BEE05:08

IMPLEMENTATION AV CUSUM/SPRT-ALGORITM FÖR KVALITETSÖVERVAKNING I REALTID AV ROBOTSVETSNING

David Grunditz

Examensarbete
Teknologie kandidatexamen i elektroteknik med inriktning mot datateknik/Dataingenjör
(Degree of Bachelor of Electrical Engineering, with emphasis on Computer Engineering)

Blekinge Tekniska Högskola

Augusti 2005

Blekinge Tekniska Högskola
Sektionen för Teknik
Avdelningen för signalbehandling
Examinator: Docent Lars Håkansson
Handledare: Civ. Ing. Carl-Axel Ohlsson, Industriellt Utvecklingscentrum i Olofström AB

Copyright © 2005 David Grunditz

ABSTRACT

In the early 1990s, Volvo Car Corporation in Olofström started a project focusing on monitoring systems for robotized Gas Metal Arc (GMA) welding. This resulted in a research project where Stefan Adolfsson, Department of Production and Materials Engineering, Lund University/ Blekinge Tekniska Högskola, presented in 1998 the doctoral thesis, Automatic Quality Monitoring in GMA Welding using Signal Processing Methods. This doctoral thesis presented a Sequential Probability Ratio Test, SPRT, concept for quality monitoring automatic robotized GMA welding.

To create a cost efficient monitoring system the *Industriellt Utvecklingscentrum i Olofström AB* presented the idea of using a traditional PC computer as a system platform. A traditional PC computer offers high performance to relatively low cost.
The main task of this thesis is to implement the SPRT concept for monitoring automatic robotized GMA welding in the LabVIEW environment on a PC and evaluate the real time capacity. A second task in the thesis is to make a survey of similar monitoring systems for robotized GMA welding.

The final implementation monitors both mean and variance of the weld voltage and the weld current. In total four different SPRT concepts have been implemented. They are modifications of the SPRT concept that is presented in the doctoral thesis. Two of the SPRT concepts were developed during this exam thesis work and are intended to reduce the risk of false alarms caused by natural systematic variations.
Since four different SPRT concepts have been implemented and every single concept monitor both mean and variance of the weld voltage and the weld current, it means that there are in total 16 SPRT algorithms working in parallel.

The evaluation of the implementation shows that an ordinary PC is sufficient for real time monitoring and that only two of the four SPRT concepts are suitable.

The result of the market research of comparable monitoring systems indicates that there exist a small number of comparable welding monitoring systems.

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

INNEHÅLLSFÖRTECKNING

FÖRORD **9**

SAMMANFATTNING **11**

KAPITEL 1 – INLEDNING **13**

1.1 Bakgrund 13

1.2 Mål 14

1.3 Syfte 14

KAPITEL 2 – MARKNADSÖVERSIKT **15**

2.1 Marknadsöversikt I 15

2.2 Marknadsöversikt II 17

KAPITEL 3 – MATERIAL, METODER OCH VERKTYG **19**

3.1 GMA-svetsning 19

3.2 Utrustning för GMA-svetsning 19
3.2.1 Svetsaggregatet 20
3.2.2 Svetspistolen 20
3.2.3 Trådmatningsverk 21
3.2.4 Trådbibo och gasbehållare 21

3.3 Vad är det som händer vid svetsning GMA-svetsning? 21
3.3.1 Ljusbågen och materialövergång 22
3.3.2 Ljusbågen 22
3.3.3 Kortbåge 23

Copyright © 2005 David Grunditz

3.3.4 Spraybåge 24

KAPITEL 4 – LABVIEW 25

4.1 Funktionalitet 25

4.2 En applikation i LabVIEW 26

4.3 Runtime 29

KAPITEL 5 – SYSTEMÖVERBLICK 31

KAPITEL 6 – VAD SKALL ÖVERVAKAS VID MIG/MAG-SVETSNING 33

6.1 Vilken övervakningsprincip skall användas? 33

6.2 Hur erhålls de fyra storheterna? 33

6.3 Kvalitetsövervakning med CUSUM eller repeterad SPRT-algoritm 34

6.4 De implementerade övervakningsalgoritmerna 35
6.4.1 SPRT1 36
6.4.1.1 Bestämning av övervakningsparametrarna, m_0, v och σ^2 36
6.4.2 SPRT2 38
6.4.3 SPRT3 39
6.4.4 SPRT4 39

6.5 FIR-filter 40

6.6 Sammanfattning 41

KAPITEL 7 – BESKRIVNING AV WELDING QUALITY CONTROL SYSTEM:S PROGRAMKOD 43

7.1 Programkodshierarkin 43

7.2 Huvudmeny 44

7.3 Algoritm, DAQ 45
7.3.1 Funktion 45
7.3.2 Algoritmberäkningar 47
7.3.3 Realisering av SPRT-algoritmen 49
7.3.4 Gränssnitt mot användare 49

7.4 Algoritm, lab och Algoritm, demo 50
7.4.1 Funktion 50
7.4.2 Användargränssnitt 52

7.5 Ref. uppdatering 53
7.5.1 Funktion 53
7.5.2 Användargränssnitt 55

Copyright © 2005 David Grunditz

7.6 Visa grafer 56
7.6.1 Användargränssnitt 56

7.7 Inställningar 57
7.7.1 Funktion 57
7.7.2 Användargränssnitt 58

7.8 Trenddata 59
7.8.1 Funktion 59
7.8.2 Användargränssnitt 60

7.9 SGL till TXT 61
7.9.1 Funktion 61
7.9.2 Användargränssnitt 62

7.10 Om WQCS 63

KAPITEL 8 – TEST AV WELDING QUALITY CONTROL SYSTEM 65

8.1 Utförande 65
8.1.1 Svetsningen 66

8.2 Test av övervakningssystemet 67

8.3 Test av övervakningsalgoritmernas realtidsegenskaper 67
8.3.1 Utfall 67

8.4 Test av SPRT 2 67
8.4.1 Utfall 68

8.5 Utvärdering av SPRT3- och SPRT4-algoritmerna 69
8.5.1 Slutsats 73

REFERENSER 77

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

FÖRORD

Examensarbetet är utfört vid Industriellt Utvecklingscentrum i Olofström AB tillsammans med deras projektteam för automatiserad svetsövervakning och i samarbete med avdelningen för signalbehandling vid Blekinge Tekniska Högskola. Speciellt vilja jag tacka min handledare Civ. Ing. Carl-Axel Ohlsson vid Industriellt Utvecklingscentrum i Olofström AB och min examinator Docent Lars Håkansson, vid avdelningen för signalbehandling, Blekinge Tekniska Högskola för all hjälp och stöd som de gett under arbetets gång.

Andra personer som betytt mycket för detta arbetes genomförande är Dr. Stefan Adolfsson vid Campus Helsingborg, Lunds Universitet som ställde upp en hel dag med att hålla ett seminarium på Industriellt Utvecklingscentrum i Olofström AB kring sin doktorsavhandling där han beskrev algoritmen för mig och Industriellt Utvecklingscentrum i Olofström AB:s projektteam. Vidare så måste jag även tacka svetsgurun Civ. Ing. Ali Bahrami, Teknikcentrum, Kronoberg för gott samarbete vid utvecklandet av de två modifierade SPRT-algoritmerna och för all hjälp som han bistod med, bl.a. programmering av robot och svetsaggregat, vid testerna av övervakningssystemet.
Sist men inte minst måste även Industriellt Utvecklingscentrum i Olofström AB:s projektteam för automatiserad svetsövervakning tackas; Civ. Ing. Bengt Nilsson, Civ. Ing. Carl-Axel Ohlsson och Ing. Gösta Strandberg för ett gott och trevligt samarbete.

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

SAMMANFATTNING

Under början av 1990-talet inledde Volvo Personvagnar i Olofström en process där övervakning av robotiserad MIG/MAG-svetsning var i fokus. Det ledde fram till ett forskningsprojekt som resulterade i att Stefan Adolfsson, Lunds tekniska högskola/Blekinge tekniska högskola, 1998 presenterade avhandlingen Automatic Quality Monitoring in GMA Welding using Signal Processing Methods [5]. I den presenterades ett SPRT-baserat övervakningskoncept för övervakning av MIG/MAG-svetsning med kort- och spraybåge.

För att få fram ett kostnadseffektivt övervakningssystem uppkom tanken att använda vanliga PC-datorer som plattform. De erbjuder hög beräkningskapacitet till förhållandevis låg kostnad. Examensarbetets mål är att implementera det SPRT-baserade övervakningskonceptet i LabVIEW på en PC och utvärdera realtidsegenskaperna. Till detta tillkom även uppgiften att kartlägga motsvarande övervakningssystem på marknaden.

Den färdiga implementationen övervakar svetsspänningens och svetsströmmens medelvärde respektive varians. Totalt har fyra olika SPRT-algoritmkoncept implementerats där alla är modifierade i förhållande till den som presenteras i ovan nämnda avhandling. Två av algoritmkoncepten är framtagna under detta arbetes gång och är avsedda för att minimera antalet falsklarm orsakade av naturliga systematiska variationer.
Eftersom fyra olika SPRT-algoritmkoncept har använts för att övervaka svetsspänningens och svetsströmmens medelvärde respektive varians medför det att totalt 16 SPRT-algoritmer har implementerats vilka arbetar parallellt.

Utvärderingen av implementationen visar att en vanlig PC-dator är tillräcklig för realtidsövervakning. Vidare visar utvärderingen också att endast två av SPRT-algoritmkoncepten är lämpliga att använda för svetsövervakning.

Marknadskartläggningen av övervakningssystem för MIG/MAG-svetsning fann ett mindre antal system som kan konkurrera.

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

KAPITEL 1 – INLEDNING

1.1 Bakgrund

I dagens svetsverklighet är den svenska robotsvetsningen utsatt för konkurrens från handsvetsare i låglöneländer och det är mycket svårt för den svenska robotsvetsningen att svara upp mot konkurrensen. Detta gäller då svetskostnaden mäts i kronor per lagd svetsmeter alternativt kronor per svetsgodsvolym och inte i förhållande till kvalitet [1]. Genom att optimera svetsprocessen för att nå en så låg kostnad som möjligt och samtidigt vidta kvalitetshöjande återgärder, kan detta konkurrera med den manuella svetsaren i låglönelandet.
En metod, som ökar kvalitén på levererade produkter, kan vara att kvalitetstesta varje enskild detalj. För detta ändamål finns det ett antal olika metoder.
Förstörande provning är en metod, som innebär en mekanisk process, som arbetar sig igenom svetsfogen för att undersöka kvalitén. Problemet är att då kvalitetsdiagnosen har gjorts är svetsfogen förstörd. I praktiken är det därför inte möjligt att använda denna metod för att säkerställa 100 %-ig kvalitet på varje detalj.
En annan metod, som kan vara lämpligare, är att använda ett realtidsövervakningssystem, som övervakar varje svetsfog under svetsprocessen.

I början av 1990-talet inleddes en process på Volvo Personvagnar i Olofström där svetsövervakningssystem för robotiserad MIG/MAG-svetsning var i fokus. Denna process ledde fram till ett forskningsprojekt vars syfte var att finna en metod för realtidsövervakning av robotiserad MIG/MAG-svetsning. År 1998 presenterade Stefan Adolfsson, Lunds Tekniska Högskola/Blekinge Tekniska Högskola, ett SPRT- (Sequential Probability Ratio Test) baserat övervakningskoncept i avhandlingen Automatic Quality Monitoring in GMA Welding using Signal Processing Methods (se referens 5).

För att få fram ett kostnadseffektivt SPRT-baserat övervakningssystem kan man kanske utgå från en PC-plattform. PC-datorer idag är mycket kostnadseffektiva med avseende på den beräkningskapacitet som erbjuds. Därför skulle det vara intressant att undersöka om det fanns möjlighet att använda dem för ett realtidsövervakningssystem. Detta examensarbete skall testa om denna möjlighet kan bli verklighet.

Copyright © 2005 David Grunditz

1.2 Mål

Examensarbetet har två delmål. Det ena är att implementera en SPRT-baserad övervakningsalgoritm för kvalitetssäkring av MIG/MAG-svetsning i LabVIEW på en vanlig PC under operativsystemet Windows och utvärdera realtidsegenskaperna. Det andra är att kartlägga likartade övervakningssystem som finns på marknaden.

1.3 Syfte

Syftet med arbetet är att ta fram material som kan användas som underlag för att kunna avgöra om ett SPRT-baserat övervakningssystem skall baseras på en PC eller ej. Genom en kartläggning av likartade system erhålls kännedom om konkurrerande system på marknaden.

Copyright © 2005 David Grunditz

KAPITEL 2 – MARKNADSÖVERSIKT

2.1 Marknadsöversikt I

För att känna till vilka konkurrenter som finns på marknaden för svetsövervakningssystem för kort- och spraybåge har en marknadsöversikt gjorts. Marknadsöversikten har gjorts genom att använda sökmotorer på Internet. De sökmotorer som använts är Google, Yahoo och Alta Vista. De sökord och sökordskombinationer som använts är:

- ”gas metal arc welding” + monitor
- ”gas metal arc welding” + quality + monitor + PC

Nedan följer en lista på övervakningssystem som hittats på Internet och som kan tänkas vara konkurrenter till ett SPRT-baserat övervakningssystem. Systemens tillförlitlighet går ej att avgöra genom att studera internetsidor, utan är något som måste provas i praktiska tester och utvärderas.

ArcSentry

Tillverkare:	Native American Technologies
Land:	U.S.A.
Hemsida:	www.natech-inc.com
Produktnamn:	ArcSentry
Beskrivning:	Samlar in spänning, ström, gasflöde, trådmatningshastighet och erbjuder upp till 42 extra kanaler för extra givare. Samplar med en frekvens på upp till 20 kHz och behandlar samplad data i tidsfönster på 50 ms. Med en P/NA3 neuralt nätverk kan systemet beräkna olika svetsdata som jämförs med olika svetsgränser som t.ex. weld size, leg lengths, actual or effective throat, inträngning mm. Kan kopplas till robot eller PLC. Systemet är ett PC-baserat system och är uppbyggt med givarhubbar dit givare är kopplade. Till varje PC är det möjligt att koppla upp till fyra givarhubbar. PC:n är utrustad med TMS320C6201-kort från Texas Instruments. Systemet

Copyright © 2005 David Grunditz

visar resultatet på datorns bildskärm samt sparar dess data på hårddisken. Operativsystem: Windows NT

ArcSentry-datorerna är kopplas till ett Factory Information System nätverk ditt resultat skickas via TCP/IP.

Övervakningsprincipen har utvecklas sedan mitt en av 1980-talet då det första övervaknings systemet togs fram.

WeldPrint

Tillverkare:	Welding Technology Innovations Pty Ltd
Land:	Australien
Hemsida:	www.wti.com.au
Produktnamn:	WeldPrint
Beskrivning:	Analyserar svetskvalitén i realtid och signalerar svetsproblem omedelbart när det uppkommer. Systemet analyserar spänning och/eller ström för att upptäcka problem.
Kommentar:	Kan vara en svår konkurrent till ett SPRT-baserat övervakningssystem.

ArcAgent

Tillverkare:	Impact Engineering
Land:	U.S.A.
Hemsida:	www.impactwelding.com
Produktnamn:	ArcAgent 2000, ArcAgent 3000, ArcClient
Beskrivning:	ArcAgent är ett realtidsövervakningssystem för GMA-svetsning avsett för både helautomatiserad och halvautomatiserad svetsning. Systemet är uppbyggt kring en inbyggd mikrokontroller. Systemet kan användas för upptäcka flera olika svetsproblem bl.a. bristande inträning, jordningsproblem, variationer i skyddsgasen, mm.
Kommentar:	Systemet ser mycket professionellt ut

ADM IV (Arc Data Monitor) mm

Tillverkare:	Computer Weld Technology, Inc
Land:	U.S.A.
Hemsida:	cweldtech.com
Produktnamn:	ADM IV samt flera andra övervakningsprodukter
Beskrivning:	ADM IV är ett övervakningssystem för bågsvetsning
Övrigt:	Computer Weld Technology, Inc tillverkar ett stort antal olika produkter för svetsövervakning och styrning svetsprocessen.

Copyright © 2005 David Grunditz

2.2 Marknadsöversikt II

I ett PC-baserat mät eller övervakningssystem är datainsamlingskortet eller mätkortet en viktig komponent. För att trygga tillgången på mätkort, i fall ett nytt måste införskaffas för att ersätta ett trasigt, är det viktigt att känna till marknaden för sådana kort, ifall huvudleverantören skulle lägga ner tillverkningen eller gå i konkurs.

Vilka andra tillverkare finns, går korten att använda tillsammans med LabVIEW osv. är frågor som måste beaktas vid en markandsundersökning. Marknadsundersökningen är gjord genom att besöka olika tillverkares hemsidor på Internet och där söka efter kort som kan användas tillsammans med LabVIEW och kan användas som ersättning till det nuvarande kortet (PCI-MIO-16E-4).

Det kort som används vid implementeringen av SPRT-algoritmen är PCI-MIO-16E-4 (NI 6040E) från National Instruments. Det är ett 16 kanaligt kort med 12 bitars upplösning på A/D-omvandlaren och en maximal samplingsfrekvens på 250 kHz då 16 kanaler används och 500 kHz då en kanal används.

Det SPRT-baserade övervakningssystemet använder endast 5 av de 16 kanalerna och de används differentiellt. Samplingsfrekvensen är 8192 Hz. Så ett ersättningskort behöver endast ha fem differentiella kanaler och klara en samplingsfrekvens på 8192 Hz.

Tillverkare:	**National Instruments**
Hemsida:	www.ni.com
Modell:	E-serien
Kommentar:	E-seriens enklare kort saknar analog triggning vilket innebär anpassning av programkoden för digital triggning samt modifierad inkoppling av triggningssignalen.

Tillverkare:	**Iotech**
Hemsida:	www.iotech.com
Modell:	DaqBoard 2000
Kommentar:	

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

KAPITEL 3 – MATERIAL, METODER OCH VERKTYG

3.1 GMA-svetsning

GMA-svetsning (Gas Metal Arc svetsning) eller MIG/MAG svetsning är en vanlig metod för att sammanfoga plåtdetaljer inom t.ex. verkstadsindustrin. Det är en smältsvetsmetod där den erforderliga värmen, som behövs för att smälta och sammanfoga plåtdetaljer, tillförs via en ljusbåge. Samtidigt tillförs ett metalliskt tillsatsmaterial i form av en wire, elektrodwire, och hela svetsprocessen skyddas av gas. Svetsmetoden tillhör gruppen bågsvetsning.

3.2 Utrustning för GMA-svetsning

Den utrustning som behövs för GMA-svetsning är svetsaggregat, svetspistol och trådmatningsverk. Där utöver behövs gasbehållare och trådbibo. Bild 3.1 visar en komplett robotcell för helautomatisk svetsning.

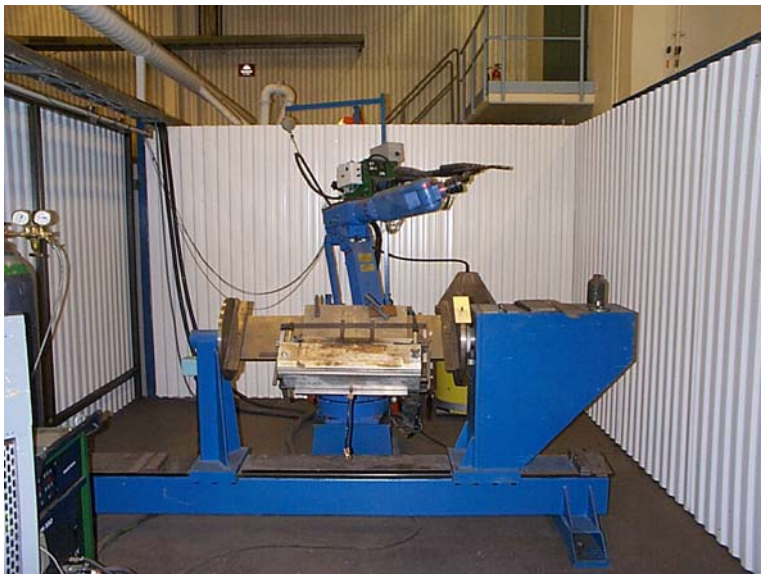


Bild 3.1 – Robotcell för GMA-svetsning

Copyright © 2005 David Grunditz

3.2.1 Svetsaggregatet

Svetsaggregatet är en enhet som består av strömkälla och styrenhet. Aggregatet omvandlar el från nätet till el för svetsning. Normalt kan den leverera en ström på ca 400 - 450 A, men det finns svetsaggregat som kan leverera högre ström [2, 3]. Tomgångsspänningen, den potentialskillnad som finns mellan anod och katod när ingen svetsning pågår, ligger mellan 15 och 70 V [2].
En viktig parameter för strömkällor är strömkällans strömkarakteristik. Detta är en graf som visar spänningen i förhållande till strömmen. Svetsaggregatet kan ha en strömkarakteristik-kurva som är svagt avtagande, svagt stigande eller horisontell [3].

Det finns olika sätt att bygga svetsaggregat för GMA-svetsning. De är byggda kring dioder, tyristorer eller transistorer, som genererar likströmmen [2]. Utöver de vanliga svetsaggregaten finns det svetsaggregat för svetsning med pulserad båge. Här pulsar svetsaggregatet svetsströmmen i stället för att låta "fysikens lagar" sköta pulsningen som är det normala vid GMA-svetsning.

Svetsaggregatets minuspol kopplas till den plåtdetalj som skall svetsas. Plåtdetaljen kallas även för arbetsstycke. Svetsaggregatets pluspol kopplas till svetspistolen.

3.2.2 Svetspistolen

Svetspistolen är det verktyg som sitter monterad på robotarmen eller som svetsaren håller i handen under svetsningen. Se bild 3.2.

(a) Svetspistol monterad på robotarm

(b) Manuell svetspistol

Bild 3.2 - Svetspistol

Den förs fram strax ovanför arbetsstycket under svetsning. Svetspistolen består av gaskåpa och kontaktmunstycke. Se bild 3.3.

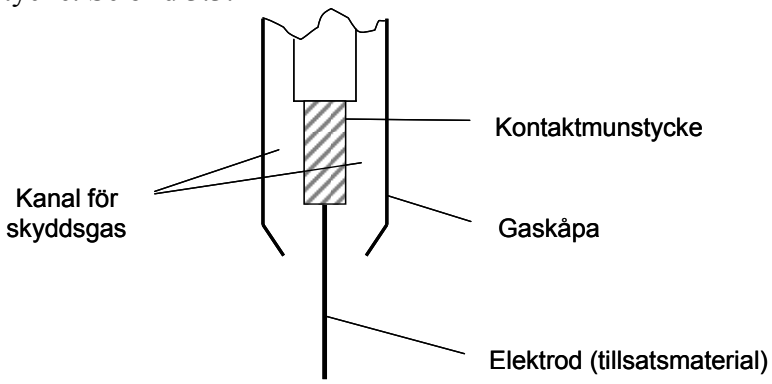


Bild 3.3 - Schematisk bild över främre delen av svetspistolen

Copyright © 2005 David Grunditz

Uppgiften för svetspistolen är att styra elektrodwiren, sörja för god elektrisk kontakt för elektrodwiren samt styra skyddsgasen. Vid svetsning ligger temperaturen på mellan 5 000°C och 7 000°C och denna höga temperatur kräver att svetspistolen kyls. Detta kan ske antingen med den omgivande luften eller med vatten som leds fram till svetspistolen.
Svetspistolen är kopplad till en special kabel som går till svetsaggregatet. I kabeln finns en ledare för ström, en trådledare som leder elektrodwiren och eventuellt slangar för kylvatten för kylning om svetspistolen kyls med vatten.

3.2.3 Trådmatningsverk

Trådmatningsverket matar kontinuerligt fram elektrodwiren i en takt som skall motsvara wirens nedsmältningshastighet. Det finns tre olika principer för trådmatning, skjutande, dragande samt Push-Pull. Skjutande matning innebär att matarverket sitter närmare trådbibon än vid dragande. Vid dragande trådmatning sitter matarverket nära svetspistolen, eller i svetspistolen. Push-Pull är en kombination av skjutande och dragande trådmatning med två matarverk, ett vid trådbibon och ett vid eller i svetspistolen.

3.2.4 Trådbibo och gasbehållare

Två andra viktiga komponenter i GMA-svetsutrustningen är trådbibon samt gasbehållaren. Trådbibon kan beskrivas som en trådrulle där elektrodwiren är upprullad och från denna matas wiren kontinuerligt till svetspistolen.
Gasbehållaren innehåller skyddsgasen som tillförs vid GMA-svetsning. Det kan vara en gasflaska vid svetsaggregatet men även en gemensam tank på en industri dit flera svetsaggregat är kopplade.

3.3 Vad är det som händer vid svetsning GMA-svetsning?

Vid GMA-svetsning sammanfogas två eller flera plåtdetaljer genom att ett begränsat område värms upp, smälter och flyter samman. Värmeenergin tillförs från en extern energikälla t.ex. ett svetsaggregat via en ljusbåge. Uppvärmningen påverkar även plåtdetaljerna utanför det smälta området. Området som påverkas av den tillförda värmeenergin delas upp i olika zoner beroende på hur stor påverkan varit [2]: smältzon, HAZ (Heat Affected Zone) och grundmaterial, se bild 3.4.

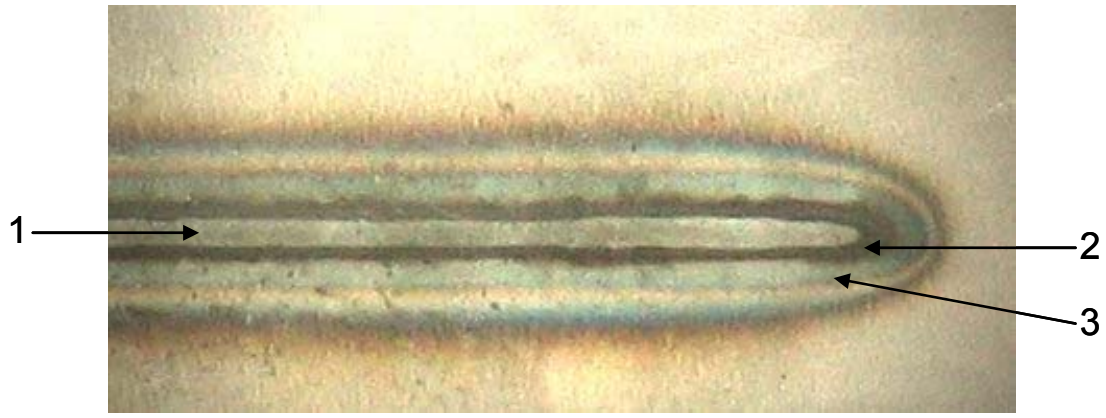


Bild 3.4 – De av svetsning påverkade zonerna. 1. smältzon, 2. HAZ (Heat Affected Zone), 3. grundmaterial.

Gemensamt för smältzonen och HAZ är atomstrukturen får en bestående förändring genom den tillförda energin.

Copyright © 2005 David Grunditz

Smältzonen är den zon där all metall, plåtmaterial och tillsatsmaterial, blir flytande och flyter samman. Området kallas i stelnat tillstånd för svetsgods, men även svetsfog.

HAZ är den värmepåverkade zonen. Området smälter inte, men plåtmaterialets atomstruktur förändras av värmen.

Grundmaterialområdet är det yttersta värmepåverkade området där atomstrukturen blir opåverkad av svetsningen.

En faktor som har en påverkan på det slutliga svetsresultatet är avkylningen. Kvalitén på svetsgodset är direkt beroende på hur snabbt avkylningen sker.

3.3.1 Ljusbågen och materialövergång

Vid all bågsvetsning tillförs värme via en ljusbåge mellan elektroden och arbetsstycket. Värmen smälter elektroden, som i droppform övergår till arbetsstycket. Vid GMA-svetsning är ljusbågen och materialövergången kopplade till varandra genom att bågen påverkar nedsmältningen och därmed materialövergången. De två vanligaste bågformerna är kort- och spraybåge och det är för dem som SPRT-algoritmen är avsedd.

3.3.2 Ljusbågen

Precis som när det åskar uppstår en ljusbåge då potentialskillnaden är tillräckligt stor mellan två punkter. Dessa punkter kallas för anod resp. katod. Anoden kopplas till svetsaggregatets pluspol och katoden till aggregatets minuspol. Vid katoden bildas elektroner som strävar efter att förflytta sig mot anoden. När elektroderna börjar förflytta sig så krockar de med atomer i skyddsgasen, som omger svetsprocessen. Vid krockarna lämnar andra elektroner atomerna och en kedjereaktion skapas. Utöver elektronerna förflyttar sig även positiva resp. negativa joner, som är mer tröga i förflyttningen. De positiva jonerna rör sig mot katoden medan negativa jonerna mot anoden.

Ljusbågen är indelad i tre olika områden, katodområde, ljusbågspelare och anodområde. I de tre olika områdena är spänningsfallet olika. I det stora området, ljusbågspelaren, är spänningsfallet litet. Spänningsfallet i ljusbågspelaren är proportionellt mot ljusbågspelarens längd. Däremot i anod resp. katod områdena är spänningsfallet desto större, se bild 3.5.

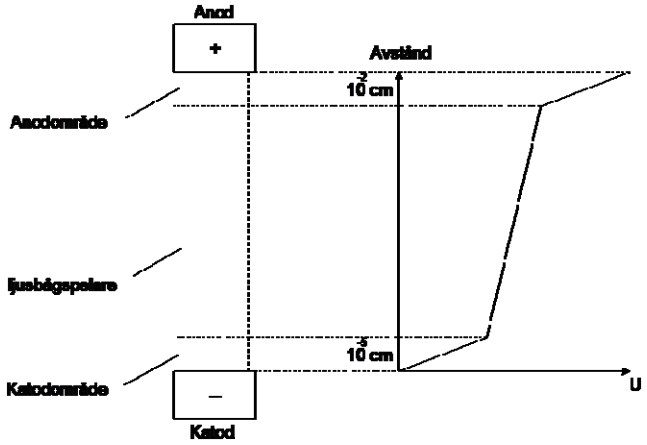


Bild 3.5 - Schematisk bild på spänningsfallet i en ljusbågens olika område

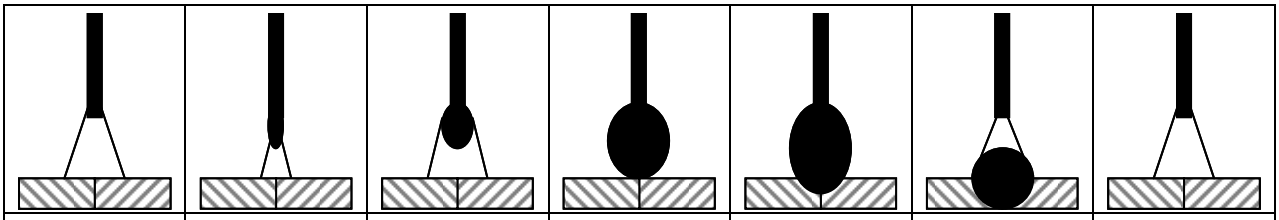
Ljusbågens färg är beroende av vilken gas som används och vilken frekvens som den elektromagnetiska strålningen har. Ljusbågens temperatur ligger mellan 5 000°C och 7

Copyright © 2005 David Grunditz

000°C. Ljusbågen tänds genom att elektroden ställs i fysisk kontakt med arbetsstycket. Då värms elektrodspetsen upp och smälter. När den smälta elektrodspetsen flyter ut försvinner den fysiska kontakten mellan elektroden och arbetsstycket och bågen är tänd. Är avståndet mellan anod och katod för stort dör bågen.

3.3.3 Kortbåge

Kortbågssvetsning används vid svetsning av tunnare plåt. Vid kortbågssvetsning är strömmen på mellan 50 – 200 A och svetsspänningen ligger på 15 – 20 V. Tändnings- och släckningsfrekvensen är mellan 50 – 200 Hz.
I kortbågens svetsprocess kortsluts strömkretsen regelbundet vilket innebär att ljusbågen släcks och tänds.
När ljusbågen är tänd värms elektroden upp så att en droppe bildas på elektrodspetsen. När droppen växt tillräckligt kommer den att vara i kontakt med både elektroden och arbetsstycket och bågen släcks. Strömmen ökar, droppen växer tills den snörps av och lämnar elektrodspetsen. När droppen snörps av tänds åter ljusbågen.
Elektrodens nedsmältning och övergång till arbetsstycket kallas för materialövergång. Bild 3.6 visar schematiskt hela svetsprocessen och bild 3.7 visar spänningen vid kortbågssvetsning.



Ljusbågen är tänd	Elektrod-spetsen börjar smälta	En droppe bildas och växer	Droppen kommer i kontakt med arbetsstycke och ljusbåg-en släcks	Droppen växer och snörps till sist av	Ljusbågen tänds igen	Ljusbågen är tänd

Bild 3.6 – Materialövergången vid kortbågssvetsning.

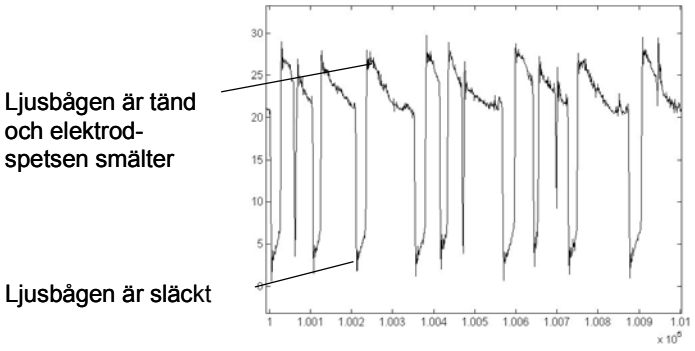


Bild 3.7 – Svetsspänningen vid kortbågssvetsning

Copyright © 2005 David Grunditz

3.3.4 Spraybåge

Spraybåge kan användas vid svetsning av material som är över 5 mm. Svetsströmmen vid svetsning med spraybåge är normalt mellan 250 och 450 A och spänningen är normalt mellan 30 och 35 V.

Svetsprocessen för spraybåge är en process där bågen är stabil och elektrodens materialövergång sker i droppform, då ljusbågens värmetillförsel är tillräckligt stor. Dropparna har en diameter som är mindre än diametern på elektrodwiren vilket innebär att ingen kortslutning sker. Eftersom dropparna är små kan man säga att materialövergången sker i sprayform. Se bild 3.8.

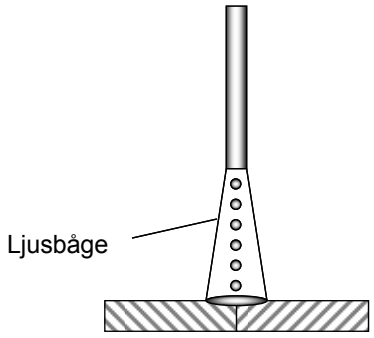


Bild 3.8 – Materialövergång vid spraybågssvetsning

Copyright © 2005 David Grunditz

KAPITEL 4 – LABVIEW

LabVIEW är en grafisk laborations- och utvecklingsmiljö för mät- och styrapplikationer. Miljön är utvecklad av National Instruments och har funnits på marknaden i många år. Idag användas miljön inom många områden som telekom-, elektronik- och halvledarindustri, flyg- och bilindustri men även inom biomedicinsk industri och den akademiska forskningsvärlden [4].
LabVIEW är uppbyggt på ett enkelt och funktionellt sätt vilket gör att icke professionella programmerare, som t.ex. forskare, kan använda miljön för att mäta och styra. Enkelheten beror på att den grafiska miljön använder grafiska programmeringssymboler som de flesta tekniker och forskare känner igen, se bild 4.1.

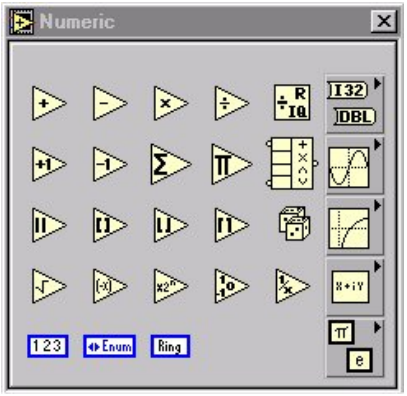


Bild 4.1 – Grafiska programmeringssymboler

4.1 Funktionalitet

LabVIEW är avsett för datorer som använder Microsoft Windows, Mac OS eller Unix/Linux. Miljön används mest under Microsoft Windows på vanliga PC-datorer, industridatorer eller PC-baserade PXI- eller VXI-system.
För ett komplett mät- och styrsystem behövs hårdvara för att mäta och styra. Vanligast är insticksskort, men även mätinstrument, t.ex. oscilloskop och funktionsgenerationer, som kommunicerar via GPIB eller LAN kan användas, se bild 4.2.

Copyright © 2005 David Grunditz

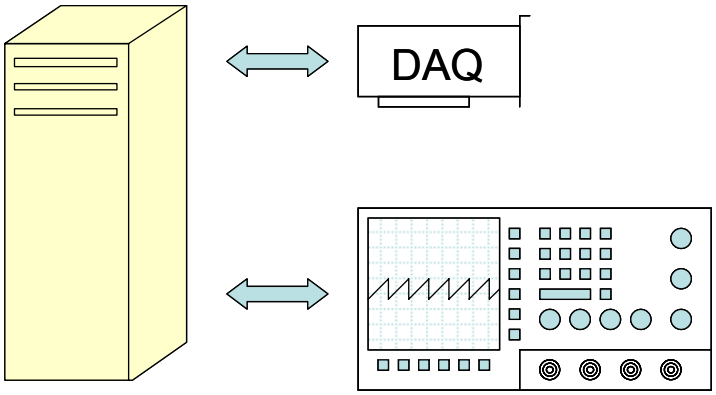


Bild 4.2 – PC-baserade mätsystem använder instickskort eller mätinstrument för att mäta och styra.

I LabVIEW finns det funktioner för bland annat:

- mjukvarubaserad signalkonditionering.
- analys av ljud, vibrationer och bild genom DSP-funktioner.
- kommunikation, t.ex. fjärrmättning över Internet.
- maskinvision.

Dessutom finns OPC-gränssnitt (OLE for Process Control) för kommunikation med styrsystem som PLC. För kommunikation och styrning av mätinstrument finns ett stort utbud av drivrutiner till mätinstrument från de flesta tillverkare som t.ex. Agilent och Rohde & Schwartz.

4.2 En applikation i LabVIEW

När en applikation utvecklas arbetar användaren med två fönster: diagramfönster och frontpanelfönster, se bild 4.3. Diagramfönstret används för programkod medan frontpanelfönstret är gränssnittet mot omvärlden. Tillsammans utgör fönstrena en VI, ett virtuellt instrument. En applikation kan bestå av en eller flera VI:er.

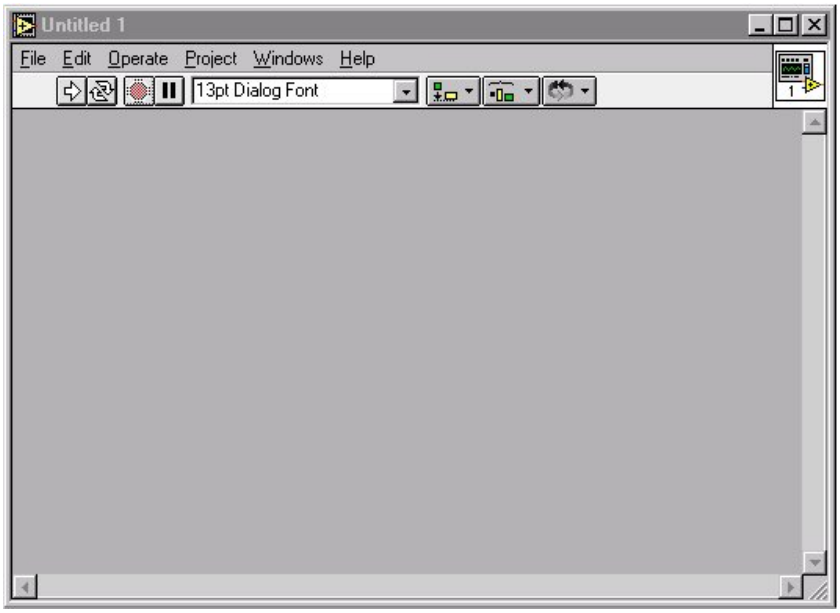


Bild 4.3 (a) – frontpanelfönster

Copyright © 2005 David Grunditz

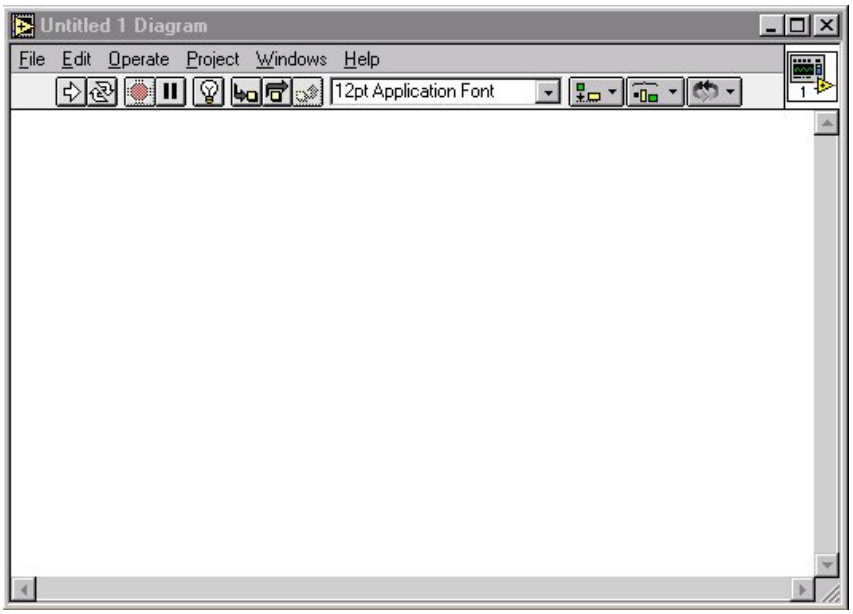


Bild 4.3 (b) – diagramfönster

En VI fungerar och används precis på samma sätt som en funktion i ett traditionellt programspråk. VI:en kan köras separat eller vara en del i ett större sammanhang där andra VI:er anropar den.

Som tidigare nämnts är LabVIEW ett grafiskt programspråk vilket innebär att varje VI har en grafisk representation som används vid anrop av VI:en. Den grafiska representationen finns uppe till höger i både diagramfönstret och i frontpanelsfönstret, se bild 4.4.

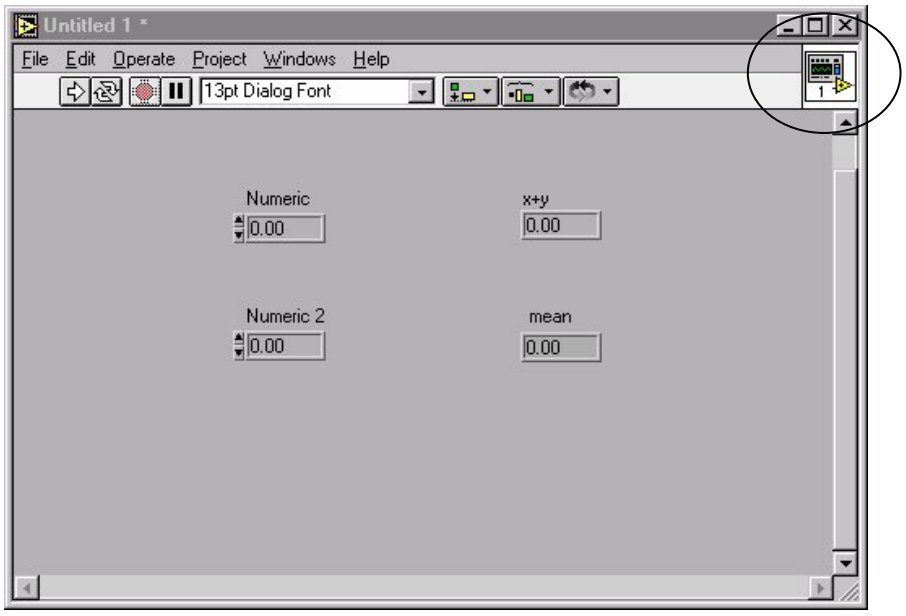
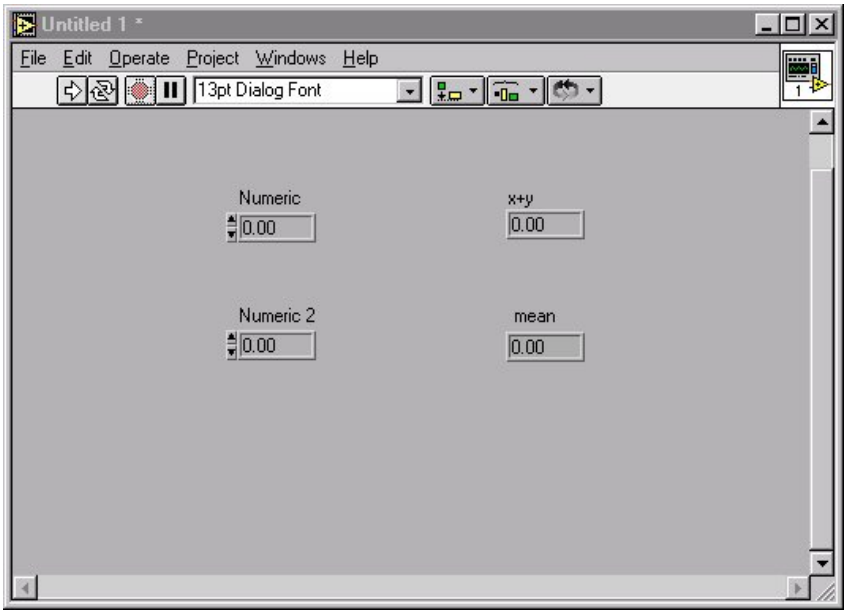


Bild 4.4 – VI:en grafiska representation

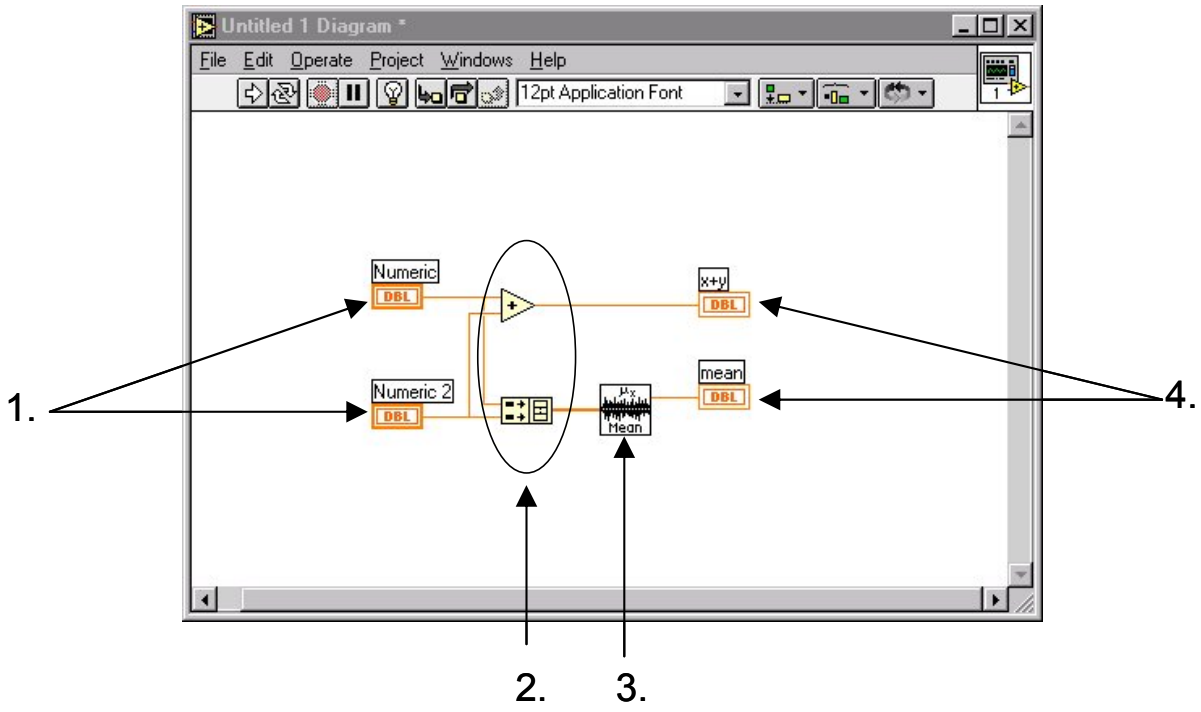
Den grafiska representationen designas av applikationsutvecklaren på ett sådant sätt att den går att skilja från andra VI:ers grafiska representation.

Copyright © 2005 David Grunditz

I en enkel LabVIEW applikation som adderar två tal och beräknar medelvärdet på talen används LabVIEWs olika programmeringsobjekt. Bild 4.5 visar VI:ens båda fönster och i bild 4.5 (b) finns den grafiska koden.



(a) – frontpaneldiagram



(b) – diagramfönster

Bild 4.5 – Enkel applikation som adderar och beräknar medelvärde

I diagramfönstret, bild 4.5 (b), finns fyra objekt (1 och 4) för kommunikation med omvärlden. De är VI:ens in- och utmattningsobjekt. Objektens gränssnitt mot omvärlden finns i frontpanelfönstret, bild 4.5 (a).

Copyright © 2005 David Grunditz

Värden som inmatningsobjekten tar emot leds till två grafiska objekt (2) som i C kallas nyckelord. Dessa objekt är inga VI:er med egen kod, utan är en del av programspråkets operationskommandon. Det översta av de två objekten adderar ihop det som kommer från inmatningsobjekten och det nedre objektet placerar det som kommer från inmatningsobjekten i en vektor, som i sin tur står i förbindelse med ett VI som beräknar medelvärdet (3). Medelvärdes-VI:en är en sub-VI som anropas först då programflödet når medelvärdesberäkningens grafiska objekt. Resultaten från additionen och medelvärdesberäkningen går till utmatningsobjekten.

Eftersom applikationen anropar medelvärdes-VI:en så har applikationen en hierarki av VI:er, se bild 4.6. Det som visas är de ingående VI:ernas inbördes relation. Motsvarande VI hierarki för övervakningssystemet ses i bild 7.1.

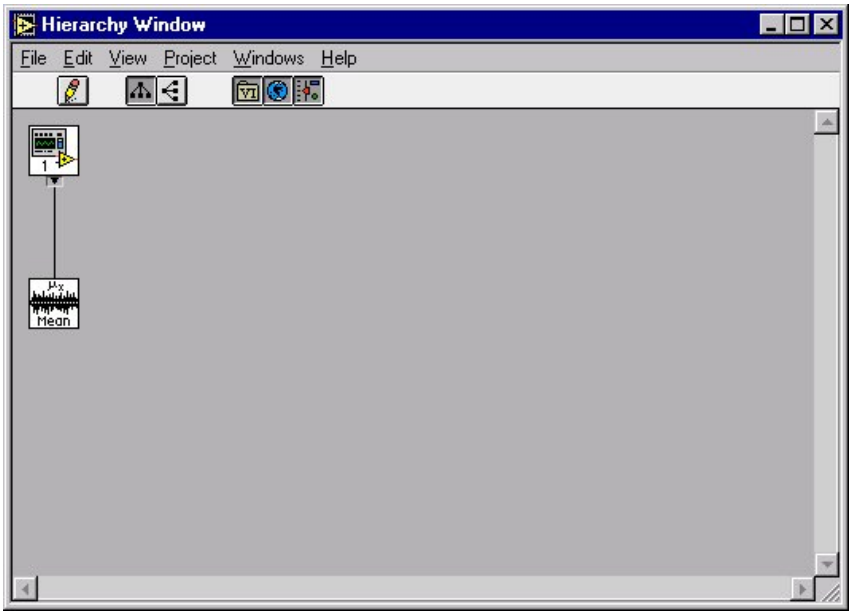


Bild 4.7 – Den enkla VI:ens VI-hierarki.

4.3 Runtime

En färdigutvecklad applikation som skall distribueras till flera användare bör vara lätt att distribuera och installera. En applikation består ofta av flera egna VI:er och skall applikationen distribueras på VI-nivå krävs att varje dator som skall köra applikationen har LabVIEW installerat. Detta kan bli väldigt dyrt om det handlar om flera datorer.
För att på ett enkelt sätt distribuera LabVIEW applikationer, utan krav på att slutanvändaren har LabVIEW installerat, finns funktionen Run Time Builder[1]. Det är ett verktyg som skapar runtime applikationer av VI-koden. Runtime applikationer är enkla att göra, enkla att distribuera och kräver inte att LabVIEW finns installerat på slutanvändarens dator. Runtime installationen på slutanvändarens dator går till på samma sätt som installationen av vanliga Windows program.

1. Funktionen är inte standard i alla versioner LabVIEW, men det är möjligt att köpa till funktionen i efterhand

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

KAPITEL 5 – SYSTEMÖVERBLICK

Svetsrobotcellen på Industriellt Utvecklingscentrum i Olofström AB, se bild 5.1, som använts för test av det PC-baserade övervakningssystemet är uppbyggd av en robot från japanska Motoman och ett svetsaggregat från danska Migatronic. Cellens uppbyggnad beskrivs schematiskt i bild 5.2. För att skydda personal finns dessutom en ljusridå som stoppar svetsningen då någon bryter ljusbågen.

Bild 5.1 – Svetsrobotcellen på Industriellt Utvecklingscentrum i Olofström AB

Copyright © 2005 David Grunditz

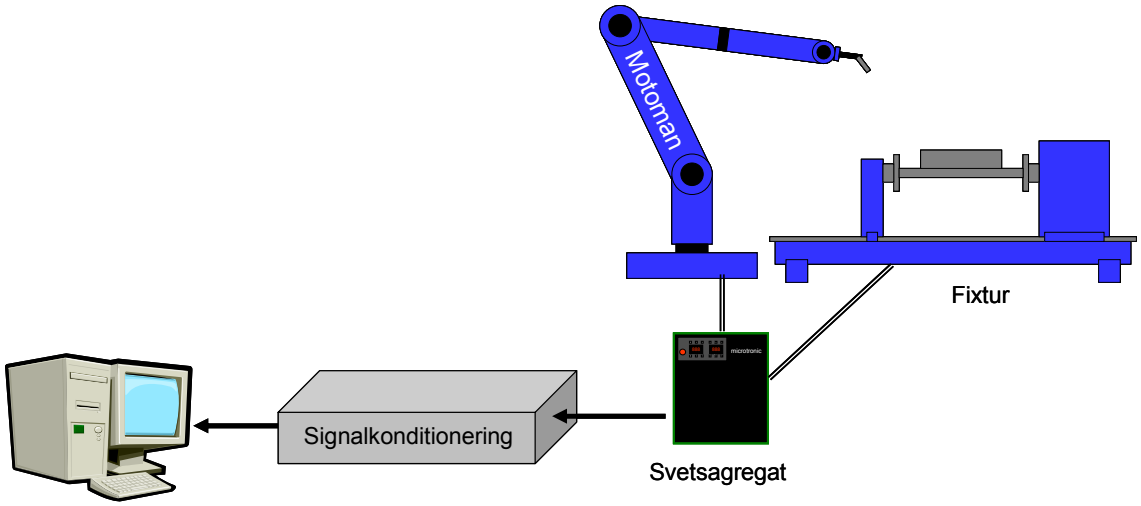


Bild 5.2 – Schematisk beskrivning av svetsrobotcellen

Svetsspänningen mäts genom spänningsdelning och svetsströmmen genom ett hallelement. Givarna sitter vid svetsaggregatet. Innan givarsignalerna når mätkortet i PC:n filtreras signalerna av ett lågpassfilter. Övervakningssystemet registrerar även trådmatningshastigheten och gasflödet. Dessa storheter övervakas ej utan är endast till för att ge kompletterande information om svetsprocessen.

Övervakningssystemets hjärna är en IBM PC med en Intel Pentium 2 processor och 64 Mbyte ram. Operativsystemet är Windows NT. I tabellen nedan summeras alla ingående komponenter.

Robot

Robot:	Motoman, YASNAC ERC

Svetsaggregat

Svetsaggregat:	Migatronic BDH 550

Signalkonditionering

Hallelement:	H.E.M.E. International, Type LC 1000 B
Spänningsdelning:	LEM, Voltage Transducer LV 25-P
Lågpassfilter:	Brytfrekvens 2.25 kHz, insignal 0 – 10 V, utsignal 0 – 5 V

Givare

Gasflödesgivare:	Swissflow, SF-800-B
Trådmatningsgivare:	Givare tillhörande svetsövervakningssystem från ADM

Dator och mjukvara

Dator:	IBM PC, Pentium 2, 300 MHz, 64 MB
Operativsystem:	Microsoft Windows NT 4.0, Service Pack 3
Mätkort:	National Instruments, PCI-MIO-16E-4
Utvecklingsmiljö:	National Instruments, LabVIEW 5.1, Full Development Version

Copyright © 2005 David Grunditz

KAPITEL 6 – VAD SKALL ÖVERVAKAS VID MIG/MAG-SVETSNING

Vad som skall övervakas vid MIG/MAG-svetsning finns det med stor sannolikhet olika uppfattningar om. De parametrar som går att övervaka är spänningen, strömmen, trådmatningshastigheten samt gasflödet. Vidare är frågan om det ska vara rådata (tidsserien), medelvärdet eller variansen som skall övervakas.
I vårt fall är det spänningens och strömmens medelvärde resp. varians som skall övervakas. Det innebär att fyra storheter skall övervakas:

- spänningens medelvärde
- spänningens varians
- strömmens medelvärde
- strömmens varians

6.1 Vilken övervakningsprincip skall användas?

Övervakningen av dessa fyra storheter skall ske med en CUSUM-algoritm (Cumulative sum; växande summa) som här fungerar som en repeterande SPRT-algoritm (Sequential Probability Ratio Test).

6.2 Hur erhålls de fyra storheterna?

Svetsspänningen och svetsströmmen kan mätas på flera sätt, men givarna bör vara placerade så nära svetspistolen och arbetsstycket som möjligt, eftersom svetskablaget kan ha en icke försumbar inverkan på de uppmätta signalerna. Signalerna från givarna går via signalkonditionering till mätkortet där signalerna samplas med en samplingsfrekvens på 8.192 kHz.

Medelvärdet och variansen för svetsspänningen och svetsströmmen beräknas på segment om 1024 sampel. Varje segment har en längd på 125 ms, vilket betyder övervakningstakten är 8 Hz. Nedan följer ekvationerna för beräkning av medelvärdet och variansen av svetsspänningen och svetsströmmen.

Copyright © 2005 David Grunditz

Spänningens medelvärde, $U_m(k)$:

$$U_m(k) = \frac{1}{1024} \sum_{n=(k-1)\cdot 1024}^{1024k-1} u(n), \qquad n = 0, 1 \ldots ;\ k = 1, 2, \ldots \quad (6.1)$$

Spänningens varians, $U_v(k)$:

$$U_v(k) = \frac{1}{1024} \sum_{n=(k-1)\cdot 1024}^{1024k-1} \left(u(n) - U_m(k)\right)^2, \qquad n = 0, 1, \ldots ;\ k = 1, 2, \ldots \quad (6.2)$$

Strömmens medelvärde, $I_m(k)$:

$$I_m(k) = \frac{1}{1024} \sum_{n=(k-1)\cdot 1024}^{1024k-1} i(n), \qquad n = 0, 1, \ldots ;\ k = 1, 2, \ldots \quad (6.3)$$

Strömmens varians, $I_v(k)$:

$$I_v(k) = \frac{1}{1024} \sum_{n=(k-1)\cdot 1024}^{1024k-1} \left(i(n) - I_m(n)\right)^2, \qquad n = 0, 1, \ldots;\ k = 1, 2, \ldots \quad (6.4)$$

6.3 Kvalitetsövervakning med CUSUM eller repeterad SPRT-algoritm

Antag att $\mathbf{x}_s=[x_s(0), x_s(1), \ldots, x_s(k)]$ är en sekvens av observationer av en av de storheter som ska övervakas, t.ex. skattningar av strömmens medelvärde. Vidare antag att observationerna $x_s(0), x_s(1), \ldots, x_s(k)$ är oberoende och normalfördelade med täthetsfunktionen:

$$p_\theta(x_s(i)) = \frac{1}{\sigma_s \sqrt{2\pi}} e^{-\frac{(x_s(i)-\theta)^2}{2\sigma_s^2}} \quad (6.5)$$

Här är σ_s observationernas standardavvikelse och θ är dess medelvärde som vid optimal svetsprocess antar värdet m_0. Däremot om svetsprocess inte är optimal så gäller det att $\theta = m_1$, där $m_0 > m_1$. Nu förutsätts det att svetsprocessen var optimal innan övervakningen startade, $\theta = m_0$, och att den försämras till $\theta = m_1$ vid ett av de $k + 1$ samplingstillfällena. För att detektera en försämring av svetsprocessen kan $k + 1$ hypoteser $H_0, H_1, \ldots, H_k$ testas där H_j definieras av:

$$\begin{aligned} H_j:\ & \theta = m_0 && \text{för}\quad 0 \le i \le j-1 \\ & \theta = m_1 && \text{för}\quad j-1 \le i \le k \end{aligned}$$

och H_0 definieras av:

$$H_0:\ \theta = m_0 \qquad \text{för}\quad 0 \le i \le k$$

Vilket är den såkallade nollhypotesen, d.v.s. ingen försämring av svetsprocessen har skett. Sannoliksförhållandet mellan hypotes H_0 och hypotes H_j ges av [5,6]:

Copyright © 2005 David Grunditz

$$\Lambda_0^k(j) = \frac{\prod_{i=0}^{j-1} p_{m_o}(x_s(i)) \prod_{i=j-1}^{k} p_{m_1}(x_s(i))}{\prod_{i=0}^{k} p_{m_o}(x_s(i))} \tag{6.6}$$

Varvid det så kallade ”log-likelihood” förhållandet kan skrivas [5,6]:

$$S_j^k = \sum_{i=j}^{k} \ln\left(\frac{p_{m_1}(x_s(i))}{p_{m_o}(x_s(i))}\right) \tag{6.7}$$

Om det så kallade ”log-likelihood” förhållandet uppfyller $S_j^k \geq h$ vid samplingstidpunkten k, då detekteras en försämring av svetsprocessen. Där h är en lämpligt vald gräns. En rekursiv uppdatering av ”log-likelihood” förhållandet ges av:

$$S_j^k = S_j^{k-1} + s_k \tag{6.8}$$

där

$$s_k = \ln\left(\frac{p_{m_1}(x_s(k))}{p_{m_o}(x_s(k))}\right) \tag{6.9}$$

Om däremot tidpunkten j då försämring av svetsprocessen sker är okänd då används följande beslutsfunktion:

$$g_k = \max_{0 \leq j \leq k} S_j^k \tag{6.10}$$

Vilken ger CUSUM eller repeterad SPRT-algoritm [5,6]:

$$g_k = [g_{k-1} + s_k]^+ \tag{6.11}$$

Där $[x]^+ = \sup(0, x)$ och om observationerna är normalfördelade kan CUSUM eller den repeterad SPRT-algoritmen skrivas [5]:

$$g_k = [g_{k-1} + \frac{\nu}{\sigma_s^2}(m_0 - x_s(k) - \frac{\nu}{2})]^+ \tag{6.12}$$

där

$$\nu = m_0 - m_1 \tag{6.13}$$

Om $g_k \geq h$ då detekteras en försämring av svetsprocessen.

6.4 De implementerade övervakningsalgoritmerna

Det har implementerats fem övervakningsalgoritmer; fyra CUSUM/SPRT-algoritmer och ett FIR-filter. CUSUM/SPRT-algoritmerna identifieras som SPRT1, SPRT2, SPRT3 samt SPRT4 och är modifierade i förhållande till den CUSUM/SPRT-algoritm som presenteras ovan. Svetsövervakning med SPRT-algoritmerna bygger på att svetsreferenser används för att

Copyright © 2005 David Grunditz

definiera övervakningsparametrarna till SPRT-algoritmerna. En svetsreferens är insamlad data från en optimal svetsning där både robot och svetsaggregat är optimalt inställda. För att täcka in de naturliga systematiska variationer som finns, bör mer än en svetsreferens användas för att bestämma övervakningsparametrarna. För SPRT1, SPRT3 och SPRT4 används endast svetsreferenser för att bestämma övervakningsparametrarna. Det är inte fallet med SPRT2 där en av övervakningsparametrarna bestäms varje gång en ny fog skall övervakas. Förloppet beskrivs ingående i avsnittet om SPRT2 nedan.

6.4.1 SPRT1

SPRT1-algoritmens beslutsfunktion ser ut enligt följande:

$$g(k) = g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0 - y(k)\right| - v\right) \tag{6.14}$$

Genom $\left(\left|m_0 - y(k)\right| - v\right)$ är det möjligt att detektera både stigande och avtagande förändringar hos $y(k)$ i förhållande till $m_{0,}$ vilket inte är fallet med original funktionen. Absolutvärdet som erhålls anger storheten på avvikelsen. Är avvikelsen större än v innebär det att svetsningen inte är optimal och $g(k)$ kommer att stiga.

6.4.1.1 Bestämning av övervakningsparametrarna, m_0, v och σ^2

Övervakningsparametrarna, m_0, v och σ^2 för övervakning av spänningens medelvärde beräknas på följande vis med början med m_0:

$$m_0 = \frac{1}{5}\sum_{n=0}^{4} m_{0_R}(n) \tag{6.15}$$

där

$$m_{0_R}(k) = \frac{1}{I}\sum_{n=0}^{I-1} U_{m_{REF_k}}(n) \qquad 0 \le k \le 4 \tag{6.16}$$

där

$$U_{m_{REF_k}}(i) = \frac{1}{1024}\sum_{n=(i-1)\cdot 1024}^{1024i-1} u_{REF_k}(n) \qquad 0 \le k \le 4 \tag{6.17}$$

och σ^2 och v bestäms på följande vis:

$$\sigma^2 = \frac{1}{5}\sum_{k=0}^{4} \sigma_R^2(k) \tag{6.18}$$

$$v = 2\sqrt{\sigma^2} \tag{6.19}$$

där

$$\sigma_R^2(k) = \frac{1}{I}\sum_{n=0}^{I-1}\left(U_{m_{REF_k}}(n) - m_{0_R}(k)\right)^2 \qquad 0 \le k \le 4 \tag{6.20}$$

När det är spänningens varians som skall övervakas bestäms övervakningsparametrarna på följande vis:

Copyright © 2005 David Grunditz

$$m_0 = \frac{1}{5}\sum_{n=0}^{4} m_{0_R}(n) \tag{6.21}$$

där

$$m_{0_R}(k) = \frac{1}{I}\sum_{n=0}^{I-1} U_{v_{REF_k}}(n) \qquad 0 \le k \le 4 \tag{6.22}$$

där

$$U_{v_{REF_k}}(i) = \frac{1}{1023}\sum_{n=(i-1)\cdot 1024}^{1024i-1}\left(u_{REF_k}(n) - U_{m_{REF_k}}(i)\right) \quad 0 \le k \le 4 \tag{6.23}$$

σ^2 och v bestäms på följande vis:

$$\sigma^2 = \frac{1}{5}\sum_{k=0}^{4}\sigma_R^2(k) \tag{6.24}$$

$$v = 2\sqrt{\sigma^2}$$

där

$$\sigma_R^2(k) = \frac{1}{I}\sum_{n=0}^{I-1}\left(U_{v_{REF_k}}(n) - m_{0_R}(k)\right)^2 \qquad 0 \le k \le 4 \tag{6.25}$$

Den kompletta SPRT1-algoritmen för övervakning av spänningens medelvärde har följande utseende:

1. $g(k-1) = 0$
2. $y(k) = \frac{1}{1024}\sum_{n=(k-1)\cdot 1024}^{1024k-1} u(n)$
3. $g(k) = g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0 - y(k)\right| - v\right)$
4. if $g(k) \le 0$ then $g(k) = 0$
5. if $g(k) \ge h$ then Alarm, $g(k) = h$
6. Goto 2.

För att övervaka spänningens varians måste punkt 2 bytas ut mot

$$y(k) = \frac{1}{1023}\sum_{n=(k-1)\cdot 1024}^{1024k-1}(u(n) - \overline{u}(n))^2 \tag{6.26}$$

där

$$\overline{u}(k) = \frac{1}{1024}\sum_{n=(k-1)\cdot 1024}^{1024k-1} u(n) \tag{6.27}$$

Copyright © 2005 David Grunditz

samt även övervakningsparametrarna m_0, σ^2 och v måste bytas till dem som är avsedda för spänningens varians. Övervakningen av strömmens medelvärde resp. varians görs på motsvarande sätt som ovan.

6.4.2 SPRT2

Skillnaden mellan SPRT1 och SPRT2 är hur övervakningsparametrarna skattas. Tanken med SPRT2-algoritmen är att försöka hantera sådana förändringar i svetsprocessen där $y(k)$ i ekvation 6.14 driver iväg mot m_1 trots att svetsen är helt riktig. SPRT2-algoritmen löser det genom att i varje svets som övervakas använda de 5120 första samplena till att bestämma m_0. Den kvarvarande övervakningsparametern skattas utifrån svetsreferenser precis som för de andra SPRT-algoritmerna dock med den skillnaden att σ^2 bestäms genom att ta medianen istället för medelvärdet på referensernas varians, σ_R^2.

Beslutsfunktionen för SPRT2 är enligt följande:

$$g(k) = \begin{cases} 0 & 0 \le k \le 5 \\ g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0 - y(k)\right| - v\right) & k > 5 \end{cases} \tag{6.28}$$

Övervakningsparametern m_0 för SPRT2 skattas på följande vis:

$$m_0 = \frac{1}{5}\sum_{n=0}^{4}\bar{u}(n) \tag{6.29}$$

där

$$\bar{u}(i) = \frac{1}{1024}\sum_{n=(i-1)\cdot 1024}^{5119}u(n) \tag{6.30}$$

Övervakningsparametrarna σ^2 och v skattas från svetsreferenserna på följande vis:

$$\sigma^2 = \frac{\max\limits_{0\le k\le 4}\left(\sigma_R^2(k)\right) + \min\limits_{0\le k\le 4}\left(\sigma_R^2(k)\right)}{2} \tag{6.31}$$

$$v = 2\sqrt{\sigma^2}$$

där

$$\sigma_R^2(k) = \frac{1}{I}\sum_{n=0}^{I-1}\left(U_{m_{REF_k}}(n) - m_{0_R}(k)\right)^2 \qquad 0 \le k \le 4 \tag{6.32}$$

Den kompletta SPRT2-algoritmen för övervakning av spänningens medelvärde har följande utseende:

1. $g(k-1) = 0$

2. $y(k) = \frac{1}{1024}\sum_{n=(k-1)\cdot 1024}^{1024k-1}u(n)$

Copyright © 2005 David Grunditz

3. $$g(k) = \begin{cases} 0 & 0 \le k \le 5 \\ g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0 - y(k)\right| - v\right) & k > 5 \end{cases}$$

4. if $g(k) \le 0$ then $g(k) = 0$

5. if $g(k) \ge h$ then Alarm, $g(k) = h$

6. Goto 2.

6.4.3 SPRT3

Vid svetsning är det inte ovanligt att det skall svetsas runt hörn, eller längs cirkelbågar. När en robot programmeras är det svårt att få svetspistolen att hålla samma vinkel i förhållande till arbetsstycket och ha samma avstånd till arbetsstycket längs hela den sträcka som skall svetsas. Ändras vinkeln mellan arbetsstycket och svetspistolen eller avståndet mellan arbetsstycket och svetspistolen påverkas svetsspänningen och svetsströmmen. Det är repetitivt eftersom roboten rör sig på samma sätt vid varje ny svets. För att hantera detta måste en modifiering av SPRT-algoritmens beslutsfunktion göras. För denna anpassning finns det två olika paradigmer. Den ena versionen arbetar med ett ”rörligt” tröskelvärde $h(k)$, och den andra, SPRT4, arbetar med en rörligt m_0-värde, $m_0(k)$.

SPRT3-algoritmen arbetar mot det rörliga tröskelvärdet, $h(k)$. I övrigt är SPRT3-algoritmen identisk med SPRT1-algoritmen.
Det rörliga tröskelvärdet fås genom att göra en SPRT1-beräkning på referensdatan och addera till det statiska h värdet.

$$h(k) = g_{REF}(k) + h \tag{6.33}$$

där

$$g_{REF}(k) = g_{REF}(k-1) + \frac{v}{\sigma^2}\left(\left|m_0 - m_0(k)\right| - v\right) \tag{6.34}$$

där

$$m_0(k) = \frac{1}{5}\sum_{i=0}^{4} U_{m_{REF_i}}(k) \tag{6.35}$$

där

$$U_{m_{REF_k}}(i) = \frac{1}{1024}\sum_{n=(i-1)\cdot 1024}^{1024i-1} u_{REF_k}(n) \tag{6.36}$$

De tre övervaknings parametrarna m_0, v och σ^2 bestäms på samma sätt som för SPRT1-algoritmen.

6.4.4 SPRT4

En alternativ paradigm till den princip som används för SPRT3 är att i original beslutsfunktionen i SPRT1-algoritmen ersätta m_0 med $m_0(k)$.
Beslutsfunktionen får då följande utseende:

$$g(k) = g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0(k) - y(k)\right| - v\right) \tag{6.37}$$

Copyright © 2005 David Grunditz

där

$$m_0(k) = \frac{1}{5}\sum_{i=0}^{4} U_{m_{REF_i}}(k) \tag{6.38}$$

$$U_{m_{REF_k}}(i) = \frac{1}{1024}\sum_{n=(i-1)\cdot 1024}^{1024i-1} u_{REF_k}(n) \tag{6.39}$$

Hela SPRT4-algoritmen blir då:

1. $g(k\text{-}1) = 0$
2. $y(k) = \sum_{n=(k-1)\cdot 1024}^{1024k-1} u(n)$
3. $g(k) = g(k-1) + \frac{v}{\sigma^2}\left(\left|m_0(k) - y(k)\right| - v\right)$
4. if $g(k) \le 0$ then $g(k) = 0$
5. if $g(k) \ge h$ then Alarm, $g(k) = h$
6. Goto 2

6.5 FIR-filter

Utöver SPRT-algoritmerna är fyra FIR-filter implementerade. Ett för varje storhet d.v.s. spänningens medelvärde och varians samt strömmens medelvärde och varians. Filtren har ingen direkt övervakande funktion i implementationen men genom att granska resultatet från dem är det möjligt att se när svetsen avviker från ett stabilt tillstånd. Filtret är enligt följande:

$$y(k) = \frac{1}{8}\sum_{i=0}^{7} x(k-i) - \frac{1}{8}\sum_{i=8}^{15} x(k-i) \tag{6.40}$$

där $x(k)$ är antingen medelvärdet eller variansen för sekvenser om 1024. Blockdiagram för FIR-filtret är enligt följande:

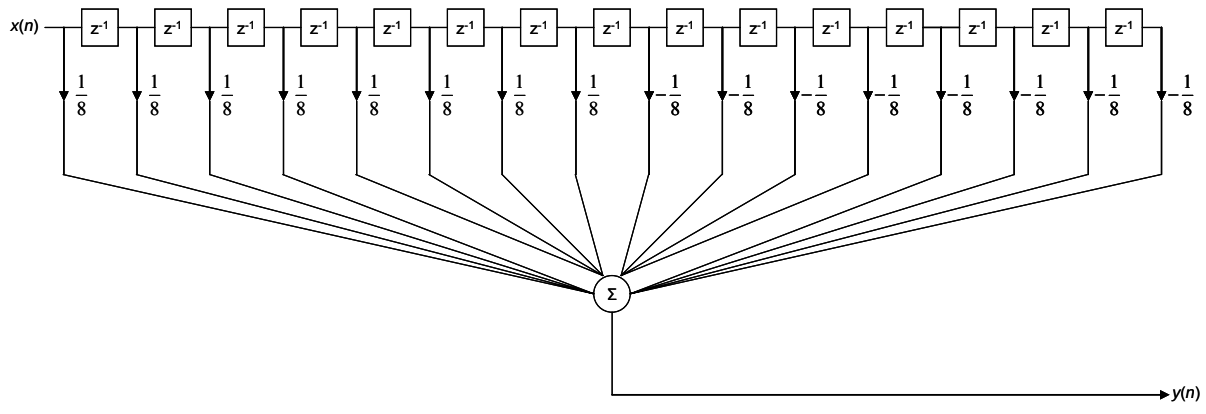


Bild 6.1 – FIR-filter

Copyright © 2005 David Grunditz

6.6 Sammanfattning

Fyra olika SPRT-algoritmer är implementerade varav två är vidareutvecklade av mig tillsammans med Ali Bahrami vid Teknikcentrum i Växjö. För SPRT2 till SPRT4 har endast den algoritm som är avsedd för att övervaka spänningens medelvärde presenterats. För övervakning av spänningens varians samt strömmens medelvärde och varians får modifieringar göras i algoritmerna så att rätt övervakningsparametrar används samt att rätt beräkning av $y(k)$ görs.

Algoritmernas lämplighet tas upp i kapitel 8 som utvärderar övervakningssystemet. För att på ett lättare sätt förstå algoritmernas funktion eller användning kallas ibland h för larmnivå och $[m_0 - v, m_0 - v]$ för fönster eller övervakningsfönster.

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

KAPITEL 7 – BESKRIVNING AV WELDING QUALITY CONTROL SYSTEM:S PROGRAMKOD[1]

7.1 Programkodshierarkin

Implementationen av WQCS är inriktad på programmets huvuduppgift att övervaka svetsning och all funktionalitet som programmet erbjuder utgår från det behov en användare kan tänkas behöva. Sammanfattningsvis finns det två viktiga VI:er i implementationen och det är wqcs_v6.vi, som erbjuder användaren en meny där det är möjligt att ta sig fram till önskad funktion i programmet, samt wqa4daq_v6.vi, som innehåller all funktionalitet för realtidsövervakning av svetsprocessen. Hela WQCS:s hierarki av VI:er finns i bild 7.1. Bild 7.1 visar endast de VI:er som har implementerats, de VI:er som tillhör LabVIEW-miljön finns inte med.

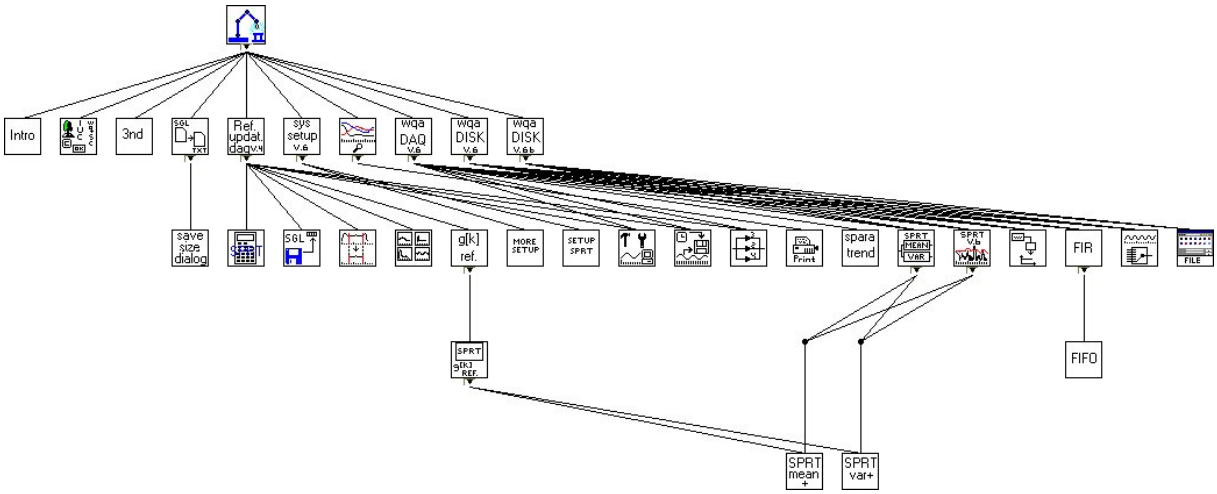


Bild 7.1 – WQCS:s programhierarki

1. Implementationen av SPRT-algoritmerna och FIR-filtrena har arbets- och programnamnet Welding Quality Control System (WQCS), på svenska ungefär Svets kvalitets kontroll system.

Copyright © 2005 David Grunditz

Implementationen består av totalt 34 olika VI:er varav 23 är sub-VI:er till 11 huvud VI:er som visas i bild 7.2

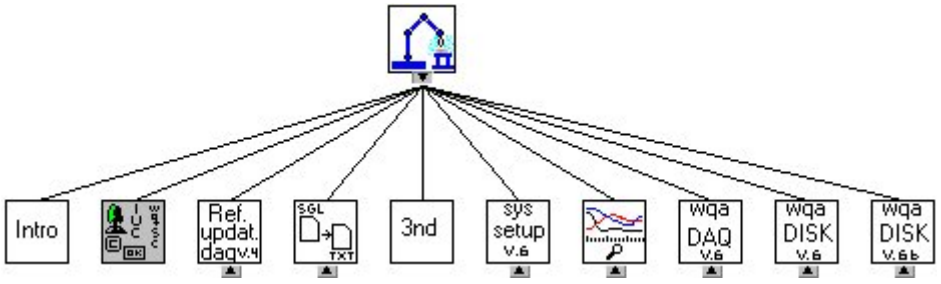


Bild 7.2 – WQCS:s 11 huvud-VI:er

Beskrivningen av implementationen utgår från huvud VI:erna och programkoden presenteras ej. Tabell 7.2 nedan beskriver enkelt huvud-VI:erna.

Grafisk representation	VI-namn	Avsnitt	Funktion
	wqcs_v6.vi	7.2 Huvudmeny	Huvudmeny
Intro	intro.vi	-	Programintroduktion
	Om_prog_wqcs.vi	7.10 Om WQCS	Visar information om programmet
Ref. updat. dagv.4	refupdate_v6.vi	7.5 Ref. uppdatering	Uppdatering av svets-referenser
SGL TXT	sgl to txt.vi	7.9 SGL till TXT	Konverteringsverktyg från SGL-filer till text-filer
3nd	showtrend.vi	7.8 Trenddata	Visa svetstrenden
sys setup v.6	systemsetup_2.vi	7.7 Inställningar	Inställningar för övervakningen
	visar_grafer.vi	7.6 Visa grafer	Visar grafer som finns lagrade på hårddisken
wqa DAQ v.6	wqa4daq_v6.vi	7.3 Algoritm, DAQ – Online	Realtidsövervakning
wqa DISK v.6	wqa4disk_v6_demo.vi	7.4 Algoritm, lab och Algoritm, demo	Demo av realtidsmätning
wqa DISK v.6b	wqa4disk_v6_lab.vi	7.4 Algoritm, lab och Algoritm, demo	SPRT-analys i efterhand

Tabell 7.1 WQCS:s 11 huvud VI:er.

7.2 Huvudmeny

Överst i implementationens hierarki finns huvud-VI:en, wqcs_v6.vi, som anropas när WQCS startas. Denna VI utför flera uppgifter som att hämta data och annan systeminformation som finns lagrad på hårddisken, erbjuda användaren ett menygränssnitt, se bild 7.3, för de olika programfunktionerna, samt när programmet avslutas sparas all nödvändig systeminformation,

Copyright © 2005 David Grunditz

som behövs nästa gång programmet körs. Via menyn kommer användaren åt alla sub-VI:er som visas i bild 7.2, utom den som är längst till vänster. Den VI:en ger användaren information om programmet när programmet startar och är en programintroduktion.

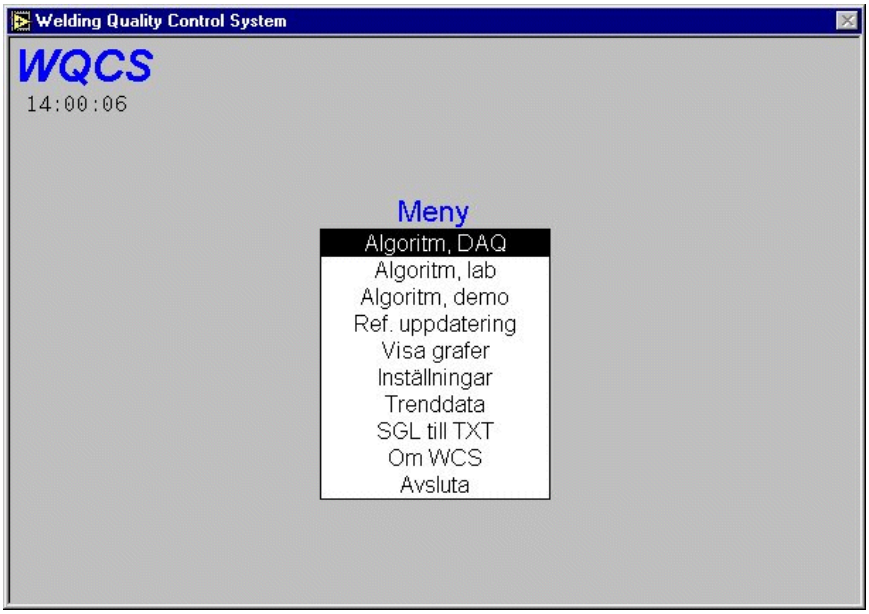


Bild 7.3 – WQCS:s huvudmeny

Användaren tar sig vidare i menyn genom att dubbelklicka med musen på önskat alternativ.

7.3 Algoritm, DAQ

Genom att i huvudmenyn välja det översta alternativet, *Algoritm, DAQ* (DAQ = Data Acquisition ≈ datainsamling), kommer, wqcs_v6.vi att anropar wqa4daq_v6.vi. wqa4daq_v6.vi är huvud-VI:en för svetsövervakningen. Övervakningen sker i realtid och all data som samlas in lagras som filer på hårddisken, så eventuell djupare analys kan göras vid senare tillfälle.

7.3.1 Funktion

wqa4daq_v6.vi är uppbyggd för att kunna övervaka svetsningen i realtid. Blockschemat nedan, bild 7.4, visar programflödet. Hela programuppbyggnaden bygger på att SPRT-algoritmerna arbetar med block på 1024 sampel, vilket motsvara 125 ms då samplingsfrekvensen är 8.192 kHz.
Det betyder att programmet under denna tid måste hämta data från mätkortet, spara den samt göra SPRT-beräkningen, som i praktiken innefattar 16 SPRT-beräkningar, se bild 7.4. I flödesschemat är det blocken fem till nio.

När wqa4daq_v6.vi anropas är det första som sker att diagrammen och dess menyer initieras. Det innebär att menyerna får rätt menytext och eventuellt skräp i diagrammen rensas bort (block 1 i flödesschemat).
Det följs av att programmet väntar på att användaren antigen skall sätta programmet i övervakningsläge, genom att trycka på knappen *Starta algoritm*, eller återgå till huvudmenyn. Väljer användaren att sätta programmet i övervakningsläge aktiveras mätkortet som inväntar en triggningssignal från robotens styrsystem. Triggningssignalen skickar roboten när svetsprocessen har startat.

Copyright © 2005 David Grunditz

För att förhindra att svetsövervakningssystemet låser sig vid fel på roboten finns det en timeout som löser ut. Det sker om ingen triggningssignal skickas och då släpper programmet mätkortet och återgår till att vänta på ny inmatning från användaren.

Skickar roboten en triggningssignal börjar mätkortet att sampla data. Samplad data hämtas kontinuerligt av programmet, för att mätkortets minnesbuffert inte skall bli full och orsaka samplingsfel. Inhämtade data är triggningssignalens spänningsnivå, svetsspänningen, svetsströmmen, gasflödet samt trådmatningshastigheten. Triggningssignalens spänningsnivå används för att detektera när svetsprocessen avslutats så att svetsövervakningen kan avbrytas. Den uppmätta svetsspänningen och svetsströmmen har en spänningsnivå på 0 – 5 V. För att de skall få rätta värden multipliceras de med kalibreringsfaktorer, block 7 i flödesschemat.

All insamlad data sparas i fyra filer. Ur filernas filnamn går det att utläsa vilken storhet som finns i filen samt klockslag och datum då svetsövervakningen startade. Det datum och den tid som är inbakat i filnamnet är inte samma som det som operativsystemet registrerar och lägger till som fil attribut. Filformatet som används är ett binärt filformat som LabVIEW använder kallat SGL.

Svetsövervakningen sker i blocket 9 i flödesschemat, som beskrivs i detalj i avsnittet 7.3.2 Algoritmberäkningar nedan. Efter beräkningarna plottas de av användaren valda kurvorna och programflödet går åter till block 5 och hämtar ny data från mätkortet. Detta förlopp pågår tills triggningssignalen blivit låg.

Copyright © 2005 David Grunditz

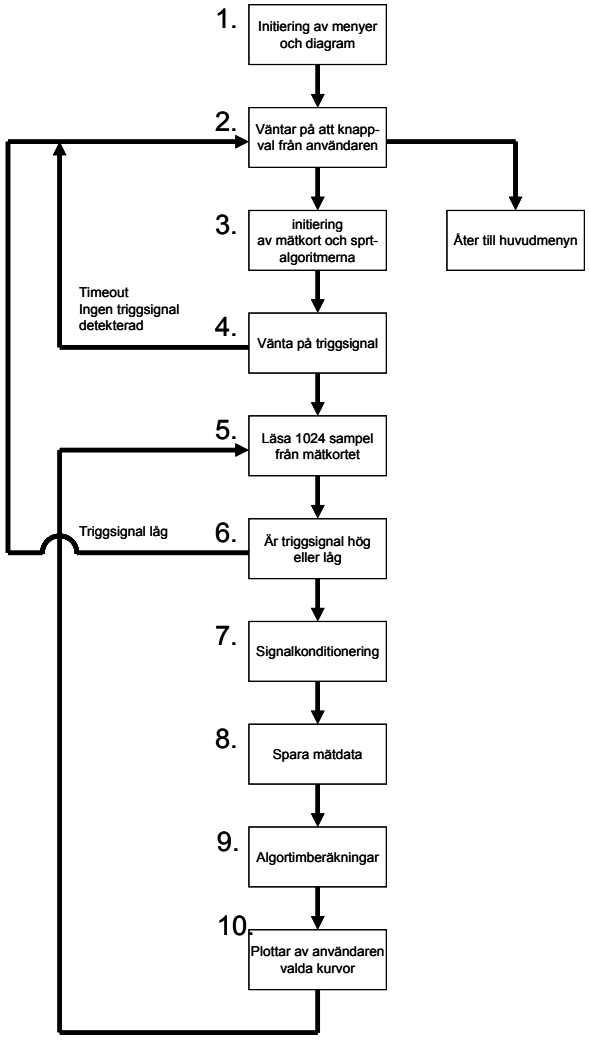


Bild 7.4 – wqcs_v6.vi:s flödesschema

7.3.2 Algoritmberäkningar

Implementationen av SPRT-algoritmerna och FIR-filtret innebär att under en övervakningscykel på 125 ms skall 16 SPRT-algoritmer och fyra FIR-filter funktioner beräknas. Till detta kommer konvertering av gasflödets och trådmatningens indata, som är i pulstågsform, till tolknings bara värden.

Övervakningsparametern m_0 till SPRT2-algoritmen beräknas under svetsövervakningens första 625 ms, vilket motsvarar fem loopar i flödesschemat ovan. Under denna tid sker ingen svetsövervakning med SPRT2-algoritmen, utan övervakning sker endast med SPRT1-, SPRT3-, SPRT4-algoritmerna.

Konvertering av gasflödet och trådmatningshastigheten sker genom att signalernas frekvensspektra bestäms och ur dem få fram den dominerande frekvensen, som multipliceras med en omvandlingsfaktor. Gasflödet och trådmatningshastigheten presenteras i l/min respektive mm/s. Algoritmberäkningarnas flödesschema visas i bild 7.5.

Copyright © 2005 David Grunditz

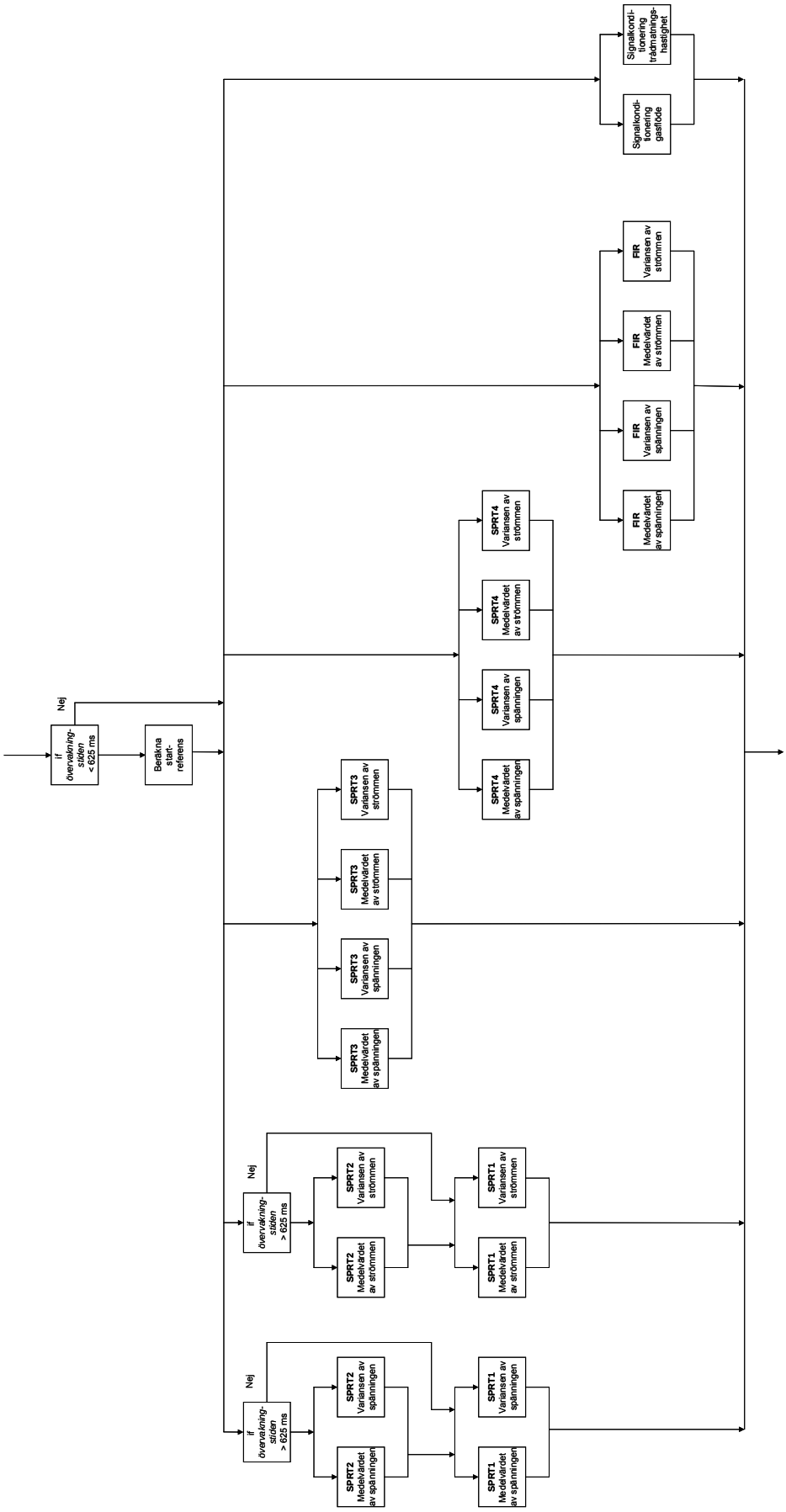


Bild 7.5 – Algoritmberäkningarnas flödesschema.
Block 9 i wqcs_v6.vi:s flödesschema, se bild 7.4.

Copyright © 2005 David Grunditz

7.3.3 Realisering av SPRT-algoritmen

Realiseringen av SPRT1-algoritmen för övervakning av medelvärdet i LabVIEW visas i bild 7.6. På vänster sida i diagramfönstret är alla inmatningsobjekten, h (larmgränsen), g(k-1), s^2 (σ^2, variansen), v, m_0 och rådata om 1024 sampel. Hela SPRT1 beräkningen görs i ett formelobjekt där formeln anges. På höger sida i diagramfönstret finns alla utmatningsobjekten, $g(k)$, $g(k)+-$, $e(k)$, $m(k)$. $g(k)+-$ ger information om på vilken sida av m_0 som $m(k)$ är när ett svetsfel detekteras. Den enda skillnaden mellan $g(k)$ och $g(k)+-$ är att $g(k)$ är absolutvärdet av $g(k)+-$. $e(k)$ är absolutvärdet av differensen mellan $m(k)$ och m_0.

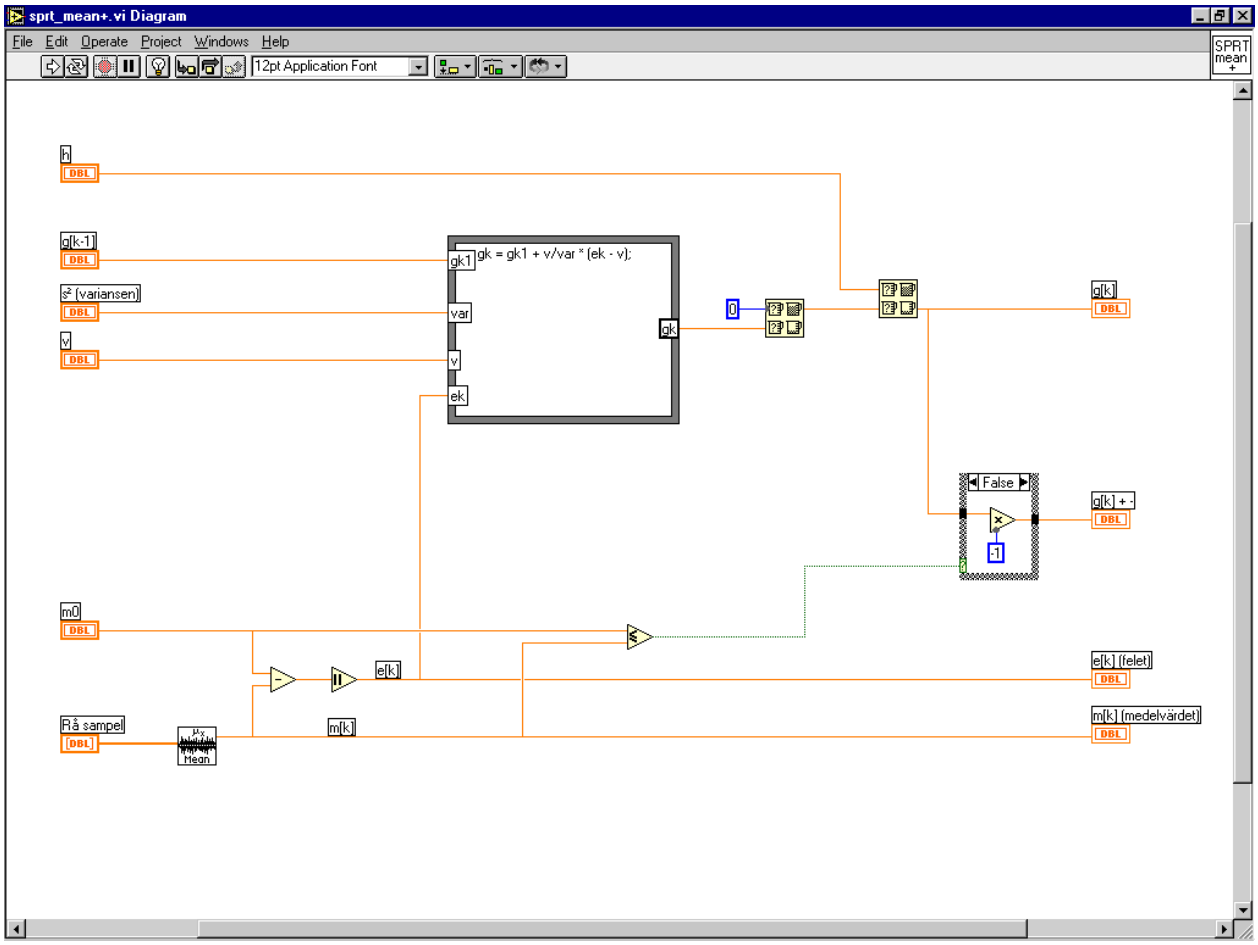


Bild 7.6 - LabVIEW implementeringen av SPRT1-algoritmen

SPRT2- SPRT3- och SPRT4-algoritmernas implementeringen är motsvarande, men har modifierats där algoritmernas egenskaper har krävt det.

7.3.4 Gränssnitt mot användare

Användargränssnittet besår av fyra diagram, bild 7.7. Till varje diagram finns en diagrammeny där det som skall plottas väljs. Totalt finns det 70 olika plottnings alternativ, som bland annat visar resultat från övervakningsalgoritmerna. Till varje SPRT-beräkning finns det fyra olika plottningsalternativ att välja mellan, $g(k)$, $g(k)+-$, $e(k)$ samt $m(k)$. För FIR-algoritmerna går det endast att välja att visa utdata från FIR-beräkningarna och för gasflödet och trådmatningshastigheten är det endast möjligt att plotta deras flöde eftersom de inte övervakas.

Copyright © 2005 David Grunditz

Varje diagram har ett antal funktioner för att zooma i diagrammen, samt ändra upplösningen på x- respektive y-axeln. Dessa funktioner finns lokaliserade antingen ovan för diagrammen eller under diagrammen.
För att användaren skall kunna sätta övervakningssystemet i övervakningsläge måste användaren trycka på knappen *Starta algoritmen*, som finns överst i mitten. Då sätts övervakningssystemet till att vänta på triggningssignalen från robotens styrsystem.
Det är tänkbart, om övervakningssystemet integreras i en produktionsmiljö, att ersätta knappen med en extra signal från roboten eller en PLC, så att övervakningssystemet automatiskt sätts i övervakningsläge och inväntar triggningssignalen.

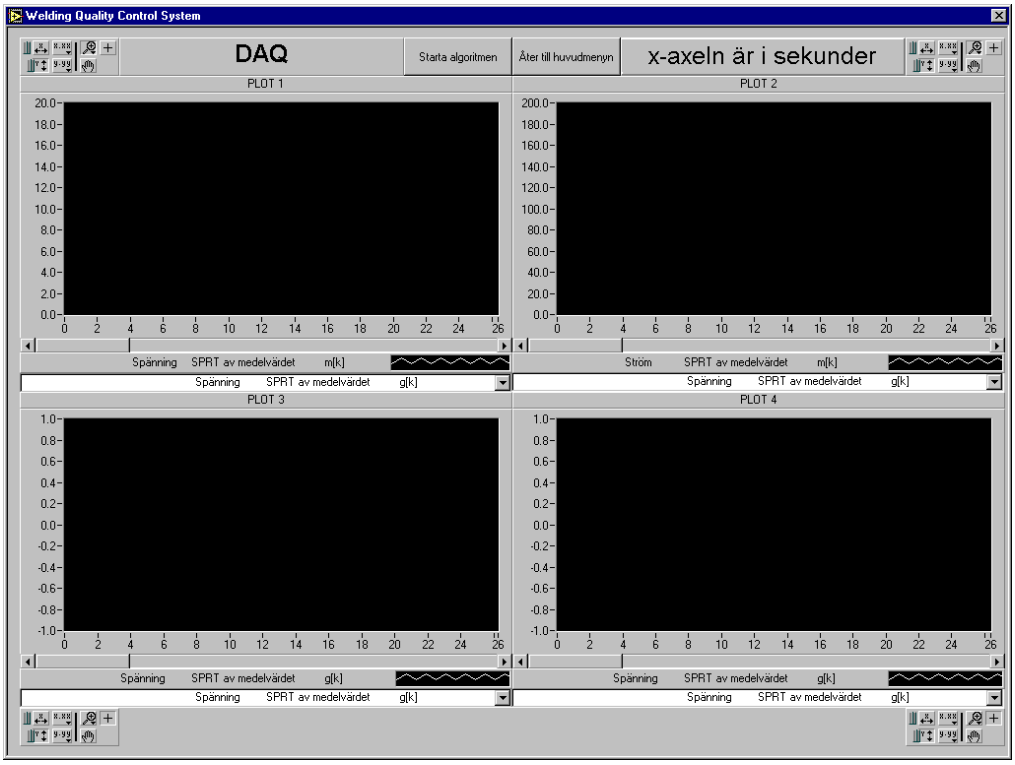


Bild 7.7 – wqcs_v6.vi användargränssnitt

7.4 Algoritm, lab och Algoritm, demo

I huvudmenyn, se bild 7.3, finns två menyalternativ, *Algoritm, lab* och *Algoritm, demo*. De erbjuder användaren att i efterhand analysera svetsar som finns lagrade på hårddisken.
Det som skiljer alternativen åt är att det ena alternativet, *Algoritm, lab*, främst är avsett för laborationsändamål, vilket i praktiken innebär efteranalys av svetsar. Det andra alternativet är till för att demonstrera hur svetsövervakningssystemet fungerar under en realtidsövervakning av en svetsprocess, när det inte finns möjlighet att övervakna en verklig svetsprocess. Användargränssnitten mot användaren är nästan helt identiskt med användargränssnittet för realtidsövervakning, *Algoritm, DAQ*. Det som skiljer är att *Algoritm, lab* har fyra knappar extra, en för varje diagram, för att spara den data som har använts för att plotta en specifik kurva i ett diagram. Dessa data sparas i det binära filformat SGL.

7.4.1 Funktion

Programflödet för wqadisk_v6b_lab.vi och wqadisk_v6_demo.vi har delar gemensamt med wqa4daq_v6.vi:s programflöde. Det väsentligaste som skiljer är att wqadisk_v6b_lab.vi och wqadisk_v6_demo.vi inte hämtar sampel från mätkortet utan hämtar sampel från hårddisken. Till detta kommer speciell funktionalitet som

Copyright © 2005 David Grunditz

wqadisk_v6b_lab.vi använder för att plotta data i diagrammen samt spara data från diagrammen. Flödesschemat för wqadisk_v6b_lab.vi och wqadisk_v6_demo.vi finns i bild 7.8 respektive bild 7.9.
Båda VI:erna använder samma svetsövervakningsprinciper som används för realtidsövervakningen, som beskrivs i avsnittet 7.3.2 Algoritmberäkningar ovan.

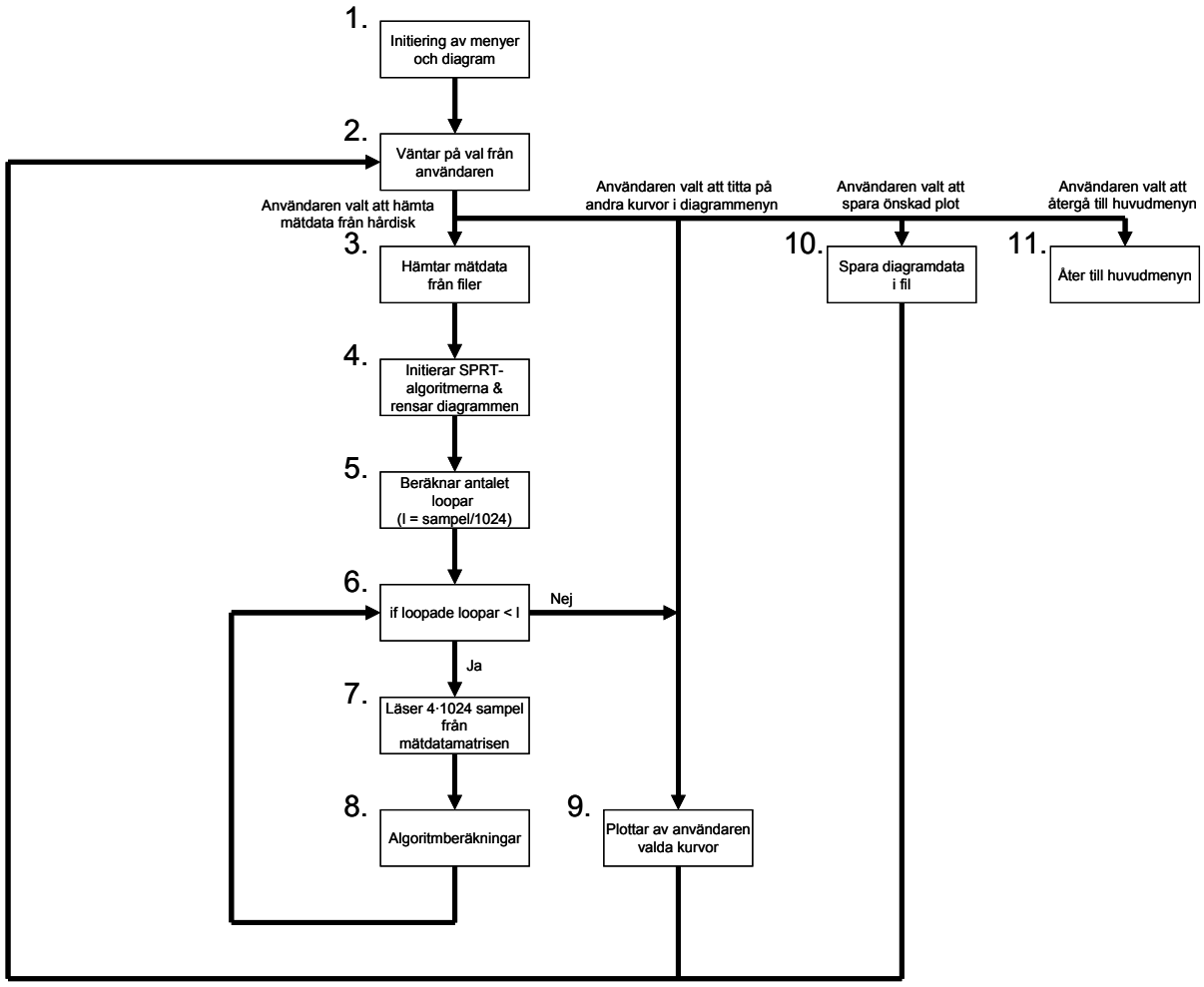


Bild 7.8 – wqadisk_v6b_lab.vi:s flödesschema.

Copyright © 2005 David Grunditz

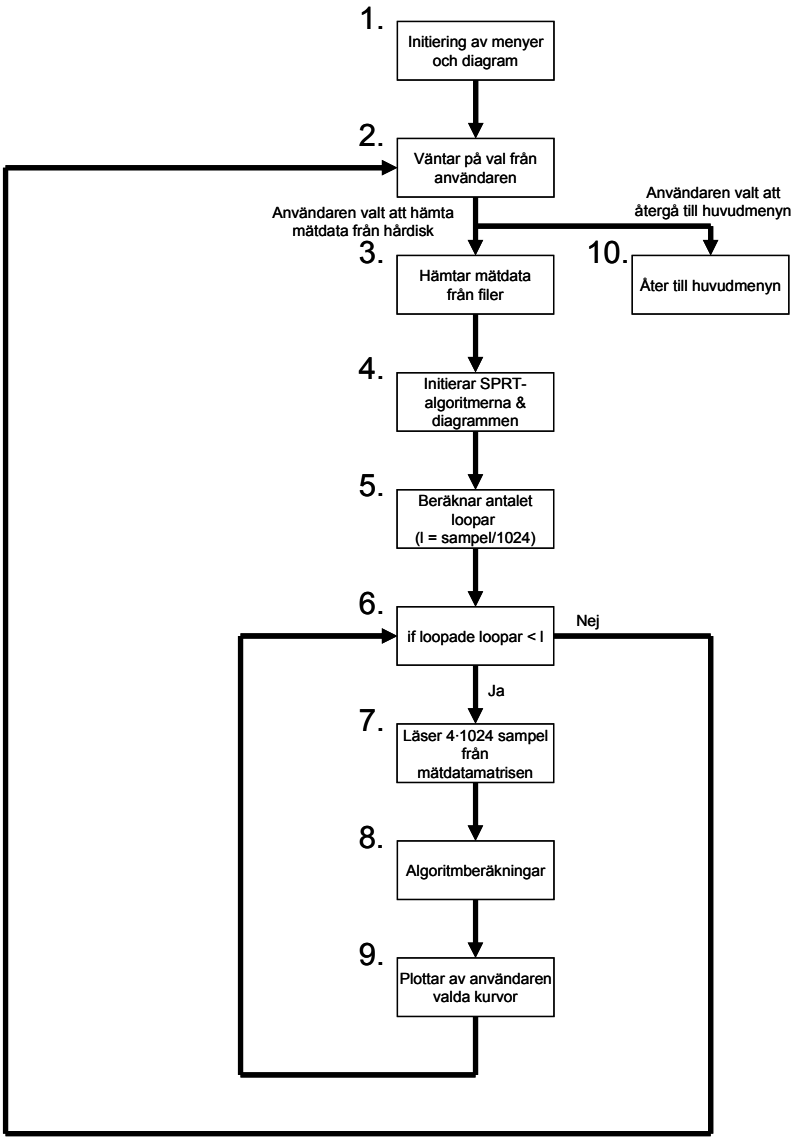


Bild 7.9 – wqadisk_v6_demo.vi:s flödesschema.

7.4.2 Användargränssnitt

Som tidigare nämnts är användargränssnitten nästan identiska med det användargränssnitt som används för realtidsövervakning. Det som skiljer är att när användaren trycker på knappen *Starta algoritmen* så ombeds användaren att välja filer med mätdatasampel, istället för att programmet initierar mätkortet och inväntar på triggningssignalen.

wqadisk_v6b_lab.vi har fyra extra knappar för att spara diagramdata, se bild 7.10. När användaren trycker på någon av dessa knappar ombedes användaren att ange ett filnamn för data som skall sparas.

Copyright © 2005 David Grunditz

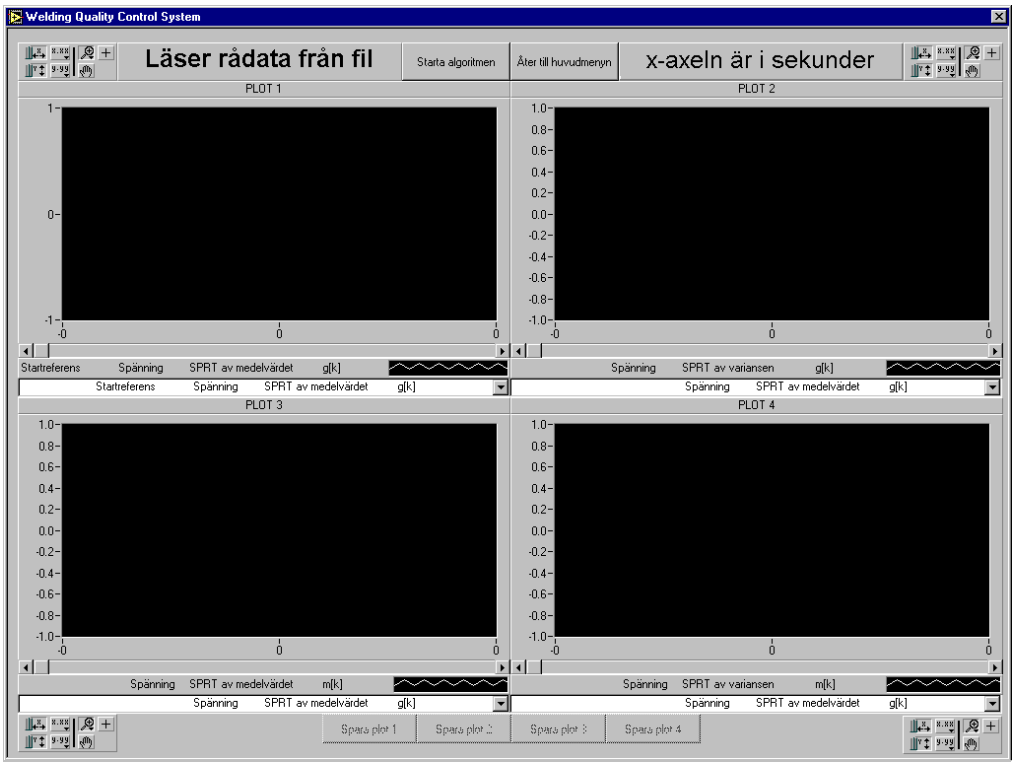


Bild 7.10 – wqadisk_v6b_lab.vi:s användargränssnitt

7.5 Ref. uppdatering

Övervakningskonceptet med SPRT-algoritmerna innebär att systemet skall vara självlärande, dvs. användaren skall ej mata in några övervakningsparametrar. Istället skall alla övervakningsparametrar bestämmas utifrån väl valda svetsreferenser. I svetsövervakningssystemet finns det i huvudmenyn ett alternativ för att bestämma övervakningsparametrarna, *Ref. uppdatering*, se bild 7.3.
Det erbjuder full funktionalitet för att i realtid mäta fem svetsreferenser eller hämta svetsreferenser från hårddisken.
För att få bra värden på övervakningsparametrarna måste användaren välja ut fem lämpliga referenser, ta bort svetsstart och svetsslut på dem och slutligen acceptera det presenterade resultatet på övervakningsparametrarna. Av de fem referenserna skapas en idealsvetsfog och utifrån den bestäms övervakningsparametrarna till SPRT-algoritmerna.

7.5.1 Funktion

refupdate_v6.vi:s flödesschema kan delas upp i tre delar. En del för att framställa svetsreferenser genom realtids mätningar, en del där svetsreferenserna hämtas från hårddisken samt en gemensam del där övervakningsparametrarna beräknas, se bild 7.11.
Det första användaren ställs inför är att välja mellan att använda svetsreferenser som finns lagrade på hårddisken eller att mäta nya. Därefter kommer programmet antingen ta den vänstra eller den högra vägen genom flödesschemat. Det de båda vägarna har gemensamt är att användaren efter varje uppmätt eller hämtad referens kan avbryta och återgå till huvudmenyn. När alla fem referenserna är tillgängliga inleds beräkningen av övervakningsparametrarna. Det är en process som börjar med att användaren tar bort delar av svetsens början respektive slut för att ljusbågens instabilitet inte skall påverka övervakningsparametrarna. Det sker genom att användaren anger hur många sampel som skall tas bort i början respektive i slutet av svetsen.

Copyright © 2005 David Grunditz

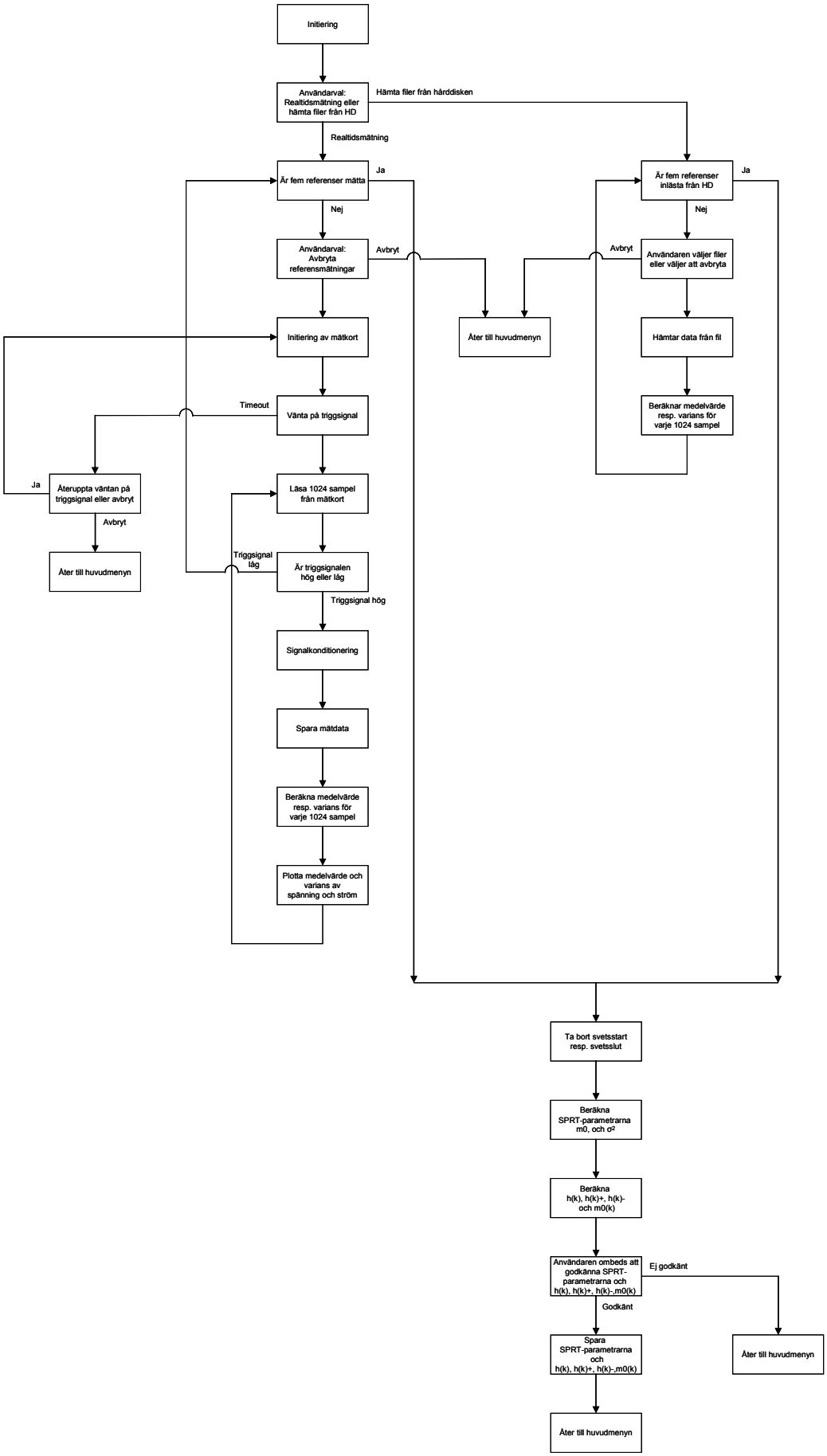


Bild 7.11 – refupdate_v6.vi:s flödesschema

Copyright © 2005 David Grunditz

Efter detta bestäms övervakningsparametrarna m_0 och σ^2, men för SPRT2 bestäms endast σ^2. För SPRT3 och SPRT4 beräknas även $h(k)$, $h(k)+$, $h(k)-$ och $m_0(k)$. Bestämningen av v parametern, som är $v = 2 \cdot \sqrt{\sigma^2}$, sker när användaren aktiverar de nya övervakningsparametrarna.
För att användaren skall ha kontroll på bestämningen av övervakningsparametrarna har användaren möjlighet att se värdena på de enskilda parametrarna, samt i diagram se $h(k)$, $h(k)+$, $h(k)-$ och $m_0(k)$. Godtar användaren ej resultatet är det endast att avbryta uppdateringsprocessen och återgå till huvudmeny. Godtar användaren resultatet så ersätts systemets ursprungsvärden för övervakningsparametrarna med nya och värdena sparas. Det skall dock påpekas att de beräknade övervakningsparametrar, m_0 och σ^2, oavsett om användaren godkänt dem eller ej, sparas i speciella filer som ej har med svetsövervakningen. Dessa filer är till för att sammanställa statistik över lyckade och misslyckade uppdateringar av svetsövervakningsparametrarna.

7.5.2 Användargränssnitt

Under hela processen med att ta fram svetsreferenser, från hårddisken eller från realtids mätningar, har användaren möjlighet att i diagram se svetsarnas medelvärde och varians för både spänningen och strömmen. Accepteras de ej är det som tidigare nämnts bara att välja avbryt och återgå till huvudmenyn.
Accepteras alla referenser ombeds användaren att ta bort svetsens början och slut. Användargränssnittet för det visas i bild 7.12 och består av fyra diagram samt två rutor längst ned till höger där antalet sampel som skall tas bort anges.

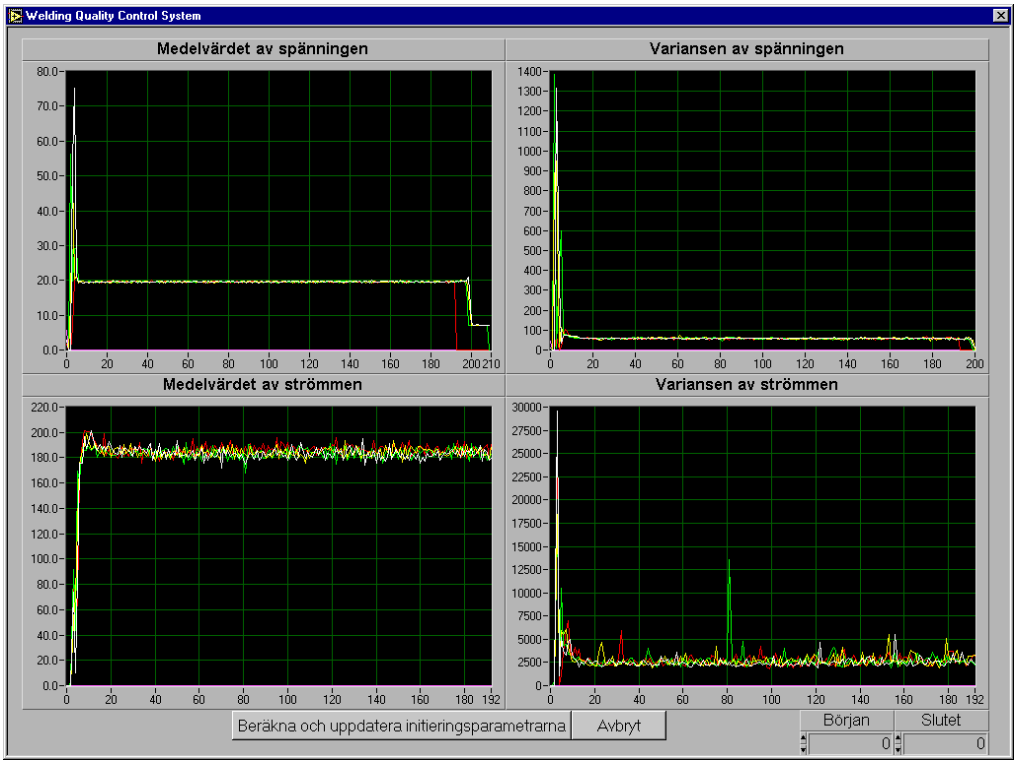


Bild 7.12 – refupdate_v6.vi:s användargränssitt för att ta bort svetsstarten och svetsslutet

När alla parametrar är beräknade får användaren som tidigare nämnts möjlighet att se resultatet, för att acceptera eller förkasta resultatet. $h(k)$, $h(k)+$, $h(k)-$ och $m_0(k)$ som används

Copyright © 2005 David Grunditz

av SPRT3 och SPRT4 visas i ett användargränssnitt där det är möjligt att detaljstudera resultatet genom zoomfunktioner, se bild 7.13.

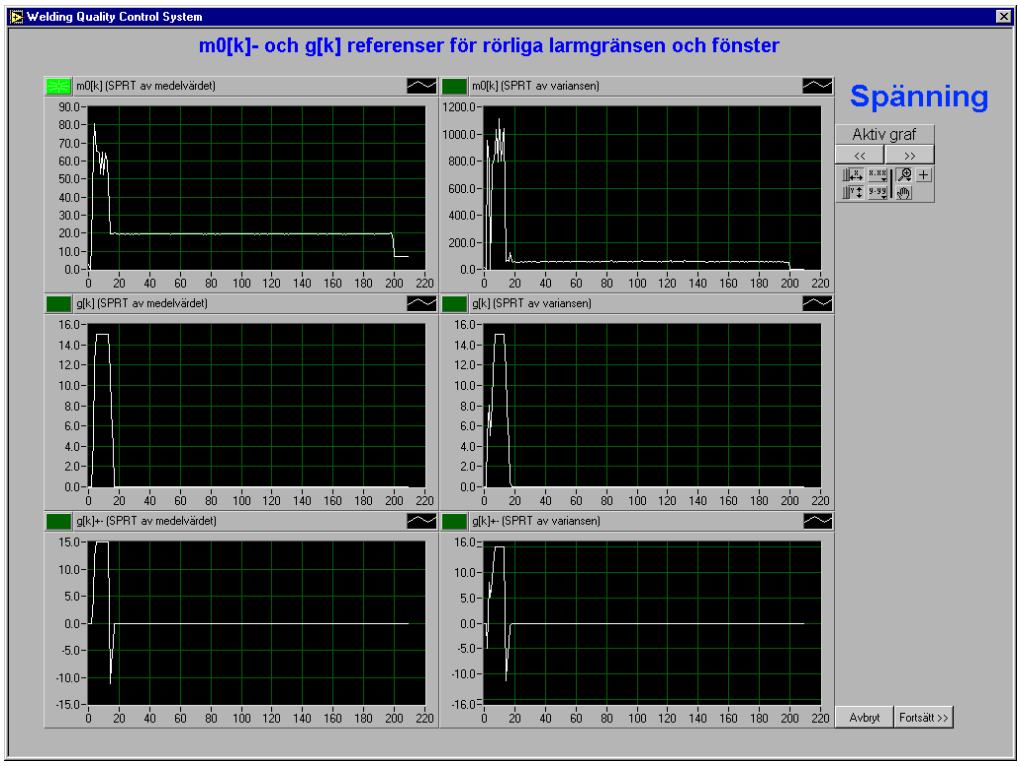


Bild 7.13 – refupdate_v6.vi:s användargränssnitt som visar $h(k)$, $h(k)+$, $h(k)-$ och $m_0(k)$

7.6 Visa grafer

Ibland är det intressant att studera den bakomliggande rådata för att försöka finna en förklaring till ett visst utfall. Genom att i huvudmenyn välja *Visa grafer* är det möjligt att titta på rådata, men även data från *Algoritm, lab*, som sparats. Det är möjligt att öppna upp till tio filer samtidigt för att jämföra dem och skriva ut dem på papper.

7.6.1 Användargränssnitt

Användargränssnittet erbjuder zoom möjligheter för att det i detalj skall vara möjligt att titta på exempelvis rådata. För att hålla isär graferna i diagrammet så har kurvorna olika färger och i nedre högra hörnet finns en färgbeskrivning där färg och fil kopplas ihop. Se bild 7.14

Copyright © 2005 David Grunditz

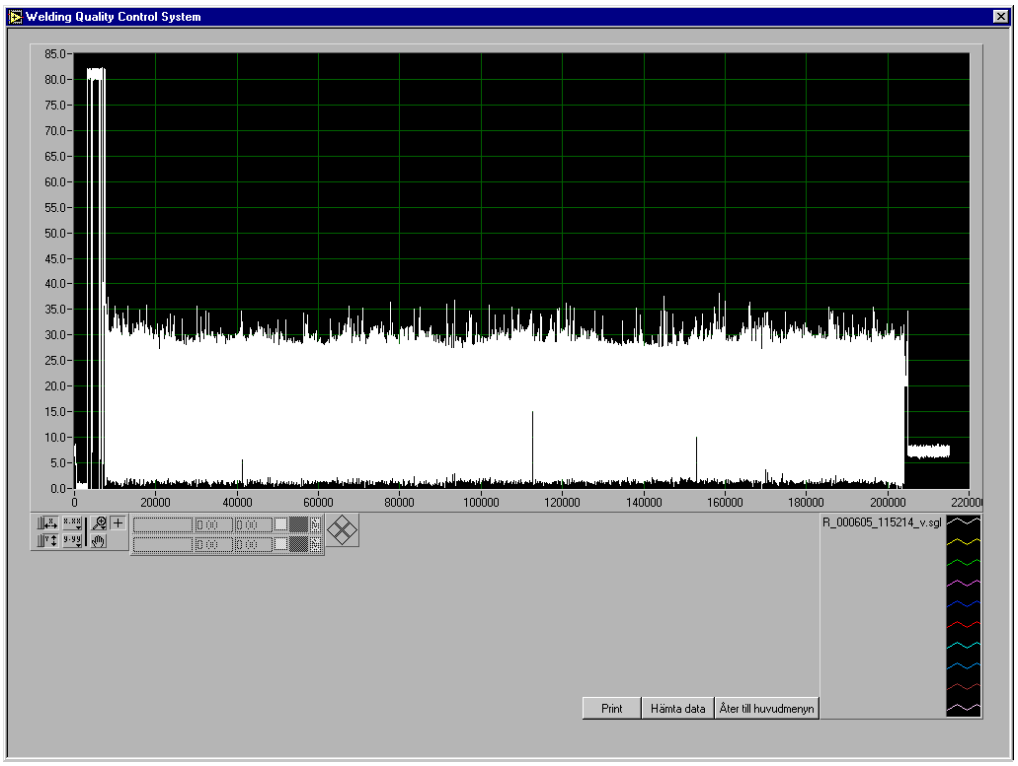


Bild 7.14 – Användargränssnittet för Visa grafer

7.7 Inställningar

Det kan finnas tillfällen då de beräknade övervakningsparametrarna behöver justeras. Exempel på det kan vara svetsning av detaljer där kvalitetskravet är lågt och en stor spridning accepteras. För att undvika att övervakningssystemet registrerar dem felaktigt måste övervakningsparametrarna justeras.

Det är möjligt att ändra övervakningsparametrarna genom att i huvudmenyn välja *Inställningar*. Härifrån nås övervakningssystemets samtliga inställningar. Det är inställningar för SPRT1- och SPRT2-algoritmerna, och indirekt även SPRT3- och SPRT4-algoritmerna, samt inställningar för x-axeln. De inställningar som kan ändras är övervakningsparametrarna m_0, σ^2, v och h för SPRT1-algoritmen och σ^2, v och h för SPRT2- algoritmen. En förändring av h *och* m_0 påverkar SPRT3-algoritmens larmgräns genom att h_{SPRT1} är en offset för $h_{SPRT3}(k)$. Sänks h_{SPRT1} så sänks SPRT3-larmgärns. Motsvarande gäller för m_0 som för SPRT4-algoritmen är en offset till övervakningsfönstret. Sänks m_0 så sänks SPRT4-algoritmens fönster med motsvarande.
För x-axeln är det möjligt att välja vilken representation som x-axelns skall ha, sampel eller sekunder.
Övervakningsparametrar som erhållits genom svetsreferenser kallas för systemets ursprungsvärden. De aktiveras genom att användaren trycker på knappen *Ändra till defaultvärden* i inställningarna för spänning respektive ström. Samtliga rutor som har ett värde som avviker från ursprungsvärdet kommer att få ett nytt värde. För att acceptera användningen av ursprungsvärdena trycker användaren därefter på *OK*.

7.7.1 Funktion

När användaren väljer *Inställningar* i huvudmenyn anropar wqcs_v6.vi systemsetup_2.vi. systemsetup_2.vi inleder med att beräkna alla *v* parametrarna, se bild 7.15.

Copyright © 2005 David Grunditz

Därefter ombeds användaren att välja vilka inställningar som skall ändras. Valet står mellan spänningens, strömmens och övriga inställningar, eller att återgå till huvudmenyn.
Vid återgång till huvudmenyn utan att några inställningar gjorts kastas alla beräknade *v* parametrar.
Väljer användaren att ändra spänningens, strömmens eller de övriga inställningarna, visas ett nytt fönster med möjlighet att förändra nuvarande värden. Ändrar användaren en inställning och kvitterar med att trycka på knappen *OK* kommer systemets inställningar att uppdateras, vilket annars inte sker.

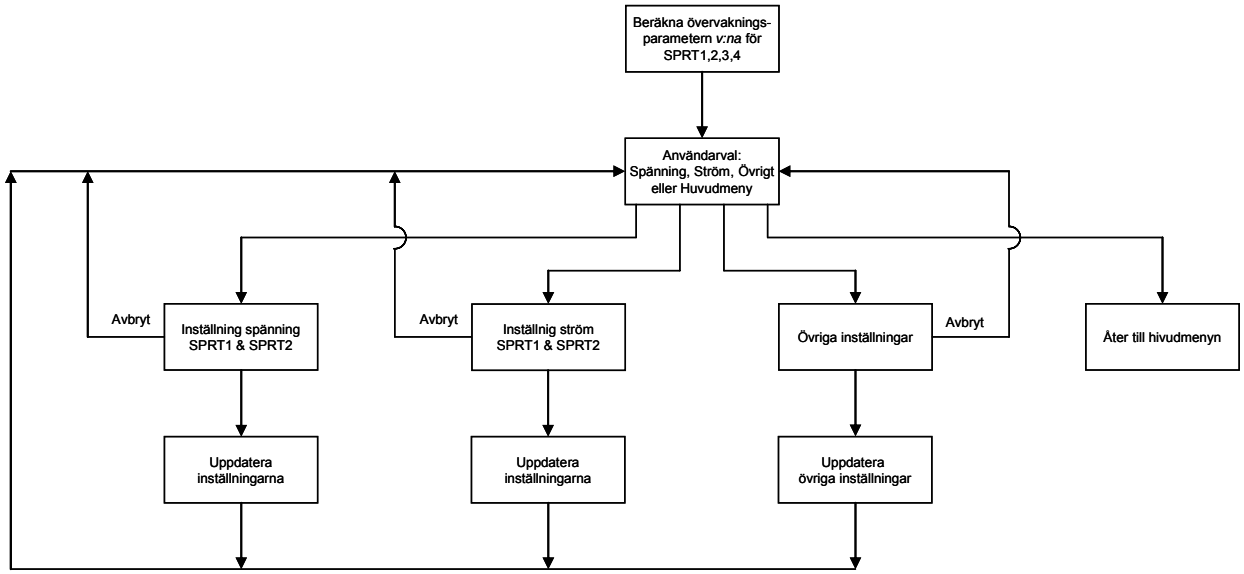


Bild 7.15 – systemsetup_2.vi:s flödesschema

7.7.2 Användargränssnitt

Användargränssnittet för inställningar består totalt av fyra olika fönster, ett för inställningarnas huvudmeny, två fönster för inställningar av övervakningsparametrar, samt ett fönster för övriga inställningar. De två fönster som är till för att ställa in spänningens respektive strömmens övervakningsparametrar är nästan identiska. Bild 7.16 visar fönstret för inställning av spänningens övervakningsparametrar.

Copyright © 2005 David Grunditz

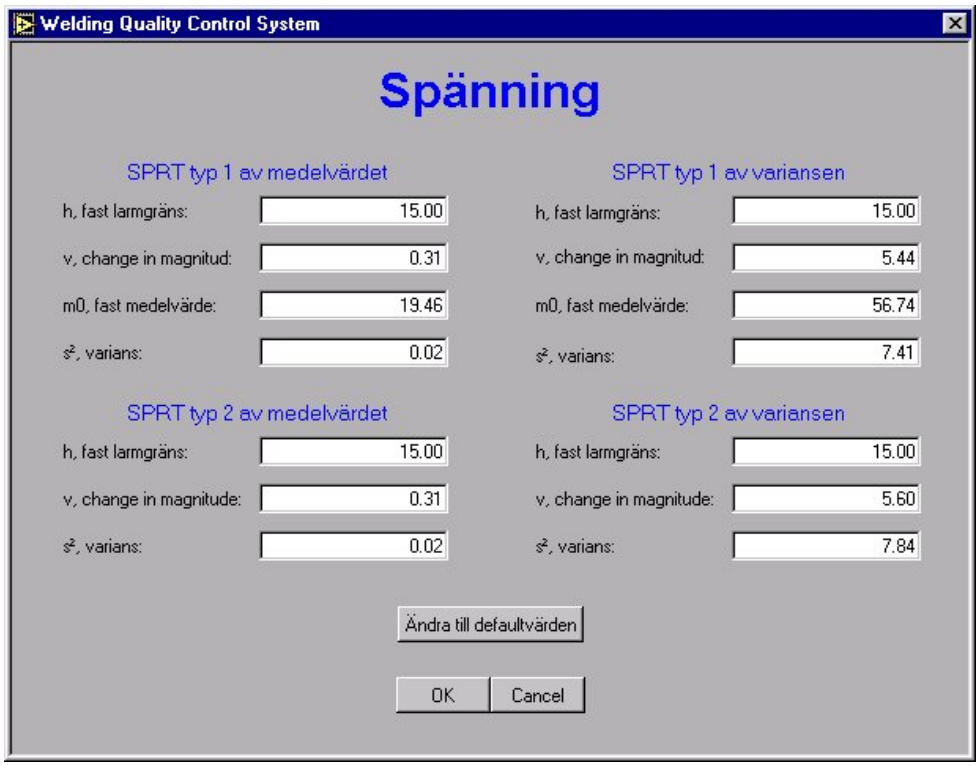


Bild 7.16 – Spänningsinställningarnas användargränssnitt

7.8 Trenddata

En typ av förändringar som kan ställa till problem är förändringar som på kort sikt knappt är märkbara men på lång sikt har betydelse för svetsprocessen. Den här typen av förändringar går att fånga upp genom att plotta svetstrender. Övervakningssystemet erbjuder denna funktion, men utan övervakning.

Efter varje realtidsövervakning beräknas ett trendvärde i form av medelvärde respektive standardavvikelse av svetsens svetsspänning och svetsström. Alla trendvärden som beräknas under realtidsövervakning presenteras i två diagram och två tabeller. Det finns möjlighet att spara trenddata i textformat.

7.8.1 Funktion

När wqcs_v6.vi anropar showtrend.vi hämtas trenddata från en trendfil. Dessa data anpassas för att visas i tabellform och för att vara möjlig att sparas i textformat. Flödesschemat för showtrend.vi visas i bild 7.17.

Copyright © 2005 David Grunditz

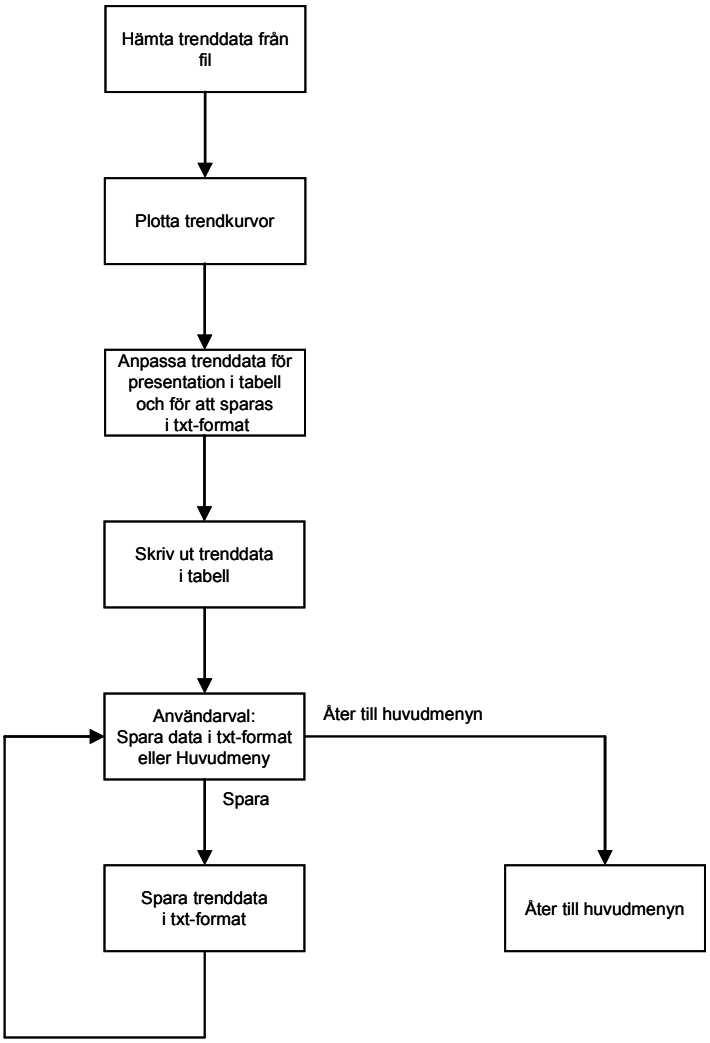


Bild 7.17 – showtrend.vi:s flödesschema.

7.8.2 Användargränssnitt

Användargränssnittet, se bild 7.18, för trenddata är uppbyggd kring två diagram som har varsin tabell där information om varje trendvärde presenteras. Tabellen ger information om medelvärdet och standardavvikelsen samt klockslag och datum då värdena registrerats.

Copyright © 2005 David Grunditz

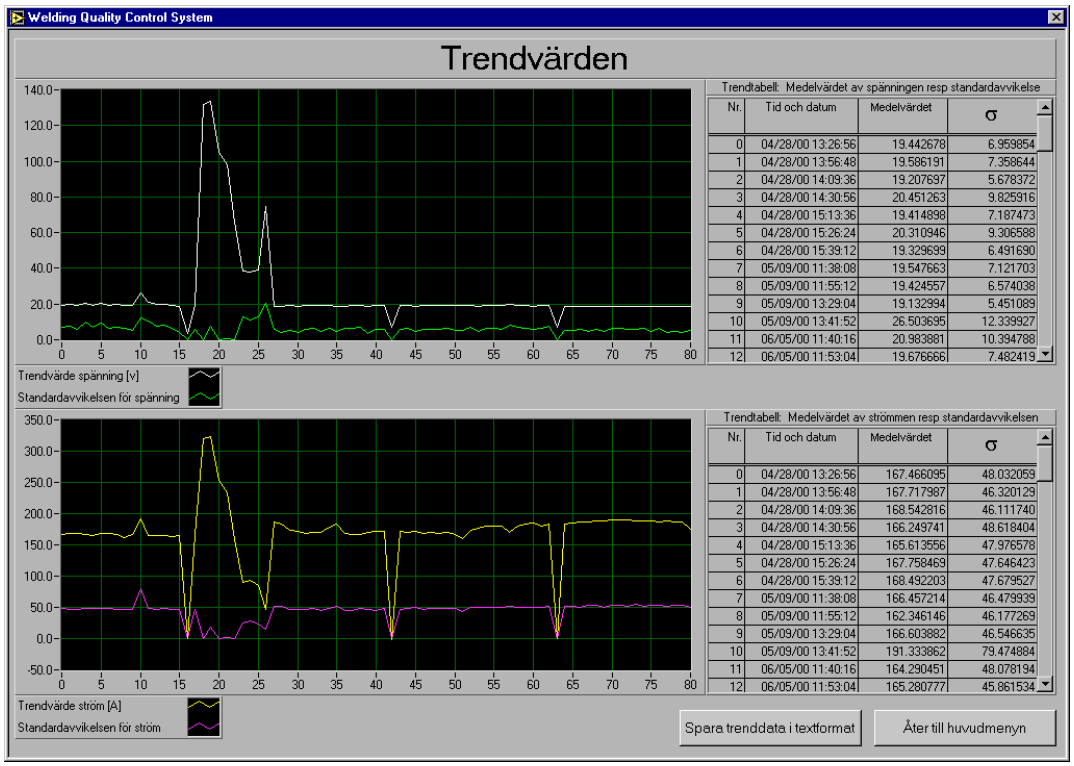


Bild 7.18 – Användargränssnitt för visning av trenddata

7.9 SGL till TXT

För att data som övervakningssystemet samlar in skall vara hanterbart, sparas all data i LabVIEWs binära filformat SGL. Om data sparas i ASCII-format blir filerna ohanterbart stora. Det finns tillfällen då det är motiverat att insamlad data sparas i ASCII-format, exempelvis för att läsa in mätdata i Matlab. Därför har övervakningssystemet en konverteringsfunktion där antingen hela SGL-filer eller delar av SGL-filer sparas om i ASCII-format i en text-fil.

7.9.1 Funktion

När användaren väjer i huvudmenyn att konvertera data från SGL-format till textformat anropar wqcs_v6.vi sgl to txt.vi. sgl to txt.vi inleder programflödet, se bild 7.19, med initiering av diagrammet, som visar den data som skall konverteras. Detta följs av en väntan på aktivitet från användaren som ställs inför valet att öppna en SGL-fil, spara data i textformat (dock inte före konvertering av den första filen) eller att återgå till huvudmenyn. Användaren måste alltid börja med att läsa in en SGL-fil för att få tillgång till möjligheten att spara i textformat. Är en SGL-fil inläst och användaren väljer att spara får användaren ange hur mycket data som skall sparas om. För varje extra sampel som sparas i textformat desto mer otymplig blir filen, så det är en avvägningsfråga hur mycket data som måste konverteras.

Copyright © 2005 David Grunditz

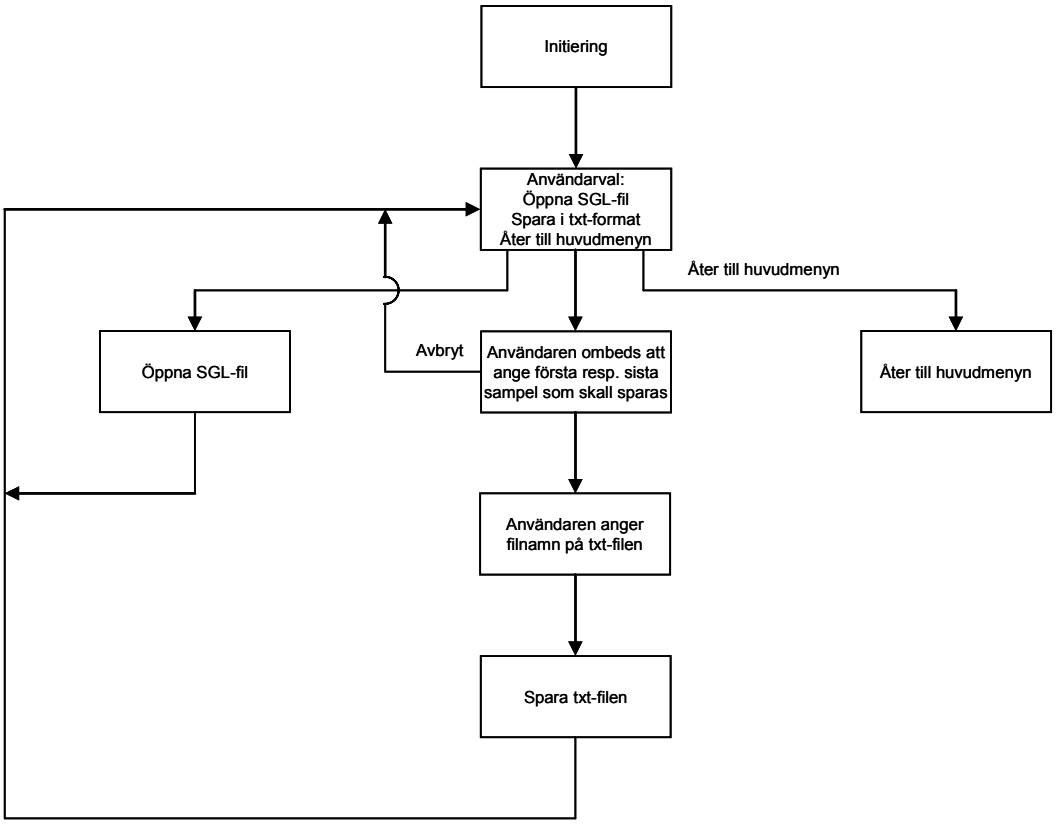


Bild 7.19 – sgl to txt.vi. sgl to txt.vi:s flödesschema

7.9.2 Användargränssnitt

Användargränssnittet består av ett diagram med zoommöjligheter för att man i detalj skall kunna avgöra vilka delar som skall konverteras, om alla data inte skall konverteras. När knappen *Spara i textformat* väljs ombeds användaren att ange vilka sampel som skall sparas. Det sker genom att ange första och sista samplet i diagrammet och bekräfta med att trycka på *OK*.

Copyright © 2005 David Grunditz

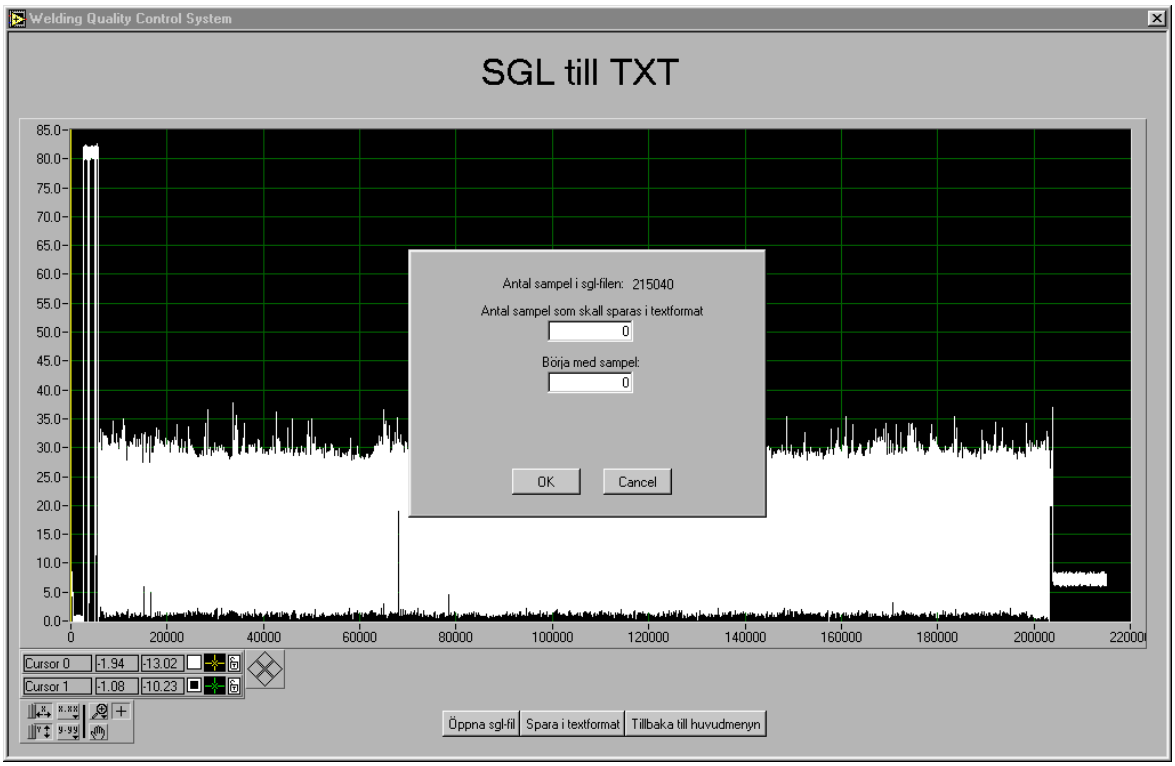


Bild 7.20 – SLG till TXT:s användargränssnitt

7.10 Om WQCS

Precis som alla program har även Welding Quality Control System ett *om* val. Genom att trycka på knappen *Om WQCS* så visas information om programmet, se bild 7.21

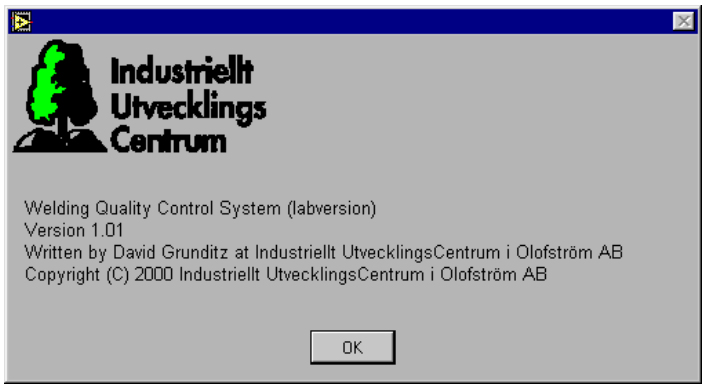


Bild 7.21 – Om WQCS

Copyright © 2005 David Grunditz

Copyright © 2005 David Grunditz

KAPITEL 8 – TEST AV WELDING QUALITY CONTROL SYSTEM

För att det skall vara möjligt att dra några slutsatser om de implementerade övervakningsalgoritmernas realtidsegenskaper så måste verkliga tester utföras. Den uppställning som använts för detta är den som finns beskriven i kapitel 5. Under testerna har även utfallet från SPRT2-, SPRT3- och SPRT4-algoritmerna studerats för att se om de är lämpliga att använd vid helautomatiserad svetsning.

8.1 Utförande

De implementerade övervakningsalgoritmernas testades genom att flera svetsprov svetsades. Svetsproven bygger på att två plåtbitar, cirka 2 – 3 mm tjocka, svetsas samman med en källfog. Den ena plåtbiten monteras liggande i svetsfixturen medan den andra stående, se bild 8.1 och 8.2.

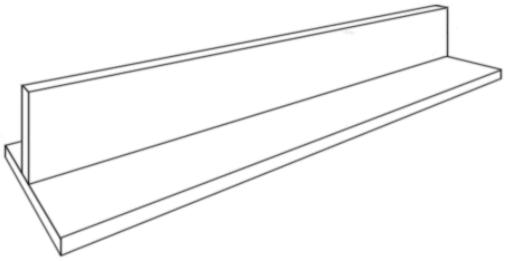
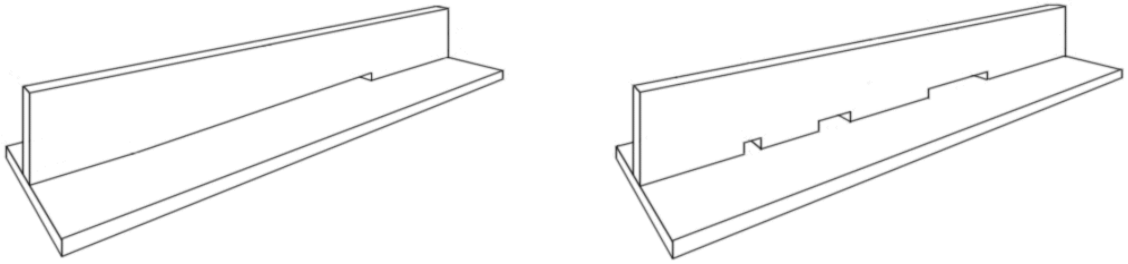


Bild 8.1 – Svetstest konfiguration; en stående plåtdetalj ovanpå en liggande plåtdetalj

Bild 8.2 – Plåtdetaljerna monterade i svetsfixturen

Den stående plåtbiten finns i tre olika versioner. En för felfri svetsning samt två för simulering av svetsfel. Plåtarna som är avsedda till för simulering av svetsfel är preparerade med en ramp eller med tre gluggar (pulser) med varierande bredd, se bild 8.3a och 8.3b.

Copyright © 2005 David Grunditz



(a) Stående plåtdetalj preparerad med ramp (b) Stående plåtdetalj preparerad med gluggar

Bild 8.3 – Stående plåtdetaljer för simulering av svetsfel

8.1.1 Svetsningen

Alla tester inleds med att den stående och den liggande plåtbiten fixeras genom att roboten häftar samman dem med två svetspunkter enligt pilarna i bild 8.5.

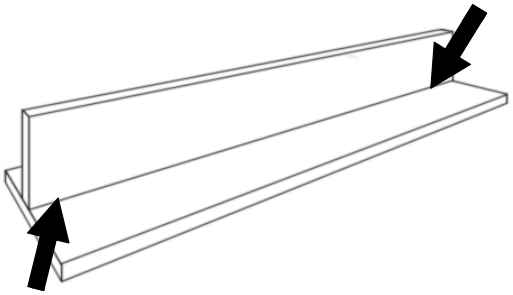


Bild 8.4 – Plåtarna fixeras med svetspunkter där pilarna pekar

Häftningen är till för att fixera de två detaljerna så att de inte skall kunna röra sig i förhållande till varandra under den slutliga svetsningen då övervakningssystemet testas.

Övervakningssystemet är inte helautomatiskt vilket innebär att det måste sättas i övervakningsläge, så att systemet reagerar på triggningssignalen från roboten och börjar övervaka svetsprocessen. Hela förloppet är enligt följande:

1. Montera plåtbitarna i fixturen

2. Lämna robotens arbetsområde.
   En industri robot i arbete har en enorm kraft som kan skada och till och med döda personer som står i robotens arbetsområde.

3. Aktivera ljusridån.
   Ljus ridån används för att stoppa robotens arbete i fall någon person av misstag skulle komma in i robotens arbetsområde.

4. Fixera plåtdetaljerna genom att roboten häftar samman dem.

5. Sätt övervakningssystemet i övervakningsläge.

6. Starta svetsprocessen, genom att trycka på en startknapp på styrskåpets kontrollpanel.

Copyright © 2005 David Grunditz

7. Svetsprocessen har startats och samtidigt har en triggningssignal skickats till övervakningssystemet som övervakar svetsningen.

8.2 Test av övervakningssystemet

För att testa övervakningssystemets egenskaper måste övervakningsparametrarna vara bestämda. De bestäms genom att låta roboten svetsa fem referensdetaljer utan några simulerade svetsfel. Det är viktigt att robot och svetsaggregat är optimerade för att lägga svetsar av högsta möjliga kvalité, så att kvalitetsspridningen mellan referenserna blir så liten som möjligt. De fem referenssvetsningarna övervaknas av övervakningssystemet, som samlar in data om referenserna och beräknar övervakningsparametrarna som skall användas till testerna.

8.3 Test av övervakningsalgoritmernas realtidsegenskaper

Testningen av övervakningssystemets realtidsegenskaper har skett genom att visuellt betrakta svetsprocessen och samtidigt jämföra med övervakningssystemets utfall. Hela testmetoden bygger på hur en användare av övervakningssystemet upplever händelse förloppet. Det är slutligen operatörens emotionella upplevelse av övervakningssystemet som avgör ifall det anses fungerar eller ej.
Att visuellt betrakta och jämföra händelseförloppet mellan övervakningssystemet och svetsprocessen är möjligt tack vare att när svetsfel uppstår ändras ljusbågens egenskaper, vilket är både synbart och i vissa fall även hörbart.

8.3.1 Utfall

Under samtliga tester som utförts har övervakningssystemet följt svetsprocessens händelseförlopp på ett tillfredställande sätt. Det har inte varit möjligt att visuellt notera någon form av eftersläpning hos övervakningssystemet. Det får anses att den PC-baserade implementationen av övervakningssystemet arbetar i realtid. Det skall även påpekas att det är 16 SPRT-algoritmer samt fyra FIR-filteralgoritmer som arbetar parallellt, om endast utfallet från fyra av dessa är möjliga att betrakta under övervakning. Det visar att den PC-baserade lösningen har godkapacitet och det är troligt att ett produktionssystem endast kommer att baseras kring en eller två av de fyra olika SPRT-algoritmerna.

8.4 Test av SPRT 2

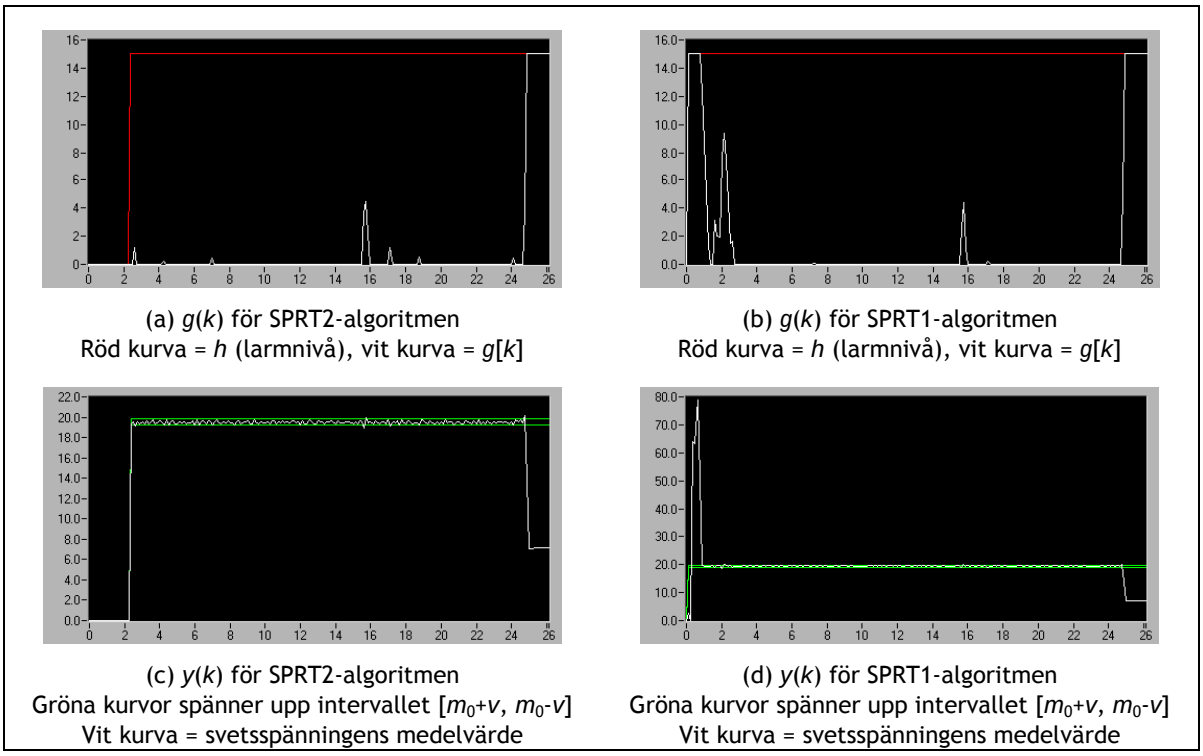
Det har inte gjorts någon noggrannare studie av hur SPRT2-algoritmen fungerar. För att göra det och dra några slutsatser måste en genomtänkt testserie tas fram, där simulering av stigande respektive minskande $y(k)$ värden genereras genom att robotens och svetsaggregatets inställningar förändras.
Dock har det under testerna av övervakningssystemet uppstått misslyckanden som beror på den mänskliga faktorn. Vid några av dem har svetsspänningen reducerats. Genom att använda dessa oönskade spänningsfall går att grovt analysera SPRT2-algoritmen.

Som tidigare nämnts så används de första 5120 samplena för att bestämma m_0. I praktiken är det de 5120 samplena mellan sampel 13312 och sampel 18432 som används för att bestämma m_0. Det beror på att de flesta sampel före sampel 13312 tillhör ljusbågständningen, som är en instabil process. Skulle m_0 bestämmas av dessa sampel skulle m_0 inte vara användbar för svetsövervakning.

Copyright © 2005 David Grunditz

8.4.1 Utfall

Trots att inga egentliga tester av SPRT2-algoritmen genomfördes så är det möjligt att dra några slutsatser genom att titta på insamlad data. Utifrån insamlad data, där svetsprocessen är normal, går det att konstatera att m_0, som bestämts av sampel från början av svetsen, tillsammans med v kommer att skapa ett fönster som ligger på ungefär på samma nivå som motsvarande fönstret för SRPT1-algoritm, se bild 8.6. Eftersom fönstret inte är identiskt med det för SPRT1-algoritmen så avviker $g_{SPRT2}(k)$ i förhållande till $g_{SPRT1}(k)$. Detta är möjligt att se i bild 8.6 där SPRT2-algoritmens $g(k)$ har några små spikar som inte finns hos SPRT1-algoritmens $g(k)$ och dessa får betraktas som försumbara.



(a) $g(k)$ för SPRT2-algoritmen
Röd kurva = h (larmnivå), vit kurva = $g[k]$

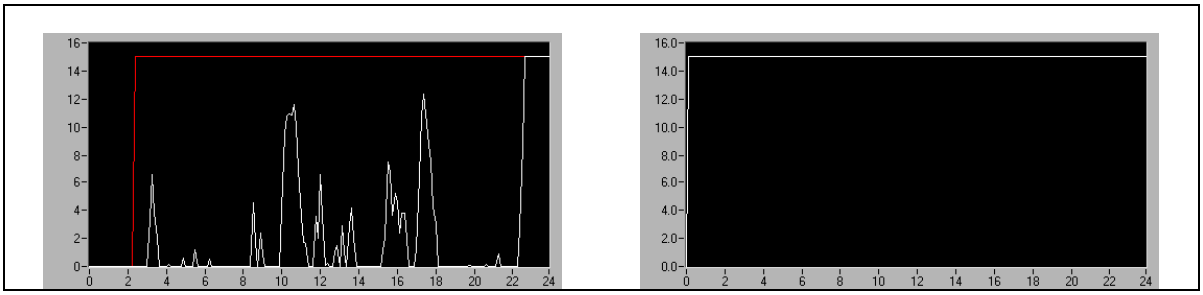
(b) $g(k)$ för SPRT1-algoritmen
Röd kurva = h (larmnivå), vit kurva = $g[k]$

(c) $y(k)$ för SPRT2-algoritmen
Gröna kurvor spänner upp intervallet $[m_0+v, m_0-v]$
Vit kurva = svetsspänningens medelvärde

(d) $y(k)$ för SPRT1-algoritmen
Gröna kurvor spänner upp intervallet $[m_0+v, m_0-v]$
Vit kurva = svetsspänningens medelvärde

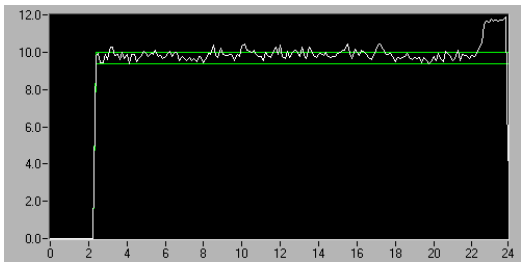
Bild 8.6 – Övervakning av svetsspänningen med SPRT1- och SPRT2-algoritmen (x-axlarna är i sekunder).

Tanken med SPRT2-algoritmen är att den skall kunna hantera förändringar där $y(k)$ har drivit iväg uppåt eller nedåt. Genom att i efterhand analysera misslyckade svetsningar där den mänskliga faktorn har missat att göra någon inställning på svetsaggregatet eller på roboten som resulterat till onormalt låg svetsspänning, är det möjligt att avgöra om SPRT2-algoritmen fungerar. Nedan i bild 8.7 så har svetsspänningen sjunkit från cirka 20 V, vid normal svetsning, till strax under 10 V. Av bilderna 8.7 a och c går det att se att SPRT2-algoritmen anpassar sig till den nya svetsspänningen och larmnivån nås först i svetsavslutningen då svetsspänningen alltid förändras för att nå tongångsspänningsnivån. I bilderna 8.7 (b) och 7.8 (d) larmar SPRT1-algoritmen under hela svetsförloppet, vilket är det riktiga i det här fallet.

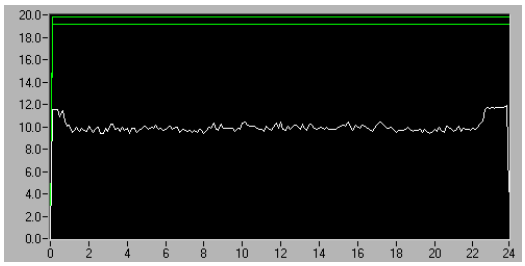


Copyright © 2005 David Grunditz

(a) $g(k)$ för SPRT2-algoritmen.
Röd kurva = h (larmnivå), vit kurva = $g[k]$



(b) $g(k)$ för SPRT1-algoritmen.
Röd kurva = h (larmnivå), vit kurva = $g[k]$



(c) $y(k)$ för SPRT1-algoritmen.
Gröna kurvor spänner upp intervallet $[m_0+v, m_0-v]$
Vit kurva = svetsspänningens medelvärde

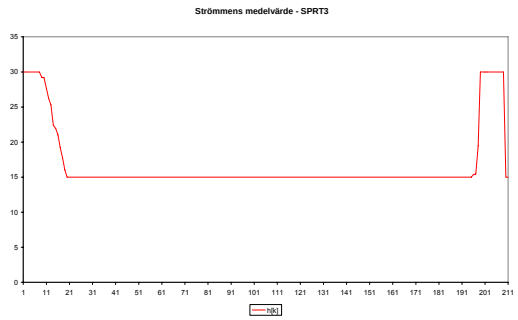
(d) $y(k)$ för SPRT2-algoritmen.
Gröna kurvor spänner upp intervallet $[m_0+v, m_0-v]$
Vit kurva = svetsspänningens medelvärde

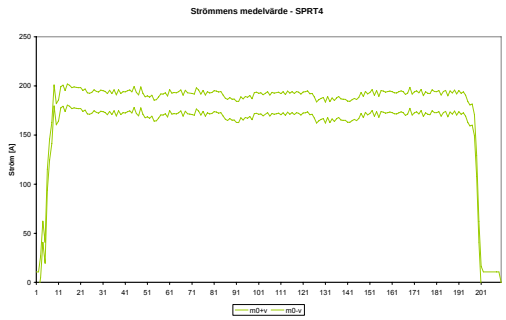
Bild 8.7 – Övervakning av spänningen med SPRT1- och SPRT2-algoriterna, då svetsströmmen sjunkit kraftigt i förhållande till idealsvetsen algoritmen (x-axlarna är i sekunder).

Detta resultat visar att tanken med SPRT 2 algoritmen fungerar, dvs. fönstret anpassas i fall den övervakade storheten stiger eller sjunker i förhållande till tidigare svetsningar. Men samtidigt visar bild 8.7 på en svaghet då den övervakade storheten förändras kraftigt på grund av ett alvarligt svetsfel. Det felet kommer inte SPRT2-algoritmen att registrera och larma för. För att hitta denna typ av fel måste SPRT2-algoritmen modifieras.

8.5 Utvärdering av SPRT3- och SPRT4-algoritmerna

Som underlag för utvärderingen har svetsdata använts från svetsningar där den stående plåtdetaljen har tre gluggar, se bild 8.3 (b). Detaljen används normalt för simulering av svetsfel, men här för att simulera naturliga systematiska variationer.
Från dessa svetsningar har svetsreferenser utvalts till underlag för beräkning av övervakningsparametrar, rörliga larmgränser och rörliga fönster. De beräknade larmgränserna och övervakningsfönstren som använts för utvärderingen visas i diagram 8.1 (a) och (b).





(a) Rörlig larmgräns, $h(k)$

(b) Rörligt fönstret, $[m_0(k)+v, m_0(k)-v]$

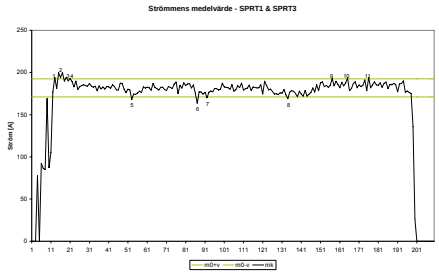
Diagram 8.1 – Diagrammen visar SPRT3-algoritmens rörliga larmgräns och SPRT4-algoritmens rörliga fönster som använts vid utvärderingen av algoritmerna. (x-axlarnas enhet är sampel)

Copyright © 2005 David Grunditz

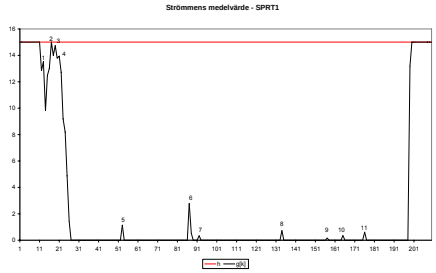
Grundidén med att bestämma SPRT1-algoritmens övervakningsparametrar utifrån svetsreferenser är att om det finns naturliga systematiska variationer så skall de påverka övervakningsparametrarna så att risken för falsklarm minimeras. Egenskaperna vid beräknandet av övervakningsparametrarna är sådana att om det finns endast en naturlig systematisk variation i svetsen och svetsen är kort, kommer variationen att ha stor påverkan på övervakningsparametrarna. Men ju längre svetsen är desto mindre påverkan kommer denna enda variation att ha på övervakningsparametrarna. Det betyder att risken för falsklarm ökar med svetsens längd.
Med SPRT3- och SPRT4-algoritmerna skall risken för falsklarm minskas genom att SPRT1-algoritmens övervakningsparametrar kombineras med rörlig larmgräns respektive rörligt fönster.

Övervakning av en detalj med simulerade naturliga systematiska variationer visas i diagramserien till höger, se diagram 8.2 (a) – (e) (Diagrammen finns även i slutet av kapitlet i stort format). Svetsströmmen som övervakas passerar ett antal gånger fönstergränserna. För SPRT1- och SPRT3-algoritmerna är svetsströmmen utanför fönstret elva gånger och för SPRT4-algoritmen tre gånger.
Den första överträdelsen inträffar precis efter att ljusbågen stabiliserats, se diagram 8.2 (a). Resultatet blir att $g_{SPRT1}(k)$ och $g_{SPRT3}(k)$ åter börjar stiga och för SPRT3-algoritmen blir $g_{SPRT3}(k)$ lika med $h_{SPRT3}(k)$, dvs. svetsfel. Denna händelse existerar inte vid övervakning med SPRT4-algoritmen, eftersom det rörliga fönstret är kompenserat för naturliga systematiska variationer.
Vid nästa överträdelse stiger åter $g(k)$, den här gången för alla tre övervakningsalgoritmerna, då inte SPRT4-algoritmen kan kompensera helt för den naturliga systematiska variationen. $g_{SPRT1}(k)$ och $g_{SPRT3}(k)$ blir lika med respektive algoritms larmgräns, dvs. svetsfel har detekterats.

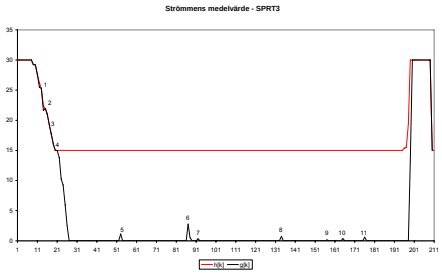
De nästkommande två överträdelserna berör endast övervakningen med SPRT1- och SPRT3-algoritmerna. Här bromsas $g_{SPRT1}(k)$



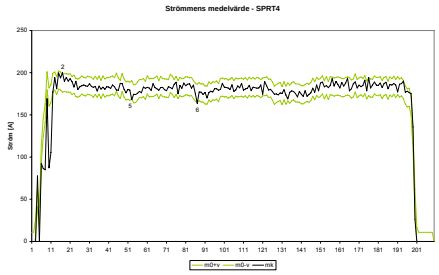
(a) - $i(k)$vid övervakning av strömmen med SPRT1-algoritmen.



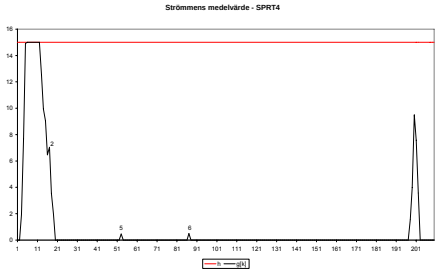
(b) – g(k)vid övervakning av strömmen med SPRT1-algoritmen.



(c) - g(k) vid övervakning av strömmen med SPRT3-algoritmen.



(d) - $i(k)$ vid övervakning av strömmen med SPRT4-algoritmen.



(e) – $g(k)$ vid övervakning av strömmen med SPRT4-algoritmen.

Diagram 8.2 - Övervakning av svetsström. Rödlinje = larmnivå. Grönt fönster = intervallet $[m_0+v, m_0-v]$ alt. $[m_0(k)+v, m_0(k)-v]$.

Copyright © 2005 David Grunditz

upp under halveringen, medan för $g_{SPRT3}(k)$, som sedan överträdelse 2 är lika med larmgränsen, händer inget eftersom $g_{SPRT3}(k)$ inte kan överstiga $h_{SPRT3}(k)$. $g_{SPRT3}(k)$ fortsätter att vara lika med larmnivån, $h_{SPRT3}(k)$, tills svetsströmmen åter går in innanför övervakningsfönstrets gränser, vid överträdelse 4.

Av de resterande överträdelserna är det endast överträdelserna 5 och 6 som berör alla tre algoritmerna. Det som skiljer är att $g_{SPRT4}(k)$ är mindre än $g_{SPRT1}(k)$ och $g_{SPRT3}(k)$, som är lika. Det beror på att SPRT4-algoritmens rörliga fönster kompenserar bäst för de naturliga systematiska händelserna.

Slutsatsen blir att SPRT4-algoritmen är bäst på att hantera naturliga systematiska variationer eftersom svetsströmmen passerar fönstrets gränser endast tre gånger. Sämst är SPRT3-algoritmen som även detekterade larm för händelserna 2, 3 och 4, och SPRT1-algoritmen är medelvägen med endast ett larm. Att notera är att SPRT1-algoritmen och SPRT3-algoritmen beter sig identiskt vid alla händelser efter händelse 4.

Men är detta verkligen sant? Fönsteröverträdelserna vid 1 – 4 är knappast större än överträdelsen vid 6 där algoritmerna inte detekterar något svetsfel. Men för de första händelserna detekteras svetsfel.

Om inte fönsteröverträdelsen vid 6 leder till något larm så bör inte heller 1 – 4 leda till larm, alltså handlar det om falsklarm. Orsaken till falsklarm beror på att $g(k)$ innan händelse 1 har ett högt värde som är så nära larmgränsen att en liten fönsteröverträdelse, orsakad av en naturlig variation, leder till att $g(k)$ ökar och blir lika med larmgränsen.

Frågan som uppkommer är: hur skall detta problem lösas? Som övervakningssystemet är nu, så övervakas hela svetsen från tändningen till släckningen av ljusbågen. Det innebär att under tändningen av ljusbågen är $g(k)$ skilt från noll och troligen lika med larmnivån. Det får till följd att $g(k)$ är skilt från noll då övervakningen av resten av svetsen, efter bågens stabilisering, skall påbörjas vilket innebär en förhöjd risk för falsklarm.
Problemet löses genom att sätta $g(k) = 0$ precis efter att ljusbågen stabiliserats. Alternativet är att inte övervaka tändningen av ljusbågen.

Genom att i Excel sätta $g(k) = 0$ under bågtändningsfasen och dra bort den påverkan som tändningen av ljusbågen har på $g(k)$, under de fyra första överträdelserna, så är det möjligt att få en uppfattning om hur övervakningen kommer att bete sig, då $g(k) = 0$ precis efter det att ljusbågen stabiliserats. Resultatet blir att SPRT1-algoritmen kommer att beräkna $g(k)$ till ett toppvärde på nästan 5 för de första händelserna och för SPRT4-algoritmen kommer händelse 2 inte ge ett större $g(k)$-värde än de övriga överträdelserna som SPRT4-algoritmen detekterar. Se diagram 8.3 och 8.4.

Copyright © 2005 David Grunditz

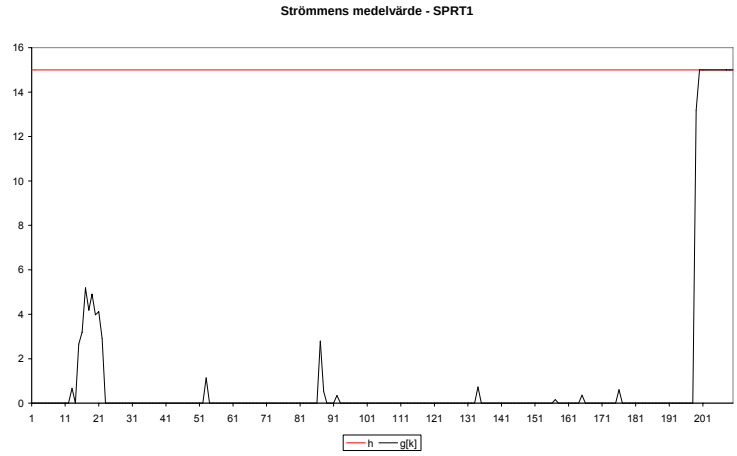


Diagram 8.3 – $g(k)$ (svart kurva) vid övervakning av strömmen med SPRT1-algortimen. $g(k)$ är modifierad i Excel så att bågtändningen inte skall ha någon påverkan på övervakningen. Röd linje = larmnivå.
(x-axlarnas enhet är sampel)

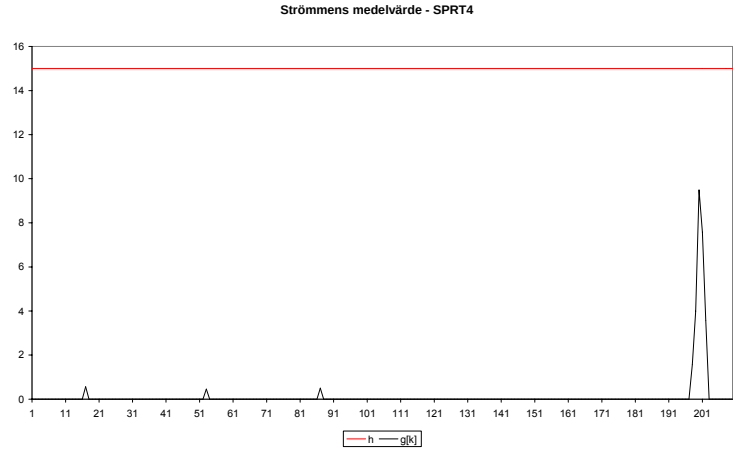


Diagram 8.4 – $g(k)$ (svart kurva) vid övervakning av strömmen med SPRT4-algortimen. $g(k)$ är modifierad i Excel så att bågtändningen inte skall ha någon påverkan på övervakningen. Röd linje = larmnivå.
(x-axlarnas enhet är sampel)

Genom att låta $g(k)$ vara lika med noll under tändningen av ljusbågen, innebär att den rörliga larmgränsen $h_{SPRT3}(k)$ kommer att vara lika med h_{SPRT1} under tändningen.
Det beror på att när $h_{SPRT3}(k)$ beräknas ($h_{SPRT3}(k) = h_{SPRT1} + g_{REF}(k)$) utifrån idealsvetsen sätts även $g_{REF}(k) = 0$ under ljusbågens tändning. Larmgränsens nivå, vid tändning av ljusbågen, har i praktiken ingen betydelse eftersom förloppet ej inte övervakas eftersom $g(k)$ är satt till noll.
Finns det inga kraftiga naturliga systematiska variationer hos idealsvetsen, som får $g_{REF}(k)$ att avvika från noll, kommer $h_{SPRT3}(k)$ att vara lika med h_{SPRT1} under hela svetsförloppet. Det innebär att då inga kraftiga naturliga systematiska variationer förekommer uppträder SPRT3-algoritmen och SPRT1-algoritmen identiskt.

I övervakningssituationer där $h_{SPRT3}(k)$ är kompenserad för kraftiga naturliga systematiska variationer och $h_{SPRT3}(k) = 2 \cdot h_{SPRT1}$, under längre tid än ett sampel, kommer SPRT3-algoritmen att larma. Det beror på att g(k) kommer att vara lika med $h_{SPRT3}(k)$, trots att avvikelsen finns i idealsvetsen. Orsaken är att när den rörliga larmgränsen $h_{SPRT3}(k)$ definieras så löper $g_{REF}(k)$ inte fritt utan begränsas av h_{SPRT1} och $h_{SPRT3}(k)$ blir är maximalt $2 \cdot h_{SPRT1}$. Det

Copyright © 2005 David Grunditz

betyder att vid övervakning av området, med den kraftiga naturliga systematiska variationen, kommer $g(k)$ att nå $h_{SPRT3}(k)$, som är för låg, och indikera svetsfel.
En lösning är att när den rörliga larmnivån bestäms låta $g_{REF}(k)$ löpa fritt uppåt och sedan addera till h_{SPRT1}. Denna idé måste testas och utvärderas innan någon slutsats om dess lämplighet kan dras.

8.5.1 Slutsats

Sammanfattningsvis är SPRT1-algoritmen och SPRT3-algorimen i praktiken identiska, så kvar är endast SPRT1-algoritmen och SPRT4-algoritmen. Av dessa två algoritmer kompenserar SPRT4-algoritmen bäst för naturliga systematiska variationer.
För att minska risken för falsklarm bör $g(k)$ sättas till noll efter att ljusbågen stabiliserats.

Den slutliga frågan som kvarstår är: vilken algoritm skall användas eller går båda algoritmerna att använda? Ett svar är att SPRT1-algoritmen är lämplig för övervakning av kortare svetsar där den naturliga systematiska variationen får större påverkan på övervakningsparametrarna. Är svetsen längre filtreras naturliga systematiska variationer bort vid beräkning av övervakningsparametrarna och kompensationen för de naturliga svetsarna måste ske mer handgripligt som i SPRT4-algoritmen där fönstren anpassas efter dessa variationer. Därför är SRPT4-algoritmen lämpligare att användas för övervakning av längre svetsar.

Copyright © 2005 David Grunditz

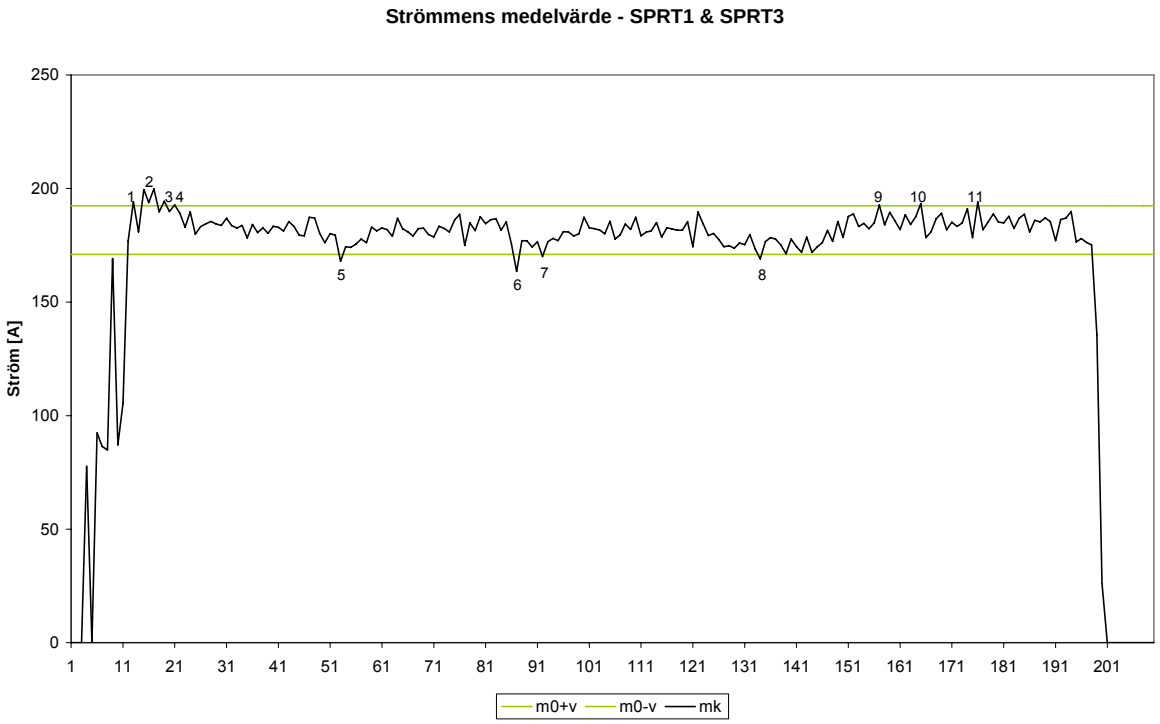


Diagram 8.2 (a) - $i(k)$ (svart kurva) vid övervakning av strömmen med SPRT1-algoritmen. Gröna kurvor spänner upp fönstret $[m_0+v, m_0-v]$. (x-axelns enhet är sampel)

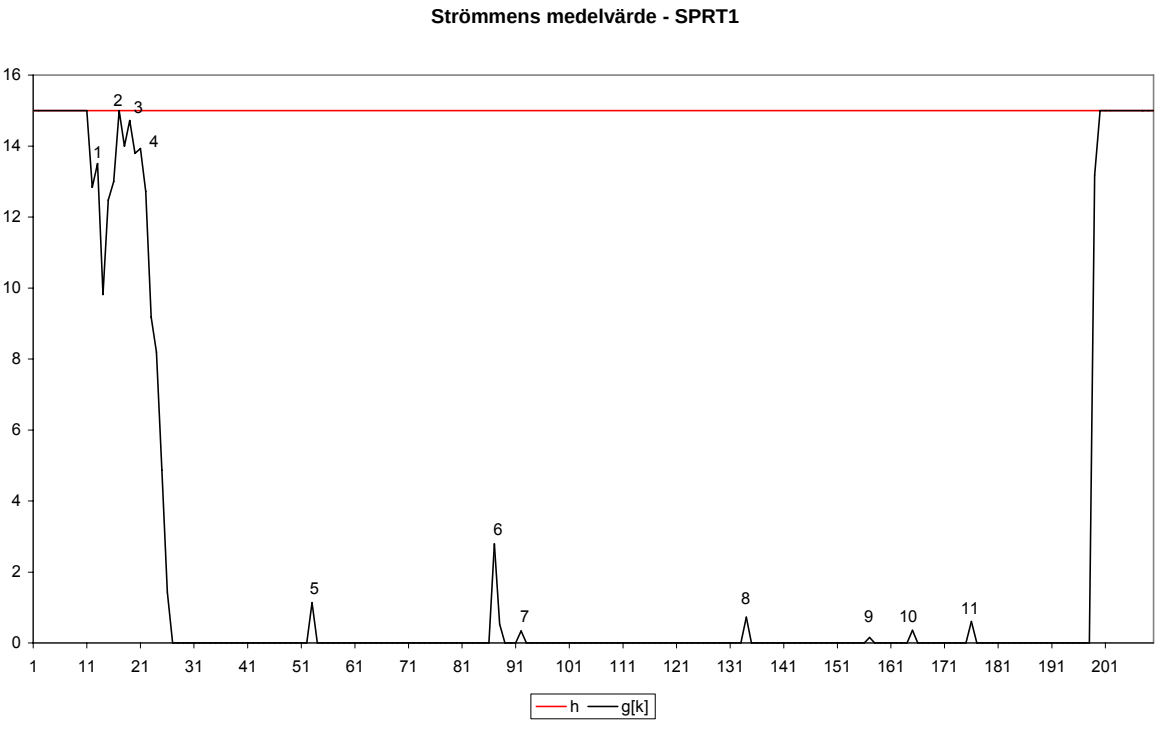


Diagram 8.2 (b) – $g(k)$ (svart kurva) vid övervakning av strömmen med SPRT1-algoritmen. Röd linje = larmnivån, h. (x-axelns enhet är sampel)

Copyright © 2005 David Grunditz

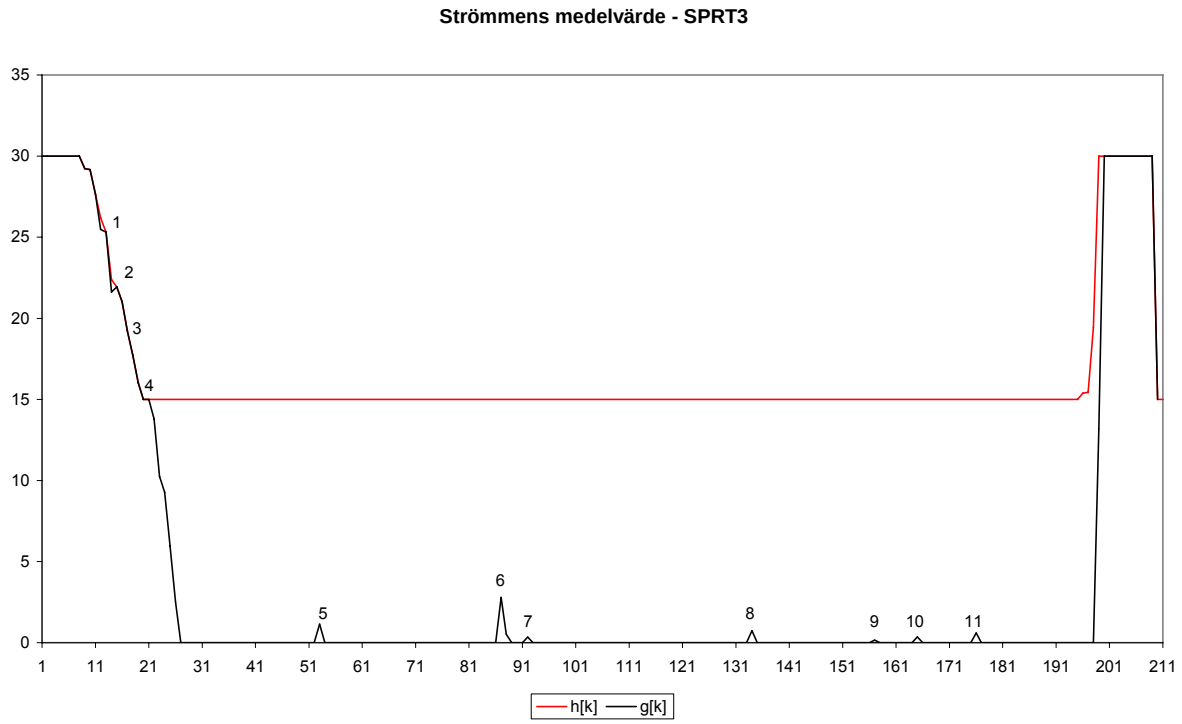


Diagram 8.2 (c) - $g(k)$ (svart kurva) vid övervakning av strömmen med SPRT3-algoritmen. Röd linje = larmnivån, h. (x-axelns enhet är sampel)

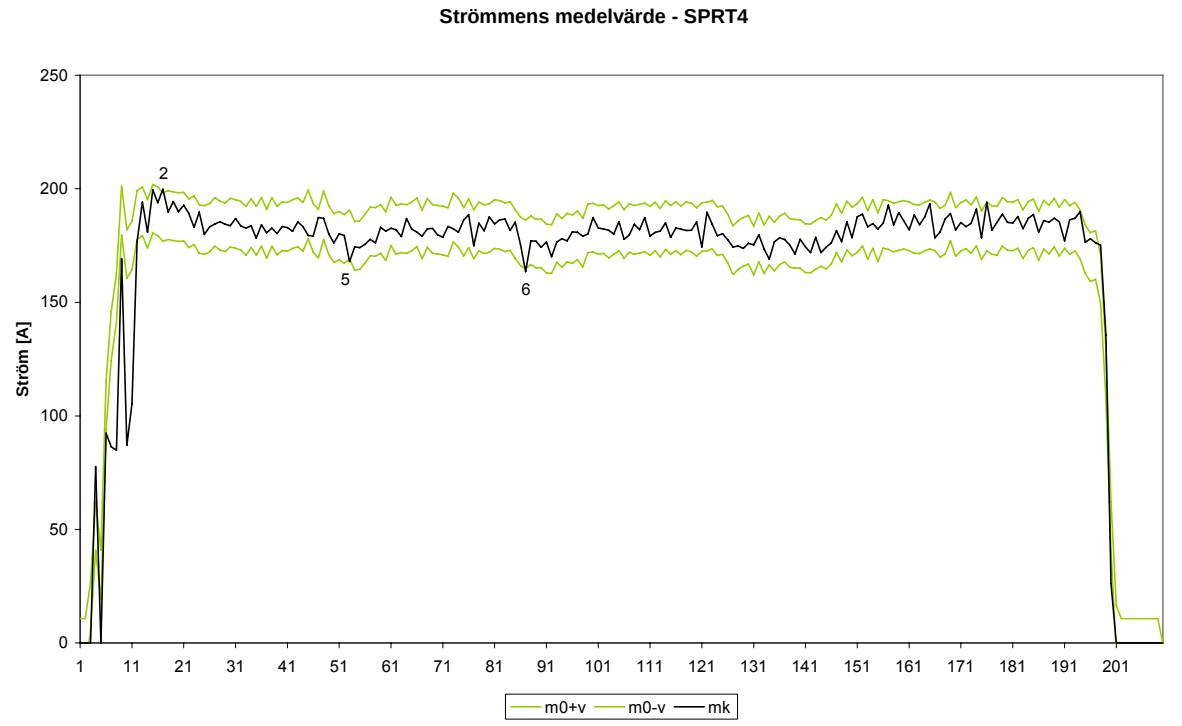


Diagram 8.2 (d) - $i(k)$ (svart kurva) vid övervakning av strömmen med SPRT4-algoritmen. Gröna kurvor spänner upp fönstret [m_0+v, m_0-v]. (x-axelns enhet är sampel)

Copyright © 2005 David Grunditz

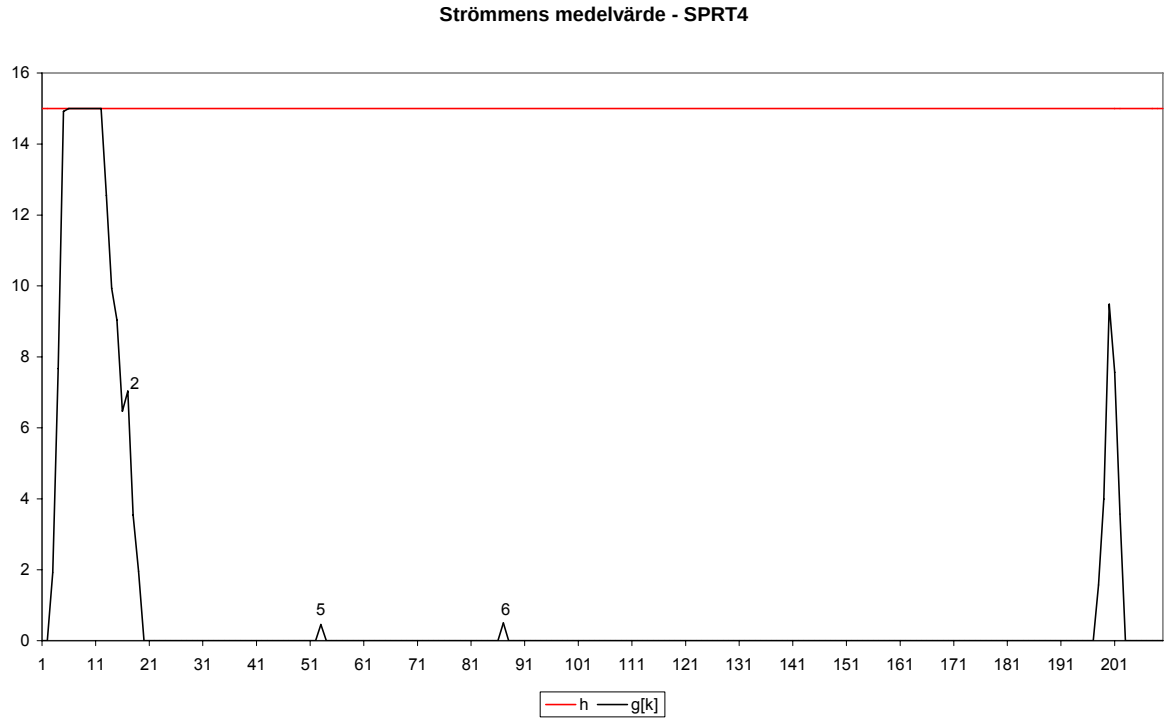


Diagram 8.2 (e) – *g*(*k*) (svart kurva) vid övervakning av strömmen med SPRT4-algoritmen. Röd linje = larmnivån, *h*. (x-axelns enhet är sampel)

Copyright © 2005 David Grunditz

REFERENSER

[1] Carl-Erik Olsson (2001): Optimering av skjuvbelastad kälsvets. I: Högproduktiv robotsvetsning och Högproduktiva svetsmetoder. Växjö: Teknikcentrum, Kronoberg. VERKTEK-seminarium, 14 februari 2001, Växjö.

[2] Air Liquide (1997): Teknikhandbok för TIG-, MIG- och MAG-svetsning. Air Liquide Gas AB

[3] Berthold Lundqvist (1980): Sandvikens handbok - Svetsning. Sandviken AB och Almqvist & Wiksell Förlag AB

[4] National Instruments Corporation (2000): The measurement and automation catalog 2001

[5] Stefan Adolfsson (1998): Automatic quality monitoring in GMA welding using signal processing methods. Lund: Department of production and materials engineering, Lund University

[6] M. Basseville & I.V. Nikiforov (1993): Detection of Abrupt Changes, Theory and Application. Prentice-Hall

Copyright © 2005 David Grunditz