen i riggen. Det finns möjlighet att vända på navet till de torsionsstyva kopplingarna. Detta gör att axlarna förlängs en bit, förmodligen så långt att adaptern blir överflödig.
- Konstruera plintar så att kopplingen kommer upp i rätt höjd för inmontering i riggen. Plintarna ska vara utformade så att det går att skruva fast kopplingen i testbänken.
- Stopp för drivpaketen så att de inte faller av testbänken när avstånden mellan dem justeras.
- Fläktarna till motorerna ger ifrån sig mycket ljud när de är i drift. Dessa är igång så länge nödstoppet inte är utlöst. Det betyder att fläktarna går när temperaturen i kopplingen justeras. För att minska ljudnivån i lokalen där testbänken är placerad, skulle fläktarna istället kunna slås på när motorstyrningarna får signalen att starta motorerna.
- Pulsgivarna som är placerade på drivpaketens utgående axlar borde kopplas in. Det skulle då finnas möjlighet att mäta skillnadsvarvtal och ingreppsvinkel på kopplingen mer exakt än vad som är möjligt idag.
- Signaler som motorstatus och nödstopp borde kopplas in, så att de går att läsa av i datorn.
- Mjukvaran i momentgivarna borde uppdateras.

- Kalibrering av samtliga givare i riggen bör utföras.
- Datorn borde inte vara kopplad till riggens matningsspänning. Det är bra att kunna starta om riggen utan att behöva starta om datorn.
- Glidpassningen på utgående axel till kopplingen kärvar.
- Riggen borde förses med en yttre signal som uppmärksammar personer runtom riggen att den är igång.

I programvaran

- Ta fram nya parametrar till temperaturregleringen.
- Läs av signalerna motorstatus och nödstopp om de kopplas in samt göra passande åtgärder i programmet.
- Implementera stöd för att kunna köra fler kopplingstyper i riggen.
- Möjlighet att automatiskt avbryta testcykeln om kopplingen är underkänd.

6. Resultat

Målet med att automatisera en testsekvens för verifikation av fyrhjulsdriftkopplingar vid olika temperaturer har uppnåtts. Tester kommer nu att kunna utföras snabbare och mer enhetligt jämfört med tidigare.

Maskinen har inte tagits i bruk än vilket främst beror på att inga revisioner på de kopplingsmodeller som är implementerade i riggen utförs för tillfället. Detta har också medfört att inga jämförande test har kunnat genomföras. Det finns alltså behov av att vidareutveckla riggen för att kunna köra fler kopplingsmodeller. Arbetet med att utveckla kommunikationen till Volvo-kopplingarna påbörjades, men blev avbrutet på grund av tidsbrist och bristfällig information.

För att säkerställa funktionen på loggning och CAN-kommunikation, där timingen är kritisk, har kontroller utförts. Dessa tester kunde verifiera att rätt frekvenser upprätthålls.

Tester har även utförts för att kontrollera funktionen hos temperaturregleringen. I nuläget fungerar den tillfredsställande, men vissa justeringar av parametrarna borde göras för att ge en stabilare reglering.

Vissa av testen som riggen är utvecklad för är inte speciellt väldefinierade, detta gjorde att vi fick göra vissa antaganden gällande rapportgenereringen. All data från riggen sparas därför även i en mätfil, vilket möjliggör eventuell manuell efterbearbetning.

Ett problem som vi tyvärr inte lyckades lösa var momentgivarna. Deras inkonsekventa beteende ställde till med en del bekymmer, men problemet löstes till viss del genom att vi enbart använde den momentgivaren som fungerade bäst.

Vi är nöjda med resultatet, dock finns det alltid saker som kan förbättras. Det hade varit tillfredställande att se hur vår automatiska testsekvens stod sig i jämförelse med de manuella test som utförs för tillfället.

Vi är övertygade om att den automatiserade testriggen kommer att utföra jobbet väl och att beläggningen på övriga riggar i labbet kommer att minska.

7. Referenser och litteraturförteckning

Jeffrey Travis, Jim Kring, (2006). LabVIEW for Everyone: Graphical Programming Made Easy and Fun, Third Edition.
Prentice Hall.

2006-2008 Vector Informatik, (skapat 2007-08-10).

http://www.vector-worldwide.com/vi_can_kommunikationsprinzip_en.,223.html. CAN

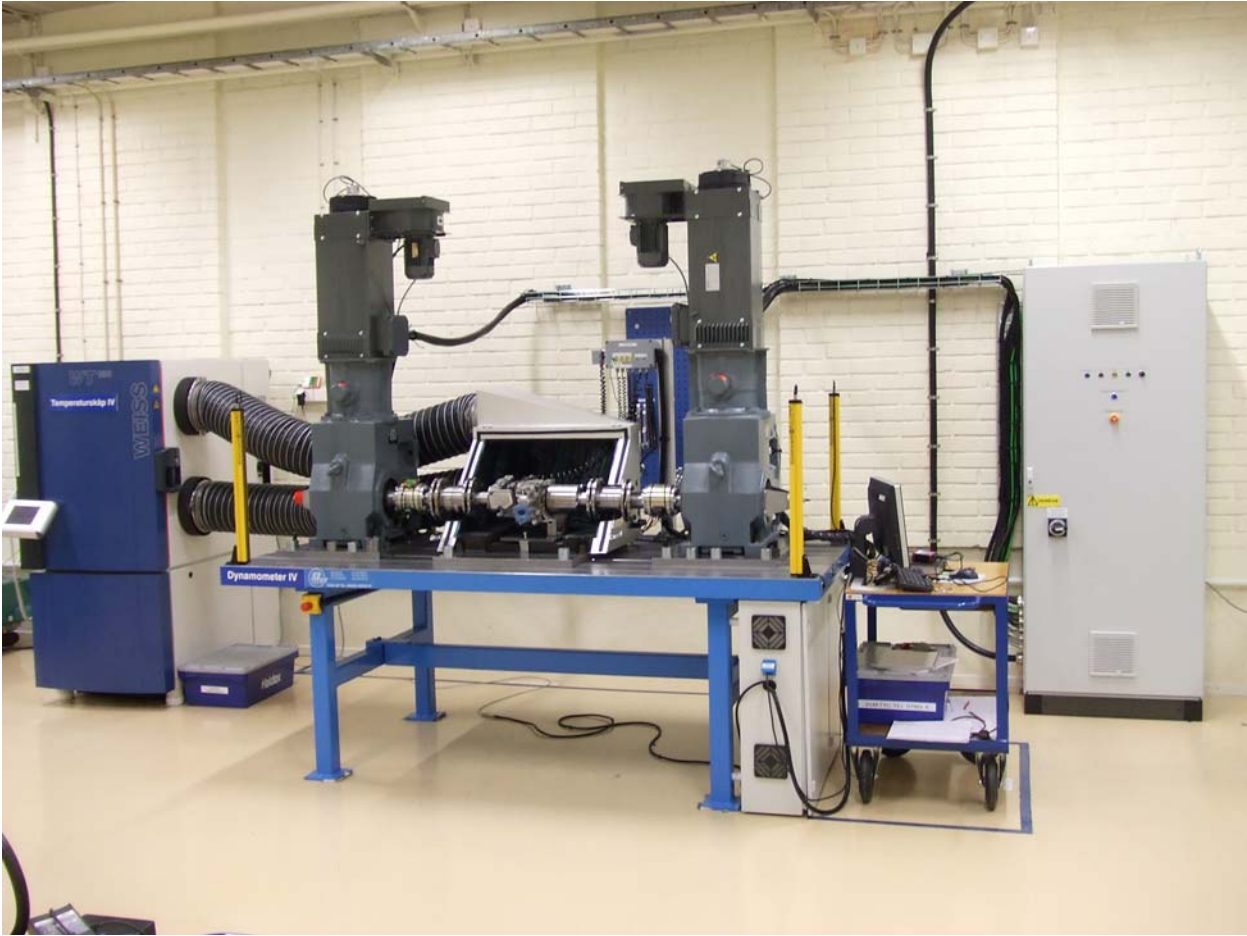
Communication Principle.

Hämtat 2007-11-03.

Bilaga A

Instruktionsmanual

Manual till Automatisk Testsekvens för DynoIV



Figur 1, Rigguppställning DynoIV

Innehållsförteckning

Innehållsförteckning	2
Bildförteckning	3
Förbered rigg inför körning	4
Testbänk	4
Temperaturskåp	5
Uppstart av automatisk temp.testsekvens för DynoIV	7
Under testets gång	10
PAUS	10
AVBRYT	11
Rapport	12
Momentgrafbeskrivning	13
Utskrift av rapport	13
Felsökning	14
Skaparecept	15
Installering av programvara	19
Kalibrering och inställningar	20

Bildförteckning

Figur 1, Rigguppställning DynoIV.....	1
Figur 2, Huvudströmbrytare.....	4
Figur 3, Temperaturhuv diktan mot fästplattan.....	4
Figur 4, Temperaturskåp med skiljeplåt.....	5
Figur 5, Anslutning för Pt100 och RS232.....	5
Figur 6, Temperaturgivare Pt100 i temperaturhuv.....	6
Figur 7, Momentgivarnod.....	7
Figur 8, Adapter.....	7
Figur 9, Bild på manöverpanel.....	8
Figur 10, Startbild.....	8
Figur 11, När programmet är igång, finns möjlighet att pausa och avbryta. Se nedan.....	10
Figur 12, ”pop-up”-ruta.....	11
Figur 13, rapport.....	12
Figur 14, skriv ut.....	13
Figur 15, utskriftsinställningar.....	13
Figur 16 receptöversikt i Excel.....	15
Figur 17, receptdel, riggstyrning.....	16
Figur 18, receptdel, styrning av koppling.....	16
Figur 19, receptdel, rapportutvärdering.....	16
Figur 20, exempel på hur ett testrecept kan se ut.....	18
Figur 21, inställningsfil.....	20
Figur 22, kalibreringsfil.....	20

Förbered rigg inför körning

Testbänk

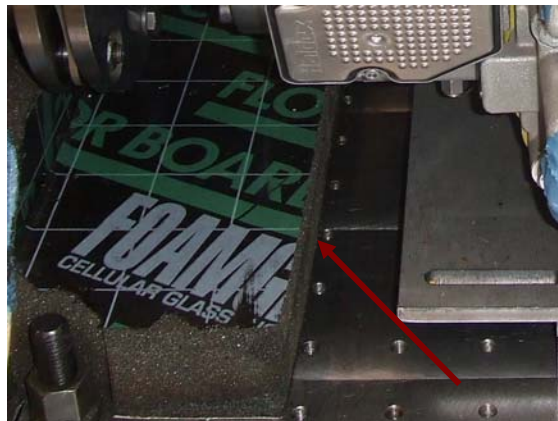
1. Börja med att slå på huvudströmmen



Figur 2, Huvudströmbrytare

2. Placera temperaturhuvn på testbänken

Se till att huvn ligger dikt an mot fästplattan för HLSC.



Figur 3, Temperaturhuv diktan mot fästplattan

3. Montera fast kopplingen

- i. Lossa muttrarna på drivtornen.
- ii. Justera drivtornen i sidled för att passa in kopplingen.
- iii. Lägg distansplintarna på fästplattan.
- iv. Montera kopplingen och adapter i riggen.
- v. Spänn fast muttrarna till drivtornen.
- vi. Isolera fästplattan med överblivet isoleringsmaterial.



*Kontrollera att roterande delar inte går emot temperaturhuvn.
Glöm ej spänna fast drivtornen. OBS!*

4. Anslut kablage

- i. Anslut CAN1-kablagen till ECU:n.
- ii. Anslut den temperaturgivare som skall användas till temperaturregleringen till T1 på givarboxen.
- iii. Om nödvändigt anslut övriga tryck och temperaturgivare till givarboxen.



Kontrollera inga kablar kan fastna i roterande delar. OBS!

Temperaturskåp

5. Förberedelse för idrifttagning av Temperaturskåpet

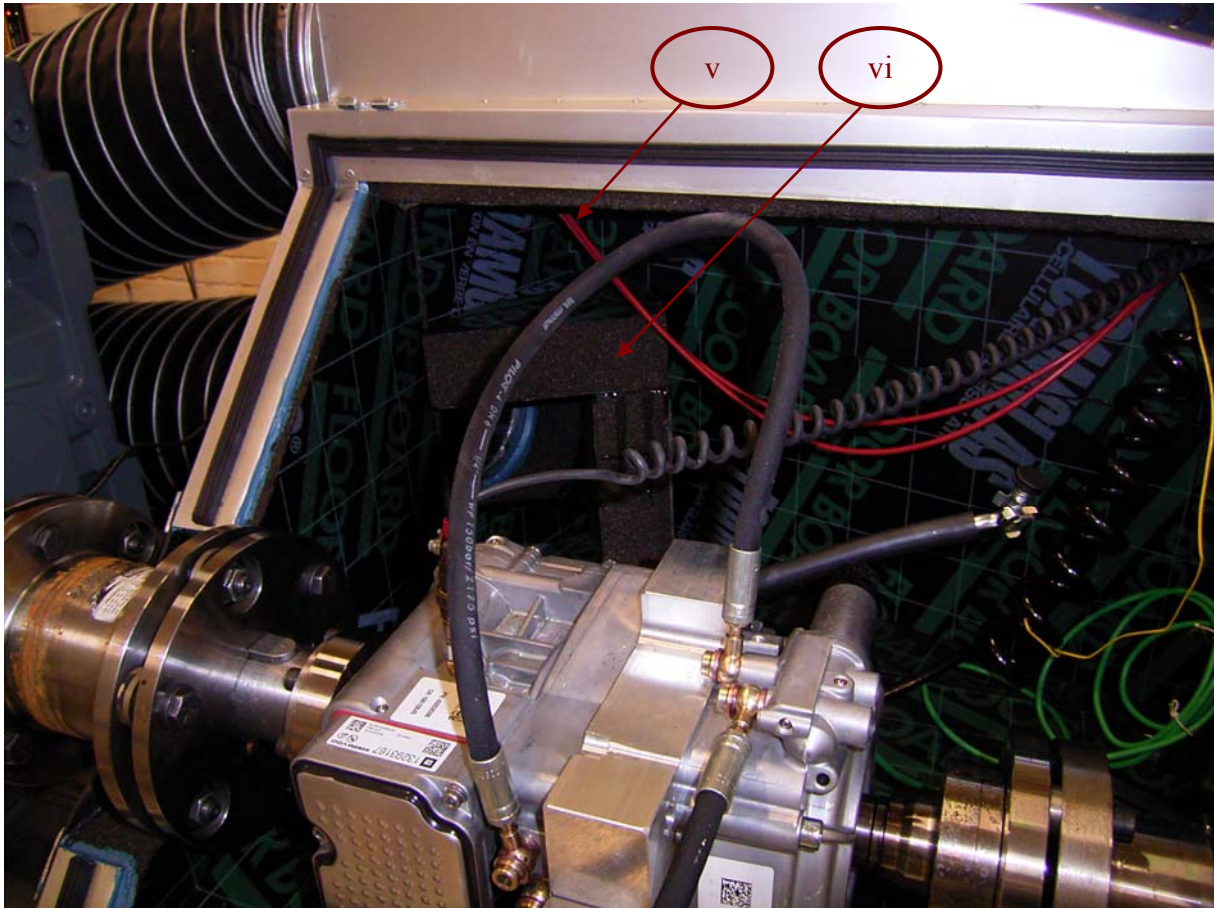
- i. Montera skiljeplåten i temperaturskåpet.
- ii. Ta bort isoleringskåporna vid anslutningsstutsarna.
- iii. Lossa spännringarna och ta bort blindskyddet.
- iv. Anslut anslutningsslangarna mellan temperaturskåpet och temperaturhuvten.
- v. Anslut mätgivaren Pt100 till temperaturskåpet, och placera givaren i utsugshålet (det övre se Figur 6) till temperaturhuvten.
- vi. Sätt in vinkeln, som vinklar av luftströmmen mot kopplingen, vid det nedre hålet (se Figur 6).
- vii. Kontrollera att RS232-gränssnittskabeln är anslutet till Temperaturskåpet.



Figur 4, Temperaturskåp med skiljeplåt



Figur 5, Anslutning för Pt100 och RS232



Figur 6, Temperaturgivare Pt100 i temperaturhuven

Kontrollera att luftflödet kan passera Pt100-givaren obehindrat.

Stäng temperaturhuven när du försäkrat dig om att inga roterande delar kan skada utrustningen.

Uppstart av automatisk temp.testsekvens för DynoIV

Innan testet körs, måste momentgivare och temperaturskåp startas.

6. Starta Momentgivare

Kontrollera att momentgivarna startar upp korrekt. Statuslampan blinkar då till en gång vid uppstart och fortsätter att blinka med ett intervall av en gång i sekunden. Om givaren inte startar ladda batterierna med adaptrarna (se Figur 8) i lådan under datorn.



Figur 7, Momentgivarnod



Figur 8, Adapter

7. Starta Temperaturskåpet

8. Kontrollera att riggen är redo för körning



Figur 9, Bild på manöverpanel

9. Starta Programmet



Figur 10, Startbild

I användargränssnittet fylls användarinformationen i. Så länge detta inte är gjort, är körknappen inaktiverad. I info-rutan visas meddelanden om vad som sker för tillfället (t.ex. om det pågår temp.reglering, vilket momentsteg som körs, om det uppstått fel etc.)

10. Fyll i användarinformation

- i. Välj "Operatör" i rullistan.
- ii. Öppna det testrecept som skall användas under "Recept".
- iii. Skriv in "kopplings-ID".
- iv. Välj "Projekt" i rullistan.
- v. Välj vilken eller vilka momentgivare som ska automatutvärderas.
- vi. Välj om temperaturregleringen ska var till eller frånslagen.

*Resultaten kommer att sparas i en katalog som heter .../"Operatör"/ "kopplings-ID".
Var därför noga med val av operatör och kopplings-id.
Programmet går inte att starta om någon av punkterna i-iv inte är ifylld.*

11. Återställ "Nödstop" på manöverpanelen

12. Tryck på "KÖR"

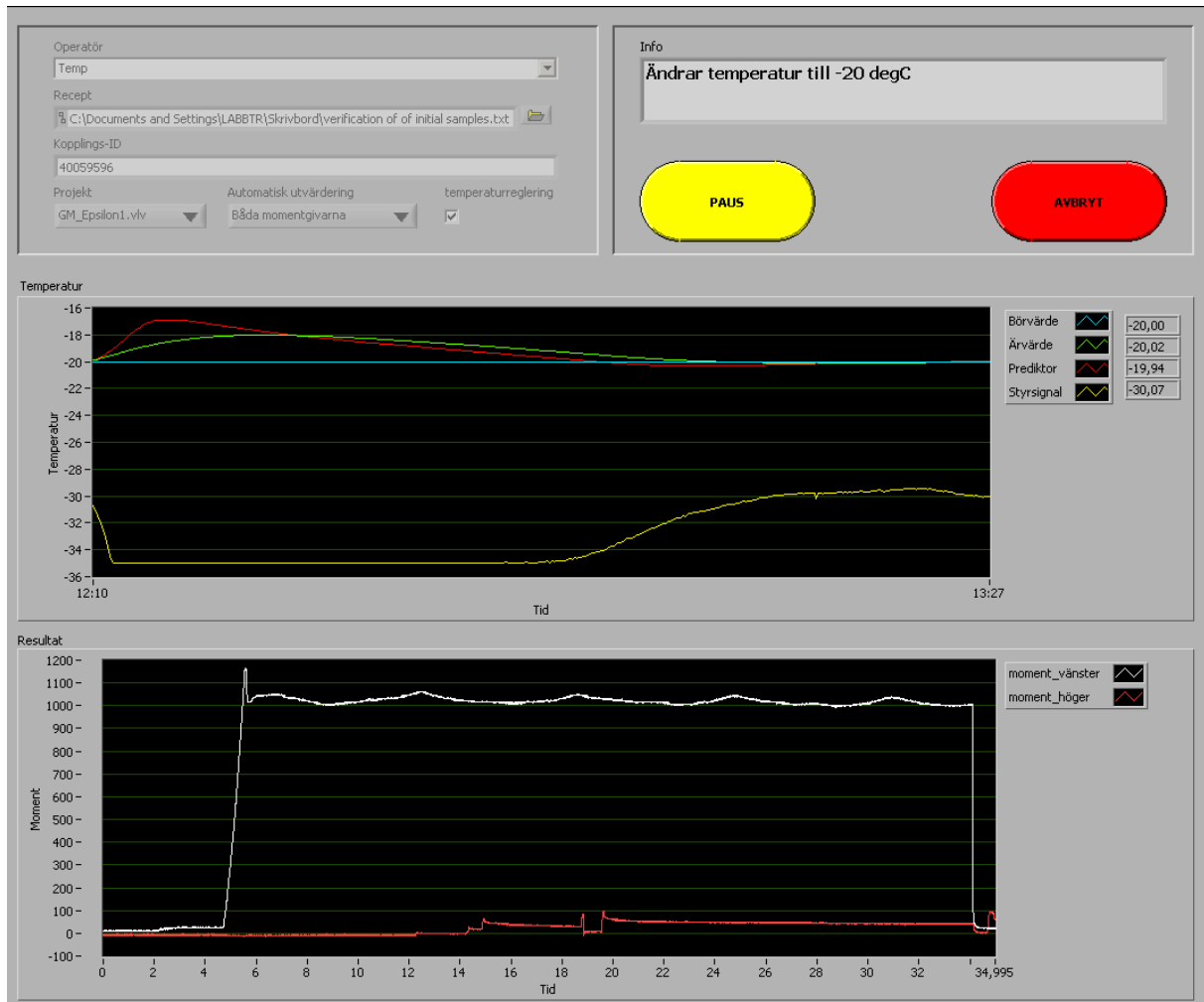
Vänta tills temperaturgrafnen börjar uppdateras eller tills rutan "Temperaturregleringen är inaktiverad" visas. Detta för att säkerställa att momentgivarna har initierats korrekt.

Under testets gång

När programmet körts igång, och temperaturen har börjat regleras, ersätts ”KÖR” och ”AVSLUTA”-knapparna med en paus och avbryt knapp.

Vidare börjar temperaturgrafan uppdateras.

Momentgrafan uppdateras inte förrän efter att en testsekvens utförts.



Figur 11, När programmet är igång, finns möjlighet att pausa och avbryta. Se nedan

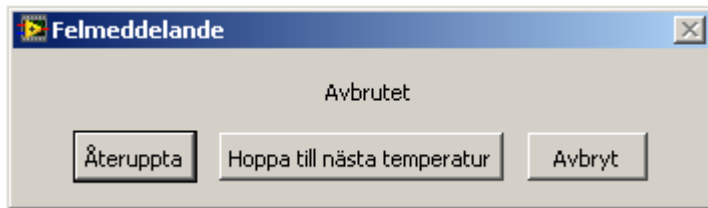
PAUS

När ”PAUS” är intryckt startas ingen ny testsekvens. Då har man möjlighet att kontrollera provobjektet.

Temperaturregleringen fortgår dock. OBS!

AVBRYT

Trycks "AVBRYT" in, avbryts testcykel, och en "pop-up"-ruta dyker upp.



Figur 12, "pop-up"-ruta

Återuppta

Aktuellt steg i testcykeln återupptas.

Hoppa till nästa temperatur

Aktuellt steg i testcykel avbryts, och nästa påbörjas.

Avbryt

Hela testcykeln avbryts.

Programmet återgår till startmenyn.

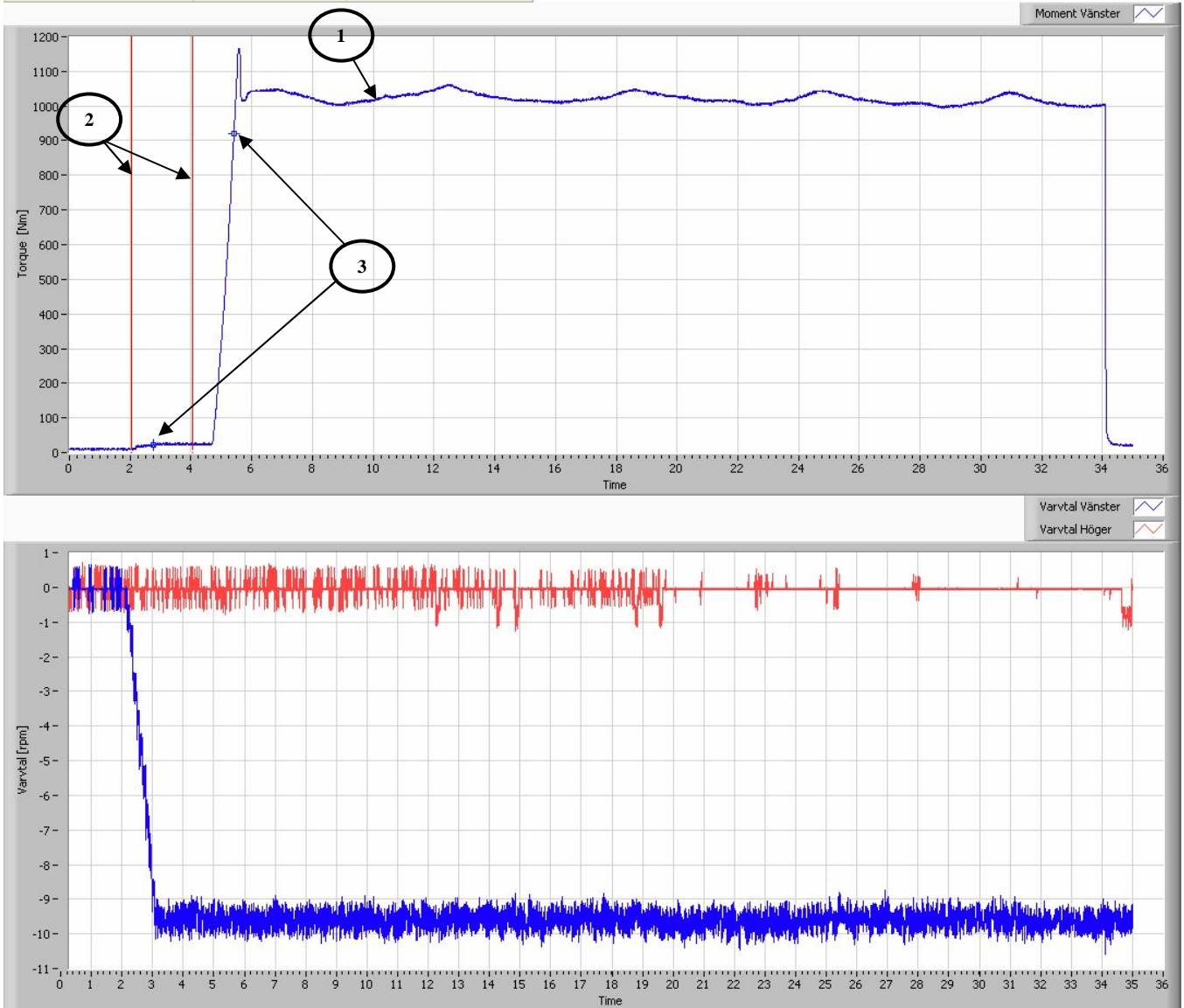


Fläkten inne i temperaturskåpet kommer att sätta igen sig med is om långa kylsekvenser körs, vid behov ta bort istäcket.

Rapport

Rapporten består av två delar, två diagram, det ena på momentkurvan och den andra på varvtalen på de två motorerna. Den andra delen är en tabell som visar vad det var för prov som kördes gränser specificerade i receptet, samt resultaten från automatutvärderingen. Längst ner står det vilket recept som kördes samt annan matnyttig information.

Kopplings-ID: 40059596 Temperatur: -20 °C Momentgivare: Vänster



	Varvtal vänster	Varvtal höger	Tidsintervall	Medelvärde	övre/undre-gräns	Max/Min	max/min-gräns	Δ Max/Min	Δ max/min-gräns	Rise/Fall-time	OK/NOK
Stabilisera varvtal på 10 varv/min	-10 rpm	0 rpm	2,037-4,057 s	24 Nm	50/0 Nm	28/19 Nm	50/0 Nm	9 Nm	50	0,734/-	OK
Time to Function TTD	-10 rpm	0 rpm	4,057-34,088 s	1021 Nm	1100/0 Nm	1062/994 Nm	1100/0 Nm	69 Nm	1100	1,361/-	OK

Dyno VI
 Operatör: Temp
 Recept: C:\Documents and Settings\LABBTR\Skrivbord\verification of initial samples.txt
 Loggföl: D:\Temp\Automatisk_Testsekvens_för_DynoIV\40059596_-20_deg C.lvm
 12:10:37 2008-05-28

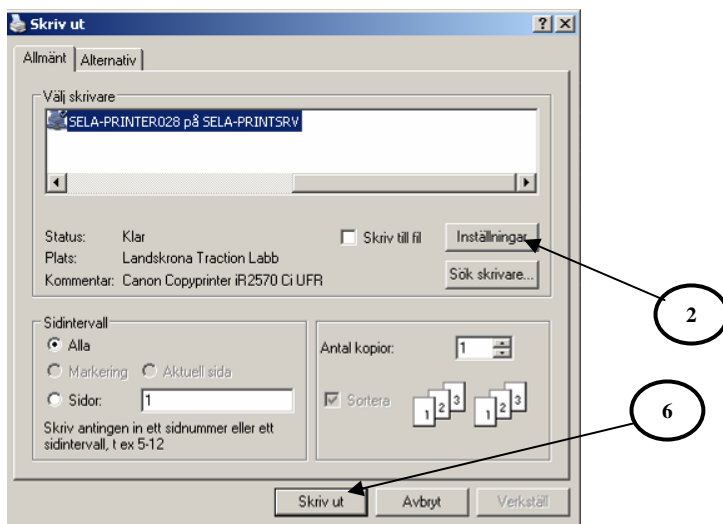
Figur 13, rapport

Momentgrafbeskrivning

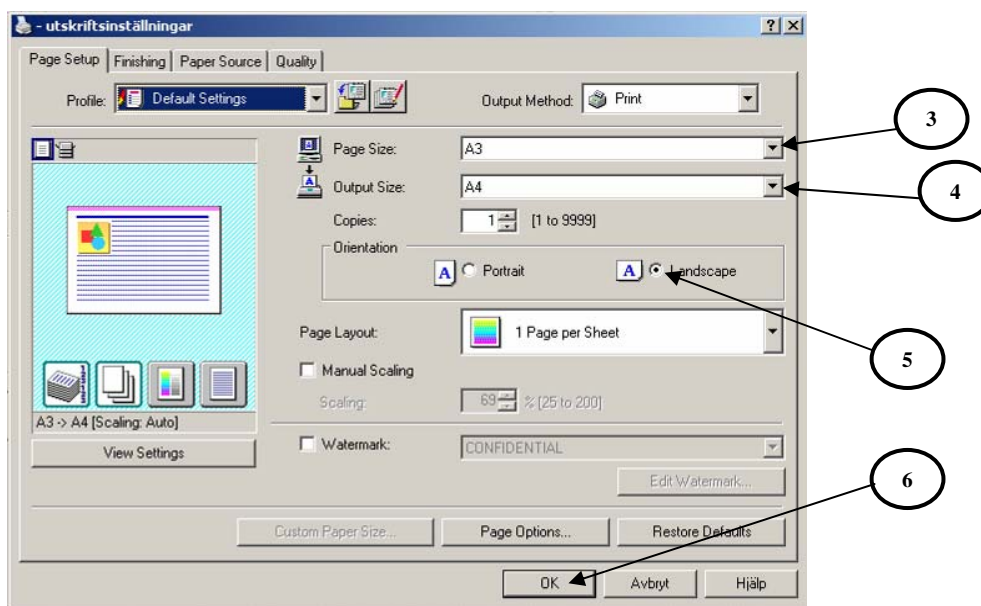
- 1) Momentkurva
- 2) Markeringar för när styrsignalen skickades till ECU:n, varje streck motsvarar en rad i tabellen.
- 3) Punkten där momentet har nått upp till 90% av momentsteget. Stig/fall-tiden räknas från det vertikala strecket till punkten. Medelvärdet av steget räknas ut en halv sekund efter punkten tills det att ett nytt steg kommer.

Utskrift av rapport

- 1) Tryck på utskriftsknappen i webbläsaren.
- 2) Tryck på ”Inställningar”.
- 3) Markera A3-format i ”Page Size:” rulllisten.
- 4) Markera A4-format i ”Output Size:” rulllisten.
- 5) Markera ”Landscape” i ”Orientation”.
- 6) Tryck ”OK” och ”Skriv ut”.



Figur 14, skriv ut



Figur 15, utskriftsinställningar

Felsökning

Tabell 1, Felkoder

Felkod:	Beskrivning:	Möjliga åtgärder:
5000	Ini-filen för kalibreringen innehåller inga värden	Fyll i kalibreringsvärdena i "kalibrering.ini".
5001	Det fanns inga vlv-filer i sökvägen	Infoga databaser och vlv-filer i sökvägen.
5002	Avbrutet	"Avbryt" har tryckts in av användaren.
5003	En eller flera temperaturer i receptet ligger utanför gränserna	Åtgärda i receptet.
5004	Fel på receptet	Kontrollera att rätt receptfil valts. Kontrollera att receptet är korrekt utformat.
5005	Höger momentgivare har inte startat	Starta om momentgivaren, batteriet kan vara slut.
5006	Vänster momentgivare har inte startat	Starta om momentgivaren, batteriet kan vara slut.
5007	Sätt på temperaturskåpet	Sätt på temperaturskåpet.
5008	Felmeddelande från temp.skåp: ---	Konsultera installations och bruksanvisningspärmen för temperaturskåp WT180/70, avsnitt 2, kap. 7.2, "felmeddelanden".
5009	Momentgivarna har stannat	Momentgivarna har "låst" sig, starta om momentgivarna.
5010	Återställ nödstopp	Motorerna har stannat under testsekvensen, återställ nödstoppet.
5011	Höger momentgivare gick inte att stoppa	Starta om momentgivaren.
5012	Vänster momentgivare gick inte att stoppa	Starta om momentgivaren.

Det kan förekomma ytterligare felmeddelanden utöver ovanstående. Försök i så fall åtgärda felet.

Skaparecept

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Inställningar	Temperatur [C]	Stabiliseringstid [min]														
2	Provnamn	ECU ON/OFF [True/False]	upprampningstid [s]	motor vänster	motor höger	stegtid	command	ManualControlValue1	ManualControlValue2	ESP	Gaspedal	EngineTorque	övre_grän_medel	undre_grän_medel	max_grän	min_grän	diff_max_min
3	inställningar		60	60													
4	Stabilisera varvtal 10 rpm	True		1	-10	0	2	5	0				12	0	12	0	12
5	Moment ansättning 100 Nm	True		1	-10	0	2	5	100				130	70	130	70	60
6	0 Nm	True		1	-10	0	2	5	0				12	0	12	0	12
7	Max Moment TTD	True		1	-10	0	10	5	1000				1100	0	1100	0	1100
8	inställningar		-20	10													
9	Stabilisera varvtal 10 rpm	False		1	0	10	2	10	0				12	0	12	0	12
10	Moment ansättning 100 Nm	True		1	0	10	2	10	100				130	70	130	70	60
11	0 Nm	True		1	0	10	2	10	0				12	0	12	0	12
12	Max Moment eLSD	True		1	0	10	10	10	1200				1320	0	1320	0	1320
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	

Figur 16 receptöversikt i Excel

Receptfilen är en tabbavgränsad textfil. Färgerna i figuren skall inte finnas i ett riktigt recept, utan finns endast av pedagogiska skäl i instruktionen.

Receptet inleds alltid med två rader, där det framgår vad kolonnerna skall innehålla, samt/alternativt annan information (se turkos ruta i Figur 16).

Det står receptmakaren fritt att fylla i vilken information han vill här, dock är det viktigt att det alltid är två rader.

En testsekvens inleds alltid med en rad som börjar med ordet ”inställningar” (se de gula cellerna på rad 3 i Figur 16). Här anges temperatur och stabiliseringstid.

Därefter följer en rad för varje momentsteg i sekvensen. Dessa rader är markerade med mörkgrått i Figur 16.

Då receptuppbyggnaden är modulär, kan receptmakaren bygga upp testcykeln genom att kombinera dessa element på valfritt sätt.

Testsekvensen som körs vid varje temperatur är här indelade i tre delar. Figur 17 (röd ruta) är styrningen av själva riggen, Figur 18 (blå ruta) är styrningen av kopplingen och Figur 19 (rosa ruta) används vid rapportutvärderingen.

När receptet är skapat ska det sparas till en textfil med filnamnstillägget .txt.

Nedan är receptet i Figur 16 indelat i tre delar för bättre översikt (se Figur 17, Figur 18 och Figur 19). I Tabell 2 och Tabell 3 står det beskrivet vad respektive kolon betyder.

Inställningar	Temperatur [°C]	Stabiliseringstid [min]			
Provnamn	ECU ON/OFF [True/False]	upprampningstid [s]	motor vänster	motor höger	stegtid
inställningar	60	60			
Stabilisera varvtal(10 rpm)	True	1	-10	0	2
Moment ansättning 100 Nm	True	1	-10	0	2
0 Nm	True	1	-10	0	2
Max Moment TTD	True	1	-10	0	10
inställningar	-20	10			
Stabilisera varvtal(10 rpm)	False	1	0	10	2
Moment ansättning 100 Nm	True	1	0	10	2
0 Nm	True	1	0	10	2
Max Moment eLSD	True	1	0	10	10

Figur 17, receptdel, riggstyrning

command	ManualControlvalue1	ManualControlValue2	ESP	Gaspedal	EngineTorque
5	0				
5	100				
5	0				
5	1000				
10	0				
10	100				
10	0				
10	1200				

Figur 18, receptdel, styrning av koppling

övre_grän_medel	undre_gräns_medel	max_gräns	min_gräns	diff_max_min
12	0	12	0	12
130	70	130	70	60
12	0	12	0	12
1100	0	1100	0	1100
12	0	12	0	12
130	70	130	70	60
12	0	12	0	12
1320	0	1320	0	1320

Figur 19, receptdel, rapportutvärdering

Tabell 2, beskrivning av inställningsparametrar

Kolonnamn	Kolonnamn i Excel	Beskrivning
inställningar	A	Viktigt att det står ”inställningar”. Varje recept måste starta med inställningar.
Temperatur	B	Temperaturen i °C -55°C → 120°C
Stabiliseringstid	C	Stabiliseringstid i minuter. Stabiliseringstiden räknar ner när temperaturen i kopplingen ligger ±1°C från börvärdet.

Tabell 3, beskrivning av inparametrar för momentsteget

Kolonnamn	Kolonnamn i Excel	Beskrivning
Provnamn	A	Här skrivs valfri information till det aktuella steget. Informationen visas i rapporten.
ECU ON/OFF	B	True → Spänningen till ECU:n slås PÅ False → Spänningen till ECU:n slås AV
upprampningstid	C	Upprampningstiden bestämmer hur många sekunder det ska ta för motorerna att komma upp i hastigheten specificerad i motor vänster, motor höger. Rampen kommer att avbrytas om tiden är längre än stegtiden.
motor vänster	D	Varvtal i [rpm]
motor höger	E	Varvtal i [rpm]
stegtid	F	Längden på steget i sekunder.
command	G	På vilket sätt ska kopplingen styras? 0 → Bilsignaler, kopplingen kommer att styras som om den statt i en bil. 2 → ström, [mA] 3 → tryck, [MPa] 5 → moment, [Nm] 6 → pump, [%] 9 → tryck & pump, [MPa]/[%] 10 → eLSD moment, [Nm] 11 → eLSD ström, [mA] 12 → eLSD Tryck, [MPa]
ManualControlvalue1	H	Här matas värdet på styrsignalen till kopplingen in, om command är 2, 3, 5, 6, 9, 10, 11 eller 12. Om command är 9 matas värdet på trycket in här.
ManualControlvalue2	I	Om command är 9 matas värdet till pumpen in här.
ESP	J	command måste vara satt till 0 0 → ESP Av 1 → ESP På

Installering av programvara

För att installera om programmet fråga Jonas Jönsson om installations CD:n. Kör det medföljande installationsprogrammet.

Det skapas genvägar till programmet på skrivbordet samt i startmenyn.

När installationen är klar skall programmet startas upp en gång för att sedan avslutas, detta för att kalibreringsfilen och inställningsfilen skall skapas.

Fyll i kalibreringsvärden i kalibreringsfilen.

Fyll i operatörsnamn och plats för sparade filer i inställningsfilen.

Sen är det bara att använda programmet.

Konfigurationsinställningar för DAQ:ar finns i rotkatalogen och importeras in via MAX. Filen heter CONFIGDATA.NCE.

Databaser för att kunna köra Epsilon 1, volvo EUCD samt PQ.mix skapas även de vid installationen.

Ett exempel på hur ett recept kan se ut kommer även det att skapas i rotkatalogen.

Manualen finns även den i rotkatalogen.

Kalibrering och inställningar

I rotkatalogen till programmet finns det två stycken filer, kallibrering.ini och inställningar.ini. I kalibreringsfilen fyller man i nya kalibreringsvärden. I inställningsfilen finns det möjlighet att fylla i namnet på olika operatörer, ändra katalogen för var mätdata ska sparas samt ändra reglerparametrar till temperaturregleringen.

```
inställningar.ini - Anteckningar
Arkiv Redigera Format Visa Hjälp
[Operatör]
1=Temp
2=Anton Sjunnesson
3=Oskar Lagerås
[Plats för sparade filer]
Sökväg=/C/
[Reglerparametrar]
proportional gain (Kc)=4,5000000000
integral time (Ti, min)=5,3000000000
derivative time (Td, min)=0,050000
```

Figur 21, inställningsfil

```
kalibrering.ini - Anteckningar
Arkiv Redigera Format Visa Hjälp
[[temperatur_1]
max=0,000000
min=0,000000
k=0,0000000000
m=0,0000000000
[temperatur_2]
max=0,000000
min=0,000000
k=0,0000000000
m=0,0000000000
[temperatur_3]
max=0,000000
min=0,000000
k=0,0000000000
m=0,0000000000
[temperatur_4]
max=0,000000
min=0,000000
k=0,0000000000
m=0,0000000000
[varvtal_vänster]
k=0,0000000000
m=0,0000000000
[varvtal_höger]
k=0,0000000000
m=0,0000000000
[tryck_1]
k=0,0000000000
m=0,0000000000
[tryck_2]
k=0,0000000000
m=0,0000000000
[tryck_3]
k=0,0000000000
m=0,0000000000
[tryck_4]
k=0,0000000000
m=0,0000000000
[moment_vänster]
k=0,0000000000
m=0,0000000000
[moment_höger]
k=0,0000000000
m=0,0000000000
[motor_vänster]
k=0,0000000000
m=0,0000000000
[motor_höger]
k=0,0000000000
m=0,0000000000
```

Figur 22, kalibreringsfil

