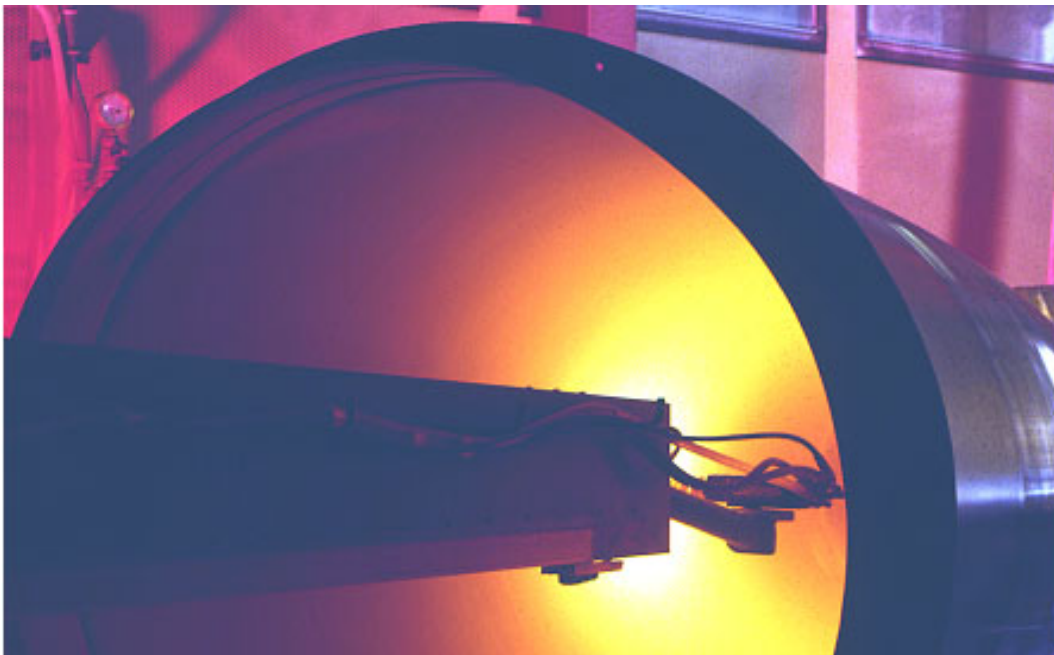


Uppgradering av robotsystem på Termisk sprutning, Volvo Aero

Charlotta Johansson

Petra Björnberg



EXAMENSARBETE

Maskiningenjör med inriktning mot produktionsteknik
Institutionen för teknik, matematik och datavetenskap

Uppgradering av robotsystem på Termisk sprutning, Volvo Aero

Sammanfattning

Examensarbetet resulterade i en mall för hur processen Termisk sprutning kan styras genom robotprogram. Dagens robotar med program behöver bytas ut och därför behövs en programstruktur som kan matchas med dagens nya robotsystem. Arbetet är avgränsat till att bara gälla strukturerad programmering på Termiskt Sprut Centers (TC) nya robotar med programspråket Rapid. Underlag till programinnehåll togs fram genom intervjuer med berörda personer på Termisk sprutning, studier av processparametrar för Termisk sprutning och befintlig specifikation på Volvo Aero Corporation (VAC).

Rapporten beskriver kortfattat processen Termisk sprutning och de fyra metoderna som används på VAC.

Litteraturstudier inom Jackson Structured Programming, Human Machine Interface och strukturerad programmering har gjorts och använts till att jämföra, utvärdera och ge förslag på hur cellen för Termisk sprutning kan förbättras.

Ett fungerande programförslag för menysystem till robot på Termisk sprutning har tagits fram. Detta skall ligga permanent i roboten för att ladda upp detaljprogram från en nätverksuppkopplad dator. Ett detaljprogram har tagits fram, dock saknas alla rörelser och kommunikation med processdator i detta. Testkörning av programmet har således endast genomförts på en robot som inte är avsedd eller utrustad för Termisk sprutning.

De nya detaljprogrammen visar sina parametrar på skärmen så att de kan kontrolleras mot operationsunderlaget. Till detaljprogrammet har en programmeringshandledning tagits fram, denna presenteras i rapporten.

Författare:	Charlotta Johansson, Petra Björnberg		
Examinator:	Per Nylén		
Handledare:	Martin Andersson Volvo Aero Corporation Kjell Hurtig Högskolan Väst		
Program:	Maskiningenjör med inriktning mot produktionsteknik		
Ämne:	Produktionsteknik	Examensnivå:	Kandidat
Datum:	2006-06-02	Rapportnummer:	2006:M12
Nyckelord:	Robot, Strukturerad programmering, JSP, Termisk sprutning, Volvo Aero Corporation, HMI		
Utgivare:	Högskolan Väst, Institutionen för teknik, matematik och datavetenskap, 461 86 Trollhättan Tfn: 0520-22 30 00 Fax: 0520-22 32 99 Web: www.hv.se		

Robot system upgrading on Thermal Spray Center, Volvo Aero

Summary

This thesis work resulted in a template for how the thermal spray process can be controlled through robot programs. The robots today and their programs need to be replaced, therefore a new structure which can be matched with the new robot systems is needed. The work is limited to only concern structured programming on the new robots on Thermal Spray Centre (TC) with the programming language Rapid. The basic data for the program was retrieved from interviews with concerned personnel at TC, from studies on process parameters for thermal spray and from existing specifications at Volvo Aero Corporation (VAC).

The thesis work briefly describes the process thermal spray and the four methods used at VAC.

A literature survey on Jackson Structured Programming, Human Machine Interface and structured programming was made and used to compare, evaluate and suggest improvements for the thermal spray work cell.

A functional suggestion for a menu system in a robot program for the thermal spray robot was presented. This will be in the robot permanently for uploading of detail programs from a network connected computer. An example of such a detail program was made but without robot movement and communication with the process computer. Testing of the programs has therefore only been made on a robot not suited for thermal spraying.

The new detail program shows its parameters directly on the screen so they can be controlled against the operation papers.

A programmer's guide to the detail program has been made which is presented in the report.

Author:	Charlotta Johansson, Petra Björnberg		
Examiner:	Per Nylén		
Advisor:	Martin Andersson Volvo Aero Corporation Kjell Hurtig Högskolan Väst		
Programme:	Mechanical Engineer in Production		
Subject:	Production Engineering	Level:	Bachelor
Date:	June 2, 2006	Report Number:	2006:M12
Keywords	Robot, Structured programming, JSP, Thermal spray, Volvo Aero Corporation, HMI		
Publisher:	University West, Department of Technology, Mathematics and Computer Science, S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00 Fax: + 46 520 22 32 99 Web: www.hv.se		

Förord

Detta examensarbete har utförts på Högskolan Väst, institutionen för teknik, matematik och datavetenskap, i samarbete med VAC avd 9933 och 9445. Examensarbetet omfattar 10p på C-nivå, och är avslutningen på vår utbildning till maskiningenjörer med inriktning mot produktionsteknik.

Examensarbetet tillkom genom ett tidigare projekt inom området Termisk Sprutning och genom samråd med Martin Andreasson och Christina Kann. Vi vill tacka dem för visat förtroende och intresse.

Vi vill framföra ett stort tack till alla som hjälpt oss med svar på frågor och med goda råd både på Termisk Sprutning och på Högskolan Väst. Ett särskilt tack vill vi ge till våra handledare Kjell Hurtig och Martin Andersson samt Mats-Olov Hansson, Krister Conradsson och Björn Kjellman på avdelning 9445.

Trollhättan den 2 juni 2006

Charlotta Johansson

Petra Björnberg

Innehållsförteckning

Sammanfattning	i
Summary.....	ii
Förord.....	iii
Innehållsförteckning.....	iv
Nomenklatur.....	v
1 Inledning.....	1
1.1 Bakgrund.....	1
1.2 Syfte och mål	1
1.3 Avgränsningar och metod.....	1
2 Företagsbeskrivning.....	2
3 Termisk sprutning.....	2
3.1 Processen.....	3
3.2 Processer på TC	3
3.2.1 Pulverflamsprutning.....	4
3.2.2 High-Velocity oxy-fuel process, HVOF	4
3.2.3 Ljusbågs/trådsprutning	4
3.2.4 Plasmasprutning	5
3.3 Cellens uppbyggnad.....	5
3.3.1 Utrustning.....	6
3.4 Funktionsbeskrivning, Cell Termisk sprutning	6
4 Förutsättningar	7
5 Tillvägagångssätt.....	7
5.1 Studiebesök Saab.....	8
6 Strukturerad Programmering.....	9
6.1 Jackson Strukturerad Programmering.....	11
7 Human Machine Interface	12
7.1 Multi Task cell VAC robotgradning.....	14
8 Resultat	14
8.1 Rekommendationer till fortsatt arbete	15
Källförteckning.....	16

Bilagor

- A. Termisk Sprutning
- B. JSP
- C. Mötesanteckningar
- D. Robotprogram Main
- E. Robotprogram Det Main
- F. Operatörshandledning detaljprogrammering
- G. Robotprogramspecifikation

Nomenklatur

Batch:	Serie av detaljer, från 2st och uppåt
DNC	Direct Numerical Control, en applikation på produktionsnätverket
Gradning:	Avlägsnande av skarpa kanter
HMI:	Human Machine Interface
IRB:	Industri Robot
Jogging	Manuell förflyttning kring robotens axlar m h a handkontrollen
JSP:	Jackson Structured Programming
Kermet	Sintrat material
Rapid:	Programspråk
S3, S4, IRC5:	System 3, System 4, System 5, ABB robotsystem
TC:	Termiskt Sprut Center
VAC:	Volvo Aero Corporation

1 Inledning

Rapporten är en summering av resultatet av examensarbetet ”Uppgradering av robotsystem på Termisk sprutning” som utförts på Volvo Aero under perioden 20 mars till 2 juni, 2006.

1.1 Bakgrund

Avdelning 9445 Termisk sprutning står inför ett arbete som innebär omfattande uppgradering av hårdvara. Dagens manipulerande utrustning består av industrirobotar med styrsystem ABB S3. Eftersom dessa är av en utgående modell medför detta svårigheter i att bibehålla tillgänglighet då service och anskaffande av reservdelar blir allt svårare.

1.2 Syfte och mål

Syftet är att utreda och dokumentera de grundläggande parametrar som måste tas om hand vid en styrsystemsuppgradering.

Målet är att presentera en mall för hur processen termisk sprutning ska styras genom robotprogram. En lösning skall presenteras som hanterar processparametrar och rörelser genom ett operatörsanpassat gränssnitt, tillsammans med en operatörsinstruktion alternativt utbildningsmaterial för operatörer. Även en mall för hur program för termisk sprutning ska struktureras enligt VAC's önskemål och tidigare upplägg i annan robotutrustning med gemensamt styrsystem skall presenteras. Ett illustrerande testprogram på befintlig utrustning med ABB S4/S5 styrsystem skall presenteras.

1.3 Avgränsningar och metod

Arbetet är avgränsat till att bara gälla strukturerad programmering på TC:s nya robotar med programspråket Rapid. Arbetet skall mynna ut i ett fungerande program som testas, dock endast på detaljer med enkel geometri, raka och plana ytor. De ordinära TC detaljerna kommer inte att programmeras. Anledningen till detta är att rörelsemodulerna kommer att skapas separat från ett off-line verktyg t ex robcad, vilket inte ingår i detta projekt.

Arbetet kommer att utföras parallellt med litteraturstudier inom områdena termisk sprutning, strukturerad programmering och HMI för att få fördjupade kunskaper inom ämnet. Fortlöpande intervjuer med berörd personal skall utföras för att identifiera krav och önskemål från TC.

Studera befintliga VAC dokument bland annat JSP för robotprocessen gradning, avdelningsinstruktioner och olika robotprogram.

Identifiering av befintliga parametrar i S3 genom studier av nuvarande robotprogram på TC för att strukturera upp och skapa ett nytt program med VACs önskade tillägg. Programmeringen kommer att ske med hjälp av strukturerad programmering enligt VACs allmänna specifikation för robotprogram.

Studiebesök på måleriet på Saab för att se hur deras sprutande utrustning fungerar.

2 Företagsbeskrivning

Volvo Aero ingår i Volvokoncernen. Företaget består av tre produktområden: Volvo Aero Services, Engine Services och Engines.

Volvo Aero Corporation som är moderbolag till Volvo Aero, ligger i Trollhättan.

I Trollhättan utvecklar och tillverkar Volvo Aero Corporation delar till flygplansmotorer, rymdraketer och gasturbiner. I Trollhättan görs också motorunderhåll på militära och civila motorer och gasturbiner.

Volvo Aero sysselsätter mellan 3000 och 3500 personer, varav cirka 2700 i Sverige, 430 i Norge och 200 i USA. Huvudkontoret ligger i Trollhättan men verksamhet bedrivs även i Stockholm, Malmö, Kongsberg (Norge) och i Boca Raton, Seattle samt Newington (USA). [1]

3 Termisk sprutning

Termisk sprutning är en process som skapar en detaljyta som har önskade material och fysikaliska egenskaper, t ex korrosionsskydd och värmeskydd [2]. Detta sker genom att material som keramer, kerment, plast och metall smälts helt eller delvis och samtidigt transporteras till detaljens yta [3].

På VAC beskrivs Termisk sprutning enligt följande (bilaga A):

”Termisk sprutning är en process där man sprutar smälta droppar av metall, keramik, plast eller karbider på en yta och på så sätt bygger upp en beläggning. Det påsprutade skiktet har vanligen helt skilda fysikaliska egenskaper i jämförelse med grundmaterialet.

Processen används idag till följande

- Slitageskyddande beläggningar vid reparation och i nyproduktion
- Korrosionsskyddande beläggningar
- Elektriskt ledande / isolerande skikt.
- Värmeskyddande skikt
- Återge felfärdiga detaljer rätt mått”[2]

3.1 Processen

Processen startar med att operatören bestämmer vilken operation som skall utföras, därefter bestäms gas- och pulverblandning genom ett så kallat recept. Pulverblandningen väljs beroende på vilka egenskaper som önskas på detaljens yta. Gasblandningen leds in i sprutmunstycket där den antänds, samtidigt leds pulvret in i lågan där det smälts och slungas iväg mot detaljen. [2]

Pulvret slungas iväg som små smälta droppar som träffar detaljytan där de fastnar och stelnar [2].

För att få rätt fäste mellan beläggning och detaljyta är det viktigt att den är fri från fett, olja och smuts. Ett vanligt sätt att rengöra detaljer är med ultraljudstvätt och avfettning. Det är viktigt att öka graden av mekanisk bindning mellan beläggning och detalj, detta görs med blästring, då ruggas ytan upp med hjälp av härdade stålkulor, aluminiumoxid eller silikonkarbid. [3] Aluminiumoxid används genomgående på TC, VAC oavsett detaljmaterial.

Termiskt sprutade skikt används även för att reparera utslitna detaljer, detaljer som blivit skadade i hanteringen och för att få rätt mått på detaljer som hamnat under minimum tolerans vid bearbetning. [3]

Fördelarna med Termisk sprutning är förbättrad teknisk prestanda och/eller ökad livslängd för komponenten. Skiktegenskaperna hos sprutade material är hårdare och har större motståndskraft mot utslitning och nötning jämfört med gjutna eller smidda legeringar. [3]

3.2 Processer på TC

På TC används fyra olika metoder inom Termisk (bilaga A:3). Tillvägagångssättet är dock lika för alla dessa processer.

När detaljen levereras till TC är den fri från smuts och olja. Detta kontrolleras vid maskering på TC, extra rengöring kan utföras på avdelningen, med aceton som avfettningsmedel. Arbetet startar därefter med maskering av de delar som inte skall beläggas på detaljen, därefter blästras detaljens yta. När det är klart flyttas detaljen till en cell för termisk sprutning, där den behandlas med värmeslag, bind- och toppskikt.

3.2.1 Pulverflamsprutning

Pulverflamsprutning (bilaga A:4) baseras på värme genererad av lågan från förbränningen av gas (oxy-acetylen, oxy-propen, oxy-hydrogen) för att smälta pulver. Partikelhastigheten är låg, 40-100 m/s, vilket ger ett skikt som har högre porositet än vad andra processer inom Termisk sprutning ger. Detta är bra när t ex turbinbladen ska slitas in för passning i turbinhuset eftersom porositeten hjälper till att försvaga den sammanhållande styrkan för skiktet och tillåter mikrosprickor vid kontakt med turbinbladet. [3]

3.2.2 High-Velocity oxy-fuel process, HVOF

HVOF system injicerar pulver i en ström av brinnande gas (bilaga A:5). Pulvret smälter eller mjukas upp och slungas mot ytan. HVOF resulterar i en hög bindstyrka med en kompakt genomtränglig mikrostruktur. [3]

Det finns även HVOF system som använder fotogen istället för gas, detta används inte på VAC.

3.2.3 Ljusbågs/trådsprutning

Ljusbågsprocessen för samman två elektriskt laddade trådar (en positiv och en negativ) med en bestämd vinkel (bilaga A:6). När de förs samman och spänning tillförs bildas en ljusbåge som smälter toppen på trådarna. Smältan pulvreras i ett ickeoxiderande gasflöde (ex kväve, helium eller argon). Partikelhastigheten blir mellan 50 och 150 m/s och temperaturen cirka 4000°C. [4]

Några av fördelarna med ljusbågsprocessen är att den är lätt att använda, underhålla och relativt lätt att lära sig. Nackdelen med ljusbågssprutningen är att skiktet ofta blir porösare med mer oxider och högre grad av osmält material kvar. Inom rymdtekniken används ljusbågssprutning för återskapande av dimensioner på olika komponenter. [4]

3.2.4 Plasmasprutning

Plasmaprocessen skapar en ljusbåge mellan en volframelektrod (katod) och en kopparanod (munstycke). Ljusbågens energi används till att separera och jonisera de gaser som används. För att förhindra att pistolen smälter kyls den med hjälp av kylvatten (bilaga A:7). [2]

En primärgas (ex kväve, argon) i kombination med en sekundärgas (ex väte, helium) värms upp och joniseras för att skapa en plasmaflamma. Plasmapistolen går på likström så när en ljusbåge bildas mellan elektrod och munstycke sluts kretsen och gaserna dislokaliseras och joniseras, plasma bildas. [4]

Några av fördelarna med plasmasprutning är den stora variationsmöjligheten med flamtemperaturer mellan 5000-25000°C och partikelhastigheter mellan 80 och 300 m/s. Det här möjliggör en mängd olika material- och partikelsammansättningar. Plasma förknippas ofta med höga temperaturer och därför kan skikt av keramer med hög smältpunkt skapas, dessa kan användas som värmebarriärer.[4]

3.3 Cellens uppbyggnad

På avdelningen för termisk sprutning finns fyra celler som är utrustade på likartat sätt. Skillnaderna består bland annat av variationer av pistoler, storlekar på maskiner och placeringar av robotar. Skillnaderna har inte någon betydelse för struktureringsarbetet. Begränsningar kommer dock att finnas när rörelseprogrammen läggs till beroende på ändlägen.

3.3.1 Utrustning

Sprutcellen består av en maskin utrustad med en IRB 2000 och ett rundbord där olika fixturer kan fästas, se figur 3.3.1. Roboten utrustas med ett sprutmunstycke, pistol och pulvermatare beroende på metod. Pulver- och gasmatning styrs av en processdator som kommunicerar med roboten. Roboten kommunicerar med en pc uppkopplad till VACs DNC nät. Cellen styrs av ett överordnat robotprogram som är skapat i S3.



Fig. 3.3.1 Sprutcell på VAC [2]

3.4 Funktionsbeskrivning, Cell Termisk sprutning

Cellerna på Termisk sprutning fungerar som CNC maskiner. Detaljen som skall sprutas laddas i och ur maskinen manuellt och allt förberedande arbete sker på likartat sätt. Det finns ingen bra automationslösning på arbetet på grund av maskeringens noggrannhetskrav. Operatören kommunicerar med cellen genom processdatorn via menyer som visar aktuella värden på bl a effekt, temperatur, gasflöde och spänning. Det finns däremot ingen kommunikation mellan operatör och robot. Det innebär att operatören inte kan se vad roboten gör eller hur fort den rör sig.

I dagsläget finns ingen kalibrering av sprutpistolens fysiska läge på roboten. Det innebär att det inte finns någon verifierbar noggrannhet på pistolens uppsättning.

4 Förutsättningar

På avdelningen 9445, Termisk sprutning finns ingen som är väl insatt i S4 eller rapid programmering

Inga programspecifikationer eller liknande dokument för Termisk sprutning finns tillgängliga utan all data måste samlas in innan strukturerings och programmeringsarbetet kan börja.

Handledare Martin Andersson på VAC kan programmera rapid, men har begränsade kunskaper om termisk sprutning.

Handledare Kjell Hurtig har programmeringsvana i rapid och kunskaper inom termisk sprutning.

Det är inte under arbetets gång fastställt vilka robotar och styrsystem som kommer att köpas in.

5 Tillvägagångssätt

Arbetet inleddes genom att följa en operatör under hela processen termisk Sprutning. För att identifiera vilka parametrar som styr processen och hur de styr den, studerades några av de nuvarande programmen och avdelningens instruktioner för att skriva ett nytt program till S3-robotar.

För att få fram en korrekt struktur, åskådliggjort som JSP, som kan godkännas studerades ett annat JSP från en VAC gradrobot. När det identifierats hur VACs standard för robotprogram ser ut och TCs särskilda processegenskaper identifierats kunde ett första JSP (bilaga B:1) skapas. Ett möte sammankallades för att få feedback på JSP1 från TC (bilaga C) Efter mötet gjordes några ändringar som resulterade i JSP2 (bilaga B:2). Utifrån det kunde skapandet av ett robotprogram påbörjas.

Parallellt med programmeringsarbetet gjordes litteraturstudier inom termisk sprutning, HMI och strukturerad programmering. Eftersom VAC använder sig av JSP har inga litteraturstudier gjorts om andra struktureringsmetoder för program. Även sökningar på mer specifika ord som S3, S4, IRC5 och rapid gjordes utan någon framgång. För att få ett så objektivt resultat som möjligt användes mycket få av VACs beskrivningar om termisk sprutning och sökningar gjordes i vetenskapliga databaser rekommenderade av Högskolan Väst.

För att få inblick i hur mycket som kan automatiseras gjordes ett studiebesök på VAC's nya "multi task cell" som kommer att fungera nästan utan operatörer. Operatörernas arbete kommer att bestå av monitorering av systemet. Systemet för realtidsövervakning beskrevs och demonstrerades. TC är även intresserade av att få en jämförelse mellan sin process och annan automationsutrustning som används för sprutning. Ett studiebesök gjordes på Saab Automobiles måleriverkstad i Trollhättan. Högskolans cell för termisk sprutning har inte varit med i jämförelsen. Dels för att

den är mer av en experimentcell och dels för att den redan står på TC där högskolans och VACs personal använder den tillsammans. Därför vet redan VAC hur den är uppbyggd och hur den fungerar.

Utöver de ursprungliga uppgifterna har även en robotspecifikation som VAC skall använda vid inköp av sina nya robotar tagits fram. Detta gjordes genom att precisera en mer allmän specifikation med exempel tagna ur VACs gradrobot. Den nya specifikationen innehåller både ett avsnitt av det nya robotprogrammet och ett framtaget JSP (bilaga B:4) utan annan text än i aktivitetsrutorna.

5.1 Studiebesök Saab

Studiebesök på Saab Automobiles måleri i Trollhättan ägde rum den 21 maj 2006.

Målet var att få kunskap om operatörsinterfacet och robotsystemet på måleriet. Guide var Thomas Zetterman, operatör på måleriet, och Mattias Axelsson, underhållstekniker, på Saab Automobile i Trollhättan. Genom frågor till dem kom följande information fram:

Beroende på respektive operatörs behörighetsnivå, skala 1-9, har respektive operatör sin egen behörighet att göra on-line ändringar i programmet. Behörighetsnivån är knuten till inloggningen på datorn. Ecopaint är ett visningssystem som kommunicerar med PLC och Bochportal.

Genom Ecopaint har Saab realtidsövervakning där sprutdata visas, t ex avstånd, karossmodell, färgmängd (ml/minut), luft-färg blandning.

Lackresultat kontrolleras m h a provplåtar som skruvas fast på en skrotbil, dessa analyseras sedan i ett laboratorium. Lackkontroll görs varje gång en helt ny bilmodell ska lackeras.

Kalibrering av sprututrustning sker m h a uppklistrade märken med pilar på väggar och på roboten se figur 5.1. Vid varje rast styrs sprutan till kalibreringsläget, maskinens pil och pilen på väggen ska peka mot varandra då maskinen är i rätt läge. Detta avläses manuellt så det krävs en skillnad på ca 1 cm för att operatören ska se att det behövs en justering av maskinens läge.



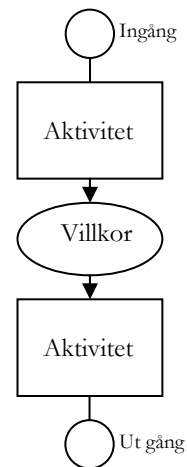
Figur 5.1
Kalibreringspilar

6 Strukturerad Programmering

Strukturerad programmering används för att få program som är lättare att testa, modifiera och åtgärda fel i jämfört med ostrukturerade program. [5]

För att bygga upp ett strukturerat program används olika sorters flödesdiagram. Den enklaste varianten finns beskriven i figur 6.1. För att få en metod som leder till strukturerade diagram har det bestämts att endast två olika sätt att skapa villkor används. Alla funktioner har endast en in- och utgång direkt kopplade mellan olika funktioner, detta kallas ”control statement stacking”. Det är också tillåtet att två eller flera olika funktioner flätas ihop. [5]

Det finns fyra enkla regler för att skapa ett strukturerat flödesdiagram, reglerna visualiseras i figur 6.2.



Figur 6.1 Enkelt flödesschema

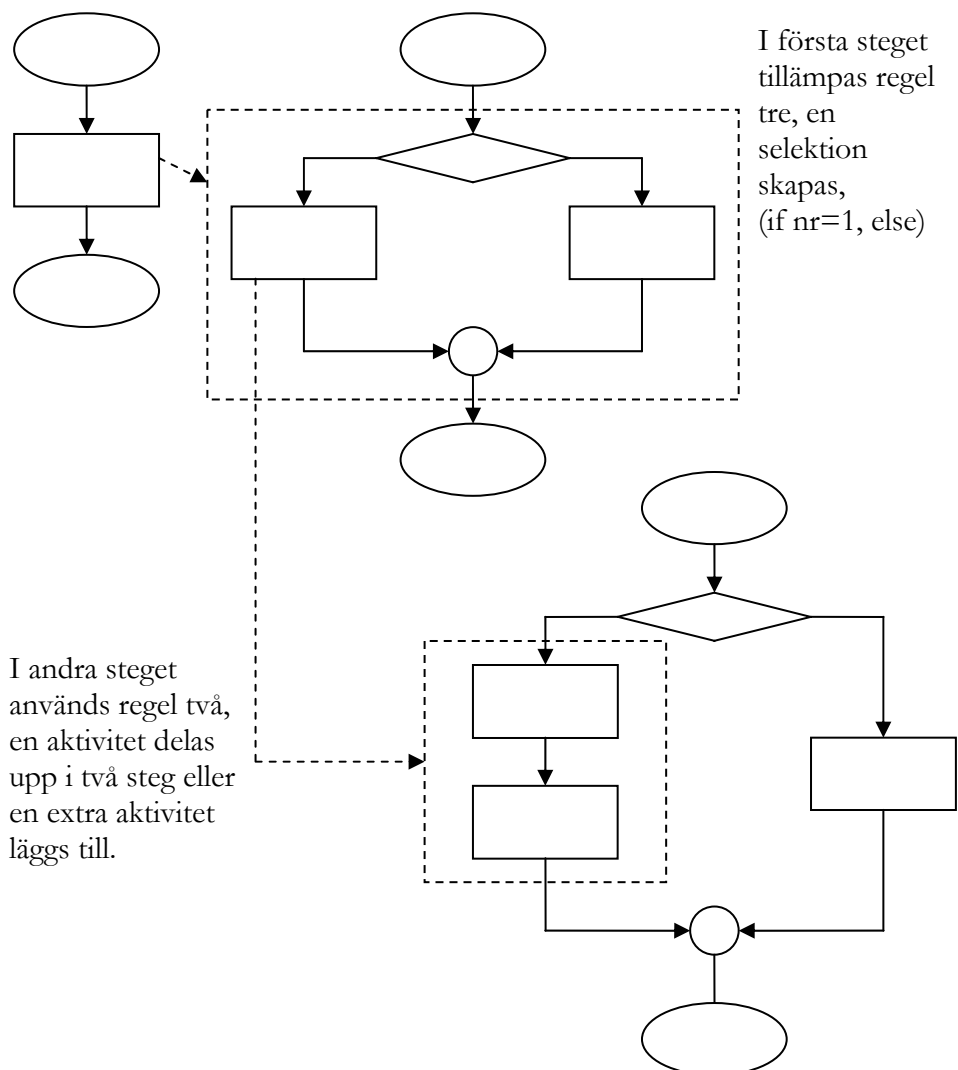
- 1 Börja med det enklaste flödesdiagrammet (figur 6.1)
- 2 Alla aktiviteter kan bytas ut mot två aktiviteter i sekvens
- 3 Alla aktiviteter kan bytas ut mot selektions- eller iterationsvillkor (if, while)
- 4 Regel 2 och 3 kan användas obegränsat antal gånger och i vilken ordning som helst.

Figur 6.2 Programmeringsregler [5]

För att tillämpa dessa regler krävs att man använder ”rektanglar” till allt, även in- och utgångar. Utförs skapandet av flödesdiagram enligt reglerna resulterar det i ett korrekt flödesdiagram som är lätt att följa.[5]

Att bara använda regel två flera gånger på det ursprungliga, enklaste, diagrammet genererar flera rektanglar i sekvens. På grund av detta kallas regel två för ”bygga på hög-regeln”. Regel tre kallas ”nesting rule”, för om den används på ursprungsdiagrammet blir den en hopvävd struktur.[5]

Kombineras reglerna två och tre (=regel fyra) kan stora och komplicerade flödesdiagram byggas upp, se figur 6.3.



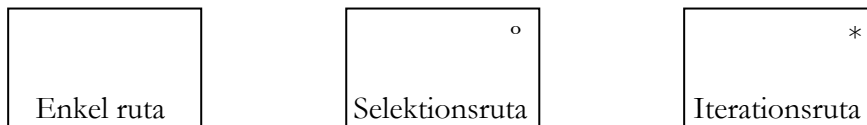
Figur 6.3 Flödesschema [5]

När reglerna följs bidrar det till att strukturerade flödesdiagram är lätta att följa och relativt lätta att generera program från. En anledning är att överlappningar, hopp (goto instruktioner), undviks. [5]

På detta sätt kan alla strukturerade diagram byggas upp och utifrån dem kan alla strukturerade program skapas. [5]

6.1 Jackson Strukturerad Programmering

Jackson introducerade sina strukturdiagram som en del av metoden ”Jackson structured programming”. I diagrammet används tre typer av rutor: en sekventiell, en selekterande och en itererande, detta illustreras i figur 6.4. Diagrammet skapas som en trädstruktur där varje förgrening kallas nod. Varje nod kan ha ett flertal underliggande rutor men en given nod har endast en överliggande ruta. [6]



Figur 6.4 Aktivitetsrutor för JSP [6]

Följande regler måste följas vid upprättande av strukturdiagram:

Alla rutor till en nod måste vara av samma typ, enkel, selekterande eller itererande.

Om en ruta är itererande får den överliggande rutan inte ha andra underliggande rutor kopplade till sig.

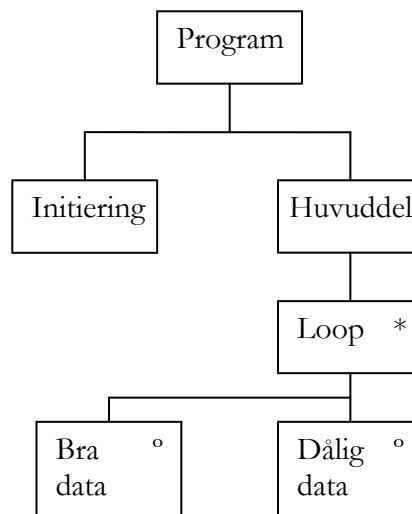
Om en underliggande ruta är valbar måste det finnas minst en ruta till, av typen selekterande, till den överliggande rutan.

Diagrammet måste ha en trädliknande struktur. Särskilt viktigt är att en överliggande ruta placeras högre upp i diagrammet än sin/sina underliggande ruta/rutor. [6]

Ett JSP illustreras i figur 6.5

Huvuddelen av informationen som är förknippad med Jackson diagram handlar om relationer mellan noder. Linjerna markerar endast involverade överliggande och underliggande rut-relationer. Följaktligen är formellt specificerade Jackson diagram en metod för att förstå vikten av dessa relationer och värdera dem i ett schema förbundna med noder. [6]

I vilket system som helst som tillhandahåller verktyg för utveckling av program kommer det en dag då de producerade programmen behöver modifieras, antingen för att rätta till fel eller för att lägga till eller ändra i systemets funktionalitet. Ingenting annat än mindre



Figur 6.5 JSP exempel [6]

korrigeringar bör vara tillåtet i programstrukturer. Ny funktionalitet och större förändringar i gammal funktionalitet kräver att ny datastruktur måste skapas och användas och att gammal datastruktur tas bort. [7]

Ändringar i en datastruktur som fortfarande är under konstruktion måste vara fritt tillåtet, men när det har blivit en programstruktur kan det bli problem, precis som för ändringar gjorda direkt i vilken programstruktur som helst. [7]

En av fördelarna med att använda en bra struktureringsmetodik som JSP är att den ger specifikationer som är lätta att följa, och därför blir det lätt att upptäcka fel och att genomföra ändringar. Om ändringarna inte är kontrollerade finns det emellertid en risk att specifikationerna blir förstörda istället för att förbättras. [7]

Det finns tre huvudgrupper av operationer i JSP. Dessa är tillägg, borttagning och ersättning. Operationen används för att skapa ändra eller ta bort någon funktion. När kravet på en ny funktion godkänts krävs att en ny datastruktur skapas och utvecklas. [7]

7 Human Machine Interface

interface ['intʃfe ¹ s] subst. 1 grännsområde, data. gränssnitt 2 samspel, anslutningspunkt, mötesplats, mötespunkt, kontaktyta, kontakter, <i>man-machine interfaces at the workplace</i> samspelet mellan människa och maskin på arbetsplatsen

Figur 7.1 Ur NE [8]

Interface definieras av Nationalencyklopedin enligt figur 7.1

För att möjliggöra mänsklig kommunikation med industrirobotar finns så kallade människa-maskin-interface. Interfacen kan bestå av både hård- och mjukvara. Genom robotens människa-maskin-interface kan roboten bland annat styras och programmeras till att utföra önskade uppgifter. [9]

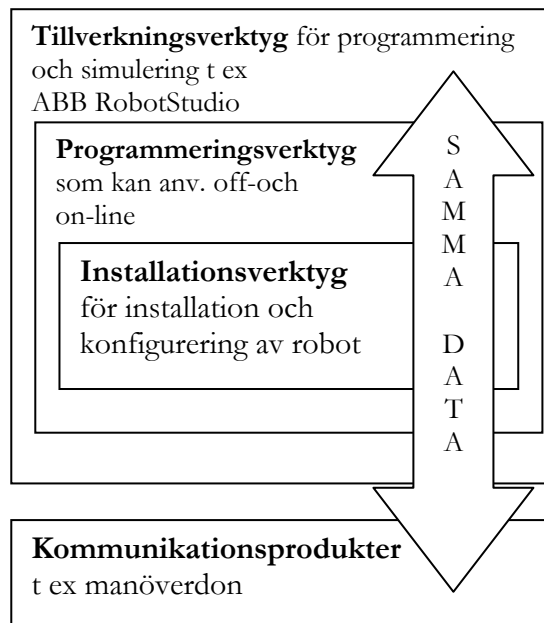
Det är viktigt att enkelt kunna skapa användargränssnitt som är specifika för de applikationer som robotcellen kommer att användas för. Det skall vara lätt att lägga till information som tillkommer efterhand med nya applikationer osv. Gränssnittet skall även kunna hjälpa till genom att hänvisa till inbyggd information i form av anvisningar. [10]

Robotprogrammering delas in i huvudkategorierna on-line-programmering och off-line-programmering. On-line-programmering innebär att roboten används för att skapa programmet. Vid off-line-programmering skapas programmet i en dator och laddas därefter in i robotens styrsystem för exekvering. Off-line-program kan testas och korrigeras på en grafiskt simulerad modell av roboten innan det körs i roboten. [9]

På VAC används Igrip och RobCAD vid off-line programmering, simulering och verifiering.

Med hjälp av den senaste teknologin inom människa-maskin-interface, robotsimulering och industriell informationsteknologi förbättras och förenklas robotanvändningen hela tiden. En viktig nyckel till förbättringar är att slå ihop off-line och on-line miljön så att det blir möjligt att använda samma verktyg i båda fallen. Användaren kan då arbeta med samma användargränssnitt och arbetsmetoder genom hela livscykeln, oberoende av om arbetet sker on-line eller off-line. [10]

Av figur 7.2 framgår det att alla verktyg som används i livscykelstödet är helt integrerade och tillgängliga från varandra. Program och data kan flyttas fritt mellan robot och virtuell robot. Användaren kan ladda ett program från robot till PC, ändra det och därefter testa det genom att använda den virtuella roboten och sedan ladda över det i roboten igen, utan att störa produktion. [10]



Figur 7.2 Dataflöde [10]

I produktionsfasen är operatören ansvarig för det oavbrutna arbetet i robotcellen. Om det uppstår fel måste han ha möjlighet att diagnostisera det, vidta korrigeringsåtgärder och återstarta produktionen så snabbt som möjligt. Underhållspersonalen ska serva och uppdatera system regelbundet. Det är viktigt att arbetet i alla dessa faser stöds av verktyg som kan användas på ett likartat sätt och arbeta med samma information både off-line och on-line. [10]

Ett grundkrav för att göra robotar enkla att använda är att anpassa programmeringsenheter, kontrollpaneler och andra operatörshjälpmedel för olika behov och olika kunskapsnivåer. Detta görs med hjälp av flexibla användarinterface som innebär att operatörsverktygen för att styra roboten i produktion ska vara så små och enkla som möjligt. [10]

7.1 Multi Task cell VAC robotgradning

På VAC pågår arbetet med att installera en cell som kan hantera ett flertal olika detaljer samtidigt. Cellen består av: ett verktygsmagasin (för 1200 Verktyg) med robot, 5 stycken fleroperationsmaskiner, portalrobot, 2 gradstationer, en manuell och en automatiserad, med robot. I- och urlastning från stationerna sker med automatik. Cellen övervakas med ett program som heter MMS Process, som visar var och vilka detaljer som befinner sig i maskinerna, i kö och vilka operationer som körs.

I maskinerna finns det uppsatta kameror som ”filmar” vad som händer i cellen. Filmen visas direkt på en datorskärm utanför maskinen. Än så länge finns ingen koppling till något verktyg, utan filmen visas och försvinner. Det vill säga att ingen feedback om robotpositioner eller dylikt sparas och kan därför inte användas i utvärderingssyfte.

Vid byte av verktyg mäts det in i en laser. Den sitter bakom roboten och är inte i vägen för när roboten rör på sig.

8 Resultat

Ett fungerande programförslag (bilaga D) för menysystem till robot på termisk sprutning har tagits fram. Detta skall ligga permanent i roboten för att ladda upp detaljprogram från en nätverksuppkopplad dator. Ett sådant har tagits fram (bilaga E), dock saknas alla rörelser och kommunikation med processdator i detta, på grund av att ingen robot utrustad för termisk sprutning på TC har rätt mjukvara. Testkörning av programmet har således endast genomförts på en robot som inte är avsedd eller utrustad för termisk sprutning.

Till detaljprogrammet har en programmeringshandledning (bilaga F) tagits fram och demonstrerats för operatörer på TC. De nya detaljprogrammen visar sina parametervärden på skärmen så att de kan kontrolleras mot operationsunderlaget

Även en ny robotprogramspecifikation för robotar på TC har tagits fram (bilaga G). Denna ingår i VAC:s robotspecifikation som skickats till leverantör.

8.1 Rekommendationer till fortsatt arbete

Vid kontakt med leverantör bör övervakningsfrågan i stort tas upp. Kan man få realtidsövervakning på roboten ev kopplad till ett verifieringsprogram? Visionkontroll av verktyg, sitter rätt pistol uppe, på rätt sätt? Sitter rätt detalj fastsatt på bordet?

Det är förberett för verktygskalibrering i programmet. robotleverantören bör få information om vad VAC, TC, vill ha för sorts kalibrering och vilka krav som ställs.

Central produktionsteknik på VAC undersöker vilka metoder för realtidsövervakning som kan passa i multitaskcellen. TC bör undersöka om inte de också vill ha något sådant. Krav på kameran är att den kan skärma bort UV ljuset från flammorna och att den inte är för känslig mot de förhållanden som finns .

Istället för att bestämma antal värmeslag kan det styras av en temperaturmätare (pyrometer), en sådan ändring går att göra i programmet.

Införande av behörighetsnivåer på operatörer/programmerare för ändringar i programmen på TC. Detta kan loggas och på så sätt får man en bra dokumentering av vad som gjorts och vem som gjort det. Behörighetsnivå kommer troligtvis att bli ett krav på TC. Det bör undersökas vilka möjligheter leverantören har att erbjuda. Detta kan jämföras med Saabs system för behörighetsnivåer.

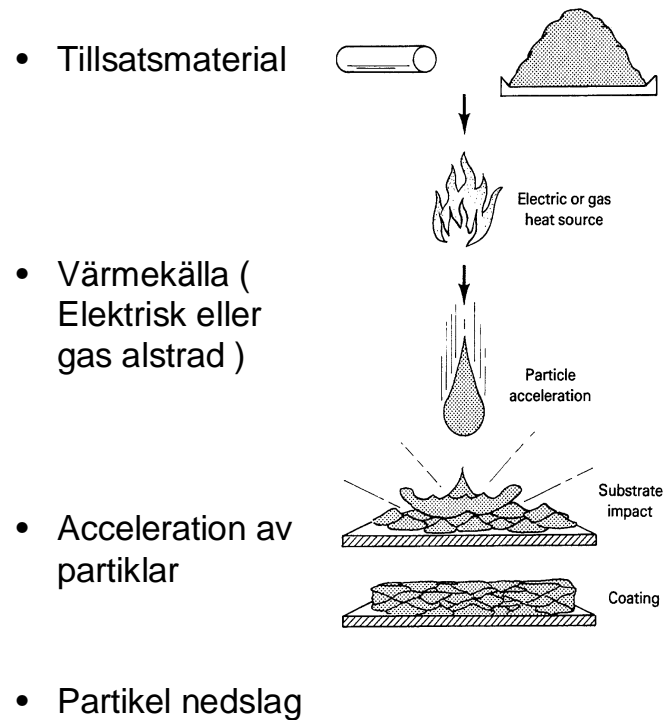
Det bör förberedas i huvudprogrammet och roboten för sensorsystemet DPV2000. DPV2000 kan mäta temperatur och hastighet på partiklar som sprutas. Med den kan bland annat optimala sprutavstånd fås fram. Den kan därför användas för att optimera detaljprogrammen.

Källförteckning

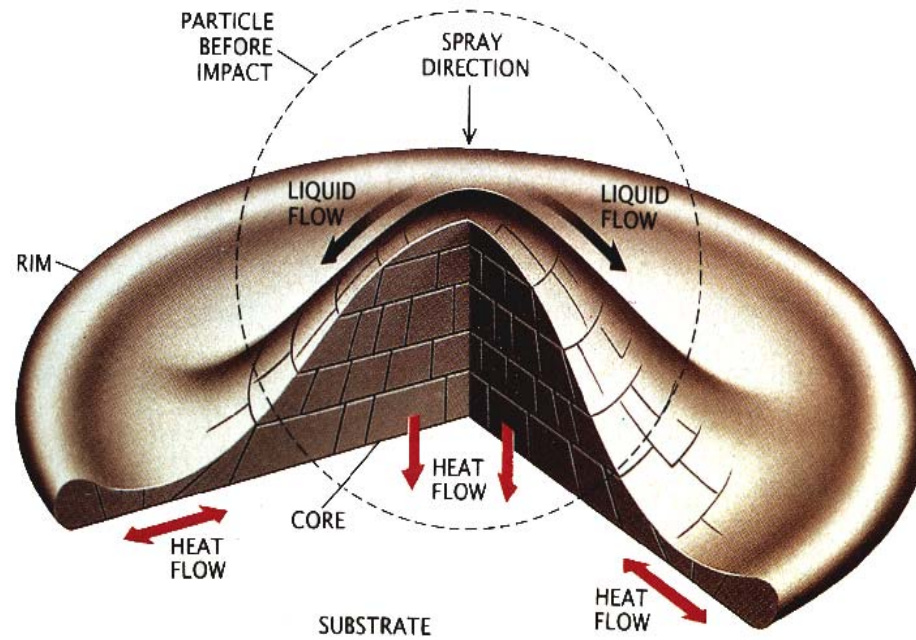
1. Volvo Aero (2004), *minifakta om volvo aero*, [Elektronisk]. Volvo Aero Tillgänglig: <http://violin.volvo.net/violinaero/corporate/en/business_functions/about_volvoaero/presentations/presentations.htm> [2006-05-16]
2. Mats-Olof Hansson(2002), *Utbildning Termisk Sprutning*, Volvo Aero. Trollhättan
3. Mitchell R Dorfman (2002), *THERMAL SPRAY BASICS*, [Elektronisk]. Advanced materials and processes Vol 160 issue 7 p 47-50. Tillgänglig: Academic Search Elite [2006-04-10]
4. Mitchell R Dorfman (2002), *THERMAL SPRAY PROCESSES*, [Elektronisk]. Advanced materials and processes Vol 160 issue 8 p 47-49. Tillgänglig: Academic Search Elite [2006-04-10]
5. H.M. Deitel & P.J. Deitel (2004). C How to program. 4 uppl. New Jersey: Prentice Hall Inc
6. Gee David M. (1998), Formal specification of visual languages, [Elektroniskt] Information and Software Technology, Vol 40 Issue 7 p 359-367, Tillgänglig: Science Direct [2006-04-06]
7. C.G: Davies &P.J Layzell (1988), *Rules to govern change in JSP-based systems*, [Elektronisk] Software Maintenance, Proceedings of the Conference on 24-27 Oct. 1988 Page(s):34 – 39 Tillgänglig IEEE Xplore [2006-05-25]
8. Sökväg till Nationalencyklopedins hemsida [Elektronisk] <http://www.ne.se/jsp/esse/web_translate.jsp?t_word=Interface> [2006-05-25]
9. E. Ilhan Konukseven, Anas Abibi (2004), Development of man machine interface software for an industrial robot, [Elektronisk] IEEE Xplore [2006-04-28]
10. Håkan Brantmark (2002), A New Concept for Human-Robot Interaction, Proceedings of the 33rd ISR (International Symposium on Robotics) October 7 – 11, 2002

A. Termisk Sprutning

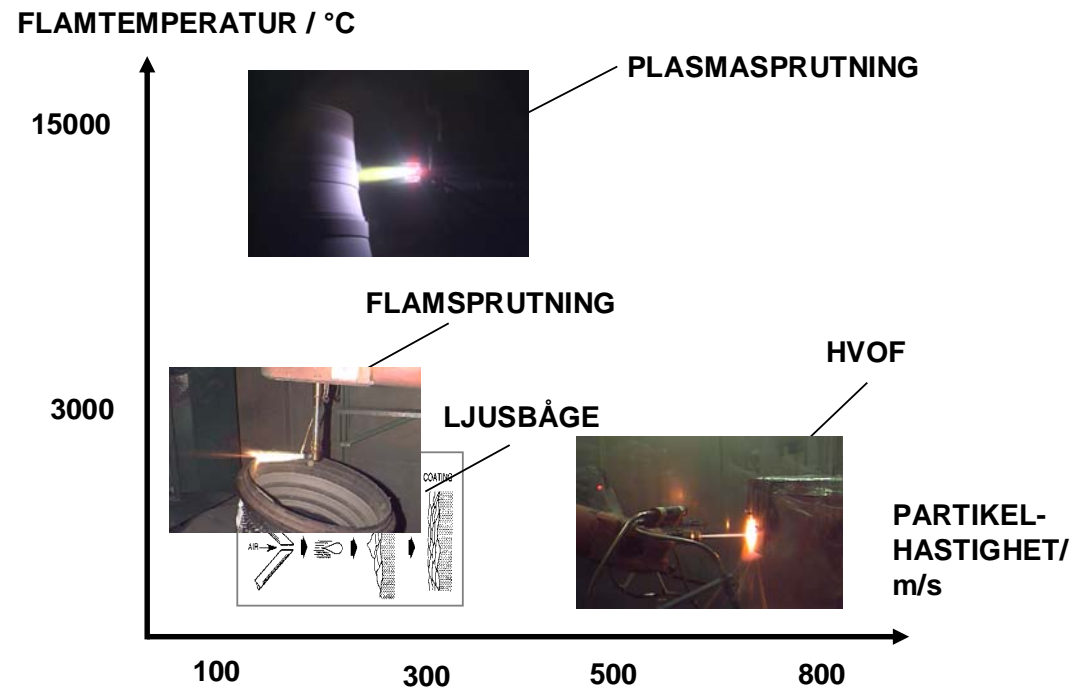
Termisk sprutning



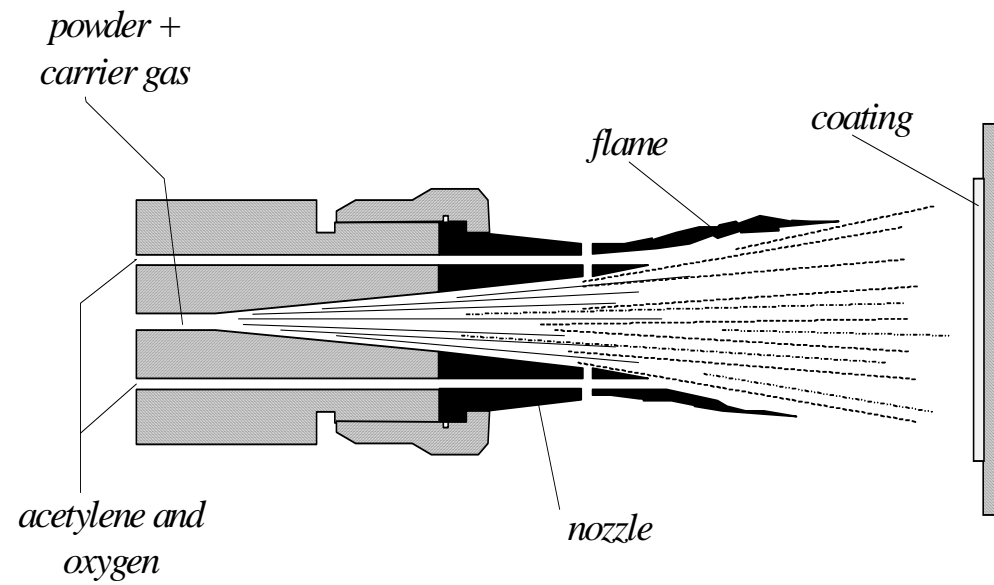
EN "SPLAT"



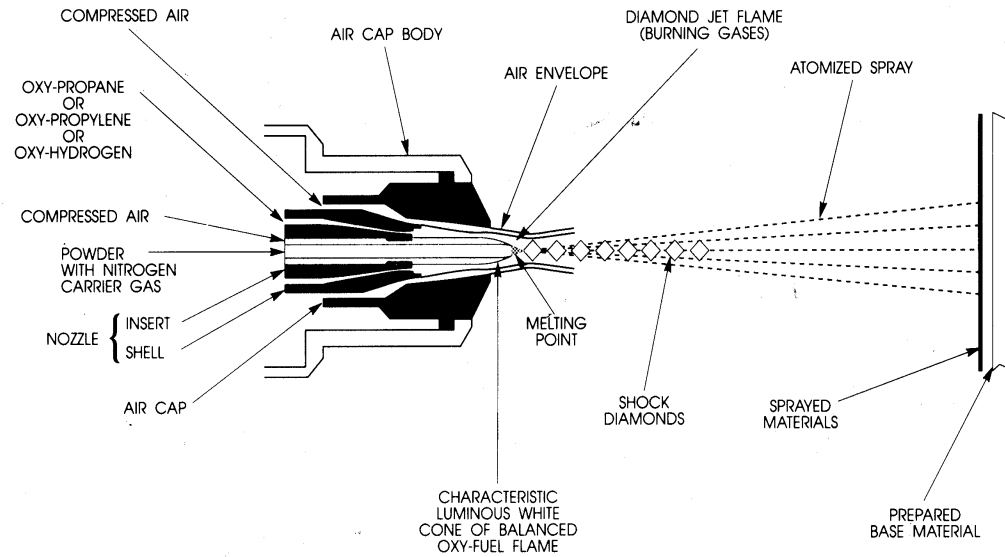
De olika processerna



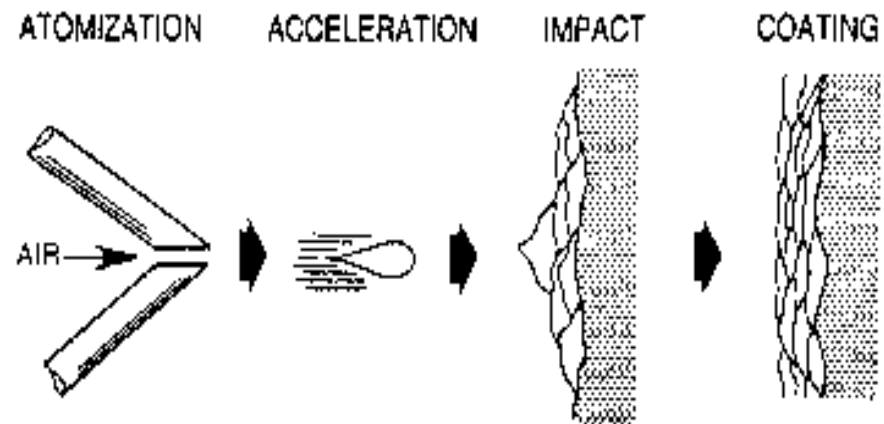
FLAMSPRUTNING



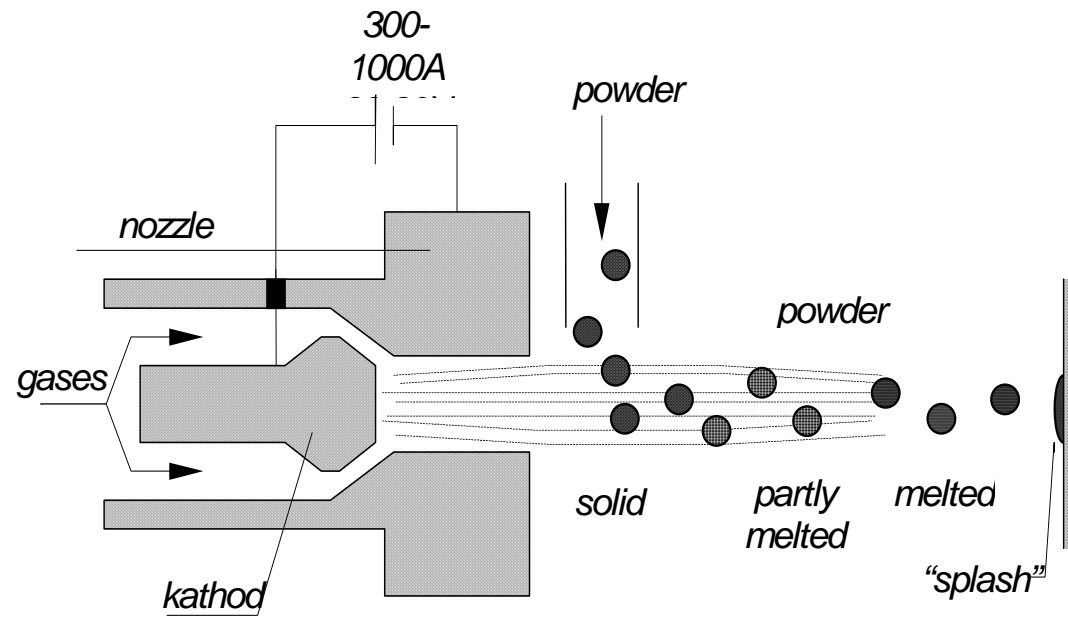
HVOF - DiamondJet



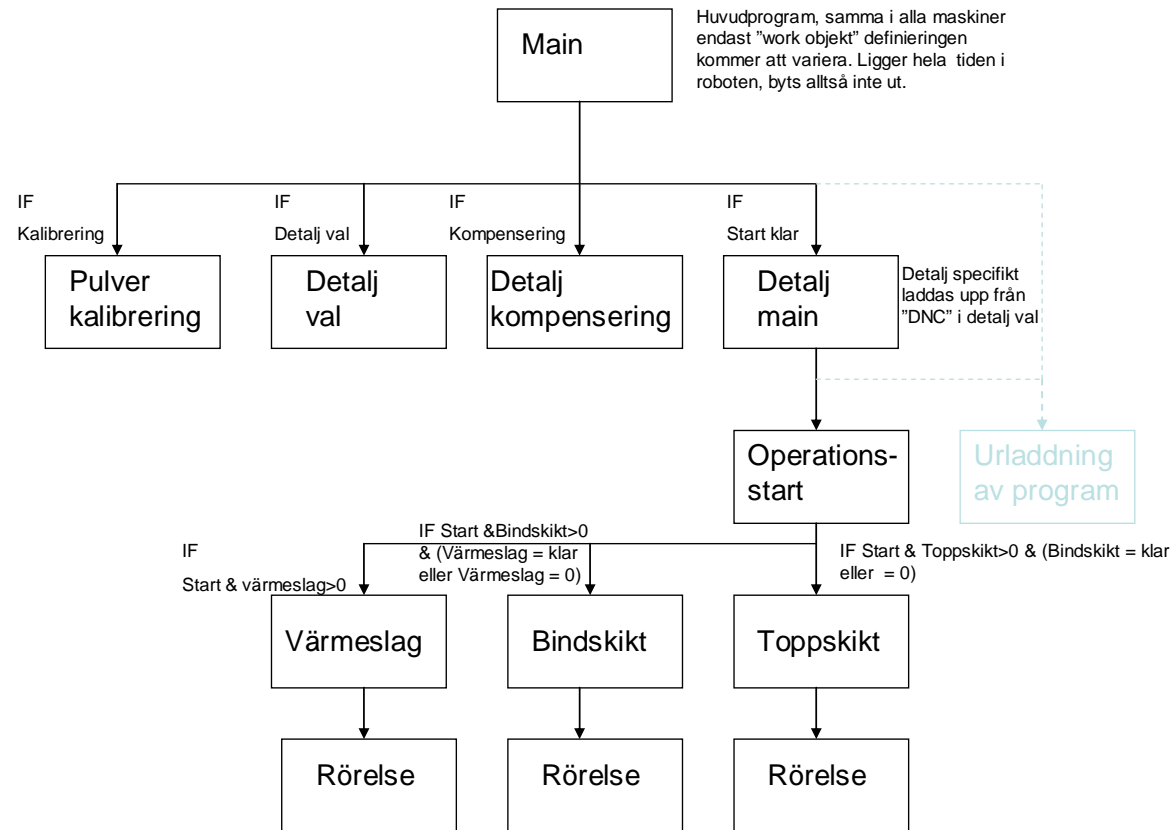
LJUSBÅGSSPRUTNING

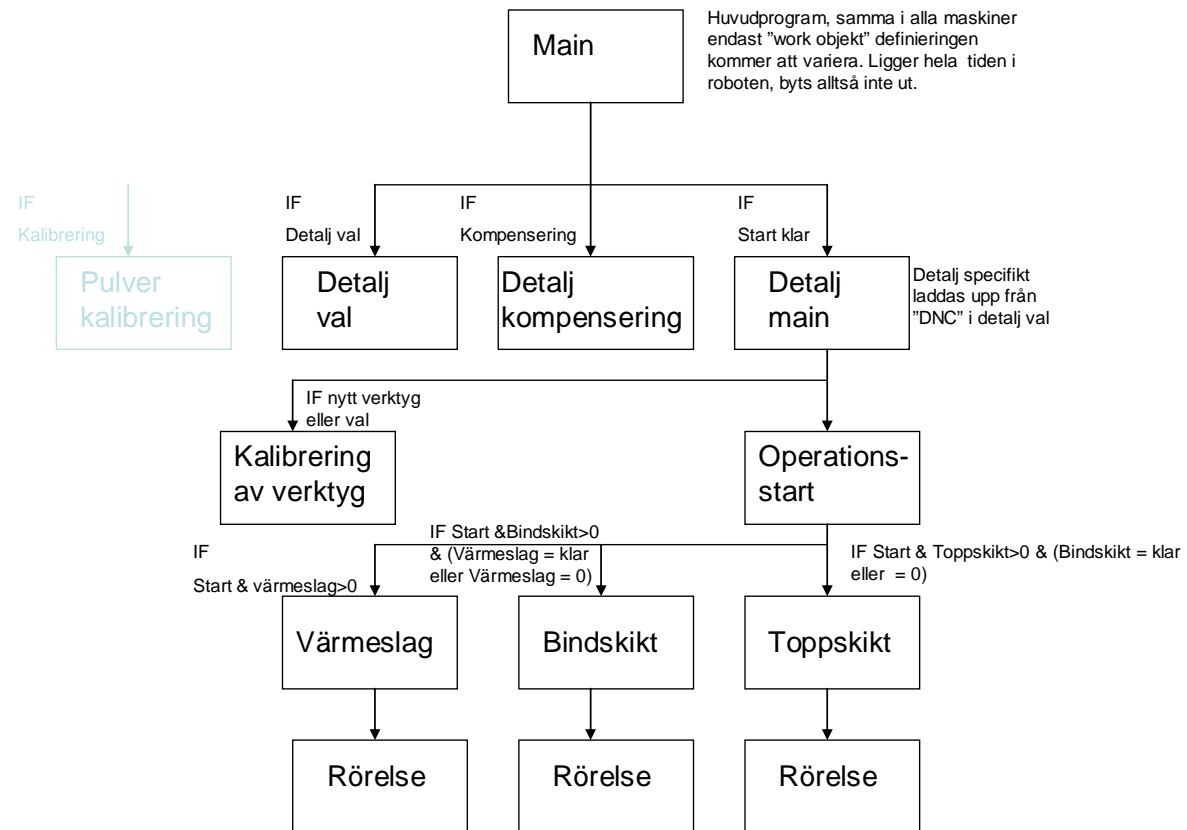


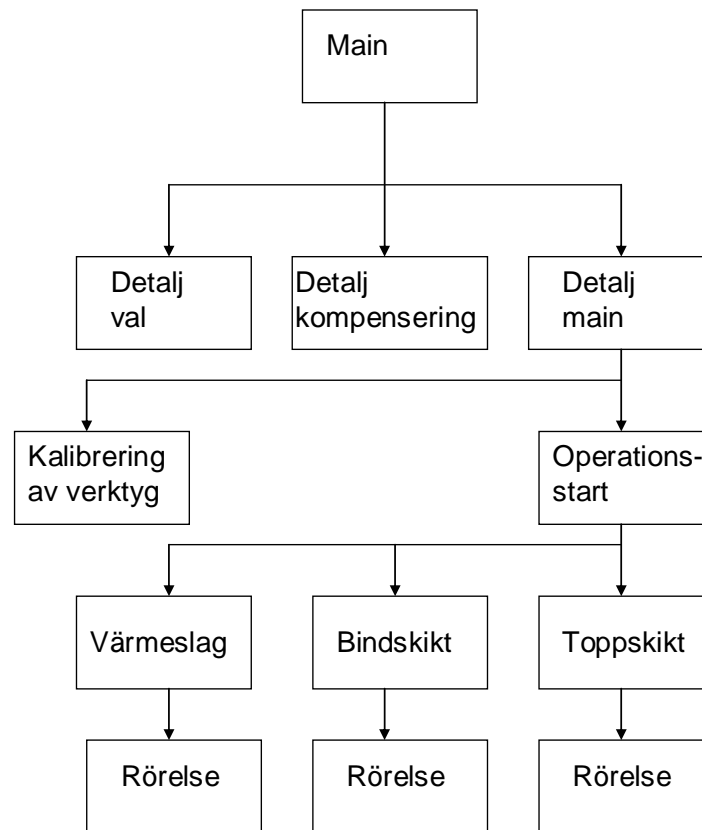
PLASMASPRUTNING

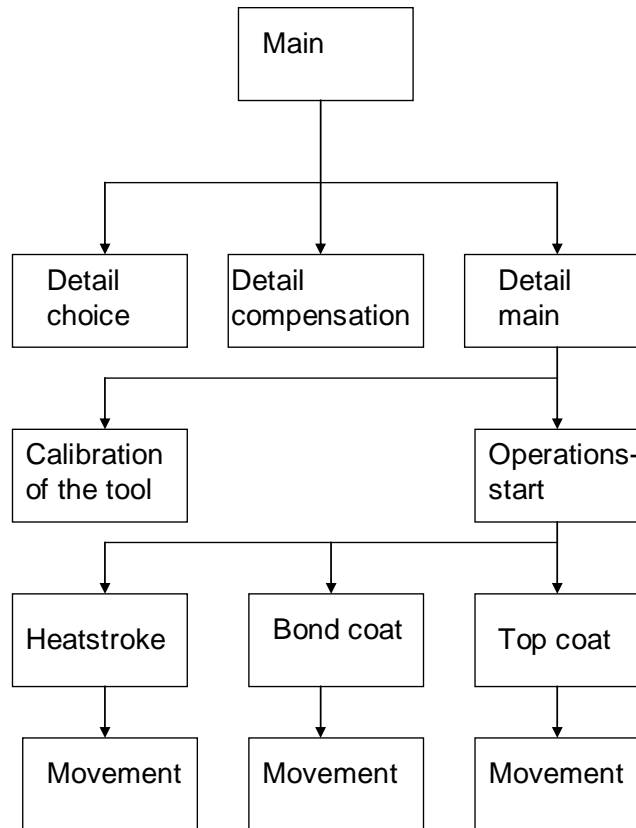


B. JSP









C. Mötesanteckningar

Närvarande: Mats-Olov Hansson, Charlotta Johansson, Krister Conradsson, Björn Kjellman, Martin Andersson, Petra Björnberg

Laddning av robotprogram

På TC kan endast ett program laddas ner åt gången i maskinen, vilket gör att programmet inte behöver laddas ur mellan två likadana detaljer. Dessutom kan ny utgåva av program inte användas förrän det blivit kvalificerat, vilket gör att ny utgåva inte kan genereras medan programmet finns i maskin. Krav från VAC o P&W är att inte ha program laddade i maskinen eftersom det finns risk att fel program laddas då flera program kan lagras i maskinen samt att det kan vara fel utgåva som används i de fall ny utgåva kan genereras under körning.

Kompensering

Punkter utanför detaljen får flyttas, kompenseras. Detta görs vid start av en ny "Batch". Kompenseringen skall bara finnas så länge som den aktuella batchen körs. När den är klar och ett nytt program laddas skall kompenseringen försvinna.

Planerat avbrott

Avbrottsrutin där punktvärden lagras. Hur rotation av rundbordet ska hanteras vid avbrott - återstart ska specificeras i strukturen.

Work objekt

Definieras från bordet. Döp detalj efter nr på o-ritning.

Robotposition

Det viktigaste för TC är hur TCP punkten träffar detaljen. För att kontrollera att pistolen monteras rätt kan man förslagsvis använda trycksensorer som mäter att z-koordinaterna är lika för tre jämt fördelade punkter på det runda munstycket.

Pistoltyp ska ligga som en funktion bredvid operationsstart. I funktionen ska toleransvärden för TCP punkten vara definierade och det ska inte gå att köra detaljprogrammet om det blir toleransavvikelse. När det är ok kan programmet upprepas med samma verktyg och funktionen kallas automatiskt in först då verktyget byts.

Verifiera detalj och utrustning i cell för valt program

Diskuterades och det förslag om kamera som togs upp är svårt att genomföra p g a att det är svårt att placera kameran på lämpligt ställe i cellen så att den kan identifiera stora och små detaljer och detaljer med inbördes små skillnader.

D. Robotprogram Main

```
MODULE ThermalSpray
```

```
CONST string strRemotePath:="pc:/PROGRAM/";  
    CONST string strLocalPath:="hd0a/HOME//";  
    VAR dir maindirectory;  
    VAR dir subdirectory;  
    VAR dir localdir;  
    VAR string strDirName:="";  
    !PERS string strSelectedDir:="PW2000_FWD";  
    VAR string strFileName:="22222";  
    VAR num nVar1;  
    VAR num nUpperZ;  
    VAR num nLowerZ;  
    VAR num nX;  
    VAR num nY;  
    !VAR num nLast_tool;  
    !VAR num nToolnumber:=4;  
    VAR num nFileNum;  
    PERS num nLast_tool:=4;
```

```
PROC main()
```

```
    rVACmain;
```

```
ENDPROC
```

```

PROC rVACmain()
!Volvo Aero Corporation;
!main program for thermal spray robot;

WHILE TRUE DO
!Start of the main loop
TPWrite "Select a new detail, press choice, all";
TPWrite "compensation will be deleted when choosing a new program";
TPWrite "Set compensation on the upper and lower point of the program, press Compensation";
TPWrite "Are you ready to start, press the begin button";
TPWrite "if you want to continue with the same operation as before, press continue";
TPReadFK nVar1, "Enter Mode!", "Begin", "Compensation", "Choice", "Continue", "";
        IF nVar1= 2 rComp;
        IF nVar1= 3 rChoice;
        IF nVar1= 1 %"rDetMain"%;
        IF nVar1= 4 %"rDetMain"%;
    ENDWHILE
ENDPROC

```

```

PROC rFrontPos()
        TPReadFK nVar1, "Enter Mode!", "Begin", "Compensation", "Choice", "Continue", "";
        !sets the compensation on the outer points of the detail;
        IF nVar1= 2 rComp;
        !makes a new program choice;
        IF nVar1= 3 rChoice;
        !Start the spray process;
        IF nVar1= 1 %"rDetMain"%;
        IF nVar1= 4 %"rDetMain"%;

```

```
ENDPROC
```

```
PROC rComp()
```

```
    TPReadNum nUpperZ, "Enter the compensation of the upper point";
```

```
    TPReadNum nLowerZ, "Enter the compensation for the lower point";
```

```
    TPReadNum nX, "Enter the compensation for the X coordinate";
```

```
    TPReadNum nY, "Enter the compensation for the Y coordinate";
```

```
    ! links to the motion module;
```

```
ENDPROC
```

```
PROC rChoice()
```

```
    TPErase;
```

```
    ! deleting the existing files from robot's ram1disk;
```

```
    !OpenDir localdir, strLocalPath;
```

```
    OpenDir localdir, "/hd0a//home";
```

```
    !WHILE ReadDir(localdir, strFileName) DO
```

```
        !IF strFileName<> "." AND strFileName<> ".." THEN
```

```
            RemoveFile "/hd0a//home/12345.prg";
```

```
                TPWrite "Nu suddas programmet";
```

```
                WaitTime 2;
```

```
        !ENDIF
```

```
    CloseDir localdir;
```

```
    !ENDWHILE
```

!to choose a new program;

```
TPReadNum nFileNum, "Enter A New Program Number:";
```

```
strFileName:=NumToStr(nFileNum,0);
```

```
!copying files from selected product directory;
```

```
TPErase;
```

```
TPWrite "Kopierar program...";
```

```
WaitTime 2;
```

```
!OpenDir subdirectory,strRemotePath;
```

```
OpenDir subdirectory,"pc:/";
```

```
!OpenDir subdirectory,"/hd0a//";
```

```
!WHILE ReadDir(subdirectory,strFileName) DO
```

```
!IF strFileName<>"." AND strFileName<> ".." THEN
```

```
!CopyFile "pc:/program/12345.prg","pc:/program/123.prg";
```

```
!CopyFile "/hd0a//home/vac_12345.prg","/hd0a//home/123.prg";
```

```
CopyFile "pc:/program/12345.prg","/hd0a//home/12345.prg";
```

```
!strLocalPath+strFileName;
```

```
!strRemotePath+strFileName+".prg",strLocalPath+strFileName;
```

```
!ENDIF
```

```
!ENDWHILE
```

```
CloseDir subdirectory;
```

```
!
```

```
TPErase;
```

```
! loading of main product program;
```

```
TPWrite "Laddar huvudprogram...";
```

```
!Load strLocalPath\File:=strFileName+"_PRG.mod";???prg
```

```
Load "/hd0a//home/12345.prg";
```

```
TPShow TP_PROGRAM;  
! execution of main product program  
!%"MainProg"%;  
!UnLoad strLocalPath\File:=strFileName+"_PRG.mod";  
ENDPROC  
  
!PROC rToolcalibration()  
! Calibrate the tools position with a probe, to be made by machine manufacturer;  
!TPWrite " The tools position is correct";  
!nLast_tool:= nToolnumber;  
!ENDPROC  
  
ENDMODULE
```

E. Robotprogram Det Main

Module detail

```
VAR num nflag1;  
VAR num nToolnumber:=4;  
  
PERS num nDetailnumber:=12121;  
PERS num nOperationNumber:= 11111;  
PERS num nIssue:= 1;  
PERS num nRecipenumber:=1;  
PERS num nHeatstrokes:= 4;  
PERS num nBondstrokes:= 3;  
PERS num nToppstrokes:=3;  
VAR num nHeated:=0;  
VAR num nBonded:=0;  
VAR num nTopped:=0;  
VAR num nRemaining_H;  
VAR num nExtraHeated:=0;  
VAR num nRemaining_B;  
VAR num nExtraBonded:=0;  
VAR num nRemaining_T;  
VAR num nExtraTopped:=0;
```



```
PROC rDetMain()
```

```
!Prints the different variables of the thermal spray process
```

```
TPERase;
```

```
TPWrite "Detailnumber: "\NUM:=nDetailnumber;
```

```
TPWrite "Operationnumber: "\NUM:=nOperationNumber;
```

```
TPWrite "Issue: "\NUM:=nIssue;
```

```
TPWrite "Toolnumber: "\NUM:=nToolnumber;
```

```
TPWrite "Recipenumber: "\NUM:=nRecipenumber;
```

```
WaitTime 5;
```

```
IF nToolnumber<> nLast_tool THEN
```

```
    !if the robot has got a new tool, go to toolcalibration
```

```
    pToolcalibration;
```

```
ENDIF
```

```
TPReadFK nflag1, "what to do?","OK","Change","Toolcalib","","abort";
```

```
IF nflag1=1 pOpStart;
```

```
!continue the process
```

```
!IF nflag1=2 pFrontPos;
```

```
!restarts the selection phase
```

```
IF nflag1=3 pToolcalibration;
```

```
!here the operator can choose to make a new tool calibration
```

```
IF nflag1=5 pFrontPos;
```

```
!the program returns to start point
```

```
ENDPROC
```

```

proc pOpStart()
    TPErase;
    TPReadFK nflag1,"Viktig fraga som MO har", "ja", "", "", "", "nej";
    IIF nflag1=2 pFrontPos;
    IF nflag1=1 THEN
        TPErase;
        IF nHeatstrokes>0 pHeating;
        IF nBondstrokes>0 pBondcoating;
        IF nToppstrokes>0 pTopcoating;
    ENDIF
ENDPROC

```

```

PROC pToolcalibration()
    !calibrates the tools position with a probe, to be made by the machine manufacturer
    TPErase;
    TPWrite "The tools position is correct!";
    WaitTime 5;
    nLast_tool:= nToolnumber;
ENDPROC

```

```

PROC pHeating()
    TPErase;
    !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
    WHILE nHeatstrokes>nHeated Do
        ldo the heating process
        !processdator prat
        pHeatmovement;
        nHeated:=nHeated+1;
    ENDWHILE
        pStopturntable;
        lenable dooropening
        TPReadNum nRemaining_H, "how many remaning heatcycles?";
    WHILE nRemaining_H>0 DO

        !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
        WHILE nRemaining_H>nExtraHeated DO
            ldo the heating process
            !processdator prat
            pHeatmovement;
            nExtraHeated:=nExtraHeated+1;
        ENDWHILE
        pStopTurntable;
        TPReadNum nRemaining_H, "how many remaning heatcycles?";
        nExtraHeated:=0;
    ENDWHILE
ENDPROC

```

```

PROC pBondcoating()
    TPErase;
    !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
    WHILE nBondstrokes>nBonded Do
        ldo the bonding process
        !processdator prat
        pBondmovement;
        nBonded:=nbonded+1;
    ENDWHILE
        pStopturntable;
        lenable dooropening
        TPReadNum nRemaining_B, "how many remaning bondcycles?";
    WHILE nRemaining_B>0 DO
        !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
        WHILE nRemaining_B>nExtraBonded Do
            ldo the bonding process
            !processdator prat
            pBondmovement;
            nExtraBonded:=nExtraBonded+1;
        ENDWHILE
        TPReadNum nRemaining_B, "how many remaning bondcycles?";
        nExtraBonded:=0;
    ENDWHILE
ENDPROC

```

```

PROC pTopcoating()
    TPErase;
    !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
    WHILE nToppstrokes>nTopped Do
        ldo the Topcoating process
        lprocessdator prat
        pTopmovement;
        nTopped:=nTopped+1;
    ENDWHILE
        pStopturntable;
        lenable dooropening
        TPReadNum nRemaining_T, "how many remaining Topcycles?";
    WHILE nRemaining_T>0 DO
        !WaitDI Start;
        ldisable door opening;
        pStartTurntable;
        WHILE nRemaining_T>nExtraTopped Do
            ldo the Topcoating process
            lprocessdator prat
            pTopmovement;
            nExtraTopped:=nExtraTopped+1;
        ENDWHILE
        TPReadNum nRemaining_T, "how many remaining Topcycles?";
        nExtraTopped:=0;
    ENDWHILE
ENDPROC

```

```
PROC pStartTurntable()
    TPWrite "snurra";
ENDPROC

PROC pStopTurntable()
    TPWrite "Stanna";
    WaitTime 2;
    TPErase;
ENDPROC

PROC pHeatmovement()
    TPWrite "Varmeslag";
    WaitTime 2;
ENDPROC

PROC pBondmovement()
    TPWrite "Bindslag";
    WaitTime 2;
ENDPROC

PROC pTopmovement()
    TPWrite "Toppslag";
    WaitTime 2;
ENDPROC

ENDMODULE
```

F. Operatörshandledning detaljprogrammering

Programmeringshandledning Detaljmodulen, Termisk Sprutning

1. Ta fram alla parametrar som behövs för detaljen.
2. Öppna start.prg som finns :::::::::::använd Notepad eller WordPad ej Word
3. Fyll i rätt värden för detaljen endast de värden som är gula här skall ändras.

! Varning Se till att inga andra tecken försvinner, det är också viktigt att inga andra tecken kommer in i programmet!

4. När det är inskrivet spara programmet som operationsnummer.prg i mappen ::::: under :::::
5. Se till att spara rörelseprogrammet enligt samma princip.
Operationsnummer_H.prg Operationsnummer_B.prg
Operationsnummer_T.prg

Module detail

```
VAR num nflag1;  
VAR num nToolnumber:=4;  
PERS num nLast_tool:=4;  
PERS num nDetailnumber:=12121;  
PERS num nOperationNumber:= 11111;  
PERS num nIssue:= 1;  
PERS num nRecipenumber:=1;  
PERS num nHeatstrokes:= 4;  
PERS num nBondstrokes:= 3;  
PERS num nTopstrokes:=3;  
  
VAR num nHeated:=0;  
VAR num nBonded:=0;  
VAR num nTopped:=0;  
VAR num nRemaining_H;  
VAR num nExtraHeated:=0;  
VAR num nRemaining_B;  
VAR num nExtraBonded:=0;  
VAR num nRemaining_T;  
VAR num nExtraTopped:=0;
```

G. Robotprogramspecifikation

General instructions

These instructions has as its aim to function as an instruction as to how Volvo Aero wants robot programs to be structured. The purpose of this is to make it possible for any programmer at any department to recognize the environment and structure when working on new programs and products. This is a necessity in order to simplify traceability and audit ability, both from internal as well as external audits. One major purpose is to make all programs, regardless of application, robot manufacturer or robot cell supplier, to create a similar structure to the largest extent possible even though different control systems are being used.

Product specific programs

All programs concerning the manufacturing process should be stored in a folder named for example "PRODUCTS". Product specific program means all programs which have to be created by a programmer when a program for a new detail or product shall be created. Under the "PRODUCTS" folder, every detail should have its own folder. If a product needs to be processed from another side, or in another setup, each operation will need its own folder. It must be clear, and documented, in which folder and program an operator must perform changes when adding or creating programs. If there is more than one robot cell connected to a master computer, product programs should be stored under folders that separates the cells with for example machine number.

Product main program

The product main program should only contain calls to sub programs to the largest extent possible. This program should include setting of correct work object, it should call to subroutines such as processing of a certain entity and engaging and retracting movements. An example can be seen below.

```
PROC DetMain()  
    !Prints the different variables of the thermal spray  
process  
    TPWrite "Detail number:"\Num:=nDetailnumber;  
    TPWrite "Operation number:      "\Num:=nOperationNumber;  
    TPWrite "Issue:                  "\Num:=nIssue;  
    TPWrite Tool number:  "\Num:=nToolnumber;  
    TPWrite Recipe number:"\Num:=nRecipenumber;  
    IF nToolnumber <> nLast_tool  
        !if the robot has got a new tool, calibrate the tool  
        pToolcalibration;  
    ENDIF  
TPReadFK nVar2 "OK", "Change", "Tool Calibration", "Abort";  
    IF nVar2 =1 rOpStart;  
!continue the process  
IF nVar2= 2 rFrontPos;  
    !restarts the selection phase  
    IF nVar2= 3 rToolcalibration  
        !here the operator can choose to make a new tool  
calibration  
    IF nVar2= 4 rFrontPos;  
    !program returns to start point
```


As seen in the program example above, subprograms are used as far as possible in the main program. Those are: OpStart, FrontPos and Toolcalibration. In addition to those, specific processing programs are called, for thermal spray robots, specific movements for different details and coatings.

Specific processing programs

Every specific processing program should contain all locations used in the processing. Variables used for processing shall be assigned their values in these programs. Examples of such are servo table rotation speed, table angle indexations, number of indexes and similar. All processing is performed from these programs along with robot and servo table movement.

Common robot cell programs

All programs that don't apply to a specific operation or product shall be stored in common folder. The robot main program, which instructs the operator to choose a certain product or operation, and programs that contains all instructions regarding servo table, I/O handling etc should be stored in structure gathered under a common program folder. Also modules containing background routines, such as monitoring functions should be here. All sub programs, such as toolcalibration, OpStart and sub program that a product specific program need to call, should be stored and downloaded into robot work memory in such a way that they are reachable from a product program at any given time. The folder structure within this common area can be created in different ways depending on application and robot manufacturer, but the structure shall always be discussed with the programmer or operator that will take on responsibility for the cell.

Additional programs

In all cases, regardless of manufacturer or application, an area for backups must exist. All backup folders must be stored in this area named with backup date.

Programs related to the cell, rather than to the manufacturing, such as calibration programs and adding/changing -programs for tools and external axis should be stored in a common utilities area.

A folder for development must exist. All programs that are not complete and tested must be stored here. Possibility to run product specific main programs from here should exist if possible.

Tool setting, calibration and creation of tools

Tool setting programs must exist, and be created in such a way that it makes it easy to change a value for a tool, example changing the TCP Z-axis offset.

A program for creating new tools should exist. When adding a new tool to the robot cell, all tool specific parameters will be entered through a text editor, or through the program. Only the values necessary for the process will be set by the operator. The values entered for a tool shall be compared to the measured values when the tool is checked before used in a processing job. The comparison shall function as a verification process, no automatic compensation needs to be done.

It is important that documentation regarding the robot cell clearly defines all settings that have to be copied and changed in the program in order to make the new tool work.

Program name

Naming of process sub routines shall be done in such a way that it makes it easy for a programmer or operator to understand its location. The name shall be associative with a tolerance number found on the detail drawing. Avoidance of possible misunderstandings is important. Naming should always be discussed with the programmer or operator whom will claim responsibility for the program.

Error handling

Besides the control systems basic error codes, specific error codes must be created by the programmer. These should contain messages that clearly state the problem. Along with this there should be documented how the operator should manage the error and how to restart the process when it has been corrected.

JSP structure example for robot

