Development of intelligent robot systems based on sensor control

*Mikael Fridenfalk*

Division of Robotics  
Department of Mechanical Engineering  
Lund Institute of Technology  
Lund University
Sensor based control systems increase flexibility and autonomy in robotics. One of the most common applications in robotics is welding, which is often limited to objects with fine tolerances. To be able to manage objects with relatively large tolerances, such as ship-hulls, sensor guided control systems have to be incorporated that are able to handle large deviations from the drawings. To be able to handle the complexity at the development of such intelligent autonomous systems, a new development environment for robot systems was designed in this work, called FUSE, that integrates software prototyping with virtual prototyping. By the application of FUSE and the methodology developed in this thesis, a seam tracking system for robotic arc welding based on arc sensing was developed and physically validated on-site at a shipyard in the European ROWER-2 project. The aim of the ROWER-2 project was to develop a robot system for welding of hull-sections of large structures such as oil tankers and cruisers. In addition, a true 6D seam tracking system was designed and validated in FUSE that allows seam tracking of seams following 3D spline curves by the real-time correction of both the position and the direction of the torch. Previous systems based on arc sensing only allow seam tracking in 2D or 3D. Similar 6D seam tracking system was also developed and validated based on laser scanning. Since kinematics singularities may disturb the sensor guided control process in robotics, a new method was also suggested that among others eliminates position errors near and on inner singularity points, based on knowledge of the kinematics and dynamics constraints of the robot system and its surrounding environment.

Keywords
Robot, sensor control, virtual sensor, simulation, software development, rapid prototyping, virtual prototyping, FUSE, ROWER-2, Asimov, robotic welding, seam tracking, arc sensing, laser scanning, 3D, 6D, Matlab, Igrip, Envision, kinematics, singularity.

Classification system and/or index term (if any)
Development of intelligent robot systems based on sensor control

Mikael Fridenfalk
Development of intelligent robot systems based on sensor control

Mikael Fridenfalk
PhD Thesis
CODEN: LUTMDN/(TMMV-1056)/1-217/2003
ISBN 91-628-5550-6

© by Mikael Fridenfalk and the Department of Mechanical Engineering, Lund University.
All rights reserved.

Printed in Sweden
KFS i Lund AB, Lund
Sensor based control systems increase flexibility and autonomy in robotics. One of the most common applications in robotics is welding, which is often limited to objects with fine tolerances. To be able to manage objects with relatively large tolerances, such as ship-hulls, sensor guided control systems have to be incorporated that are able to handle large deviations from the drawings. To be able to handle the complexity at the development of such intelligent autonomous systems, a new development environment for robot systems was designed in this work, called FUSE, that integrates software prototyping with virtual prototyping. By the application of FUSE and the methodology developed in this thesis, a seam tracking system for robotic arc welding based on arc sensing was developed and physically validated on-site at a shipyard in the European ROWER-2 project. The aim of the ROWER-2 project was to develop a robot system for welding of hull-sections of large structures such as oil tankers and cruisers. In addition, a true 6D seam tracking system was designed and validated in FUSE that allows seam tracking of seams following 3D spline curves by the real-time correction of both the position and the direction of the torch. Previous systems based on arc sensing only allow seam tracking in 2D or 3D. Similar 6D seam tracking system was also developed and validated based on laser scanning. Since kinematics singularities may disturb the sensor guided control process in robotics, a new method was also suggested that among others eliminates position errors near and on inner singularity points, based on knowledge of the kinematics and dynamics constraints of the robot system and its surrounding environment.
Acknowledgements

First of all, I would like to thank Professor Gunnar Bolmsjö for the excellent supervision and for providing for vision, knowledge and guidance throughout this work. Professor Bolmsjö has been my mentor since the age of 27 and a source of inspiration already from the times at junior high, when I first accoutered his book “Industriell Robotteknik”.

I would further like to thank Dr. Giorgos Nikoleris for valuable feedback on my work, Dr. Magnus Olsson for valuable feedback on my work and technical support, and Dr. Lars Randell for provision of technical support. I wish to thank my colleagues Per Cederberg, Lars Holst, Arne Ingemansson, Ulf Lorentzon, and my former colleagues Jonas Assarsson, Fredrik Ahlquist, Dr. Hamid Nasri and Per Hedenborn for the great support. Thanks also to one of the members of the technical staff, Mikael Vatau for inventing the acronym “FUSE”, and the former technical staff member Mats Rosengren. My colleagues have given me the best time at the Department of Robotics and the highlight was of course our philosophical discussions, covering a wide area of subjects.

I would like to further thank Dr. Anders Robertsson, Dr. Klas Nilsson and Professor Rolf Johansson at the Department of Automatic Control for the great encouragement I have received. Thanks also to Professor Gunnar Sparr at the Department of Mathematics for good advice in some mathematical issues and Professor Gustaf Olsson at the Department of Industrial Electrical Engineering and Automation for supporting me in one of my private research projects in robotics.

I would also like to thank our partners in the ROWER-2 project, especially those I had the pleasure to cooperate with, which are Tiberio Grasso, Mascia Lazzarini and Alessio Nista at Tecnomare in Venice, Italy, and Dr. Manuel Armada and Dr. Pablo González de Santos at IAI in Madrid, Spain. The ROWER-2 project was funded by European Commission under the BRITE/EURAM program.

Finally, I would like to thank my parents, friends and former teachers, which are extremely dear to me.
Abstract v

Acknowledgements v

1 Introduction 1
   1.1 Complex systems 1
   1.2 Hypothesis and research methodology 3
   1.3 Objective 3
   1.4 Ethical principals 4

2 Contributions 5

3 Application 9
   3.1 Intelligence in manufacturing systems 9
   3.2 The ROWER-2 project 10

4 Flexible Unified Simulation Environment 15
   4.1 Virtual prototyping 15
      4.1.1 Software prototyping of real-time systems 15
      4.1.2 Virtual mechanical prototyping and CAR systems 16
      4.1.3 Integration of software and mechanical prototyping 16
      4.1.4 Virtual sensors 17
   4.2 FUSE 17
      4.2.1 FUSE design 17
      4.2.2 Linking method 17
      4.2.3 Program structure 18
      4.2.4 Envision Motion Pipeline 18
      4.2.5 Flow structure 19
      4.2.6 FUSE Console 20
      4.2.7 Run-time interactive programming 20

5 Development and implementation of seam tracking algorithms 21
   5.1 Seam tracking at robot welding 21
      5.1.1 Robot welding 21
### Paper III  The Unified Simulation Environment - Envision Telerobotics and Matlab merged into one application  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>1.2 Integration of systems</td>
<td>88</td>
</tr>
<tr>
<td>1.3 Integration in robotics</td>
<td>88</td>
</tr>
<tr>
<td>1.4 FUSE</td>
<td>89</td>
</tr>
<tr>
<td>2</td>
<td>89</td>
</tr>
<tr>
<td>2.1 Materials and methods</td>
<td>89</td>
</tr>
<tr>
<td>2.2 Matlab</td>
<td>89</td>
</tr>
<tr>
<td>2.3 FUSE</td>
<td>92</td>
</tr>
<tr>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td>5</td>
<td>94</td>
</tr>
</tbody>
</table>

### Paper IV  Design and validation of a sensor guided robot control system for welding in shipbuilding  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>1.1 Robot welding</td>
<td>99</td>
</tr>
<tr>
<td>1.2 Seam tracking</td>
<td>100</td>
</tr>
<tr>
<td>1.3 Process control</td>
<td>100</td>
</tr>
<tr>
<td>1.4 Through-arc sensing</td>
<td>101</td>
</tr>
<tr>
<td>1.5 Control algorithms for seam tracking</td>
<td>102</td>
</tr>
<tr>
<td>1.6 Simulation using virtual sensors</td>
<td>103</td>
</tr>
<tr>
<td>1.7 ROWER-2 application</td>
<td>103</td>
</tr>
<tr>
<td>2</td>
<td>104</td>
</tr>
<tr>
<td>2.1 Systems development methodology</td>
<td>104</td>
</tr>
<tr>
<td>2.2 Joint profiles</td>
<td>106</td>
</tr>
<tr>
<td>2.3 Experimental setup</td>
<td>106</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
</tr>
<tr>
<td>3.1 Simulation experiments</td>
<td>107</td>
</tr>
<tr>
<td>3.2 Validation by robot welding</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>124</td>
</tr>
<tr>
<td>5</td>
<td>124</td>
</tr>
</tbody>
</table>

### Paper V  Design and validation of a universal 6D seam tracking system in robotic welding using laser scanning  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>131</td>
</tr>
<tr>
<td>2</td>
<td>133</td>
</tr>
<tr>
<td>2.1 Sensors guided robot control</td>
<td>133</td>
</tr>
<tr>
<td>2.2 Robot system</td>
<td>135</td>
</tr>
<tr>
<td>2.3 Simulation system</td>
<td>135</td>
</tr>
<tr>
<td>2.4 Calculation of position and orientation errors</td>
<td>137</td>
</tr>
<tr>
<td>3</td>
<td>137</td>
</tr>
<tr>
<td>3.1 Position control</td>
<td>137</td>
</tr>
</tbody>
</table>
3.2 Trajectory tangent vector control ....................... 138
3.3 Pitch control ........................................ 138
3.4 Code and pseudo code samples ....................... 139
3.5 Computational costs ................................ 141
3.6 Kinematics singularities ............................ 141
3.7 Initial and final conditions .......................... 141
4 Experimental results .................................. 142
4.1 Workpiece samples ................................... 142
4.2 Laser scanning ....................................... 145
5 Results and discussion ................................ 146
6 Future work ............................................ 146

Paper VI Design and validation of a universal 6D seam tracking system in robotic welding using arc sensing 151
1 Introduction ............................................ 153
2 Materials and methods ................................. 155
  2.1 Sensors guided robot control .......................... 155
  2.2 Robot system ...................................... 159
  2.3 Simulation system .................................. 159
  2.4 Calculation of position and orientation errors .... 160
3 Modeling ............................................... 161
  3.1 Position control ..................................... 162
  3.2 Trajectory tangent vector control ................... 163
  3.3 Pitch control ....................................... 164
  3.4 Arc sensing weaving interpolation .................. 164
  3.5 Code and pseudo code samples ....................... 165
  3.6 Kinematics singularities ........................... 167
  3.7 Initial and final conditions ........................ 167
4 Experimental results .................................. 168
  4.1 Workpiece samples ................................... 168
  4.2 Arc sensing ....................................... 170
5 Results and discussion ................................ 171
6 Future work ............................................ 172

Patent I Method eliminating position errors caused by kinematics singularities in robotic wrists for use in intelligent control and telerobotics applications 175
1 Background of the invention ............................ 177
2 Problem analysis ...................................... 177
3 New method .......................................... 178

APPENDIX 183

Patent II Intelligent system eliminating trajectory programming and geometrical databases in robotic welding 185
1 Introduction ............................................ 187
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Robot welding</td>
<td>187</td>
</tr>
<tr>
<td>2</td>
<td>Materials and methods</td>
<td>189</td>
</tr>
<tr>
<td>2.1</td>
<td>Sensors guided robot control</td>
<td>189</td>
</tr>
<tr>
<td>2.2</td>
<td>Robot system</td>
<td>191</td>
</tr>
<tr>
<td>2.3</td>
<td>Simulation system</td>
<td>192</td>
</tr>
<tr>
<td>2.4</td>
<td>Calculation of position and orientation errors</td>
<td>192</td>
</tr>
<tr>
<td>3</td>
<td>Modeling</td>
<td>193</td>
</tr>
<tr>
<td>3.1</td>
<td>Position control</td>
<td>193</td>
</tr>
<tr>
<td>3.2</td>
<td>Trajectory tangent vector control</td>
<td>194</td>
</tr>
<tr>
<td>3.3</td>
<td>Pitch control</td>
<td>194</td>
</tr>
<tr>
<td>3.4</td>
<td>Code samples</td>
<td>195</td>
</tr>
<tr>
<td>3.5</td>
<td>Computational costs</td>
<td>196</td>
</tr>
<tr>
<td>3.6</td>
<td>Arc sensing weaving interpolation</td>
<td>197</td>
</tr>
<tr>
<td>3.7</td>
<td>Kinematics singularities</td>
<td>197</td>
</tr>
<tr>
<td>3.8</td>
<td>Initial and final conditions</td>
<td>198</td>
</tr>
<tr>
<td>4</td>
<td>Experimental results</td>
<td>198</td>
</tr>
<tr>
<td>4.1</td>
<td>Workpiece samples</td>
<td>198</td>
</tr>
<tr>
<td>4.2</td>
<td>Laser scanning</td>
<td>199</td>
</tr>
<tr>
<td>4.3</td>
<td>Arc sensing</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>Results and discussion</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>Future work</td>
<td>201</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Complex systems

Human progress and mere survival have always been dependent on the ability to process large amounts of complex information and to respond to it likewise in a very complex manner. The reason why we often experience that our response to different situations come easily and naturally, such as when someone says “hello”, is not because reacting to such an obvious situation is an easy thing in itself, but experienced as such since the complexity of our reaction patterns are encapsulated to hide the immensely diversified process that goes on in our minds. Due to the parallel architecture of the human brain, a qualified guess is that a proper reaction to a simple greeting involves more computations than the total amount of computations performed by a modern computer today during its entire lifetime\(^1\).

As complexity in ordered form is often considered a beautiful thing, such as a mathematical formula or a piece of music, complexity may be quite terrifying when it is not ordered, such as in software "spaghetti-code".

Humans often use to decompose complex information into simple elements. This creates a feeling of understanding of the contents of the information and helps to place it in the right context along with other information stored in our minds. This concept of systemization by placing clusters of processes and information in logical and often hierarchical structures, is used among others in the OOP (Object-Oriented Programming) methodology \([20, 107, 111]\). Encapsulation of complexity is essential here. By hiding complexity, we are less bothered by large amounts of information, and are free to focus our efforts.

\(^1\)The human brain has more than 10 billion neurons with thousands of connections each operating in the millisecond range \([8, 73]\). This assumption does not however include hardware implementation of neural networks.
As an attempt to increase the development efficiency of robot systems, in this thesis a methodology was developed and validated that integrates software prototyping with mechanical virtual prototyping from the conceptual level to the design of the physical prototype. The objective of this methodology is to minimize the illusion of complexity in designing mechatronic systems, thereby saving time, effort and costs in practical applications.

Today, software prototyping and mechanical virtual prototyping are often performed in separate environments independently of each other. This reminds of separating methods from data in OOP. Within the frames of this methodology a new simulation environment was designed by the integration of Matlab\(^2\) (including Simulink and Real-Time Workshop) with the robot simulator Envision\(^3\), by linking and compiling the kernels of Matlab and Envision into a new software system, called FUSE (Flexible Unified Simulation Environment).

Presently, at software prototyping involving mathematical models which is usually the case in robotics, applications such as Matlab, Maple\([6]\) and Mathematica\([7]\) accelerate the development significantly compared to programming languages such as for instance C++. The reason is that these applications are interactive, use concise matrix notation and provide for large mathematical libraries. The difference between these applications and C++ may in this context be compared to the difference between C++ and machine code. Though low-level programming is essential for optimizing the final code, high-level programming is more suited for software prototyping.

The developed methodology was basically validated using FUSE for the design and validation of a 3D seam tracking system using arc sensing (also called through-arc sensing) in the European ROWER-2 project\(^4\). FUSE was also used for the design and validation of 6D seam tracking systems based on arc sensing and laser scanning, see papers V-VI and Appendix. Today, though there exist 6D seam tracking systems based on laser scanning, the performance of these systems are however unknown, since no evaluations of such systems have been published. In arc scanning however, there is no record as far as the author knows of any attempts to design a 6D seam tracking system, so this system opens up new possibilities to increase the intelligence in robotic arc welding without using expensive laser scanning systems.

It should be noted that this thesis does not deal with the welding process explicitly at arc welding, since the experiments in the ROWER-2 project showed that such considerations was not necessary for the implementation and val-

---

\(^2\)MathWorks, Inc.

\(^3\)Robot simulator, Delmia Inc.

\(^4\)The objective of the ROWER-2 project was to develop a robot system for automatic and adaptive welding of double-hull sections in super-cruisers and oil tankers. The idea was that the workers only had to mount the robot inside the hull-cell and would be able to supervise the welding process from a remote distance. A robot system was developed and physically validated along with a SGRC (Sensor Guided Robot Control) system for seam tracking based on arc welding. For a more detailed description about this project, see papers I and IV.
1.2 Hypothesis and research methodology

The hypothesis of this thesis is formulated as:

full integration between the software and the mechanical parts of a robot system saves significant development time, effort and costs compared to traditional methods where software prototyping and mechanical virtual prototyping are performed independently.

Integration of software prototyping and mechanical virtual prototyping tools provides for a holistic and analytical perspective in micro and macro-scale that is essential for the development of complex systems, such as intelligent sensor guided robotic systems. The research methodology is hence based on development of sensor guided control systems based on full integration between software and mechanical parts, for all sub-levels of the robot system, ranging from systems analysis and the overall seam tracking process to details in the software and mechanical systems that may play an important role in the development of the robot system.

The research methodology includes also the use of simulation tools for enhanced communication and cooperation between research and development teams spread over a wide geographical area, such as in the ROWER-2 project which included team members from several European countries.

1.3 Objective

In the manufacturing industry, high level of intelligence and flexibility saves time and effort and causes fewer interruptions in production, especially for
one-off production and building of large structures, such as ships, where relatively high tolerances are allowed in the sheet material to minimize manufacturing costs. The first objective in this work was therefore to design and validate a 3D seam tracking algorithm in the ROWER-2 project. The second was to design universal 6D seam tracking systems for robotic welding that were based on arc sensing and laser scanning. 6D seam tracking systems based on arc sensing do not exist presently and would most probably decrease the costs of 6D seam tracking to less than 1/10 of existing systems based on laser scanning.

To develop such systems, a new simulation tool was required that integrated software prototyping with virtual mechanical prototyping. Simulation is often used in robotics to save time, effort and costs. Today, in the development of new robotic systems, virtual prototyping is rarely integrated in the software development process. Such integration would however make the development of intelligent systems, including SGRC systems, less time-consuming and complex. Though such systems exist for some mechanical simulation applications by integration with mathematical and software prototyping tools such as Matlab, there exist no system that the author knows of that includes a robotic simulator. The third objective of this work was therefore to design an integrated simulation environment especially for the development of robotic systems.

Another problem that was addressed was kinematics singularities at SGRC, causing disturbance in the robot control system. Higher amount of autonomy in robotic systems increases the chance for the robot to come near such areas and may in the case of robotic welding probably ruin the weld. The final objective was therefore to suggest a new method by which such problems would be minimized for robots with RPY-wrists, which are the predominant category of wrists for industrial robots.

1.4 Ethical principals

The ROWER-2 project was examined from an ethical point of view by analysis of its effects on human life. Robot automation yields (1) solution to the problem of the lack of qualified welders in a relatively rapidly expanding economy, (2) improvement of environmental conditions for workers, (3) increased global wealth due to higher productivity and (4) increased welding quality and safety in ships. There are also negative effects, such as (1) the need of dismissing or transferring of some workers to other departments and (2) deterioration in the skills of manual welding. As the positive effects outweigh the negative however, the ROWER-2 project is viewed to be based on ethical principals.
Chapter 2

Contributions

The attached papers and patents in this thesis are briefly summarized in this section, by the order of date of submission, date of publication or category.

**Paper I: Simulation based design of a robotic system**

Simulation of the ROWER-2 robot system at the design process. The conclusions were that the workspace of the robot and the mobile platform were sufficient for obtaining the ROWER-2 specifications at seam tracking, but that the welding torch and its mounting had to be modified [50]. Further, the terms breadth-first and depth-first simulations were defined based on data structure terminologies.

**Paper II: The Asimov project**

Participation in the software development and integration of the real-time system of Asimov, a modular light weight robot with 7 degrees of freedom, potentially useful in application areas such as assistive technologies for disabled people and space [41, 49]. The purpose was to introduce and test novel techniques using virtual prototyping tools within the design process. The Asimov robot was successfully validated by physical experiments.
Paper III: FUSE

An integrated software system was built based on Envision (robot simulator, Delmia Inc.) and Matlab (MathWorks Inc.) [43]. The integration was performed by compiling these software applications and their libraries into one single simulation environment. Thereby the process of mechanical virtual prototyping and software prototyping could be closer linked, saving time and effort. Today, though such unified environments exist for generic mechanical simulation systems, such as Adams (Mechanical Dynamics Inc.), no solution does exist explicitly for robotic systems. FUSE was used in the design of the 3D seam tracking system in the ROWER-2 project and the 6D seam tracking systems described in papers V-VI and Appendix.

Paper IV: The design and validation of a robotic seam tracking system based on arc sensing

The design and validation of a robotic seam tracking system in the ROWER-2 project, based on arc sensing at arc welding for joining large sections, such as double-hull constructions in shipbuilding [44, 45]. The system was developed in FUSE, implemented in the real-time robot control system and successfully validated by physical experiments on-site at a shipyard in Venice, Italy.

Paper V: Design and validation of an intelligent 6D robot control system for seam tracking based on laser scanning

A universal 6D seam tracking system was designed based on laser scanning at robotic welding [48]. This system showed basically to be able to seam track any 3D spline curve and eliminate the need for robot programming and geometrical databases at robotic welding. Three things that presumably speak in favor of this method compared to previous ones for laser scanning, are that: 1) stability and reliability has been the highest priority here, 2) while this method has been carefully evaluated by simulation experiments, more or less identical to the real system and based on real experiments, no evaluations have been found for previous methods, and 3) since 6D seam tracking systems based on laser scanning have not been the commercial success that were anticipated, this may raise the question of what went wrong, since there exists obviously a tremendous need today for increased levels of intelligence in robotic automation.
Paper VI: Design and validation of an intelligent 6D robot control system for seam tracking based on arc sensing

A 6D seam tracking system was designed based on arc sensing for robotic arc welding [47]. This system is basically able to seam track any 3D spline curve and eliminates the need of robot programming and geometrical databases for robotic welding. Today, there exist no solutions for 6D seam tracking based on arc sensing. The current solutions are typically limited to 2D. Since systems based on arc sensing typically cost 1/10 compared to systems based on laser scanning, this system may in the future replace laser scanning systems, unless the price of laser scanners are significantly reduced.

Patent I: Eliminating position errors caused by kinematics singularities in robotic wrists

Patent application. A new method by which kinematics singularity problems may be addressed in robotics, among others by the elimination of position errors near and on inner singularity points, based on knowledge of kinematics and dynamics constraints of the total robot system including its surrounding environment [42]. One interpretation of this method is that the stronger motors in the manipulator are allowed to compensate the errors caused by weaker wrist motors. Singularity problems are especially an issue in intelligent systems and telerobotics applications, since the actual movement of the robot is not often known in advance in such systems. This new method is however also useful for traditional industrial robot applications, since it makes robot control systems less sensitive to kinematics singularities.

Appendix (Patent II)

Patent application, 6D seam tracking [46]. Here, additional code is included compared to papers V-VI and the 6D seam tracking systems are treated in parallel.
3.1 Intelligence in manufacturing systems

Higher customer demands on economy and product diversification requires higher flexibility and intelligence in the manufacturing process. Today, the greatest challenge for many manufacturers are to be able to deliver customized products that meet the needs of each client. Customization however is costly, since it often requires reconfiguration of the production line and re-programming of machines and robots. One specific case is robotic welding, where today in general only large product series are manufactured. Increased autonomy would however make one-off production as economic as production of larger series and thereby allow higher diversification of products manufactured by a single company.

High intelligence in manufacturing systems may also introduce automation in new areas. To decrease manufacturing costs for example, often materials with relatively large tolerances are allowed. In traditional robot welding, large tolerances may cause displacement of the seam, which in turn may ruin the weld. This is especially common in shipbuilding, where large hull-sections are built and where displacement of the weld may be several centimeters compared to the computer models.

Seam tracking, based on laser scanning or arc sensing is therefore often used in robotic welding. In laser scanning, a laser beam is used to retrieve the profile of the seam in advance of the welding torch. In arc sensing, the arc current is measured, while simultaneously an oscillating motion is added to the tool-tip of the torch. By retrieving the seam profile, the robot is able to correct the position of the torch continuously, and thereby insuring high welding quality. Seam tracking is presently in general performed only in 2D. Often the position
of the torch is corrected in a plane orthogonal to the direction of the seam. By increasing the adaptability of a seam tracking system to include any seam that follows an arbitrary 3D spline, compensation of both the position and orientation of the torch may be performed in full 6D, which increases the flexibility in robotic welding. By the implementation of such systems, robot trajectory programming and geometrical databases will no longer be necessary, which saves time and effort. This makes one-off production economic, and thereby also the manufacturing of customized products.

3.2 The ROWER-2 project

The application that inspired the development of the 6D seam tracking systems was the European ROWER-2 project. Though the implemented seam tracking algorithm in this project was according to the project specifications only limited to 2D, the need for such solutions become obvious, since not all seams in a ship are linear.

The objective of the ROWER-2 project was to automate the welding process in shipbuilding, specifically for joining double hull sections in super-cruisers and oil tankers. The double hull construction is imperative for such structures to ensure high safety at incidents. The joining of double hull sections is however associated with large environmental problems since the welders work in a relatively hot and enclosed environment, filled with poisonous gas. By automation of the welding process, allowing the workers to mount a robot system inside the cell and supervise the welding process from a remote distance, the work environmental problems may be solved.

According to the specifications in the ROWER-2 project, every cell is equipped with a manhole with the dimensions $600 \times 800\,\text{mm}$, through which the robot may be transported, see Fig. 3.1. Since the robot has to be transported manually, the constraint on the robot is that each part is allowed to weigh no more than $50\,\text{kg}$. Further on, the robot system is designed to operate in standard cells with predefined variations in dimension. To be able to meet the specifications, a robot manipulator was built based highly on aluminum alloy. The robot was mounted on a mobile platform with 1 degree of freedom to increase its workspace. The welding method was GMAW using through-arc sensing at seam tracking [29].

Fig. 3.2 shows the ROWER-2 robot system at the final physical validation phase and Figs. 3.3-3.5 show ROWER-2 simulations, including 3D as well as one 6D seam tracking experiment. The 6D seam tracking experiments, based on both laser scanning and arc sensing, was designed based on experience and results gained from the development and validation of the 3D seam tracking system in the ROWER-2 project.
3.2 The ROWER-2 project

Figure 3.1 A double hull-cell in ROWER-2 project with a manhole through which the robot equipment is manually transported. One of the walls and the floor have been excluded from the picture.

Figure 3.2 The ROWER-2 robot system. The system is mounted on a mockup that is about 3.5 m high, belonging to the hull-cell of a ship.
Figure 3.3 Example of a simulation performed in the process of virtual prototyping of the robot manipulator and the mobile platform. The aim of this specific simulation was to make sure that the suggested robot design was able to fulfill the ROWER-2 specifications, before it was manufactured.

Figure 3.4 Example of ROWER-2 seam tracking simulation, analyzing the behavior of the SGRC system at very large linear deviations.
Figure 3.5 Example of a 6D seam tracking experiment based on arc sensing, performed by the ROWER-2 robot.
4.1 Virtual prototyping

4.1.1 Software prototyping of real-time systems

Software prototyping is essential for the design of reliable software and increases productivity of critical and complex systems in robotics where errors may cause casualties, damage to expensive equipment or costly production delays [21, 36, 56, 133]. Software prototyping plays also an important role in software maintenance and evolution, a process that often requires more effort than the initial design process [26, 77, 101, 109]. It is in addition in some areas essential in the research work, such as in evolutionary software development, automatic programming or brain building [13, 31, 61]. There exists no specific software prototyping language today for robot applications, but efforts has been made to design generic robot programming languages [81, 99]. Instead, common programming languages are often used together with specific robot application libraries, such as the Matlab Robot Toolbox [30], used in this thesis.

An example of a high-level software prototyping environment is Matlab [3]. Matlab is an environment for numerical calculation and offers more than 20,000 functions in more than 50 application toolboxes ranging over application areas such as neural networks, signal and image processing, automatic control and database management. The programming language of Matlab is called Matlab script or simply Matlab and is interpreted in run-time.

In Matlab there exist environments for development of RT (Real-Time) embedded systems [40] such as mechatronic and robotic systems [28, 62, 78], called
Simulink and Stateflow\(^1\)\([25, 32]\). These sub-applications are RT software prototyping environments that via Real-Time Workshop \([86, 87]\) are able to convert standard or user-written RT simulation blocks to C-code, for immediate compilation and download to an embedded computer \([54, 102, 116]\). By working on a higher level of abstraction, it becomes possible to design, modify, maintain and evolve complex RT systems in a fraction of the time that it ordinary takes using traditional RT programming methods.

### 4.1.2 Virtual mechanical prototyping and CAR systems

Virtual mechanical prototyping is today essential in mechanical design \([39, 75]\). Applications for virtual prototyping provide tools for analysis covering areas such as structural mechanics, dynamics and logistics. Similar to software prototyping, there exists in addition research applications where the prototyping tool is in charge of the mechanical design process, using evolutionary prototyping methods \([128]\).

Within robotics research and manufacturing industry there exist a number of CAR (Computer Aided Robotics) systems, such as Envision (Igrip) \([1]\) and RobCad \([5]\). These graphical simulation applications are equipped with RT graphical 3D engines and kinematics and geometrical libraries covering most commercial robots. These applications provide in addition also many specialized functions, such as robot dynamics and collision detection.

In Envision, the user is constrained to a relatively limited set of mathematical functions. These functions may be expanded by auxiliary C-libraries. It is however a cumbersome job to develop mathematical models using C and C-libraries, since C was never primarily designed with matrix manipulation and the use of higher mathematics in mind. What is needed here is a mathematical engine that gives easy access to mathematical methods for modeling and experimentation in robot applications.

### 4.1.3 Integration of software and mechanical prototyping

There is today a clear trend towards integrating software products, since integration often results in new software systems that are more powerful than the sum of the component applications alone. This phenomenon is due to the non-linear nature of computer programs\(^2\).

By linking Matlab, Simulink or Stateflow to mechanical simulators, the functionality of these applications are increased. Examples of such applications are Adams \([4, 97]\) and DynaWorld Professional \([2, 64]\). Though these tools are

\(^1\)Simulink is basically continuous while Stateflow is time-discrete.
\(^2\)Integration is always per definition non-linear, since in linear systems the basic components are independent and not integrated.
in general very useful for the design and analysis of dynamic systems, they are however of limited use for robot simulation compared to the more specialized robot simulators Envision and RobCad.

4.1.4 Virtual sensors

The development of sensor guided control systems may be accelerated using virtual sensors in the simulation control loop. Since a system's state is often easily obtained in simulation, assuming that the simulation model is sufficiently accurate compared to the real system, this provides for a better foundation for analysis of well defined processes than in general is possible in physical experiments. Further on, as stated before, insertion of a sensor in a control loop may cause damage to expensive equipment, unless the behavior of the entire sensor guided control system is precisely known, which is rarely the case at the development stage.

Virtual sensors are used in many application areas, such as robotics, aerospace and marine technologies [19, 38, 55, 96, 103]. Development of new robot systems, such as for seam tracking may be accelerated by application of simulation [49, 50]. In general, the design methodology of virtual sensors may vary due to their specific characteristics. If a sensor model is not analytically definable, artificial neural networks or statistical methods may be used to find appropriate models [11, 98].

4.2 FUSE

4.2.1 FUSE design

FUSE is the joint kernels and libraries of Envision and Matlab linked into one simulation environment, using Matlab Engine [82, 83, 84, 85] and the open architecture of Envision. It is a unified environment in robotics for integrated virtual prototyping and software simulation, see Paper III. FUSE was implemented on IRIX, SGI and validated by the development and simulation of the 3D and 6D seam tracking algorithms, presented in Papers I, IV-VI and Appendix. The objective of FUSE is to be used during the whole virtual prototyping process of robot design, both for mechanical virtual prototyping and for software development all the way from concept to automatic C-code generation for final compilation and download to RT embedded computers, see chapter 6.

4.2.2 Linking method

Matlab provides direct access to its environment for integration with other applications by the API (Application Programmer's Interface), called the Matlab
Engine Library. This library gives Envision full access to Matlab's environment as a virtual user. The integration is performed using the open architecture structure of Envision by compiling and linking its application kernel and shared libraries [35] with Matlab, using pipes on Unix and ActiveX on Microsoft Windows. The open architecture structure of Envision is presented in Fig. 2, paper III. The communication between the process of Envision and Matlab was experienced by the author as both reliable and fast. On Unix, if the Matlab process is executed on a remote computer, remote shell (rsh) is used instead of pipes. This makes it possible to run distributed Matlab applications from Envision over the network.

Matlab functions may also be called from other applications by automatic translation and compilation of Matlab code using the Matlab Compiler Suite. Since not all Matlab functions are supported, and compilation is required each time a change is made in the Matlab code, this method was however not considered for the design of FUSE, in favor of Matlab Engine, which provides for all Matlab commands in run-time.

4.2.3 Program structure

In the same way that Envision has run-time access to the functionality of Matlab, including Simulink, Stateflow and Real-Time Workshop, Matlab may use the main functionality of Envision by sending RT Command Line Interpreter (CLI) commands [33] for execution in Envision. CLI is a command language that interacts with Envision as a virtual user. Since Envision is also a virtual user from the perspective of Matlab with full access to the Matlab environment, none of these applications will automatically take charge unless they are explicitly programmed to. In the implementation of FUSE by which the SGRC algorithms in the ROWER-2 project were simulated and developed, Matlab was in charge via Envision Motion Pipeline [34], using Envision basically as a clocking device, inverse kinematics calculator and graphical engine. This was sufficient for the development of SGRC algorithms in FUSE, but if FUSE is however intended to be used as an on-line and RT application, there will be need for external clocking for the synchronization of FUSE to the external RT process.

4.2.4 Envision Motion Pipeline

Envision MP (Envision Motion Pipeline) consists of the sequence of functions that are executed in order to cause motion in a device with multiple DOF (Degrees of Freedom). These functions include inverse kinematics, motion planning and execution, dynamic analysis and simulation. MP is written in C and the user may partially or completely replace the source-code delivered by Delmia with user-written code. Each robot in Envision may be assigned
4.2 FUSE

Each time MP is modified, FUSE has to be rebuilt. FUSE, according to the present implementation, uses MP to link Envision with Matlab. When the user loads a workcell into Envision containing a robot that uses a specific MP component connected to Matlab, Matlab Engine is instantly activated.

4.2.5 Flow structure

![Flow diagram](image)

Figure 4.1 In the present implementation of FUSE, Matlab is in charge, acting on sensor information received from Envision. Envision is used mainly as a graphical engine, inverse kinematics calculator and clocking device. When Envision asks Matlab to perform a calculation, the Envision process becomes blocked in Unix and Envision waits until Matlab is finished before it proceeds further.

In our applications, FUSE was clocked by MP and Matlab was addressed every time MP was executed, see Fig. 4.1. The execution frequency of MP depends on the processor power of the computer it runs on. If some part of the process however slows down the system, such as a workcell with too many polygons or heavy Matlab calculations, the execution will adapt to the delays in the total system instead. If RT execution is not required, MP is used as the process timer, why the calculations are allowed to take the time needed to finish a job without effecting the outcome of the simulation. When Envision asks for the calculation of the next joint values of the robot, MP is executed, which makes a call to Matlab to take over the process. When Matlab is assigned a task by Envision, it blocks the Envision process in Unix. This allows the Matlab process to finish its task, before Envision continues the execution.

Envision has reading and writing access to all workspace variables in Matlab at run-time. In this context, Envision may also import CLI strings from Matlab for execution. It is hence not obvious which application that should be in charge. The robot control system and basically the intelligence is presently implemented in the Matlab environment. As illustrated in Fig. 4.1 all sensor information
is provided by the graphical environment of Envision and passed to the Matlab workspace for calculation, each time Matlab is called. The inverse kinematics of the ROWER-2 robot is calculated by user-written C-code implemented in MP. Though Envision provides for many generic inverse kinematics algorithms, a user-written implementation may in some cases improve the execution speed or enable development and validation of new inverse kinematics algorithms.

### 4.2.6 FUSE Console

The Matlab prompt in FUSE is literally replaced by the MP. FUSE console in Fig. 4.2 may thereby be used for run-time communication between the user, and the Matlab and Envision environments within FUSE.

![FUSE Console Diagram](image)

**Figure 4.2** Commands written in FUSE Console are executed in the FUSE environment in run-time.

### 4.2.7 Run-time interactive programming

Since Matlab is an interpreted language, programs written in Matlab within FUSE may be modified in run-time as the simulation progress. The same is not applicable for C-code written in MP, which has to be recompiled each time the code has been modified.
Chapter 5

Development and implementation of seam tracking algorithms

5.1 Seam tracking at robot welding

5.1.1 Robot welding

Robot automation increases industrial productivity and frees man from unhealthy and monotonous labor. In the future robots are also anticipated to contribute to education, entertainment, health care, mining, building, household and many other areas [88]. One application area today is automatic robot welding which improves the welding quality and the environmental conditions for welders. This applies especially to automatic welding in shipbuilding where large structures are welded, including the joining of double-hull sections [65, 67, 104, 105, 110].

5.1.2 Seam tracking

Seam tracking [18, 22, 27, 50, 95] is essential for automation in shipbuilding at manufacturing large passenger and cargo ships, such as super-cruisers and oil-tankers, where high tolerances in the sheet material are allowed to minimize manufacturing costs. The study of the patents issued during the last century shows that research within this application area goes back at least 40 years, covering seam tracking systems based on among others mechanical, inductive, electrical, infrared, optical and electron beam sensors. Today the predominating sensors are laser scanners [70, 92, 118, 123] and arc-sensors [12, 29, 112, 120], both for industrial applications and in research with few exceptions, such as in [106, 66].
5.1.3 Arc sensing

Seam tracking using through-arc sensing, or simply arc sensing, was introduced at the beginning of the 1980s. In this method the welding current is used to estimate the position of the torch. Since the approximate distance from the electrode to the workpiece may be calculated analytically by a simple formula, the geometrical profile of the workpiece may by this method be obtained by weaving the torch across the seam, see Fig. 5.1.

![Figure 5.1](image)

**Figure 5.1** Definition of TCP, and the orthonormal coordinate system noa. Weaving is performed in n direction, o is opposite to the direction of welding and a is the direction of approach.

5.1.4 Laser scanning

Vision systems are basically only used to estimate the position of the workpiece, prior of seam tracking [71, 112, 45]. During seam tracking, due to the intensity of the arc light, usually only optical sensors such as laser scanners are used. A laser scanner is in general built using a diode laser and a mirror, scanning the beam across the seam, which is reflected back from the workpiece and analyzed by an optical system containing a CCD camera. To minimize interference of the laser light with the welding process, a laser with an appropriate wavelength has to be chosen [10] and a robust method adopted for the extraction of the seam profile [69].

5.1.5 Comparison between arc sensors and laser scanners

Laser scanners are expensive, accurate and decrease the workspace of the robot since they have to be mounted on the torch. Arc sensors are inexpensive and inaccurate. Since welding of large structures requires relatively
low accuracy, this may be one explanation why the majority of the patents that have been issued during the last 10 years for seam tracking at arc welding [15, 60, 63, 113, 119] are based on arc sensing. Another explanation is that while the accuracy of laser scanners are sufficient for seam tracking, large improvements are still needed to increase the robustness of the systems based on arc sensors. Today, improvements on laser scanners are primarily focused on making them less expensive [57, 58, 72, 122, 124].

5.1.6 Power source settings and adaptive control

Besides the seam geometry in seam tracking, considerations have to be made of process related welding parameters [9, 127]. The welding process contains many parameters, such as arc voltage, current, wire feed speed and wire consumable. The objective is to determine feasible parameters for welding before seam tracking starts. This may be performed experimentally or by the use of knowledge based systems [17, 126]. If it is not possible or desirable to keep these settings constant throughout the seam at seam tracking for maintained high welding quality [53, 59, 76, 119], for instance due to the characteristics of the power source [14, 114, 45] adaptive control may be introduced into the sensor guided control loop. To increase the intelligence or robustness in seam tracking, many seam tracking systems using arc sensing are based on neural networks [74, 93, 94, 121] and fuzzy control [16, 68, 125, 129]. Since the physical experiments in the ROWER-2 projects were successful using analytical methods alone, the implementation of these were however not considered.

5.2 Implementation of seam tracking in the ROWER-2 project

In the European ROWER-2 project, also presented in Papers I and IV, the objective was to develop a robot system for automatic and adaptive welding of double-hull sections in super-cruisers and oil tankers. The idea was that the workers only had to mount the robot inside the hull-cell and would be able to supervise the welding process from a remote distance. A robot system was developed and physically validated along with a SGRC system for seam tracking, using COM-ARC III [130] as an initial reference for such design.

5.3 6D seamtracking

5.3.1 Background

Seam tracking is in general only performed in 2D by moving the torch with constant velocity in $x$ direction while a continuous correction is performed in
y and z directions to keep constant relative distance from TCP to the seam walls, see Fig. 5.1. In 3D seam tracking another DOF is added to the SGRC system, performed by for instance correction in negative o direction, or modification of an orientation axis. The seam tracking system implemented in the ROWER-2 project is essentially a 2D system, plus an additional DOF due to the vector booster method, that also modifies the direction of the nominal trajectory, see Paper IV. Examples of 3D systems where laser scanners and arc sensors are used are [131, 132, 37].

There is today a great need for 6D seam tracking systems. Such systems are theoretically able to correct the TCP in not only x, y and z directions, but also around roll, pitch and yaw\(^1\), and is per definition able to follow any continuous 3D seam with moderate curvature.

There exist no record today of the design of any 6D seam tracking system based on arc sensors. There exist however records of suggestions of systems based on laser scanners [63, 90, 91] and force sensors [23, 24]. In additions, there are companies that claim to have commercial systems based on laser scanning for 6D seam tracking. Since no evaluations of these systems have been found by the author, and these systems have not become the commercial success that one would anticipate judging from the needs of the market, we have to question how robust these systems actually are. A preliminary study of the few publications that exist in this area showed that the stability problems have not been prioritized in the solutions. This makes this thesis, to the knowledge of the author, the first publication to present evaluation data for 6D seam tracking systems based on arc sensing and laser scanning. These new systems are, to the knowledge of the author, also the first 6D seam tracking systems based on the same, where stability and reliability issues have been prioritized already from the initial development stage.

5.3.2 Development and validation of 6D SGRC systems

The design and development of the 6D seam tracking systems presented in this thesis was performed mainly in FUSE, both for the fundamental operation of systems based on laser scanning and arc sensing. While a virtual generic arc sensor was used in the arc sensing experiments, a virtual model of the M-SPOT [108] was used at laser scanning.

Regarding the system based on laser scanning, the performed simulation was considered so accurate compared to real physical validation, that no physical validation was considered as necessary. The performance of the seam tracking system using arc sensors however is dependent on the power-source. Since the use of the Migatronic power-source showed to be successful in the ROWER-2 project despite intrinsic arc control, the implementation of the seam track-

\(^1\) Roll, pitch and yaw are in this thesis defined as rotation around z, y and x, in sequential order.
5.3 6D seamtracking

The 6D seamtracking system is estimated to be successful for any automation-friendly power-source.

In the case of the laser scanning system, the computational costs were estimated to less than $250 + 50N$ floating point operations (without consideration to memory operations), where $N$ is the number of curve fitting points, which for a typical value of $N = 10$ gives 750 operations per control cycle. At a rate of maximum 30 scans per second, this gives less than 22,500 floating point operations per second.
Chapter 6

Systems development methodology

6.1 Development methodologies

6.1.1 Software development

At the end of the last decade, only less than half of software projects were delivered and the average cost and time overruns were 200% [89, 117]. Further, only a small fraction of delivered software was actually used. This problem may be addressed by implementation of software methodologies, acting as the infrastructure behind all development process at an organization, increasing thereby the chances of a good quality product by decreasing the overall complexity of the software engineering effort [80, 117].

A software methodology may be defined as a collection of methods applied across the software development life cycle and unified by some general, philosophical approach, where a method is defined as a disciplined process for generating a set of models that describe various aspects of a software system under development, using some well-defined notation [20].

An example of a software development methodology is described in [20]. The steps in this methodology are (1) conceptualization, or establishing core requirements, (2) analysis, or development of a model of the desired behavior, (3) design, or creation of an architecture, (4) evolution, or evolvement of the implementation and, (5) maintenance, or management of post-delivery evolution.
6.1.2 Mechanical virtual prototyping

Simulation in the mechanical design process saves time, effort and expenses. It reduces the need of physical prototypes in the design process, decreases the risk of design failure and improves communication between the design teams and its members [50, 49, 115]. Currently, a typical mechanical prototyping methodology includes the following major steps: (1) conceptual design, (2) simulation, (3) detailed design, (4) manufacturing of a physical prototype and (5) experimental validation [79].

6.1.3 Systems development methodologies in robotics

A typical systems development methodology in robotics, using the mechanical virtual prototyping scheme described above, is presented in Fig. 6.1. In general, software prototyping is not integrated with mechanical virtual prototyping. In this scheme, the software development methodology is structurally adapted to the suggested mechanical virtual prototyping methodology.

![Diagram showing the link between software development and mechanical virtual prototyping](image)

**Figure 6.1** Presently, the link between software development and mechanical virtual prototyping is practically non-existent in the development methodology of robot systems.

6.2 Formulation of a new methodology

A complex system that works is invariably found to have evolved from a simple system that worked. A complete system designed from scratch never works and
cannot be patched up to make it work. You have to start over, beginning with a working simple system [52].

Since in many mechatronic systems, such as advanced airplanes and SGRC systems, software and mechanical design are highly dependent on each other, it is essential to use a methodology that integrates these already from start [100], see Fig. 6.2.

The idea of including software prototyping in robot design is here taken another step further by using Matlab and Simulink as the software prototyping environment, including automatic C-code generation by RTW. This makes software development more high-level oriented which is essential in both design, maintenance and evolution of robot systems. Robot systems are in general implemented on real-time embedded computers. Since many professionals regard software and hardware integration in embedded systems as the most critical bottleneck in the whole design process of a real-time system [51], and since development of real-time systems is one of the most resource-consuming steps in the design process of a robotic system, inclusion of RTW in the methodology accelerates the development process significantly.

Figure 6.2 The suggested methodology includes software prototyping in robotics design along with automatic C-code generation for instant compilation and download to the final target.

The design process of a robot system is here divided into the following five steps:

1. A conceptual model of the robot system is iteratively developed, including both mechanical parts and software.

2. A simulation model is built covering the mechanical and the software system. If the design iterations fail, return is made to the conceptual
3. Blueprints are produced for the mechanical system. In the software system, Matlab code is converted to C/C++ code or Simulink blocks.

4. The physical robot and mechanical accessories are manufactured. The manually translated C/C++ code or the automatically generated C-code from Simulink blocks are compiled and downloaded to the final target.

5. The robot system is tested to meet the specifications formulated in the beginning of the development process. If the specifications are met, the iteration process ends, otherwise return is made to a previous step. In practice, to avoid the redesign of the mechanical system, much effort is spent on mechanical virtual prototyping.

### 6.3 Application in the ROWER-2 project

Although the suggested methodology was not applied in its fullest extent in the ROWER-2 project, it did minimize the development effort in the design of the 3D SGRC system, see Paper IV. The most essential concept of the suggested methodology is the simulation of the robot manipulator and accessories together with the software, which in this application saved both time and effort. Figure 6.3 displays the application of FUSE in the Rower-2 project.

![Diagram](image)

**Figure 6.3** The methodology was applied in the ROWER-2 project by the integration of a model of the SGRC system with the virtual prototype of the mechanical manipulator and the mobile platform.

The applied methodology contained the following steps:

- Drafts were made of the robot system based on the specifications.
- The basic design of the robot manipulator and the mobile platform were verified as suggested by the designers. Due to initial simulation results
the orientation of the torch was modified and the mobile platform was decided to only have one DOF. At a later stage a model of the SGRC was developed integrated with a model of the mechanical design of the system.

- Detailed blueprints for the mechanical parts were produced based on the virtual prototypes. The software model of the SGRC system was manually translated from Matlab to C++ to be included in the robot control system.

- The physical robot manipulator and mobile platform were built and mechanically validated. The software was compiled and downloaded to the embedded computer. Since compilation is a relatively fast process compared to the manufacturing of the mechanical parts of the robot, the software system could be iterated several times between simulation and experimental validation.

- The robot system was experimentally validated along with further simulations and software modifications until the specifications were reached.

### 6.4 Development of the 6D seam tracking systems

The major development work of the 6D seam tracking systems was performed according to the methodology used in the ROWER-2 project, as described above. In the case of the arc sensing system, it was developed and perfected in FUSE prior to the final translation from a combination of Matlab and C-code to pure Matlab code. The arc sensing system implemented in pure Matlab code was used as a template for the design of the laser scanning system, by some modifications of the template and the addition of a virtual laser scanner.
Chapter 7

Discussion

It is remarkable that a robot development environment such as FUSE is not available on the market already, since similar systems exist for integration of generic mechanical simulation systems with calculation engines such as Matlab and Simulink. Mechanical simulators lack however the many special features provided by robot simulators such as Envision and RobCad. The idea to keep most of the development work of a new robot system within the same environment and to be able to generate real-time C-code, for immediate compilation and download to the embedded computer of the robot, is appealing for anyone who develops new robotic systems.

It is most probable that the 6D seam tracking systems presented in this thesis would never have been developed if FUSE would not have existed. To develop robotic systems in Matlab only, is simply too tedious, since the VRML toolbox has to be used and the robot systems modeled from scratch. On the other hand, to perform development of such systems in Envision would have required time-consuming C-programming, without the flexibility and the high-level approach that is provided by Matlab. The time and effort that a traditional approach would have required would probably had been discouraging enough for the author to drop such enterprise already from start. This shows actually how important tools can be for development work, not only for the performance of the work itself, but also as a source of inspiration, which is confirmed by the expression “the road is the goal”.

From this standpoint it is clear that the road in this work comprised the development methodology as described in chapter 6 and the integrated development environment FUSE. On this foundation, an efficient development process could take place as part of a bigger team for robot systems development with sensor guidance control. To meet both time and cost constraints in such development work it is important to be able to concurrently develop such a system including mechanical system, control system and parts related to user inter-
faces and control of the application process including sensors. To meet goals
defined in the ROWER-2 project, the methodology and FUSE environment was
developed and validated. As a result of this work, generalized sensor guidance
algorithms were developed for 6D control including management of inverse
kinematics problems close to singular areas.

The validation experiments of the 6D seam tracking systems showed that by
pre-scanning, any radius of curvature could be seam tracked in laser scan-
ning, theoretically including sharp edges. It was further estimated that both
systems could be able to manage a radius of curvature below 200 mm at real
time control, to be on the safe side, using any standard industrial robot ma-
ipulator. In reality however, it is highly probable that much smaller radius
of curvatures are managed by these systems at real-time, perhaps even radius
of curvatures below 50 mm for a modern robot manipulator. As a next step,
physical validations would be necessary to find the real limitations of these
systems for any chosen industrial robot.

Concerning the new method for elimination of position errors caused by in-
er kinematics singularities in robotic wrists, preliminary experiments showed
that it works according to the formulated theory. If it is fully implemented,
including considerations of the whole robot system and its surrounding en-
vIRONMENT, this system is theoretically to be much more efficient that existing
methods today for dealing with inner singularity problems.
Chapter 8

Conclusions

As pointed out in chapter 7, the integrated approach in the development work has clearly demonstrated that the hypothesis holds. The methodology and FUSE enable simulations of SGRC systems, which require integration of both software and the mechanical models. Such integration allows for deeper analysis of the interaction of the robot control system with its environment, based on virtual sensors with predefined behavior, which in turn allows for development of systems that would have required much time and effort using traditional systems development methodologies in robotics. The methodology itself was validated by the development and the physical validation of the 3D seam tracking system implemented in the ROWER-2 project and in the development and evaluation of the 6D seam tracking systems based on laser scanning and arc sensing.

Moreover, the following conclusions related to the objectives of the thesis are made based on the research, development and validation work presented in this thesis:

- The design of a unified simulation environment that integrates software prototyping with virtual prototyping of the mechanical parts of the robotic systems. This is especially useful for the development of intelligent robotic systems, interacting with the surrounding environment based on sensor information.

- The design and validation of a seam tracking system in the ROWER-2 project for robotic welding using arc sensing. The specifications of this project were successfully fulfilled based on physical experiments.

- It was shown by systematic simulation experiments based on physical experiments in the ROWER-2 project, that it is possible to design robust 6D seam tracking systems based on laser scanning or arc sensing that in
real-time are able to seam track a seam following an arbitrary 3D spline curve with a radius of curvature below 200 mm. In the non-destructive “pre-scan” mode, the laser scanning system is expected to manage very small radius of curvature, theoretically even sharp edges.

Since 6D seam tracking based on arc sensing has not existed before, this makes arc sensors very attractive, since in real-time seam tracking, such system is able to basically do the same job as the ten times more expensive systems based on laser scanning.

- In addition to above, a new and novel method was designed that is able to decrease problems caused by kinematics singularities in robotic systems, which is especially important for intelligent and telerobotic systems, where the motion of the robot is always predefined in advance. According to simulation of the basic functionality of this system within the scope of this thesis, the method worked according to the theory and in some cases large disruptions caused by inner kinematics singularities were entirely eliminated.
Chapter 9

Future work

The 6D seam tracking systems were developed using simulation tools, based on physical experiments in the ROWER-2 project. According to these simulations, by pre-scanning, any radius of curvature may basically be managed in the case of laser scanning, theoretically including sharp edges. At real-time, to be on the safe side, it was suggested that the systems were able to manage a radius of curvature below 200 mm at real time control, using a standard robot manipulator. In reality however, by the experience gained in the ROWER-2 project, much smaller radius of curvature should be managed by these systems at real-time, perhaps below 50 mm for a modern robot manipulator. Physical validations should therefore be performed to find the real limitations of these systems.

In addition, also the remaining features of the suggested method for elimination of position errors caused by kinematics singularities should be implemented and evaluated.
Bibliography


Bibliography


[133] M. V. Zelkowitz and L. Rus. The role of independent verification and validation in maintaining a safety critical evolutionary software in a
CONTRIBUTIONS
Paper I

Simulation based design of a robotic welding system

Mikael Fridenfalk, Magnus Olsson, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

In Proceedings of the Scandinavian Symposium on Robotics 1999, October 14-15, Oulu, Finland
Simulation based design of a robotic welding system

Mikael Fridenfalk, Magnus Olsson, Gunnar Bolmsjö

Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

Abstract

Simulation and a virtual prototyping method is applied in the design process of a robot system for arc-welding of ship hulls. Simulation is used for the identification of design problems at an early stage, by an iterative problem-solving approach, integrating the work of designers and simulation engineers all the way from idea to product. This is of vital importance especially in shipbuilding where the complicated geometrical parts of the ship hull makes it difficult to develop robot systems that are valid for all hull-cells. Initially, the robot system has to be adapted to a specific reference cell, representing a general cell hull in a submarine or a supertanker. An embryonic, well functioning virtual sensor for welding with weaving using through-arc sensing has been created and integrated with the simulation environment.

KeyWords: arc-welding, robot, simulation, shipbuilding, virtual prototyping, virtual sensor, virtual reality.

1 Introduction

New areas in welding large structures in ship-building include joining large sections such as double-hull constructions. Joining these sections create great problems for a manual welder since the welding takes place in a closed area with associated work environmental problems. The accessibility to the working area is limited to a man-hole and the use of robots for welding such structures requires new robot design that are adapted for the task as well as the additional requirements of one-off production.

The present paper will describe work and result related to design of a robot system for joining ship sections, specifically welds that must be made inside
the structure. Within the scope of this work there are some obvious reasons for automation that include not only quality and productivity issues, but work health for the welders where the welding is performed in a closed environment.

Robot welding in shipbuilding is a relatively new application area in comparison with robot welding in automobile industry. Due to geometrical complexities of ship hulls [1], especially in the fore and aft of the ships, programming is usually an awkward task. Simulation and off-line programming tools are most useful in this aspect for virtual prototyping and recollection of CAD-geometries for robot welding [2, 3, 4, 5, 6, 7, 8].

Additional problems include the difficulty of transportation and assembly of the robot to the working place and the design or selection of a robot system with a working space that is large enough for the hull panels. In present case the main issue is to build a robot that is light enough to be transported and assembled manually, is fixable in the reference hull cell and that has the reachability and power to perform the required welding operations.

The design of a new robot in the present case is fundamental for obtaining the defined prerequisites of this project, formulated by the shipyards. The robot that will be designed and built reminds strongly of a Motoman K3. The differences are mainly that the new robot will have extended range and will be made of aluminum alloy for the achievement of high performance characteristics and low weight.

The physical requirements on the robot system besides of what has been mentioned above are that it can be taken inside the structure through a man hole of \(600 \times 800\, \text{mm}\), weighting not more than 50 kg per part and assembled inside the cell. The structure of the cell is such that the robot must be mounted on a mobile platform, see Fig. 4.

Based on requirements defined by shipyards, the design work includes the use of Envision TR\(^1\) for analysis and evaluation of the robot performance during welding tasks. Of importance is the robustness of the system that relates to the ability to cope with the range of geometrical dimensions as well as fluctuations in tolerances. This must include the capability to perform the task within the joint limits of the robot system (including the movable robot base) and avoidance of kinematics singularities within the motions of the weld task that may jeopardize the result.

The paper will focus on the following:

1. Virtual design of a robot system using simulation tools such as Envision TR.
2. A systems description of the robot system and the application area specification.

\(^1\)Deneb Robotics Inc.
3. Methods to simulate the robot system including sensor interaction. In this case specific macros will be included that uses search techniques with the torch (mechanical/electrical contact) and weaving (tracking using through-the-arc sensing principle). Simulation of the system including the sensor interaction and tolerances is an important part to validate the robustness of the design.

Work presented within this paper is part of an on-going BRITE/EURAM project ROWER-2.

2 Simulation development process

The use of simulation in the process of the design of complex mechanical systems, decrease the risk of design errors and saves time, money, equipment and extensive specialist knowledge [9]. The process of innovation is drastically empowered by the aid of virtual prototyping, since building of a physical prototype often slows down the innovation process and is usually quite expensive. Further more, a physical model do not allow parameterization, an indispensable feature that allows mapping of the areas of robustness in a fast and efficient way, which is fundamental for optimization.

It is a disadvantage to build a physical model in one place and to convey the information back to co-designers in an another place of the world. An optimal design process requires a functioning model of the system to be available during the whole design iteration process. The transportation of a physical model from one place to another is unthinkable and it is not always possible to gather the necessary assembly of experts under the banner of one single city or country, yet less in one building. In this work the simulation technique is used as a tool to support the design process. During a pre-study, functional requirements and design parameters were identified to be used in the first conceptual design for simulation.

After a study of the nature of virtual prototyping, the conclusion is that there exist in a sense two mutually opposite simulation strategies: the “breadth first simulation” strategy [10] and the “depth first simulation” strategy [11, 12]. Breadth first simulation consists of five major steps, see Fig. 1. The building process implies the design of a simple but functional CAD-model of the entire mechanical system. The model should include the basic parts of the mechanical system and the kinematics relationships between the parts. The testing of the model reassures that the model is working accordingly. Validation implies that the model is tested against real data. If the model fails to match reality, it has eventually to be refined and the procedure of validation is repeated until the simulation match with real data. Parameterization and definition of variables prepare the model for experiments and robustness tests. Optimization implies the identification of the areas of robustness and determination of the
parameters that give the most efficient or robust system configuration.
The second strategy, depth first simulation is the crawl-walk-run method that in the most extreme form implies full simulation of every part of the system, before the integration of the entire system is made.

The normal approach to simulation is often an unconscious combination of these two strategies, often in practice a random combination of these two. However, a careful examination should be made after the pre-study to identify the critical parts of the system that need to be analyzed and modeled in more detail. This is important to speed up the iterative process during the design of the system.
3 Simulation environment

Envision TR, running on SGI, was used as the primary tool during this work for all simulation tasks. Pro/ENGINEER\(^2\) was used for modeling the modified K3 robot. Envision TR is a Deneb product for telerobotics and offers a graphical real-time simulation environment with the most sophisticated features for simulation, control and virtual prototyping of robots.

The internal programming language, called Graphical Simulation Language (GSL) [13] is a script language used to execute tasks within the program. Linking the simulation environment with the real world is possible by the use of Low Level Telerobotics Interface (LLTI) and Axxess. LLTI [14] provides an environment in which real-world devices may be programmed and controlled directly from the simulated workcell. LLTI automatically maintains the accuracy and integrity of the simulated workcell according to subtle changes in the real-world environment through the use of real-time sensory feedback.

For more advanced telerobotics applications, Envision TR also provides a flexible API (Application Programmer Interface) framework called Axxess [15], in which it is possible to link the kernel of Envision TR with user-code written in C. It is also possible to access any functionality within Envision, whether it is the user interface, the geometric database, the graphics display or any other functionality and to replace the source code of Envision TR with user-written code.

The Axxess library of functions provides a comprehensive set of routines that the developer may use to build his own application, a feature that has been used during this work to build a basic and straight-forward sensor model; an embryo of a virtual sensor to be developed in future work. The Axxess open architecture framework is based on Dynamic Shared Objects (DSOs), an Executable and Linking Format object file (ELF), relocatable at run time, and very similar to an executable program but with no main routine. It has a shared component, consisting of shared text and read-only data, a private component, consisting of data and an indirection table also know as the Global Offset Table (GOT). It has further several sections that hold information necessary to load and link the object and a liblist, the list of other shared objects referenced by this object. Most UNIX (including IRIX used by SGI) software vendors supply their libraries in DSO form. In Windows NT, these libraries are in DLL format.

4 Experimental

A model was created of the welding robot, the mobile platform and the double hull reference cell based on a conceptual design of the robot system, see Fig. 3.

\(^2\)Parametric Technology Corp.
Welding paths, specified by the shipyard were created and attached to the hull cell.

Figure 3 Reference cell used in the design process.

Since the welding robot is similar to the Motoman K3 robot, see Fig. 4, a model of the K3 robot was used as the base structure of the welding robot. The implementation of the inverse kinematics of the welding robot was easily realized by inheritance of the optimized inverse kinematics routines of the Motoman K3 model.

All the parts of the K3 model were redesigned to fit the new specifications and a welding torch model was designed and attached to the robot. The model of the welding torch in Fig. 5, was designed in accordance with the first suggestion of the co-designers of the robot system. Early simulation work proved however that the welding torch was almost aligned with the sixth axis of the robot, creating a singularity. The weakness of the first torch was that the angle between its welding axis and the sixth axis of the robot was too small, or more specifically only 7 degrees. It was suggested that a welding torch with an angle between 25-45 degrees should be used, see Fig. 4.

To be able to reach all stiffeners, the welding robot was mounted on a mobile platform (MP), movable in the vertical direction, see Fig. 6. This platform enhanced the number of the degrees of freedom for the robot from 6 to 7, increasing thereby its reachability. Since the MP does not move during welding however, it will take no active part in the welding process.

A reference cell with the dimensions of $1950 \times 2600 \times 3500$ mm was used in
Figure 4 Robot design for joining large structures.

Figure 5 The welding torch model in accordance with the first suggestion of the co-designers.
the simulation. In practice every ship has a huge amount of different types of
cells. The reference cell represents a typical cell in any ship and the main idea
is that once the robot system is fully adapted to the reference cell, it will also
be able to operate, with some programming modifications, in the other cells
as well.

The recommended design process for this project at this particular phase, is
to first set up a priority list over the parameters in the robot system that may
be modified. In some cases for instance, the robot is already selected and
may not easily be exchanged. In this project the robot is not built yet, but
the design process has been terminated. It is in the present case not possible
to make any suggestions for improvements, unless a serious flaw is found
in the robot design. Yet another reason for not changing the robot design,
is that the design of industrial robots in general are highly optimized and
the possibilities of making an improvement of the design of a K3 robot, for
instance, is quite limited in comparison with the labor that such an attempt
requires.

In this project, the design of the MP and the selection of the welding torch
is however currently under progress. The parameters of interest for the MP
are among others, dimensional values and values of maximum torque in dif-
ferent directions. Some important parameters for the welding torch are the
dimensional values and the maximum current capacity.

The application of the breadth first simulation in this context might very well
be the fact that it was decided before the start of the project, on a draft level,
how the robot system should practically be realized. No consideration was
made to the very details of the realization of the project. The application of
depth first simulation in its pure form might be the development of the robot
manipulator in advance and without influence of the design of the rest of the
robot system.

An important factor in choosing simulation strategy is to identify interde-
pendent parameters within and between the sub-systems. The principal rule
is, wherever there exist independent parameters to be optimized, depth-first
simulation is applicable. On the other hand, when the parameters to be op-
timized are dependent on each other, breadth first simulation is the method
of choice.

At present the design of the MP is in the iteration and optimization phase, see
Fig. 7. Different suggestions has been made for the design of the MP. These
include the L-shaped platform, envisaged to allow the robot manipulator to use
its full mobility. In order to minimize the weight and maximize the stiffness of
the MP, also a diagonal MP has been considered. Note that since two walls has
to be welded in each cell, the MP has to be reversed in direction or switched
to its mirror counterpart. If the MP is not able to switch direction, a pair of
MPs has to be designed, each for one direction.
Figure 6 The mobile platform (MP) in simulation.

Figure 7 A sketch of the MP and the parameters to be optimized, top view.
At the beginning the suggested values for simulation were 500 mm both for D and H. After selection of the first weldgun, D and H were changed to 718 and 397 mm respectively. The conflict of interest between different parameters is very likely to occur in the design of robot systems. In the current case a torch had to be selected and the dimensions of the MP determined to avoid (during welding):

- collision with the hull,
- collision with the MP or itself,
- singularities.

All simulation work was carried out using the collision detection and the singularity notification features of Envision TR. Since the torch geometry was the cause of singularity and collision problems, a new torch was selected and the parameters D and H were iterated once again. The best values found for the new torch at present, are D = 617 and H = 450 mm. There are of course some restrictions to the dimensions of the MP. The MP itself consists of an overhead, carrying the L-shaped platform. The overhead, called GLC-10\(^3\), is a commercial linear guide system, able to move a specific payload. Though it has a high dynamic capacity and maximum speed, its momentum capacity is however quite modest. Since large values of D and H might overload the linear guide, not every value of D and H is acceptable. At present the new values achieved from the simulation are examined by the sub-designers of the MP to ensure that the linear guide is not overloaded.

5 Implementation of virtual sensor

One of the reasons to develop a virtual sensor model was to attain independence of the welding equipment used in this project. Since the co-designers of the robot system are spread over different countries in Europe, the equipment may not at all be present when needed. Other factors are the possibilities to test and verify the functionality of a sensor that will be used in the system. Thus, controller software as well as sensor functionality can be developed and tested [16, 17].

Adding to all the advantages that has been related to the virtual prototyping method, the sensor model also increases the possibilities to optimize the design conditions of the robot system in a systematic manner, much due to the feature of parameterization and variable definition in virtual prototyping.

The foundation of a virtual sensor model was laid during this work. A simple linear sensor model was integrated with the kernel of Envision TR. The virtual sensor measures the current in the welding equipment, a technique called

\(^3\)NIASA, Spain
through-the-arc sensing. The principle behind through-the-arc sensing is based on the assumption that the current is inverse proportional to the distance of the welding electrode and the work piece during weaving. In reality, the relation between the current and the distance is non-linear and stochastic (including noise).

It is thus possible to dispense with torch sensors such as laser-beam sensors for distance measuring. Since a vision camera will be placed on the welding torch, there is no room for an additional sensor. A hall sensor that monitor welding current, do not have to be placed on the robot and saves crucial space. The weaving is a sine-curve motion perpendicular to the bisects of the planes of the workpiece, see Fig. 8. The weaving motion is executed by the GSL-script below:

```
PROGRAM wmot
VAR
  current : real
  max_current_comp : real
BEGIN MAIN
  $weave = on
  $WEAVE_AMPLITUDE = 10
  $WEAVE_FREQUENCY = 4
  $speed = 10
  $MOTYPE = AUX1
  current = 50
  max_current_comp = 25
  MOVE TO ww2
END wmot
```

The main programming however, has been done in C and linked with the kernel of Envision TR. The flow diagram in Fig. 9 gives the primary structure of the code.

## 6 Results

The first design suggestion of a robot for arc-welding in ship hulls was tested and modified by use of simulation and the virtual prototyping method. Important but not insurmountable obstacles for realization of the robot design were found in the first design iteration and the information was communicated back to the designers of the sub-systems. The problems found relate to the welding torch and the mobile platform. Simulation showed that the welding torch was almost aligned with the sixth joint of the robot, creating a singularity. The change in the simulation environment of the torch angle to 25-45 degrees proved to be a better solution. An embryonic virtual sensor
was created measuring the weld current in weaving. The virtual sensor was integrated with the simulation environment. The concept of the "breadth first simulation" and the "depth first simulation" strategies was introduced.

7 Future work

In future work, the embryonic virtual sensor model is planned to be extended by the introduction of neural networks or another similar method. The model will be based on empirical data from the welding equipment.

Besides the use of simulation performed in this work, it could also be used within the following areas in future work:

- Examination of workcell with updated information about the geometry of the robot and the MP.
- Dynamic analysis of the joint torques and forces on the robot and the MP.
- Introduction of link flexibility in simulation and analysis of eigenfrequencies at weaving.
- Simulation of the welding cables to avoid undesired collisions and twisting accidents.
- Simulation of vision system.
- Integration of the welding software and the virtual model for software debugging.

Acknowledgment

The ROWER-2 project is funded by European Community under the BRITE/EURAM program.

References


Figure 8 *Simulation of robot weaving with virtual sensor.*
IF MIN OF THE WEAVING CYCLE HAS BEEN REACHED, LOOK FOR MAX.

MEASURE MINIMUM DISTANCE BETWEEN TOOL-TIP AND WORKPIECE.

IF DISTANCE < MIN. DISTANCE, USE TRESHOLD VALUES.

IF MAX OF THE WEAVING CYCLE HAS BEEN REACHED, CALCULATE AVERAGE CURRENT BY INTEGRATION. THEN LOOK FOR MIN.

IF MIN OF THE WEAVING CYCLE HAS BEEN REACHED, LOOK FOR MAX.

CALCULATE ADJUSTMENTS OF THE TOOL-TIP RELATIVE TO THE NOMINAL TRAJECTORY.

SAVE DATA IN LOG-FILE. THE FILE CAN BE OPENED IN MATLAB FOR ANALYSIS.

Figure 9 Present C-code linked with the kernel of Envision TR.
Paper II

Virtual prototyping and experience of modular robotics design based on user involvement

Mikael Fridenfalk, Ulf Lorentzon, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

In Proceedings of the AAATE 99 Conference,
November 1-2, 1999, Düsseldorf, Germany
Virtual prototyping and experience of modular robotics design based on user involvement

Mikael Fridenfalk, Ulf Lorentzon, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

Abstract

Robotics in rehabilitation provides considerable opportunities to improve the quality of life for physically disabled people.

However, practical results are limited, mainly due to the lack of development of new robotics concepts where people are working together with robots in highly unstructured environments. A way to meet these conditions is to apply a modularized approach that allows for customization of the robot on an individual basis.

This paper highlights the user requirements and how they affected the technical demands in the robotics design. In the design process the “virtual prototyping” method was used. The robot design is in a technical validation phase and the results related to the user requirements are promising. The presented work shows a technique that make it possible to customize a robot called Asimov for a disabled user depending on the individual need.

1 Introduction

Robotics with a high degree of autonomy is a challenge in robotics research and can advance the technology within the health care area. The present status of robot systems, in general those within industrial automation, is that they are more or less used as programmable automatic machines.

Most robots used in rehabilitation today have similarities with industrial robots, such as the RT-series robots and SCORBOT which originally were developed for educational purposes and have been used within projects for disabled people [1, 2] often with adaptations for rehabilitation purposes as Handy-1 [3] which is used to assist in eating [4] and DeVar which uses a PUMA robot for assisting disabled at home or at vocational workplaces [5].
An example of a commercial robot designed for mounting on a wheelchair is the relatively advanced designed Manus robot which can be considered to be the most successful robot for disabled people of its kind [6].

New designs are on the way that will introduce the use of compact and flexible arms as well as new drives/actuators. Examples of this include the Tou soft (flexible) assistant arm [7], the pneumatically driven Inventaid arm [8, 9] and the compliant actuator Digit Muscle [10]. The interest in combining a wheelchair and a manipulator is increasing, not only the manipulator itself, but also through the enhancements of the wheelchair control by providing it with sensors and a control system as with any mobile robotic base [11]. There are currently research within the MOBINET project which will integrate innovative design solutions of a mobile base with a robot arm. Developments described in this paper form a subsystem within MOBINET [12, 13, 14].

This paper will address specific problems associated with the development of the robot system Asimov for disabled people and the technology behind it.

An important issue related to the human machine interface is the relatively complex control pattern needed to define a robot motion even with a proportional joystick with two degree of freedom. The different characteristic configurations including six degree freedom in space require additional user interaction similar to developments in adapting user interfaces for omni-directional wheelchairs [15, 16].

The main contribution shown in this paper can be summarized as (1) the design solutions resulting in a modular approach of customizing a robot and (2) the method of applying virtual prototyping techniques with extensive use of advanced simulation tools during the design process and for validation of the functionality.

2 Modular design and virtual prototyping

The starting point to create a robot for disabled people was to identify a set of user requirements that can form a basis for technical specifications. The requirements were formed by rehabilitation centers in cooperation with potential end-users such as people injured in working life and people with high spinal injuries. Based on the user requirements and the resulting technical specification, solutions were generated and tested against the specified needs, see table 1.

It is clear that any robot to be used by a disabled person will be a challenge in terms of usefulness related both to the manipulative functionality and to the human machine interface. However, the aim was not to identify a user specification and design one robot that would fit any disabled person but to develop innovative solutions that make it possible to customize a robot for disabled people on an individual basis. The customization aspect is important
Table 1 *Technical specification.*

<table>
<thead>
<tr>
<th>Technical Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight (robot arm)</td>
</tr>
<tr>
<td>Payload</td>
</tr>
<tr>
<td>Working range</td>
</tr>
<tr>
<td>Maximum joint speed</td>
</tr>
<tr>
<td>Maximum Cartesian speed</td>
</tr>
<tr>
<td>Power</td>
</tr>
<tr>
<td>Configuration</td>
</tr>
</tbody>
</table>

since there are different needs based on both the individual requirements and the disability which in itself may need specific considerations including changes over time.

One way to meet these specifications was to design a robot where each joint is an independent module. Each joint module consists of one or several motors and the peripherals needed for driving the joint(s) and make communication with the rest of the arm possible. The complete robot will be built from a number of standardized joint modules and connecting links. The modular approach makes it possible to adapt the robot to a specific user and situation since it will allow a great flexibility to the configuration and size of the robot. For example a robot mounted on a desk could have a 5-axis configuration with short connecting links which would give a small working space. A robot mounted on a wheelchair would need a larger working range and more joints to be able to reach a desk surface or a book shelf. It must also be able to fold up in a position within the chair’s outer limits. With the Modular approach it is possible to adapt the robot for these applications.

To be able to design and analyze a complex system such as a robot it is important to use modern techniques where simulations and mechanical design run in parallel with the development of the controller hardware and software.

A design method that lately have been introduced is the virtual prototyping method where the designers work closely together with simulation engineers and exchange models and experiences during the whole process. Thus, a team of experts is able to cooperate in the design process and continuously assess a large number of features by using virtual prototypes and simulation models [17].

A virtual prototyping system, as described in figure 1, consists of a solid modeler and a simulation software. The models created in the CAD system are transferred to the simulation system and the complete mechanism is assembled and simulated. The simulation software makes it possible to analyze and calculate vital engineering information, such as motor torques and interlink forces. The virtual prototyping method makes it possible to de-
velop and test a large number of virtual prototypes at low cost thus reducing development time and costs. Within this project, the softwares used were PRO/Engineer for the solid modeling and Envision TR for the simulations.

![Diagram of the virtual prototyping system]

**Figure 1** A schematic showing the principals of virtual prototyping. [18]

When using the virtual prototyping method the designers are able to explore a wider range of possible solutions without making a single physical prototype until the design is finalized. This makes it possible to explore unorthodox design solutions that normally would have been rejected at an early stage of the design process. The visualization of the models in the simulation system makes it possible for people without an engineering background to get a fairly good understanding of the model which is useful in discussions with future users and customers.

### 3 Technical development

The Asimov robot, described in figure 2, is a telerobotic arm with 7 joints, primarily designed to be attached to a wheelchair (from the manufacturer Permobil in Sweden) and controlled by a joystick. To fulfill the specifications formulated in table 1, the following measures had to be taken to obtain maximum efficiency, minimum weight and minimum power consumption:

- Components of lightweight material
Figure 2 Schematic depiction of the robot system. One of the modules is highlighted and extracted from the manipulator.
- Selection of brushless D.C. motors
- Selection of planetary gear boxes

The power source of the robot and the wheelchair is a dual set of standard car batteries (12 VDC) providing for 24 VDC. Since modular design require that the motors and the electronics have to be enclosed into the minimal space of the robot arm, special electronic circuits have been designed. The principal scheme of the electronic system of a single module is presented in figure 3.

![Figure 3 Scheme of the information flow in the electronics.](image)

Every module contains a microcontroller from Permobil and is connected by the CAN-bus to the main computer. Only four wires are running through the robot arm: two for the power-supply and two for the CAN-bus communication. The motion control unit contains besides a motion generator, also PID-control and generates pulse width modulated (PWM) signals that runs the motor after the commutation order defined by the programmable array logics (PAL) and amplification (H-Bridge). There are some feedback signals from the motor where the hall signal is essential for motor commutation and the encoder signal for the motor positioning.

The information flow is also presented at a higher level in figure 2 and needs no comments besides that the program input refers to Envision TR.

## 4 Validation of technical specification and user requirements using VR-tools

Virtual prototyping was used as design process which helped to make sure that there would not occur any miscommunication between the end-users and the development team concerning the product.

The use of virtual prototyping tools showed to be a successful strategy of visually clarifying the potential of the project (see figure 4). After establishing a list of requirements, based on consultation with potential users, a refined
model was produced, that should be able to, at least in theory, to fulfill the user requirements.

The models of Envision TR [19], precisely represent the actual geometry, motion and dynamic characteristics associated with the real world system. Envision TR delivers (with the dynamic option) dynamic data such as forces and torques for arbitrary revolute and prismatic joints. While kinematics modeling gives a good hint of how the physical body of the robot should be dimensioned, the dynamic modeling deliver information essential for motor dimensioning.

This information simplified and accelerated the iterative process of design throughout the project and secured that great consideration of the user requirements was taken. Furthermore, the simulator was used to, in real-time, generate and perform the motion control of the physical robot during technical validation and testing.

The major problem during the testing proved to be some instability and oscillation. The problem was however temporarily solved by the replacement of the base motor with a higher torque.

At start, Asimov was designed to have 8 degrees of freedom, excluding the gripper, for optimum reachability. At a later stage the removal of one motor was scrupulously examined in Envision TR and a new and better solution was found, a task that would have required much more time and calculations to accomplish, without the ability of VR-prototyping.

The possibility to develop entire motion procedures and working scenarios
using simulations and VR-tools showed to be a useful working method for
cross-disciplinary teamwork. Fields of interests for Asimov are such opera-
tions as opening/closing doors, putting/taking a book from a shelf, grabbing
a glass and drink from it, etc. A number of successful simulations have been
performed concerning this kind of operations.

5 Results

A modular robot was designed by the virtual prototyping method and a seven
axis prototype has been built. The physical robot showed to be very well
correlated to the virtual model. The initial technical validation of the robot
proved well to match the basic initial specifications.

The weight of the robot is approximately 13 kg and the working range about
1.5 m. The maximum joint speed surpass the specified. It is possible to
increase the ratio of the reducers by a factor of four and still preserve the
maximum joint speed of 90 degrees/s. Some instability problems that were
identified will most likely be solved in the next generation of Asimov by the
implementation of adaptive control and the reduction of the size of the robot
arm.

The power consumption proved to be very low, about 50 W at typical perform-
ance, which is quite insignificant in comparison with the power consumption
of the rest of the wheelchair.

6 Discussion

Although much was achieved by the virtual prototyping approach, there is
still need for improvements of the next generation of Asimov. Below follows
a list of suggestions, based on an interview with a potential user of Asimov
(during a workshop held recently) and experimental results received from the
trial run of the robot:

- Reduction of the length of the robot arm by a factor of 2/3.
- Reduction of the size of the electronics. The present controller card and
  the motion controller circuits should be replaced by a microcontroller
  chip. This work is in progress.
- Introduction of current monitoring for the motor phases. This enables
torque control and eliminates the motor fuses.
- Study the possibility to increase of the gear box ratio by a factor of 2-3.
• Implementation of full modularity and unlimited rotations by introduction of slip ring connectors.
• Introduction of adaptive control.
• Reconfiguration of the robot arm to resemble a human arm.

Other more innovative solutions are the replacement of the CAN-bus with radio communication and the implementation of a “plug and play” strategy, where the modules could be connected and unconnected by a simple operation. The modules would instantly recognize the new configuration, change the kinematics model and update the adaptive control.

Acknowledgments

This work was funded by AMS (The Swedish National Labor Market Board), RFV (National Social Insurance Board), NUTEK (Swedish National Board for Industrial and Technical Development) and the European Commission within the TMR Programme, project MOBINET.

References


Paper III

The Unified Simulation Environment - Envision
Telerobotics and Matlab merged into one application

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

In Proceedings of the Nordic Matlab Conference,
October 17-18, 2001, Oslo, Norway
The Unified Simulation Environment - Envision Telerobotics and Matlab merged into one application

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University
Box 118, SE-221 00 Lund, SWEDEN
www.robotics.lu.se

Abstract

The kernels of Matlab and Envision Telerobotics (TR) were merged on the SGI platform, taking advantage of Matlab Engine and the open architecture of Envision TR, designing a new robot simulation environment called the Flexible Unified Simulation Environment (FUSE). Envision TR is a time-continuous graphical robot simulation program from Delmia Inc.

In FUSE, Matlab is able to take advantage of geometrical libraries for most commercial robots on the market and the optimized graphical kernel for real-time 3D applications offered by Envision TR. Envision TR is gained by full access to Matlab's vectorized calculation environment accelerating either the speed of execution at run-time, in some cases above the speed that compiled C-code offers, or at debugging and software simulation, offering the capability to modify the software at run-time, altering the program flow instantly in the middle of program execution. In addition, development of new robot systems are optimized by the ability of using specialized Matlab toolboxes in diverse application areas, such as statistics, automatic control and artificial neural networks, minimizing the software development time, effort and risks taken in major robot projects.

FUSE was used in the ROWER-2 project, a European Union project aimed to develop a robot system for automatic arc-welding of hulls in tanker ships. The seam-tracking and sensor control software developed in FUSE were implemented in the target QNX embedded system.

KeyWords: Robotics, Simulation, IGRIP, Envision TR, Matlab Engine, software development, systems development, virtual sensor.
1 Introduction

1.1 Integration of systems

Integration of systems saves time, effort and money. The requirement of universality makes software and hardware products to either improve their compatibility with current standards or other products, or to increase their market shares enough to be able to establish their own standards. One example is Matlab (www.mathworks.com), an application for numerical calculations and one of the standard applications used for scientific research and development. Though Matlab is regarded to be user-friendly and to possess many optimized mathematical tools, the real strength lies however in the open architecture and provision of an environment where the users easily make addition to the application [5,6]. Matlab has also established compatibility with many software and hardware standards on the market during its history, aiding the process of data analysis, software prototyping and automatic code generation for among others, embedded systems [7].

In contrast to the strategy to gain market shares by offering a high amount of compatibility and open architecture, some companies have established universal standards by using their market dominance. These software are usually not provided with open architecture and do not promote seamless integration with similar applications on the market, neither do they in general offer compatibility with file formats from other competing products. They offer however as a rule a great amount of integration and automation within their respective product family.

1.2 Integration in robotics

There is presently a great need in robotics for seamless integration of CAD/CAM/CAE and robot simulation applications into one single and working solution. There exists however no such system today. Within the application area of robot simulation the most common applications are IGRIP (www.delmia.com) and RobCAD (www.tecnomatix.com). These programs offer graphical real-time environments and geometrical libraries for most common commercial robots on the market. They also provide for kinematics models for all provided robot models, as well as many accessories, such as welding torches and fixtures.

1.3 FUSE

The design of the Flexible Unified Simulation Environment (FUSE), done by merging the open architecture version of IGRIP, called Envision Telerobotics (TR), with Matlab, is in no wise an attempt to create a unified system integrating robot simulation and with CAD/CAM/CAE. It is a unified system consisting
of the integration of robot simulation with software prototyping. Since the main development time designing a new robot, as a rule is spent on systems development, the integration of robot simulation with a software prototyping environment seemed to be the most efficient way to improve the overall productivity in robotics design.

1.4 Objective

We designed FUSE to be used in the development of a number of sensor control systems for seam-tracking at arc-welding in the European ROWER-2 project. The aim of the ROWER-2 project was to develop a robotics system for automatic welding of hulls in ships [4,8], mainly super-sized cruisers and oil tankers. The sensor control models were originally based on the concept of the Yaskawa COM-ARC III [9].

2 Materials and methods

2.1 Envision TR

Envision TR, is a real-time graphical robot simulation program. It contains, apart from a nearly optimal graphical engine, also full geometrical and kinematics libraries covering most commercial robots and many robot accessories on the market. Figure 1 displays the Graphical User’s Interface (GUI) of Envision TR, including the robot Galileo designed specially for the ROWER-2 project by Tecnomare, Italy.

In contrast to robot simulation programs from Delmia such as IGRIP, Envision TR is shipped with open architecture, and provides the source code for a significant part of the application. Figure 2 shows the open architecture structure of Envision TR. The user may modify or replace many libraries with user written code and add new features to the application. The application is created by compiling and linking all libraries to Envision TR.

2.2 Matlab

Matlab is an application and a programming language widely used for numerical calculation in education and research, both in university and in the industry. The extensive libraries offered by Matlab makes it usable for software development and testing within a wide range if engineering fields. Programs written in Matlab may be converted to C/C++ and compiled for faster execution or integration with other products. The Application Programmers Interface (API) of Matlab, also called Matlab Engine may be linked to user-written software in C/C++.
Figure 1 *Envision TR GUI*
Figure 2 Envision TR Open Architecture Structure. The libraries marked with an asterix are not modifiable by the user.
2.3 FUSE

We designed FUSE application kernel by compiling and linking Matlab Engine with Envision TR and their share libraries [1,2,3], also known as Dynamic Shared Objects (DSOs) or Dynamic Link Libraries (DLLs). The compilation was performed on the SGI, IRIX platform, compiling for the o32 compiler architecture and FUSE was successfully tested for use on an O2 and an Octane computer using 32 bit and 64 bit architectures respectively. The same merging procedure was theoretically possible, but was not carried out, on PC Windows environments, but also on all other platforms that supported Envision TR and Matlab Engine.

Envision TR was modified and adapted to the new application structure by modification in the Motion Pipeline[2]. The Motion Pipeline, written in C, was part of the libdnbusr.so DSO and was used for calculation of new positions for the robot joints about 30 times per second. Envision TR and Matlab is a server-client system. Envision TR plays the role of a passive client and Matlab the role of an active server. During the calculations, the Motion Pipeline is made to halt and have to wait for Matlab to finish its task before it is allowed to continue the execution. If the calculations in Matlab takes more than about 1/30s, the simulation rate becomes dependent on Matlab instead of Envision TR. By merging these application kernels, Matlab is received full access to the graphical engine of Envision TR and the geometrical robot libraries. And vice versa, Envision TR is given access to the Matlab environment, including all available Matlab Toolboxes. This includes also automatically full integration of Envision TR with Simulink, Stateflow and consequently also with Real-Time Workshop (www.mathworks.com).

This section is concluded by an example showing how to transfer a transformation matrix (4x4) from Envision TR Motion Pipeline (C-code) to the Matlab workspace, within the FUSE environment. The reverse process is performed in a similar manner. It should be remarked, that though the transformation matrix in the example below, per definition always had to be transposed in Envision TR, with respect to the first and second index parameters, it ended up right in the Matlab environment automatically by the transfer.

```c
/*--------------Sample code--------------*/
/*--------------Declaration--------------*/

static Engine *ep;
...
mxArray *P_init_mxArray = NULL,
... double P[4][4];

/*--------------Assignment---------------*/
```
/* Assigning numerical values to P */

/*---Transferring a 4x4 Matrix from Motion Pipeline to Matlab---*/

/* Create and load P_init into Matlab */
P_init_mxArray = mxCreateDoubleMatrix(4, 4, mxREAL);
mxSetName(P_init_mxArray, "P_init");
for ( i = 0; i < 4 ; i++ ) {
    for ( j = 0; j < 3 ; j++ ) {
        P[i][j] = initial_location->xform[i][j];
    }
}
/* Assigning perspective/scale vector 
   for explicit order notification */
P[0][3] = 0; P[1][3] = 0;
P[2][3] = 0; P[3][3] = 1;
memcpy((void *)mxGetPr(P_init_mxArray),
       (void *)P, sizeof(P));
engPutArray(ep, P_init_mxArray);
/*---------------------------------------*/

3 Results

We integrated Envision TR with Matlab on the SGI platform, resulting in the
new application kernel FUSE. The fusion was successful and was made by
linking and compiling the source code of Envision TR with Matlab Engine. This
new application was fully stable and the only time it crashed was occasionally
at the startup process due to some basic bugs in Matlab v 6.0 R12 for IRIX,
SGI. This bug may most likely be removed in the next version of Matlab.

FUSE can theoretically run in two major modes. In the real-time execution
mode, Matlab's highly vectorized functions were used to improve execution
speed, even in some cases compared to optimized C/C++ code. In the soft-
ware prototyping mode, modification in the software in the middle of program
execution led to instant change of the program flow. In both modes it was
of course possible to use the wide range of specialized Matlab toolboxes in
diverse application areas, such as statistics, automatic control and artificial
neural network, to minimize the software development time, effort and risks
taken in major projects.

In general, FUSE led, compared to traditional robot system development tools,
also to the need of fewer programmers and administrators, cutting the costs,
minimizing management and thereby increased the control over the code,
which indeed also resulted in safer programs. The improvement of productiv-
ity that FUSE was provided for, made us to develop, instead of only one sensor
algorithm, a great number of sensor control algorithms in the ROWER-2 project, many of them more competent and robust than any similar product on the market. The prototype of one of the sensor control algorithms was finally selected, translated to C++ and implemented on a Industrial PC board, a PIA-662 with QNX operating system.

4 Discussion

The full integration of Envision TR with Matlab shows on one level to save much time and effort in development work. In practice however, the total development time and effort seems in a way not to be reduced but rather increased. The reason is that FUSE encourages development of complex solutions, based on experiments and analysis that usually never is performed due to lack of time and human resources. In our case, we would have most likely only developed one single algorithm for seam-tracking and sensor control and we would have not taken any risks using more advanced and complex solutions than absolutely necessary.

The strength of FUSE is in this context that it encourages the user to use high-tech solutions since it makes software prototyping powerful and fun. Though the software development rate is in general increased by many times using FUSE, the documentation rate still remains the same. In the development of the sensor control algorithms for instance, the documentation rate is presently estimated to be more than five times slower than the actual development time for the software, which is a bottleneck. The good news is however that if future software prototyping environments will be like FUSE, perhaps the real programmers may in the future not dislike real programs any longer, but only the documentation of real programs.

5 Acknowledgements

The ROWER-2 project is funded by European Commission under the BRITE/EURAM program.

References


Paper IV

Design and validation of a sensor guided robot control system for welding in shipbuilding

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

In International Journal for the Joining of Materials,
Volume 14, Number 3/4, December 2002
Design and validation of a sensor guided robot control system for welding in shipbuilding

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Abstract

New areas in welding large structures in shipbuilding include joining large sections such as double-hull constructions. Joining these sections create great problems for a manual welder since welding takes place in a closed area with associated work environmental problems. The accessibility to the working area is limited to a man-hole and the use of robots for welding such structures requires new robot design that are adapted for the task as well as the additional requirements of one-off production.

This paper will describe research work and results within the ROWER-2 project. The aim of the project was to design a robot system for joining ship sections in the final stage when ship sections are to be assembled together in dry dock. Due to a high degree of manual work involved in the assembly procedure of the ship, the project addressed both productivity and quality issues. An important part within the project was to develop control algorithms for seam tracking during welding based on through-arc sensing. The aim was to be able to cope with tolerances in the joints after manual setup and tack welding of the structure.

A special software system, FUSE, was developed for this purpose that seamlessly integrates commercial available software tools such as Matlab and Envision (robot simulator). Simulation in FUSE showed that the major part of the development of sensor guided robot control algorithms should be performed by simulation, since it cuts time, expenses and efforts, especially when software simulation is included in the methodology.

1 Introduction

1.1 Robot welding

Since manual labor is a highly limited resource, especially when it comes to skilled craftsmen, robot automation is essential for future industrial expansion. One application area is presently robot welding, by which the welding
quality and the environmental conditions for welders are improved and the productivity is increased. This applies especially to robot welding in shipbuilding [23, 24, 25] where large structures are welded, including the joining of large double-hull sections.

1.2 Seam tracking

Seam tracking [5, 7, 8, 13, 19] is essential for automation in shipbuilding for manufacturing of large passenger and cargo ships, such as super-cruisers and oil-tankers, where high tolerances in the sheet material are allowed to minimize manufacturing costs.

A great number of Sensor Guided Robot Control (SGRC) systems for seam tracking at arc welding have been developed. The patents within this application area during the last 40 years indicates that there is a clear tendency that old methods using mechanical, inductive, electrical and infrared sensors are becoming less important along with the use of electron beams and camera systems. Today laser scanners and arc-sensors mainly replace these systems.

Systems based on laser scanners and arc sensors differ in accuracy, geometry and price. Laser scanners provide for a more accurate signal than arc sensors, which contain much noise due to the interference of the welding process. On the other hand, laser scanners have to be mounted on the torch, decreasing the workspace of the robot. Laser scanners are also significantly more expensive than arc sensors, which perhaps is one of the reasons why the majority of the patents that have been issued during the last 10 years for seam tracking at arc welding [3, 16, 17, 26, 27] are based on through-arc sensing, while systems based on laser scanners are hardly even represented.

1.3 Process control

Besides seam geometry at seam tracking, considerations have to be made of process related welding parameters [1, 29]. The welding process contains many parameters, such as the arc voltage, wire speed and wire material. The aim is to determine feasible parameters for a welding procedure before seam tracking. This may be performed experimentally or by the use of knowledge based systems [4, 28]. If it is however not possible or desirable to keep these settings constant throughout the seam [14, 15, 18, 27], for instance due to the characteristics of the power-source, adaptive control may be introduced into the seam tracking procedure for maintaining the desired welding quality.
The usual methods used for automated arc welding are gas metal arc welding (GMAW), flux-cored arc welding (FCAW) and submerged arc welding (SAW). In GMAW, metal parts are joined together by heating them with an arc established between a continuous, consumable filler metal electrode and the workpiece. The filler metal is either transferred to the workpiece in discrete drops under the influence of electromagnetic forces and gravity or in the form of molten electrode produced by repetitive short-circuiting.

Through-arc sensing was introduced in the beginning of the 80th and is described by among others G. E. Cook et al. [9]. According to experimental results, the approximate relationship between arc voltage \( V \), arc current \( I \) and the nearest distance between the electrode and the workpiece \( l \), for an electrode extension ranging between 5-15 mm, is expressed by the equation:

\[
V = \beta_1 I + \beta_2 + \frac{\beta_3}{I} + \beta_4 l
\]

where the constants \( \beta_1 - \beta_4 \) are dependent on factors such as wire, gas and the power-source. Theoretically, if the power-source is adjusted for keeping the current at a constant level, and succeeds to do so, \( V \) will be a linear function of \( l \). Practically, the voltage and current readings of the arc contain much noise, why the signal data has to be filtered by a low-pass filter.

In through-arc sensing the welding is performed parallel to the seam-walls, see Fig. 1. By weaving the arc across the weld joint, the geometrical profile of the workpiece is obtained, since the distance from the tooltip perpendicular to the nearest wall, is a function of the arc current and the voltage, as approximately expressed in Eq. 1.

**Figure 1** Figure 1: Definition of Tool Center Point (TCP), and the orthonormal coordinate system noa. Weaving is performed in \( n \) direction, \( o \) is opposite to the direction of welding and \( a \) is the direction of approach.

1.4 Through-arc sensing
1.5 Control algorithms for seam tracking

Template matching

The first suggested method in [9] is template matching. In this method the width and centering corrections are made proportional to \( e_a \) and \( e_n \), where \( t(x) \) and \( s(x) \) are the template signal and the measured arc signal as a function of displacement \( x \) with respect to the center of the weld joint. The template signal is the measured arc current at welding, when the position of the workpiece is optimal. \( A \) denotes in Eq. 2 and 3 the weaving amplitude:

\[
e_a = \int_{-A}^{A} |t(x) - s(x)|dx
\]  

(2)

\[
e_n = \int_{-A}^{0} |t(x) - s(x)|dx - \int_{0}^{A} |t(x) - s(x)|dx
\]  

(3)

The template signal may here be analytically or empirically determined. Other examples of error calculations are by using the integrated difference and the integrated difference squared errors. Control in \( a \) and \( n \) directions in Fig. 1 is performed by comparing the average value of \( s(x) \) at, and near the weave center position with a reference value.

Differential control

The second method consists of differential control. It is computationally more simple and has proven to be quite reliable for control in \( a \) and \( n \) directions. Sampling is only made at the turning points in the weaving trajectory. Measuring the arc-signal, i.e. the current in the case of GMAW, FCAW or SAW, the error \( e_a \) in a direction will be proportional to the difference between the average current sampled at the center of the oscillation \( i(0) \), and the reference current value \( I_{ref} \):

\[
e_a = K_a[i(0) - I_{ref}]
\]  

(4)

In similar manner, the difference between the two samples is proportional to the magnitude of the error \( e_n \) in \( n \) direction:

\[
e_n = K_n[i_{+A} - i_{-A}]
\]  

(5)
where \( i_{+A} \) and \( i_{-A} \) are the average measured current at a pair of adjacent extremepoints and \( A \) is the weaving amplitude. The parameters \( K_a \) and \( K_n \) are dependent on the weld joint geometry and other process parameters such as shielding gas and wire feed rate. Since these parameters will be known in advance, \( K_a \) and \( K_n \) may be defined for any welding application.

1.6 Simulation using virtual sensors

Virtual sensors are presently used in many application areas, such as robotics, aerospace and marine technologies [6, 10, 20, 22]. Development of new robot systems, such as for seam tracking may be accelerated by application of simulation [13, 12]. In general the design methodology of virtual sensors may vary due to their specific characteristics. If the characteristics are not known for the design of analytical sensor models, artificial neural networks may be used [2, 21].

1.7 ROWER-2 application

The objective of the European ROWER-2 project was to automate the welding process in shipbuilding, specifically for joining double-hull sections in supercruisers and oil tankers. According to the specifications the workers have to able to mount the robot system inside the hull-cell and supervise the welding process from a remote distance. Every hull-cell should be equipped with a manhole, through which the robot may be transported, see Fig. 2. Since the robot has to be transported manually, the constraint on the robot is that each part is allowed to weigh no more than 50 kg. Further on, the robot system is designed to operate in standard hull-cells with predefined variations in dimension. To be able to meet the specifications, a robot manipulator was built based highly on aluminum alloy. The robot was mounted on a mobile platform with 1 degree of freedom to increase its workspace.

The method chosen for automatic welding in the ROWER-2 project was GMAW using through-arc sensing at seam tracking [9]. A simple seam tracking algorithm was developed at an early stage of the project [13]. By the application of the Flexible Unified Simulation Environment (FUSE) [11], this algorithm was optimized and evolved into many new algorithms. FUSE is an integrated software system based on Envision (robot simulator, Delmia Inc.) and Matlab (MathWorks Inc.). The algorithms were initially designed to contain the basic functionality of the Yaskawa COM-ARC III sensor [30] but were further developed to meet the ROWER-2 specifications along with others added by the authors.

Since one of the simple algorithms showed by simulation to meet the ROWER-2 specifications, it was chosen for implementation in the project. The imple-
mentation was performed in C++, running on a QNX-based\textsuperscript{1} embedded system. The algorithm was further developed to be able to handle long welds by using linear interpolation. A method was additionally developed to compensate for the power-source controller that interfered with the seam tracking process by disabling control in negative $a$-direction of the TCP. An automatic delay detection for synchronization between the control system and the data from the arc-sensor was also designed to secure control in $n$-direction of TCP.

2 Materials and methods

2.1 Systems development methodology

The methodology is based on the assumption that software development in robotics is the part that requires most time, money and effort at development of robot systems. To optimize the procedure of robot design from virtual prototyping and systems development to the final integration of the robot in a production line, a new software system was designed that can be used during the whole process. FUSE fills the gap that traditionally exists between CAD/CAM/CAE, robot simulation systems and software simulation environments.

\textsuperscript{1}QNX is an operating system for embedded controllers.
Figure 3 presents the methodology that was developed and used for the design and validation of a seam tracking algorithm in the ROWER-2 project. In this figure design of model denotes the design of a model that is focused on the functionality of the algorithm. Simulation and verification of the model denotes the process of estimating the potential and finding the limitations of the algorithm. Software structure simulation is not bound to the model itself but to the implementation of the model at a later stage.

It is an awkward task to debug complex algorithms after they have been implemented in a robot system. At an early development stage, however, a detailed simulation of the execution flow, supported by automatic testing programs will most likely isolate the problems. The test programs may be used to systematically find the limits of the algorithm and make sure that it behaves as expected. Any significant change in the algorithm structure in the verification phase implies similar changes in the simulation model of the software to mirror the program flow in the robot system. If the algorithm has been developed with care using software simulation to begin with, such modification
will require minimal effort.

By physical validation, deficiencies may be found in the simulation model. If the specifications are not met, the model is modified and new model and program structure simulations are performed. When the specifications are met by physical validation, the process is terminated by a final evaluation of the simulation model using the optimized parameters found by physical validation.

2.2 Joint profiles

The simulation was performed on fillet and V-groove joints, according to the specifications in the ROWER-2 project. Due to these, the algorithm had to be able to make compensations for deviations of $\pm 300$ mm in y and z directions (see Fig. 4) during a 15 m long weld. This was redefined as a maximum deviation of $\pm 20$ mm per meter weld, or $\pm 2\%$ expressed in percentage. Profiles of fillet and V-groove joints are presented in Fig. 4. Since the start point of the seam is always found by image processing before seam tracking is performed, no special consideration is taken at start.

![Fillet and V-groove joints](image)

**Figure 4** Fillet and V-groove joints were used in simulation and physical validation. The lengths of the sample plates in the experiments were 600 mm (orthogonal to the plane of the figure).

2.3 Experimental setup

The functionality of the seam tracking model and the essential features of the program structure were simulated in FUSE on SGI workstations and manually translated to C++ for implementation in the robot system running on QNX OS, using a real-time industrial PC. As power-source for welding, a Migatronic BDH STB Pulse Sync 400 for MIG/MAG was chosen operating together with a Planetics Mars-501 push-pull unit. The push-pull unit is suited for welding
cables up to 25 meters between the units and another 16 meters between the pull unit and the welding torch. OK Autrod 12.51 was used as welding wire together with 80Ar/20CO₂ shielding gas. Figure 5 displays the ROWER-2 system.

![The ROWER-2 robot system. The system is mounted on a mockup that is about 3.5 m high, belonging to the hull-cell of a ship. The metallic box near the torch is a camera that is used to identify the start point of the seam.](image)

**Figure 5** The ROWER-2 robot system. The system is mounted on a mockup that is about 3.5 m high, belonging to the hull-cell of a ship. The metallic box near the torch is a camera that is used to identify the start point of the seam.

### 3 Experimental results

#### 3.1 Simulation experiments

**Overview**

A number of seam tracking algorithms were developed and implemented in FUSE. The differential algorithm, which was considered to be the easiest one to implement and required the lowest amount of computational power, yet satisfying the ROWER-2 specifications, was chosen for implementation and physical validation in the ROWER-2 project. The simulation and validation
work presented in this section is based on the differential algorithm presented in the introduction.

**FUSE simulations**

The simulations in FUSE were performed using FUSE Wizard, see Fig. 6. The wizard lists the available algorithms and makes suggestions of parameters for the selected one before the start of the simulation.

The first algorithm in the wizard is a simple differential algorithm using distance measurements directly retrieved from the FUSE environment. This was an early prototype of the differential algorithm, but worked principally in the same way as the calibrated differential, implemented in the robot system. The calibrated differential algorithm uses a virtual arc-sensor and is calibrated by FUSE Wizard. At calibration, the value of the main current is calculated and saved. The calibration value may be changed before the start of any new experiment. In simulation the same calibration value showed to work properly for both fillet and V-groove welds. In physical experiment however, different calibration values for fillet and V-groove welds were used for optimal performance.

The two last methods in the wizard input dialog are two algorithms that use statistical methods to find the 2D profile of the seam, perpendicular to the direction of motion. The last algorithm is in addition able to adapt to the orientation of the joint perpendicular to the direction of motion by rotation of the TCP around the $o$-axis.

Initial simulations showed that the differential algorithm was theoretically able to meet the specifications. By physical experiments, the model was modified due to changes of parameters such as welding speed, weaving amplitude and frequency. These data, along with other data such as nominal voltage, were optimized by a series of tests performed by an experienced welding specialist tuning the welding system. To evaluate the simulation model compared to real experiments, a final series of simulations were performed using the power-source parameters that had shown to give high quality welds.

According to the evaluation, the theoretical limits for the algorithm is a deviation in the interval between $-10\%$ and $30\%$ with $K_a = 0.01$ and $K_n = .005$. This is better than $\pm 2\%$ (specifications). The conclusion is that the ideal case, when the current has very low amount of noise and Eq. 1 is valid, the maximum allowed deviation is $\pm 20\%$ (moving the offset of the nominal welding trajectory to the middle). The asymmetrical performance ($-10\%$ to $30\%$) is most likely related to the present control system.

Figures 7-17 present a few simulations performed for evaluation purpose after the verification of the algorithm by series of real experiments. In all these experiments, $K_n$ was $50\%$ of $K_a$, which is an empirically derived optimal value.
Figure 6 The FUSE Wizard. For each algorithm a special set of parameters are suggested that may be modified by user input before simulation. New algorithms may be added or removed from the list throughout the development process.
verified by robot welding experiments.

The criterion for a good weld is that the final result should at ocular examination be similar to a straight weld without seam tracking. To be able to produce such result, the smallest gains that still made it possible to meet the specifications had to be found, minimizing the instability that occurred during SGRC. At too low gains, the algorithm does not compensate enough. On the other hand, if the gain is too large, instability will occur resulting in low welding quality. So the trick is to find the largest gains for $K_a$ and $K_n$ both for positive and negative deviations and both for fillet and V-groove welds at which the seam tracking remains stable and converges smoothly to the seam. At these gains the maximum possible deviations are experimentally found by increasing the deviation step by step until the algorithm has reached the limits of its performance.

**Vector booster method**

To increase the performance of the algorithm, a feature was implemented called the vector booster. The vector booster method consists of a pre-calculation of seam direction at the beginning of seam tracking and subsequent alteration of nominal trajectory according to the initial prognosis for the rest of the seam. This method showed to increase the performance of the algorithm by a factor of up to two. The vector booster is not optimized for dealing with small deviations, but rather extreme deviations, many times larger than the specifications, see Figs. 7-8. The vector booster does theoretically increase the welding quality at large deviations, due to the need of lower gains in the SGRC system. With larger expected deviations, higher gains have to be set by the operator. And high SGRC gains may in turn decrease the welding quality due to higher instability at seam tracking, by causing small oscillations.

3.2 Validation by robot welding

**Overview**

Initial simulations showed that the algorithm used less than 1% of the computational power it was assigned, which equals 1/2000 of the total computational power of the real-time embedded system, so the efforts to produce efficient code was successful. The simulation and validation loop described in Fig. 3 was repeated twice and concluded with the evaluation simulations. The first physical experiments consisted of a number of fillet, V-groove and flat surface welds performed by the robot, with and without seam tracking. The primary task was to find the set of parameters for the power-source that resulted in good welding quality.

The second robot welding experiments were performed on author requests
Figure 7 Extreme deviations like this (60%) are not used in practice. According to specification, the algorithm has to handle a deviation of 2%, which is 30 times smaller than the one in the picture. The picture demonstrates however the effect of the vector booster, working especially well for large deviations. Although the simulation succeeded, a deviation of 30% or smaller is recommended using the vector booster for maximum quality of the weld.

Figure 8 The same experiment as previous figure. The vector booster showed to be of limited interest at moderate deviations. The method inspired however the design of the power-source compensation component in the algorithm, described later in this section.
Figure 9 The seam tracking algorithms were first developed using workpieces like this one.

Figure 10 At a later stage, when it turned out that all seams in practice were straight, only such workpieces were used for simulation experiments. A V-groove weld simulation is displayed in this picture.

Figure 11 Seam tracking with deviations by 30% in y and z directions, using the vector booster. Left: at $K_a = 0.010$ the seam tracking process is very stable, which is a prerequisite for high welding quality. Right: $K_a = 0.020$ gives high instability.
Figure 12 Seam tracking with deviations of -20% (left) and -10% (right) in y and z directions, with $K_a = 0.010$, using the vector booster. The process is fully stable, and at -10% the convergence is good.

Figure 13 The same as previous picture, but without the vector booster. As expected the convergence is even slower at -20% than before (left), but is still very good at -10% (right).

Figure 14 Seam tracking with deviations of 30% (left) and 40% (right) in y and z directions with $K_a = 0.010$. There is no instability and the convergence is fine, but there is some clippings at 40%. The theoretical positive limit without using the vector booster is thus 30%.
Figure 15 Seam tracking with deviations of 20% in y and z directions, using the differential method with direct measurement in FUSE. $K_a = 0.010$. The process is stable and the convergence is fine.

Figure 16 Seam tracking with deviations of 20% in y and z directions, $K_a = 0.010$. The stability is high, but in the left picture the weaving is too near one of the walls. In the right picture convergence is fine, since the vector booster is used. In both experiments current calibration was used, which is default for the algorithm.
**Figure 17** Seam tracking with deviations of 10% (left) and −10% (right) in y and z directions. $K_a = 0.010$. Good results, but these are the limits in positive and negative directions.

**Figure 18** Example of fillet weld.
and consisted of about 25 fillet and 25 V-groove joints. Also as a final evaluation of the implemented algorithm 10 fillet and 10 V-groove welds were carried out. In addition, an uncounted number of fillet and V-groove welds were performed to find good parameter settings for the power-source. The overhead experiments showed to be many compared to the pure seam tracking experiments. About 120 fillet and 75 V-groove joint workpieces were estimated to have been used during the two experimental occasions. These overhead experiments showed to be very important for the development of the algorithm. Without precise parameter settings of the power-source seam tracking would have worked, but without producing high quality welds.

**Anomalies caused by power-source**

In theory, seam tracking requires that the relation between arc voltage and current is known, such as by Eq. 1. Usually the voltage is held constant, while current changes due to the distance between wire and workpiece. In synergic welding however, both current and voltage are modified by the control system of the power-source, causing disruption in the control system of the algorithm. Early experiments with the Migatronic power-source, using the pure synergic mode showed that this mode disabled algorithm control in the negative approach direction of the TCP.
Initial seam tracking experiments using manual mode showed that the current constantly decreased throughout the weld, making compensation in negative direction impossible. A thorough examination of the current sensor showed that the current measurements were both stable and sufficiently accurate for the application and that the decrease of current throughout the weld was not due to measurement errors. The conclusion was therefore that the Migatronic power-source controller was most likely controlling the current also at manual mode. The reason is assumed to be that modern power-sources also include some adaptive control in manual mode to assist humans in performing high quality welds.

**Compensation for power-source control**

Addition of a constantly increasing offset to the nominal trajectory solved the problem caused by the adaptive behavior of the power-source. The solution had similarities with the vector booster method, but the modification of the trajectory increased constantly in negative direction of the approach axis.

To be able to handle negative deviations, \( K_d \) had to be doubled. The method was tested for fillet joints and showed to be a reliable and permanent solution to this problem. Since the same principal is valid for fillet as V-groove welds, no experimental series were considered necessary to prove the validity of the power-source compensation for V-groove welds.

**Power-source parameter settings**

The following data was primarily acquired and logged during the second experimental series consisting of 80 fillet and V-groove experiments: (1) objective, (2) label, (3) \( K_a, K_n \), (4) deviations in \( y \) and \( z \) directions, (5) welding speed, (6) weaving frequency and amplitude, (7) weaving shape, see Fig. 20, (8) nominal voltage and current, (9) wire speed, (10) offset magnitude calculated by the vector booster in \( x, y \) and \( z \) directions, (11) a detailed description of the results. The optimal parameters that were experimentally found, giving high welding quality, are presented in Table 1.

**Review of fillet welds**

Example of typical fillet and V-groove samples are displayed in Figs. 18 and 19. Information from these initial experiments was used for the modification of the algorithm, followed by a new series of simulations. Some selected fillet and V-groove experiments are presented by photos in Figures 21 and 24. These experiments are further commented and reviewed below.

1013. Deviation in \( z \) direction by 8%, followed by multipass welding.
Sine
Sinesquare
Square

Sawtooth
Triangle
Ellipse

Figure 20 The implemented weaving shapes in the algorithm. Sinesquare was chosen for fillet and sawtooth for V-groove welds for the achievement of maximum welding quality after a series of experiments and consultation with welding expertise. The sinesquare consists of a sine wave with an amplitude of 2 units, truncated at 1 unit.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Fillet weld</th>
<th>V-groove joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal voltage (V)</td>
<td>28</td>
<td>28.5</td>
</tr>
<tr>
<td>$K_a$</td>
<td>0.015</td>
<td>0.012</td>
</tr>
<tr>
<td>$K_n$</td>
<td>50% of $K_a$</td>
<td>50% of $K_a$</td>
</tr>
<tr>
<td>Weaving frequency (Hz)</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Weaving shape</td>
<td>Sinesquare</td>
<td>Sawtooth</td>
</tr>
<tr>
<td>Welding speed (mm/s)</td>
<td>8</td>
<td>3.5</td>
</tr>
<tr>
<td>Wire speed (m/min)</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Weaving amplitude (mm)</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1 Recommended parameters using the Migatronic power-source, derived by experiments.
Figure 21 Pictures of some selected fillet welds.
Since $K_a$ and $f$ were too small, seam tracking failed by a deviation of 2.3%. $K_a = 0.05$, $f = 1.5$ Hz.

1026. Deviation in $y$ direction by 4%. $K_n$ was too high and caused oscillations in $n$ direction. $K_a = 0.05$, $f = 3$ Hz.

1027. Reference welding without seam tracking, nearly optimal with $f = 3$ Hz. The distinct edges of the weavings indicate too high nominal voltage, cutting too deep into the workpiece.

1032. Deviations in $y$ and $z$ directions by 8%. High welding quality, but this is the limit for positive deviations. $K_a = 0.025$, $f = 3$ Hz.

1057. Deviations in both $y$ and $z$ directions by 2% (specifications). High welding quality in both root and multipass layers. $K_a = 0.0125$, $f = 3$ Hz.

1062. Final test. Deviation in $y$ direction by 2% (specifications), using power-source compensation. High welding quality. Some instability occurred at the beginning of the seam and the stick-out was 3 mm too large at the end of the seam. $K_a = 0.015$, $f = 3$ Hz.

1063, 1065-1066 Final tests. Deviations of -2% for 1063 and 2% for 1065-1066 in $y$ and $z$ directions (specifications), using power-source compensation. High welding quality. $K_a = 0.015$, $f = 3$ Hz. Multipass welding in 1066. The stick-out was 3 mm too large throughout the seam for 1063 and 3 mm too large at the end of 1065-1066. Due to interference with the root layer at multipass welding, the last of the three layers deviated by 2 mm at the end of the seam.

Additional information and summary of the selected fillet experiments in Fig. 21 follows below:

1. High weaving frequency gives fast control response. When the weaving frequency was increased from 1.5 Hz in 1013 to 3 Hz in 1032, the algorithm performance (compensation ability) was doubled.

2. The experiments proved that $K_n$ should be 50% of $K_a$ or less for stable control, which was previously found by simulation.

3. In the presented fillet experiments, $K_n$ was 50% of $K_a$ in all cases except in 1013 where it was 75%. Due to problems in the newly developed robot system (regarding tuning of the motor control unit) the highest weaving frequency that could be used was 2-3 Hz. At 3 Hz, the weaving amplitude was not able to exceed 3 mm despite the setting of 5 mm.

4. The fluctuation that occurred in the beginning of the seam in 1062 was due to the applied automatic calibration method. It was however found that the same reference current value could be used in the algorithm for all fillet welds, eliminating the fluctuations at start.

5. The stick-out (of the wire) was in some experiments 3 mm larger than desired, according to the welding specialist. This did not effect welding
quality in the root layer, but effected the multi-pass layers, since subsequent layers interfered with the root layer. This interference was not larger than about 2 mm, but for high welding quality at multipass welding the algorithm should be fine-tuned to be able to reduce the stick-out some millimeters. One way to achieve this is to slightly increase the reference current value. The power-source compensation should also be amplified to avoid collision of the torch with the workpiece.

6. The most important result from this experimental series was that power-source compensation showed to work properly.

![Figure 22](image.png)

**Figure 22** Current samples from the first 6 seconds of a fillet weld experiment, using power-source compensation. The unit of the horizontal axis is ticks, which equals 1/50 seconds. The current values are filtered by an active 4th order Bessel low-pass filter with $f_0 = 10$ Hz, added to the current sensor device, delivered from Migatronic. Though some transient spikes occurred in the initial part of the welding, the remaining data had relatively low level of noise. The lower figure is a magnification of a selected area of the upper.
Figure 23 Average current samples throughout the weld, using power-source compensation. The unit of the horizontal axis is setpoints, which in short welds is equal to one weaving cycle, but in longer is an integer multiple of weaving cycle. The upper figure displays $I_a$, which is the average current over each setpoint. The compensation in the TCP approach axis is made by multiplying this factor by the gain in a direction, $K_a$. In the middle and lower figures average currents $I_{n1}$ and $I_{n2}$ in the surroundings of the turning points of the weaving are displayed. The compensation in $n$ direction is basically determined by $K_n$ multiplied with the difference of $I_{n1}$ and $I_{n2}$.
Review of V-groove welds

Below follows comments and a review of the selected V-groove welding experiments presented in Fig. 24.

2031. Deviation in $y$ direction by -2%. High welding quality. $K_a = 0.015, f = 2$ Hz.

2033. Deviations in $y$ and $z$ directions by 2%. Seam tracking failed due to small compensation caused by the average current estimator. A 10 mm too large stick-out caused bubbles in the weld and a poor welding quality. $K_a = 0.020, f = 2$ Hz.

2041. Deviations in $y$ and $z$ directions by 2%. High welding quality. Anti-transient current limiters were added to the algorithm and a 230 A threshold was added to the reference current estimator. $K_a = 0.012, f = 2$ Hz, $U = 29$ V.

2043. Final test. Deviations in $y$ and $z$ directions by 2%. High welding quality. Basically the same experiment as above, except for $U = 28.5$ V, for enhanced welding quality. $K_a = 0.012, f = 2$ Hz.

2047. Final test. Deviation in $y$ direction by 2% with two multipass layers on top of the root layer. High welding quality. $K_a = 0.015, f = 2$ Hz.

Figure 24 Pictures of some selected V-groove welds.
4 Conclusions

The ROWER-2 specifications were fulfilled by the development and physical validation of a seam tracking algorithm distinguished by:

- Stable control in $a$ and $n$ directions for fillet and V-groove joints (negative $a$ direction for V-grooves based on fillet weld experiments).
- Ability to perform multipass welding using interpolation.
- Maintaining high welding quality throughout the seam by good power-source parameter settings and the use of power-source compensation.
- Design and use of an auto-phase analyzer, calculating the total delay in the robot system for accurate control in $n$ direction (A new version of the FFT algorithm was designed and implemented. This special version worked faster and was simpler to implement for this specific task than standard FFT).
- Design and implementation of the vector booster method.
- Automatic reference current calibration.

5 Acknowledgements

The ROWER-2 project was funded by European Commission under the BRITE/EURAM program. Thanks also to the project teams at Tecnomare (Venice), IAI (Madrid) and Fincantieri (Venice) for support in development of the ROWER-2 system and experimental studies.

References


Paper V

Design and validation of a universal 6D seam tracking system in robotic welding using laser scanning

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Submitted to the Journal of Industrial Robot, January 2003
Design and validation of a universal 6D seam tracking system in robotic welding using laser scanning

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Abstract

This paper presents the design and validation of a universal 6D seam tracking system that reduces the need of accurate robot trajectory programming and geometrical databases in robotic laser scanning. The system is able to follow any 3D spline seam in space with relatively small radius of curvature by the real-time correction of the position and orientation of the welding torch, using a laser scanner in robotic arc or laser welding. This method is further useful for the seam tracking of any seam with a predefined signature of the seam profile, also at non-welding applications, and is able to seam track objects with very small radius of curvature at pre-scanning, including theoretically also sharp edges.

The 6D seam tracking system was developed in the FUSE simulation environment, integrating software prototyping with mechanical virtual prototyping, based on physical experiments. The validation experiments showed that this system was both robust and reliable and should be able to manage a radius of curvature less than 200 mm. In the pre-scanning mode, a radius of curvature down to 2 mm was managed for pipe intersections at 3 scans/mm, using a laser scanner with an accuracy of 0.015 mm.

1 Introduction

Robot automation increases productivity, the product quality and frees man from involuntary, unhealthy and dangerous work. While computational power has increased exponentially during the last decades, the limitations in flexibility constitute today a bottleneck in the further evolution of robotic systems.
One application for intelligent systems is seam tracking in robotic welding. Seam tracking is among others used for the manufacturing of large ships such as cruisers and oil tankers, where relatively high tolerances in the sheet material are allowed to minimize the manufacturing costs [3, 12]. Seam tracking is today typically only performed in 2D, by a constant motion in \( x \) and compensation of the errors in \( y \) and \( z \) directions, see Fig. 1. There exist solutions for higher autonomy that include seam tracking in 3D where compensation is also performed in \( o \) direction or around an orientation axis. These limitations make however seam tracking only possible for workpieces with relatively simple geometrical shapes.

![Diagram of Tool Center Point (TCP) and the orthonormal coordinate system](image)

**Figure 1** Definition of Tool Center Point (TCP) and the orthonormal coordinate system \( \{n, o, a\} \). Here, \( o \) is opposite to the direction of welding and perpendicular to the plane \( \Omega \), and \( a \) is the direction of approach. \( \{x, y, z\} \) is a local coordinate system, defined for the workpiece.

The next step in the evolution of sensor systems in robotic welding is the introduction of a full 6D sensor guided control system for seam tracking that is able to correct the TCP in \( x, y, z \) and around roll, pitch and yaw. Such ultimate system is per definition able to follow any continuous 3D seam with moderate curvature. This system has many similarities with an airplane control system, where a change in either position or orientation effects all other degrees of freedom.

Though seam tracking has been performed in 2D for at least 40 years, the hypothetical design of 6D seam tracking systems was probably first addressed in the beginning of the last decade by suggestions of models based on force sensors [5, 6] and laser scanners [17, 18]. It is however difficult to evaluate these systems, since no experimental results are presented, neither any explicit control systems for creating balance between the subsystems.

The contribution of this paper is the invention of a universal 6D seam tracking system for robotic welding, validated by simulation experiments based on...
physical experiments, that proved that 6D seam tracking is possible and even very robust for laser scanning [13]. On a more detailed level, the contributions of this paper are considered to be the introduction of: (1) differential control in laser scanning, using the same techniques as used in arc sensing [7], (2) trajectory tangent vector control using differential vector interpolation, and (3) pitch control by vector interpolation. These components constitute the foundation of the 6D seam tracking system.

The authors have found only a few references from the literature that describe similar work, which is the introduction of a trajectory tangent vector by curve fitting [17, 18]. There exist however some essential differences between how such vector was used and implemented. The differences consist of (1) curve fitting by 2nd instead of 3rd degree polynomials, for faster computation and still high control stability, (2) using an analytic solver for curve fitting of 2nd degree polynomials developed and implemented for this system, increasing the calculation speed further and (3) using differential vector interpolation instead of direct use of the trajectory tangent vector, which showed to be essential for maintaining control stability.

A 6D seam tracking system increases the intelligence and flexibility in manufacturing systems based on robotic welding using laser scanning. This reduces the need for accurate robot trajectory programming and geometrical databases. The simulation results (based on physical experiments) showed that the initial objective to design a 6D seam tracking system was reached that could manage a radius of curvature down to 200 mm. Further on, if low-speed seam tracking is performed without welding, to find the geometry of the workpiece, there is basically no limit how small radius of curvature this system is able to manage.

2 Materials and methods

2.1 Sensors guided robot control

Many systems have been developed during the last decades for seam tracking at arc welding. The patents and scientific publications within this application area show that laser scanners and arc sensors today replace older systems. Laser scanners have in general high accuracy but are expensive and have to be mounted on the torch, thereby decreasing the workspace of the robot. This problem is partly solved by introduction of an additional motion that rotates the scanner around the torch to keep its relative orientation constant with respect to the seam profile. In addition, it should be mentioned that there exist solutions for digital image processing of the workpiece before welding. The problems are however, that small changes in the surrounding light may disturb analysis. Further on, 3D spline curves requires the camera to follow the seam all the way to be able to capture segments that otherwise would have remained
hidden. This applies also to very long seams. In practical applications today, solutions based on pure image processing by a CCD camera are successfully used to find the initial point where 2D seam tracking, based on laser scanning or arc sensing should start [12].

**Laser scanning**

In laser scanning, the seam profile is extracted by periodically scanning the laser beam across a plane perpendicular to the direction of the seam. The laser scanner is mounted on the torch in front of the direction of welding. As a general rule, the control system is based on full compensation of the position errors, which is equal to a constant value of $K_1 = 1$, as defined in Fig. 2.

![Figure 2 Cross section, plane $\Omega$. Vector interpolation used for position and pitch control at laser scanning. The input from the laser scanner system is the intersection point of the seam walls $v_0$ and the unit vector $v_1$ on $\Omega$ between the seam walls. $D$ denotes the distance between $v_0$ and $p'$.](image)

**Arc sensing**

Another method used for seam tracking is “through-arc” sensing or simply arc sensing [7]. Here, the arc current is sampled while a weaving motion is added to the TCP in $n$ direction, see Fig. 1. Since the distance from TCP to the nearest seam wall may be approximately calculated as a function of the current, it is possible to compensate for deviations in the seam by analysis of the current measurements. Arc sensors are inexpensive, do not decrease the workspace of the robot, but are compared with other sensors for seam tracking relatively inaccurate due to the stochastic nature of arc welding. Since welding of large structures in general require a relatively low accuracy however, arc sensors are a competitive alternative to laser scanners. Arc sensing may also be used as a complementary sensor to laser scanners in some cases where the laser scanner is unable to access the seam track.
Virtual sensors

The development of sensor guided control systems may be accelerated using virtual sensors in the simulation control loop. Since a system's state is often easily obtained in simulation, assuming that the simulation model is sufficiently accurate compared to the real system, this provides for a better foundation for process analysis than in general is possible by physical experiments. Further on, insertion of a sensor in a control loop may cause damage to expensive equipment, unless the behavior of the entire sensor guided control system is precisely known, which is rarely the case at the development stage.

Virtual sensors are used in many application areas, such as robotics, aerospace and marine technologies [4, 9, 16, 19, 21]. Development of new robot systems, such as for seam tracking may be accelerated by application of simulation [14, 15]. In general, the design methodology of virtual sensors may vary due to their specific characteristics. If a sensor model is not analytically definable, artificial neural networks or statistical methods may be used to find appropriate models [2, 20].

2.2 Robot system

The design of the 6D seam tracking system was based on experiments and results from the European ROWER-2 project [1]. The objective of this project was to automate the welding process in shipbuilding by the design of a robotic system, specifically for joining double-hull sections in super-cruisers and oil tankers. Figure 3 shows the robot system developed in the ROWER-2 project and the corresponding simulation model in FUSE (described below). The implementation of the 3D seam tracking system was performed in C/C++, running on a QNX-based embedded controller (QNX is a real-time operating system for embedded controllers). The robot was manufactured partly in aluminum alloy to decrease the weight for easy transportation and mounted on a mobile platform with 1 degree of freedom for increased workspace. The robot system was integrated and successfully validated on-site at a shipyard.

2.3 Simulation system

The development and implementation of the 6D seam tracking system was initially performed in the Flexible Unified Simulation Environment (FUSE) [11]. FUSE is an integrated software system based on Envision (robot simulator, Delmia Inc.) and Matlab (MathWorks Inc.), including Matlab Robot Toolbox [8]. The integration was performed by compiling these software applications and their libraries into one single software application, running under IRIX, SGI. The methodology developed for the design of the 6D seam tracking system was based on the integration of mechanical virtual prototyping and software
Figure 3 The design of the robot system for 3D seam tracking based on arc sensing in the ROWER-2 project (left) was based on simulations in FUSE (right). The 6D seam tracking system presented in this paper was developed based on experience and results gained from the ROWER-2 project.

protoyping. The 6D seam tracking simulations in FUSE were performed by using the robot developed in the ROWER-2 project, see Figs. 3, and a ABB S4 IRB2400.16 robot. After the design and successful validation of a 6D seam tracking system for arc sensing, the virtual prototyping and software models, partly written in C-code and partly in Matlab code, were translated to pure Matlab code. The Matlab models were modified to work also for laser scanners by the addition of a virtual model of the Servo-Robot M-SPOT laser scanner system. The translation was necessary, since it saved development time for the implementation of the evaluations system, for the accurate calculation of position and orientation errors during simulation.

Figure 4 Example of a 6D seam tracking experiment performed in Matlab. Each laser scan is plotted at each control cycle while seam tracking is performed 180° around a pipe intersection. The workpiece object corresponds to the object in the upper experiment in Fig. 7.
2.4 Calculation of position and orientation errors

Error is defined as the difference between desired and current pose $P$. The position errors were calculated by projection of $\epsilon_a$ and $\epsilon_n$ on a plane $\Omega_t$ perpendicular to the desired trajectory while intersecting current TCP, $P_t$. Thereby, per definition $\epsilon_o = 0$. The orientation error (the orientation part of $P_\delta$) is calculated by:

$$P_\delta = P_t^{-1} \cdot P_{\Omega_t}$$

where $P_t$ is the current TCP and $P_{\Omega_t}$ is the desired TCP for any sample point $t$. The errors around $n$, $o$ and $a$ were calculated by the subsequent conversion of the resulting transformation matrix to roll, pitch and yaw, here defined as rotation around, in turn, $a$, $o$ and $n$. Rotation in the positive direction is defined as counterclockwise rotation from the perspective of a viewer when the axis points towards the viewer.

3 Modeling

In this section the design of the 6D seam tracking system is described. The principal control scheme of this system is presented in Fig. 5. The control system is based on three main components, position, trajectory tangent vector and pitch control. Pitch denotes rotation around $o$. The main input of this control system is the $4 \times 4$ transformation matrix $P_{N-1}$, denoting current TCP position and orientation. The output is next TCP, $P_N$. $N$ is the number of samples used for orientation control. The input parameters are $N$, the proportional control constants $K_1$, $K_2$ and $K_3$, and the generic sensor input values $\xi = [v_0 \ v_1]$ and $\zeta = v_1$, where $v_0$ and $v_1$ are defined in Fig. 2. The proportional constants $K_1$, $K_2$ and $K_3$ denote the control response to errors in position, trajectory tangent vector and pitch control. $N$ is also used in trajectory tangent vector control and is a measure of the memory size of the algorithm. The seam profile presented in Fig. 1 is a fillet joint. Since $v_0$ and $v_1$ may be extracted for any joint with a predefined signature, no other joints were required in the experiments.

3.1 Position control

The 6D seam tracking algorithm starts by the calculation of next position for the TCP. Next position $p_N$ is calculated by vector interpolation based on $K_1$ in Fig. 2, between current position $p_{N-1}$ and the position $p'$ extracted from the laser scanner.
3.2 Trajectory tangent vector control

The positions $p_1...p_N$ are used for trajectory estimation by curve fitting, see Fig. 6. To decrease calculation time, an analytical estimator was devised for 1st and 2nd degree polynomials that showed to work much faster than traditional matrix multiplication followed by inversion techniques using Gauss elimination. The estimation of the trajectory at $N$ gives $o'$. Since an attempt to abruptly change the direction of $o$ from $o_{N-1}$ to $o'$ would most probably create instability in the system, the motion between $o_{N-1}$ and $o'$ is balanced by vector interpolation using $K_2$. The scalar $\kappa_2$ is not explicitly used in the calculations, but used for visualization of the calculation of $o_N$ in Fig. 6.

3.3 Pitch control

The first step in pitch control is performed by vector interpolation between $a_{N-1}$ and $a'$, using $K_3$, see Fig. 2. In a second step, $a_N$ is calculated by orthogonally projecting $a''$ on a plane perpendicular to $o_N$ and normalization of the resulting vector. Finally, $n_N$ is calculated by the cross product $o_N \times a_N$.  

Figure 5 The principal scheme of the 6D seam tracking control system. $P_{N-1}$ and $P_N$ denote current and next TCP. $P_N$ is calculated for each control sample designated for the seam tracking motion.
Figure 6 Trajectory tangent vector control is performed by the least square curve fitting of the last N trajectory points to a 2nd degree polynomial curve for x, y and z, followed by vector interpolation. Here N = 5. X, c0, c1 and c2 denote vector entities. The trajectory tangent vector F is for a 2nd degree polynomial equal to c1 + 2Nc2.

3.4 Code and pseudo code samples

The actual implementation of the 6D seam tracking system showed to be straightforward. Most of the development effort was spent on the design of the surrounding environment of the seam tracking system, including a semi-automatic testing and evaluation system. In simulation, the plane Ω is assumed to intersect the TCP point. In reality however, the laser scanner is mounted on the torch, often several centimeters ahead in the direction of motion. The transformation to such system at real-time control is performed by storing the trajectory data for use with a time delay. A pseudo code representation of the 6D seam tracking system in Fig. 5, follows below:

1. $p_N = p_{N-1} - r_w \cdot o_{N-1} + K_1 \cdot (v_0 + D \cdot v_1 - p_{N-1})$, where $v_0, v_1$ and $D$ are defined in Fig. 2 and $r_w$ is the nominal displacement in the direction of welding during a control sample.

2. Calculation of $x_N$ by shifting $x_{N-1}$ one step to the left with $p_N$: $x_{N-1} = \begin{bmatrix} p_0 & p_1 & \ldots & p_{N-2} & p_{N-1} \end{bmatrix}$, $x_N = \begin{bmatrix} p_1 & p_2 & \ldots & p_{N-1} & p_N \end{bmatrix}$.

3. Estimation of the trajectory tangent vector $F$ by least square curve fitting of the parameterized 3D curve $X = C \cdot T$ with a parameter $t$ to three independent second-degree polynomials (for x, y and z). $F = \frac{\partial X}{\partial t} \bigg|_{t=N}$ with $X = C \cdot T$, where $C = \begin{bmatrix} c_0 & c_1 & c_2 \end{bmatrix}$ and $T = \begin{bmatrix} 1 & t & t^2 \end{bmatrix}^T$ (Note that $A^T$ is the transpose function for a vector or matrix A and has no correlation with the vector T). This gives $F = C \cdot \frac{\partial T}{\partial t} \bigg|_{t=N} = C \cdot \begin{bmatrix} 0 & 1 & 2t \end{bmatrix} \bigg|_{t=N} = C \cdot \begin{bmatrix} 0 & 1 & 2N \end{bmatrix}$. $C$ is calculated by the estimation of the parameters $c_i$ for $i = \{x, y, z\}$, where $c_i = (A^T A)^{-1} (A^T p_i)$ and $p_i$ are the row vectors of $x_N$, which may also be written as $x_N = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$. 
The matrix $A$ consists of $N$ rows of $[1 \ t \ t^2]$ with $t = 1 \ldots N$, where $t$ is the row number. It should be mentioned that $C$ may be directly calculated by the expression $C = ((A^T A)^{-1} (A^T x_N^T))^T$, but we have chosen to use the notation above for pedagogical reasons.

4. $o_N = \text{normalize}(o_{N-1} - K_2 \cdot (\text{normalize}(F) + o_{N-1}))$. Normalization of a vector $v$ denotes $v$ divided by its length.

5. $a'' = a_{N-1} - K_3 \cdot (a_{N-1} + v_1)$.

6. $a_N = \text{normalize}(a'' - \text{dot}(o_N, a'') \cdot o_N)$, where $\text{dot}(o_N, a'')$ is the dot product $o_N^T \cdot a''$.

7. $P_N = \begin{bmatrix} o_N \times a_N & o_N & a_N & p_N \end{bmatrix}$

The initial value of $x$ is determined by linear extrapolation with respect to the estimated direction at the beginning of the seam. The elements of $F(i)$ with $i = \{1, 2, 3\}$ for $\{x, y, z\}$ were here calculated by the Matlab function $\text{pfit}((1:N)', x(i,:)'$) (the input parameters $x$ and $y$ in this function are $[1 \ 2 \ldots N]^T$ and $b_i$ as defined above). The pfit function was developed in this work using Maple (Waterloo Maple Inc.) and gives an analytical solution to 2nd degree polynomial curve fitting for this application, that works 8 times faster for $N = 10$ and 4 times faster for $N = 100$ than the standard Matlab function $\text{polyfit}$. The analytical function pfit was implemented according to below:

```matlab
function k = pfit(x,y)

n = length(x); sx = sum(x); sy = sum(y); sxx = x'*x; sxy = x'*y; sx3 = sum(x.^3); sx4 = sum(x.^4); sxx = sum(x.^2.*y);
t2 = sx*sx; t7 = sx3*sx3; t9 = sx3*sxx; t12 = sxx*sxx;
den = 1/(sx4*sxx*n-sx4*t2-t7*n+2.0*t9*sx-t12*sxx);
t21 = (sx3*n-sxx*sx)*t15;
k = 2.0*n*(sx3*sxx-t2)*t15*sxx-t21*sxy+(sx3*sxx-t12)*t15*sy-...
t21*sxx+(sx4*n-t12)*t15*sxy+(t9-sx4*sxx)*t15*sy;
```

The calculation speed was optimization by the application of Cramer’s rule by an analytical inversion of $A^T A$ in Maple and the elimination of $c_0$. In the function above, $x$ and $y$ are $N \times 1$ column vectors, $x^\prime$ equals $x^T$, sum($x$) is the sum of the vector elements of $x$, and a point before an operator such as the exponential operator $^\prime$, denotes elementwise operation. This implementation may be further optimized by reusing terms that only depend on $x$ in pfit (which is equal to $t$ outside pfit). Since such optimization is however outside the scope of this paper, we have chosen to keep this presentation as simple as possible.
3.5 Computational costs

The computational costs for the 6D seam tracking system based on laser scanning (without consideration to memory operations or index incrementation) was preliminary estimated to less than $250 + 50N$ floating point operations for the implementation presented in this paper, which for $N = 10$ gives 750 operations per control cycle. At a rate of maximum 30 scans per second, this gives less than 22,500 floating point operations per second.

3.6 Kinematics singularities

In sensor guided control, the manipulator may involuntarily move into areas in which no inverse kinematics solutions exist, generally called kinematics singularities. It is however possible to minimize inner singularity problems caused by robotic wrists. A novel method is suggested in [10] which was designed for the 6D seam tracking system, but works basically for any industrial robot with 6 degrees of freedom. In this method, the stronger motors, often at the base of the robot, assist the weaker wrist motors to compensate for any position error. This method allows also for smooth transitions between different configurations.

In addition to this, in one-off production or for generation of a reference program, pre-scanning and trial run may be performed before welding. In the first step, low-speed (or high-speed) seam tracking is performed to calculate the trajectory of the TCP, following the seam. In the second step, a trial run is performed in full speed to check the performance of the robot, and finally the actual welding is performed.

3.7 Initial and final conditions

In the experiments, seam tracking was initiated from a proper position already from start. The start position for seam tracking is in general found either by image analysis using a camera or by seam tracking itself. Since previous positions are unknown at start, an estimation is made of previous $N - 1$ positions, by extrapolation of the trajectory in the direction of $o$. This is the reason why in some experiments, higher orientation errors occurred at the beginning of seam tracking. The condition to stop seam tracking may be based on the position of the TCP. The stop signal may be triggered by definition of volumes outside which no seam tracking is allowed.
4 Experimental results

4.1 Workpiece samples

The position and orientation errors at seam tracking were calculated based on simulation experiments of basically 8 different workpieces, see Figs. 7-11. It should be mentioned that the experiments in the bottom of Fig. 8 and Fig. 9 are not actually possible in practice due to collision problems, but are performed for the analysis of the control system. The workpieces in Matlab were approximated by lines orthogonal to the direction of the weld and the resolution was defined as the nominal distance between these lines. The resolution was here set to 0.02 mm. The reason why no noise was added to the profile obtained from the laser scanner (except for perhaps decreasing the number of measurement points compared to the M-SPOT, which may use up to 512 points per scan), was because large number of measurements allow for high-precision estimations of $v_0$ and $v_1$, with a constant offset error less than the accuracy of the laser scanner. Since the 6D seam tracking system corrects the position and the orientation of the TCP relative to the extracted profile, such error will not influence the behavior of the control system more than highly marginally, by basically adding a small offset error to the experimental results in Figs. 7-11.

![Figure 7](image)

**Figure 7** Seam tracking, laser scanning. Pipe intersection (top) and workpiece for isolating control around a (bottom). $K_1 = 1.0, K_2 = K_3 = 0.8, N = 10$. Error denotes the difference between desired and current pose.

The workpieces are categorized into two groups, continuous and discrete. In
Figure 8 Seam tracking, laser scanning. Isolating rotation around $n$, following inner and outer curves. Note that in this figure, the case in the bottom is not very realistic, but only included for the completeness of the experiments. $K_1 = 1.0, K_2 = K_3 = 0.8, N = 10$.

Figure 9 Seam tracking, laser scanning. The same as in previous figure, except for $K_2 = 0.1$ and $r = 50$ mm. This shows that $K_2 = 0.1$ works fine as well for continuous seams with moderate radius of curvature. Further, the same experiment was also performed with $r = 20$ mm, which showed to be basically identical to this figure, but with saturation at 12 degrees instead of 4. Note that while other laser scanning experiments were performed on the verge of instability to find the limits of the seam tracking system, using the M-SPOT laser scanner, the system is highly stable at moderate curvatures using low values for $K_2$. 
Figure 10 Seam tracking, laser scanning. The step is introduced at $x = 0$ mm. $K_1 = 1.0, K_2 = 0.1, K_3 = 0.8, N = 10$.

Figure 11 Seam tracking, laser scanning. In this figure, the case in the bottom denotes a step of $-30^\circ$ performed around $o$. $K_1 = 1.0, K_2 = 0.1, K_3 = 0.8, N = 10$. 
the experiments presented in this paper, the range was $0 - 180^\circ$ for continuous, and 12 mm for discrete workpieces. In seam tracking of a continuous workpiece, a sudden change is interpreted as a curve with a small radius of curvature, see Figs. 7-8, while in seam tracking of a discrete workpiece a sudden change is regarded as a disturbance, see Figs. 10-11. The settings of the control parameters decide alone the mode of the control system: continuous or discrete. For seam tracking of objects with a large radius of curvature, the discrete mode works well for both continuous and discrete workpieces, see Fig. 9.

The control parameters $K_1, K_2, K_3$ and $N$ used in the presented experiments showed to work well for each category, continuous and discrete. These settings were found by more than 50 laser scanning simulation experiments in Matlab, and large deviation from these parameters showed to create instability in the system, making the control components working against each other instead of in synergy, resulting in divergence from the seam and ultimately failure at seam tracking.

The seam tracking system is theoretically able to handle any angle $\alpha$ (in Fig. 2) between 0 and $180^\circ$. In practice however, a small angle may cause collision between torch and workpiece. For simplicity, $\alpha$ was set to $90^\circ$ for all experiments except for pipe intersections.

4.2 Laser scanning

The reference laser scanner, M-SPOT, used in the experiments has a sweeping angle of $28^\circ$, an accuracy better than 0.02 mm at the distance of 100 mm and 1.5 mm at a distance of 1 m. It is able to scan at a frequency of up to 40 Hz with a resolution of maximum 512 points per scan. The virtual model of the M-SPOT that was designed, were programmed to have an accuracy of 0.02 mm and was mounted on the torch at a distance of approximately 80 mm from the workpiece. The resolution was set to 11 points per scan for a sweeping frequency of $5^\circ$ (the only exception was steps in the $n$ direction, see Fig. 11, where 31 points per scan with a sweeping angle of $10^\circ$ was used instead) and the scanning frequency to 30 Hz, which showed to be fully sufficient for successful seam tracking.

In fact, the simulation experiments proved that the accuracy of M-SPOT was higher than needed for seam tracking. The accuracy is so high in close range (100 mm) that only robots with very high performance are able to use the accuracy of this sensor to its full extent, when seam tracking is performed in real-time. The experiments presented in Figs. 7-8 and 10-11 show the limits of performance for the 6D seam tracking system, when it is at the verge of instability. For larger radius of curvature and a lower value for $K_2$, i.e. for normal operation, Fig. 9 gives perhaps a better indication of the efficiency of the 6D seam tracking system. Since the performance of a robot is dependent on cur-
rent joint values, it is further not possible to translate these values directly to joint positions, velocities and accelerations. What these experiments proved however, was that it was the performance of the robot that constituted the limitations in seam tracking based on laser scanning.

The laser scanning simulations presented in this paper were performed with a constant welding speed of 10 mm/s (which equals 3 scans/mm at a scanning frequency of 30 Hz) and a virtual TCP, 2 mm away from the intersection point of the seam walls. Empirical experiments showed that by moving the TCP from previously \( D = 15 \) mm in Fig. 2 to only 2 mm, the control ability of the 6D seam tracking was highly improved, especially in seam tracking of objects with a small radius of curvature. Though the parameter settings \( K_1 = K_2 = K_3 = 1 \) showed to be unstable in laser scanning, \( K_1 = 1, K_2 = K_3 = 0.8 \) showed to be relatively stable for continuous welds. For discrete workpieces, \( K_2 \) was set as low as 0.1 to maximize control stability.

5 Results and discussion

A 6D seam tracking system was designed and validated by simulation that is able to follow any continuous 3D seam with moderate curvature, using a laser scanner, and is able to continuously correct both position and orientation of the tool-tip of the welding torch along the seam. Thereby the need for robot trajectory programming or geometrical databases is practically eliminated, except for start and end positions.

Simulations based on the M-SPOT laser scanner showed that 6D seam tracking was theoretically possible of objects less than a radius of curvatures of 2 mm. Seam tracking in real-time requires however a robot with very high performance. It is however possible to perform seam tracking without welding to find the geometry of the workpiece, and perhaps even checking that the trajectory is within the workspace of the robot, before actual welding is performed. Thereby it will be possible to perform welding of objects with small radius of curvature, theoretically also including sharp edges. This is much better than the initial objective to design a 6D seam tracking system that could manage a radius of curvature down to 200 mm, which is still considered as relatively small for robotic welding in for instance shipbuilding applications.

6 Future work

Though the 6D seam tracking system showed to be very robust, optimization of the algorithm is still possible. The control system worked well using proportional constants alone, but if needed, for instance PID or adaptive control may be included to enhance the performance. As an example, the control
parameter $K_2$ could be defined as a function of the “radius of curvature” (or more specifically as a function of for example $c_2$ in Fig. 6) of the 2nd degree polynomial used for calculation of the trajectory tangent vector, thereby enable the system to automatically change between normal operation, used for maximum control stability, and extraordinary operation with fast response, used for management of small radius of curvatures.

References


Paper VI

Design and validation of a universal 6D seam tracking system in robotic welding using arc sensing

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Submitted to the Journal of Advanced Robotics, Januari 2003
Design and validation of a universal 6D seam tracking system in robotic welding using arc sensing

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Abstract

This paper presents the design and validation of a novel and universal 6D seam tracking system that reduces the need of accurate robot trajectory programming and geometrical databases in robotic arc welding. Such sensor driven motion control together with adaptive control of the welding process is the foundation for increased flexibility and autonomous behavior of robotic and manufacturing systems. The system is able to follow any 3D-spline seam in space with moderate radius of curvature by the real-time correction of the position and orientation of the welding torch, using the through-arc sensing method.

The 6D seam tracking system was developed in the FUSE simulation environment, integrating software prototyping with mechanical virtual prototyping, based on physical experiments. The validation experiments showed that this system was both robust and reliable and is able to manage a radius of curvature less than 200 mm.

1 Introduction

Robot automation increases productivity, the product quality and frees man from involuntary, unhealthy and dangerous work. While computational power has increased exponentially during the last decades, since many new application areas require operation in relatively unstructured environments, the limitations in flexibility constitute today a bottleneck in the further evolution of robotic systems.
One application for intelligent systems is seam tracking in robotic welding. Seam tracking is among others used for the manufacturing of large ships such as cruisers and oil tankers, where relatively high tolerances in the sheet material are allowed to minimize the manufacturing costs [3, 12]. Seam tracking is today typically only performed in 2D, by a constant motion in \( x \) and compensation of the errors in \( y \) and \( z \) directions, see Fig. 1. There exist solutions for higher autonomy that include seam tracking in 3D where compensation is also performed in \( o \) direction or around an orientation axis. These limitations make however seam tracking only possible for workpieces with relatively simple geometrical shapes.

![Figure 1](Image)

**Figure 1** Left: Definition of Tool Center Point (TCP) and the orthonormal coordinate system \([n, o, a]\). Weaving, if any, is performed in \( n \) direction in arc sensing, \( o \) is opposite to the direction of welding and perpendicular to the plane \( \Omega \), and \( a \) is the direction of approach. \([x, y, z]\) is a local coordinate system, defined for the workpiece. Right: The optimal position for seam tracking in arc sensing.

The next step in the evolution of sensor systems in robotic welding is the introduction of a full 6D sensor guided control system for seam tracking that is able to correct the TCP in \( x \), \( y \), \( z \) and around roll, pitch and yaw. Such ultimate system is per definition able to follow any continuous 3D seam with moderate curvature. This system has many similarities with an airplane control system, where a change in either position or orientation effects all other degrees of freedom.

Though seam tracking has been performed in 2D for at least 40 years, the hypothetical design of 6D seam tracking systems was probably first addressed in the beginning of the last decade by suggestions of models based on force sensors [5, 6] and laser scanners [17, 18]. It is however difficult to evaluate these systems, since no experimental results are presented, neither any explicit control systems for creating balance between the subsystems.

The major contribution of this paper is the invention of a universal 6D seam
A 6D seam tracking system increases the intelligence and flexibility in manufacturing systems based on robotic welding using laser scanning. This reduces the need for accurate robot trajectory programming and geometrical databases. The simulation results (based on physical experiments) showed that the initial objective to design a 6D seam tracking system was reached that could manage a radius of curvature down to 200 mm.

2 Materials and methods

2.1 Sensors guided robot control

Many systems have been developed during the last decades for seam tracking at arc welding. The patents and scientific publications within this application area show that laser scanners and arc sensors today replace older systems. Laser scanners have in general high accuracy but are expensive and have to be mounted on the torch, thereby decreasing the workspace of the robot. This problem is partly solved by introduction of an additional motion that rotates the scanner around the torch to keep its relative orientation constant with respect to the seam profile. Arc sensors are inexpensive and compared with other sensors for seam tracking, relatively inaccurate due to the stochastic nature of arc welding. Since welding of large structures in general requires a relatively low accuracy however, arc sensors are a competitive alternative to laser scanners. In addition, it should be mentioned that there exist solutions for digital image processing of the workpiece before welding. The problems are however, that small changes in the surrounding light may disturb analysis.
Further on, 3D spline curve requires the camera to follow the seam all the way to be able to capture segments that otherwise would have remained hidden. This applies also to very long seams. In practical applications today, solutions based on pure image processing by a CCD camera are successfully used to find the initial point where 2D seam tracking, based on laser scanning or arc sensing should start [12].

**Figure 2** Arc sensing. Left: Cross section, perpendicular to the approach vector a. The weaving is added to the interpolated motion between \( P_{N-2} \) and \( P_{N-1} \), including changes in position and orientation. Right: Cross section, plane \( \Omega \). By the measurement of the current from start to end, the profile of the workpiece is approximately determined.

**Arc sensing**

In “through-arc” sensing or simply arc sensing [7], the arc current is sampled while a weaving motion is added to the TCP in \( n \) direction, see Fig. 1. Weaving is useful in arc welding due to process related factors and was used already before the introduction of arc sensing in the beginning of the 1980s. In arc sensing, by keeping the voltage constant, it is possible to calculate the distance from TCP to the nearest seam wall at each current sample, and thereby during a weaving cycle obtain the profile of the seam. According to experimental results [7] the approximate relationship between arc voltage \( V \), arc current \( I \) and the nearest distance between the electrode and the workpiece \( l \) for an electrode extension ranging between 5-15 mm, may be expressed by the relation:

\[
V = \beta_1 I + \beta_2 + \beta_3/I + \beta_4 l
\]  

(1)
where the constants $\beta_1 - \beta_4$ are dependent on factors such as wire, shielding gas and power-source. In practice the voltage and current readings of the arc contain much noise, why the signal has to be low-pass filtered before it is used for feedback control. An example of such filtered signal, sampled in the ROWER-2 project is presented in Fig. 3.

**Figure 3** Example of a filtered arc signal at seam tracking, sampled in the ROWER-2 project. Though anomalies are common at the ignition of the arc, only minor deviations from this pattern showed to occur during continuous seam tracking. The unit of the horizontal axis is ticks, which equals 20 ms. The current values were filtered by an analogue active 4:th order Bessel low-pass filter with $f_0 = 10$ Hz before sampling.

In automatic control, error is defined as the difference between desired (input) value $u$ and current (output) value $y$. The control algorithms in arc sensing are based on methods such as template matching or differential control [7]. In template matching the errors may for instance be calculated by integrating the difference of the template $u$ and the arc signal $y$, where $A$ denotes the weaving amplitude:

$$E_a = \epsilon_a = \int_{-A}^{A} (u(n) - y(n))dn$$  \hspace{1cm} (2)

$$E_n = \epsilon_n = \int_{-A}^{0} (u(n) - y(n))dn - \int_{0}^{A} (u(n) - y(n))dn$$  \hspace{1cm} (3)

Another example includes using integrated difference squared errors. The equation above use $n$ as variable to simplify the notation. Actually, more specifically, the integration between $-A$ and $A$ is performed in negative $o$ direction, using an auxiliary variable $\psi$ denoting the state of weaving in $n$ direction, integrated over $[\psi_{-A}, \psi_A]$, where $\psi_{-A}$ and $\psi_A$ denote the turning points of weaving. In differential control, which has proven to be quite reliable for control in $a$ and $n$ directions, current sampling is performed at the turning points of the weaving. Here, the errors in $a$ and $n$ directions are obtained by:
\[ \epsilon_a = I_{ref} - \frac{i_{+A} + i_{-A}}{2} \]  \hspace{1cm} (4) \\
\[ E_a = K_a \epsilon_a \]  \hspace{1cm} (5) \\
\[ \epsilon_n = i_{+A} - i_{-A} \]  \hspace{1cm} (6) \\
\[ E_n = K_n \epsilon_n \]  \hspace{1cm} (7)

where \( I_{ref} \) is a reference current value and \( i_{+A} \) and \( i_{-A} \) are the average measured current values at a pair of adjacent extreme points. The parameters \( I_{ref}, K_a \) and \( K_n \) are dependent on the weld joint geometry and other process parameters such as shielding gas and wire feed rate. Since these parameters will be known in advance, \( K_a \) and \( K_n \) may be defined for any welding application.

An example of how \( E_a \) and \( E_n \) are used for position control follows below:

\[ p_N = p_{N-1} + E_n n - r_w o + E_a a \]  \hspace{1cm} (8)

where \( p_{N-1} \) and \( p_N \) denote current and next positions and \( r_w \) is nominal displacement during a weaving cycle, also in Fig. 1, which is equal to welding speed (without consideration to the weaving motion) divided by weaving frequency.

**Virtual sensors**

The development of sensor guided control systems may be accelerated using virtual sensors in the simulation control loop. Since a system’s state is often easily obtained in simulation, assuming that the simulation model is sufficiently accurate compared to the real system, this provides for a better foundation for process analysis than in general is possible by physical experiments. Further on, insertion of a sensor in a control loop may cause damage to expensive equipment, unless the behavior of the entire sensor guided control system is precisely known, which is rarely the case at the development stage.

Virtual sensors are used in many application areas, such as robotics, aerospace and marine technologies [4, 9, 16, 19, 21]. Development of new robot systems, such as for seam tracking may be accelerated by application of simulation [14, 15]. In general, the design methodology of virtual sensors may
vary due to their specific characteristics. If a sensor model is not analytically definable, artificial neural networks or statistical methods may be used to find appropriate models [2, 20].

2.2 Robot system

The design of the 6D seam tracking system was based on experience and results from the European ROWER-2 project [1]. The objective of this project was to automate the welding process in shipbuilding by the design of a robotic system, specifically for joining double-hull sections in super-cruisers and oil tankers. Figure 4 shows the robot system developed in the ROWER-2 project and the corresponding simulation model in FUSE (described below). The implementation of the 3D seam tracking system was performed in C/C++, running on a QNX-based embedded controller (QNX is a real-time operating system for embedded controllers). The robot was manufactured partly in aluminum alloy to decrease the weight for easy transportation and mounted on a mobile platform with 1 degree of freedom for increased workspace. The robot system was integrated and successfully validated on-site at a shipyard.

![Robot System](image)

Figure 4 The design of the robot system for 3D seam tracking based on arc sensing in the ROWER-2 project (left) was based on simulations in FUSE (right). The 6D seam tracking system presented in this paper was developed based on experience and results gained from the ROWER-2 project.

2.3 Simulation system

The development and implementation of the 6D seam tracking system was initially performed in the Flexible Unified Simulation Environment (FUSE) [11]. FUSE is an integrated software system based on Envision (robot simulator, Delmia Inc.) and Matlab (MathWorks Inc.), including Matlab Robot Toolbox [8].
The integration was performed by compiling these software applications and their libraries into one single software application, running under IRIX, SGI. The methodology developed for the design of the 6D seam tracking system was based on the integration of mechanical virtual prototyping and software prototyping. The 6D seam tracking simulations in FUSE were performed by using the robot developed in the ROWER-2 project and a ABB S4 IRB2400.16 robot, see Figs. 4 and 5. After the design and successful validation of the 6D seam tracking system for arc sensing, the virtual prototyping and software models, partly written in C-code and partly in Matlab code, were translated to pure Matlab code. The translation was necessary, since it saved development time for the implementation of the evaluation system, for the accurate calculation of position and orientation errors during simulation.

Figure 5 Examples of 6D seam tracking experiments performed in FUSE, Envision, using a model of the ROWER-2 robot.

2.4 Calculation of position and orientation errors

Error is defined as the difference between desired and current pose $P$. The position errors were calculated by projection of $\epsilon_a$ and $\epsilon_n$ on a plane $\Omega_t$ perpendicular to the desired trajectory while intersecting current TCP, $P_t$. Thereby, per definition $\epsilon_o = 0$. The orientation error (the orientation part of $P_0$) is calculated by:

$$P_5 = P_t^{-1} \cdot P_{\Omega_t} \tag{9}$$

where $P_t$ is the current TCP and $P_{\Omega_t}$ is the desired TCP for any sample point $t$. The errors around $n$, $o$ and $a$ are calculated by the subsequent conversion of the resulting transformation matrix to roll, pitch and yaw, here defined as rotation around, in turn, $a$, $o$ and $n$. Rotation in the positive were defined
as counterclockwise rotation from the perspective of a viewer when the axis points towards the viewer.

3 Modeling

In this section the design of the 6D seam tracking system is described. The principal control scheme of this system is presented in Fig. 6. The control system is based on three main components, position, trajectory tangent vector and pitch control. Pitch denotes rotation around $o$. The main input of this control system is the $4 \times 4$ transformation matrix $P_{N-1}$, denoting current TCP position and orientation. The output is next TCP, $P_N$. $N$ is the number of samples used for orientation control. The input parameters are $N$, the proportional control constants $K_1$, $K_2$ and $K_3$, and the generic sensor input values denoting $\xi = [\epsilon_a \epsilon_n]$ in Eqs. 10-11 and $\zeta = [k_1 k_2]$ in Figs. 2-7. The proportional constants $K_1$, $K_2$ and $K_3$ denote the control response to errors in position, trajectory tangent vector and pitch control. $N$ is also used in trajectory tangent vector control and is a measure of the memory size of the algorithm.

![Figure 6](image)

**Figure 6** The principal scheme of the 6D seam tracking control system. $P_{N-1}$ and $P_N$ denote current and next TCP. Next TCP is calculated at the beginning of each weaving cycle, and the motion between $P_{N-1}$ and $P_N$ is interpolated.

The seam profile presented in Fig. 1 is a fillet joint. Since all standard seam profiles are basically L-shaped, the generic parameters $\xi$ and $\zeta$ may be estimated by the same profile extraction methods used for fillets joints. An alternative to fillet joint is for instance the V-groove, which mainly contains a gap between the seam walls. In arc sensing, since electrical current always...
Figure 7 Cross section, plane $\Omega$, corresponding to the upper seam wall in Fig. 1. The measured distance from TCP to nearest wall is always perpendicular to the wall. The geometrical relation between the measured distance and the geometrical shape of the workpiece is calculated by the least-square estimate of the coefficients $k_1$ and $k_2$ in Fig. 2. Note that since both position and the orientation of TCP are interpolated between $P_{N-2}$ and $P_{N-1}$, this figure is only an approximation of the trajectory between $P_i$ and $P_j$. Note also that $\gamma < 0$ for $k > 0$.

takes the shortest path between TCP and the seam walls, see Fig. 2, such gap has in general small influence on the measurements in arc sensing. In addition, since using other shapes than sinus, such as sawtooth and square wave does not effect seam tracking as much as it effects the welding process in arc sensing, only sine waves were used in the simulation experiments.

3.1 Position control

The 6D seam tracking algorithm starts by the calculation of next position for the TCP. The position is calculated by the differential method in Eqs. 10-11, rewritten as a function of distance instead of current (with $d$ defined in Fig. 1), calculated by using Eq. 1:

$$
\epsilon_a = s(i_A) + s(i_{-A}) - 2d
$$

(10)

$$
\epsilon_n = s(i_{-A}) - s(i_A)
$$

(11)

Since the distance to the workpiece is approximately inverse proportional to the current, this may explain the main differences between Eqs. 4 and 6 and Eqs. 10-11. Also here, $E_a = K_a \epsilon_a$, $E_n = K_n \epsilon_n$ and $p_N$ is calculated by Eq. 8.
In the ROWER-2 project, simulation and physical experiments showed that $K_n$ should be 50% of $K_a$ for maximum control stability, both for Eqs. 4-6 and Eqs. 10-11. Here, $K_a = K_1$ and $K_n = K_1/2$. Since the position is controlled by keeping the distance from TCP to the nearest seam wall constant at the turning points of the weaving, the desired distance from TCP to the intersection point of the seam walls in Fig. 1 was derived as:

$$D = A + d\sqrt{1 + \tan^2(\alpha/2)} \over \tan(\alpha/2)$$

(12)

This equation is valid for any angle $\alpha$ different to an integer multiple of $\pi$, where $D$ is instead equal to $d$. This relation was primarily used for the calculation of the position errors in evaluation of the 6D arc sensing system.

### 3.2 Trajectory tangent vector control

The positions $p_1...p_N$ are used for trajectory estimation by curve fitting, see Fig. 8. To decrease calculation time, an analytical estimator was devised for 1st and 2nd degree polynomials that showed to work much faster than traditional matrix multiplication followed by inversion techniques using Gauss elimination. The estimation of the trajectory at $N$ gives $o'$. Since an attempt to abruptly change the direction of $o$ from $o_{N-1}$ to $o'$ would most probably create instability in the system, the motion between $o_{N-1}$ and $o'$ is balanced by vector interpolation using $K_2$. The scalar $K_2$ is not explicitly used in the calculations, but used for visualization of the calculation of $o_N$ in Fig. 8.

**Figure 8** Trajectory tangent vector control is performed by the least square curve fitting of the last $N$ trajectory points to a 2nd degree polynomial curve for $x$, $y$ and $z$, followed by vector interpolation. Here $N = 5$. $X$, $c_0$, $c_1$ and $c_2$ denote vector entities. The trajectory tangent vector $F$ is for a 2nd degree polynomial equal to $c_1 + 2Nc_2$. 
3.3 Pitch control

Pitch control is performed in two steps. First, \(a_{N-1}\) is orthogonalized with respect to \(o_N\). This step could also have been performed after the second step (which then becomes the first step). Empirical experiments showed however that the performance of the algorithm is slightly improved by pre-orthogonalization. In the second step the new coordinate system is rotated around \(o\) by \(\delta \phi\):

\[
\delta \phi = -K_3 \cdot (k_1 + k_2) \tag{13}
\]

Here, \(\delta \phi\) is in radians, and \(k_1\) and \(k_2\) are estimated by the least square method, based on sampling of the arc current during a half weaving cycle, see Figs. 2-7. The index \(i\) in these figures denote arc current samples. Since current always takes the shortest path between TCP and the nearest seam wall (homogeneous conducting surface) and the distance between the TCP and the seam walls may be calculated by Eq. 1, the full extraction of the seam profile is theoretically possible. With \(\sin \gamma = \partial s/\partial n\) and \(k = -\partial s/\partial n\) in Figs. 2-7, the following relation is deduced:

\[
k = -\sin \gamma \tag{14}
\]

The orientation of the walls projected on plane \(\Omega\) is calculated from the vector components of \(s_i\):

\[
s_i = |s_i|((\cos \gamma \cdot a_i + \sin \gamma \cdot n_i) \tag{15}
\]

where \(|s_i|\) is the distance calculated from the current measurement, using Eq. 1. Eq. 14 and \(\sin^2 \gamma + \cos^2 \gamma = 1\) give in turn the relation:

\[
\cos \gamma = \sqrt{1 - \sin^2 \gamma} = \sqrt{1 - k^2} \tag{16}
\]

which finally yields the direction of the current, projected on plane \(\Omega\):

\[
s_i = |s_i|((\sqrt{1 - k^2} \cdot a_i - k \cdot n_i) \tag{17}
\]

3.4 Arc sensing weaving interpolation

The interpolation between two poses such as \(P_{N-1}\) and \(P_N\) is usually performed by vector interpolation of position \(p\) and quaternion interpolation of the orientation \([n,o,a]\). Since most of the calculation time in arc sensing was spent on
quaternion interpolation, a new method was adopted that empirically showed to work much faster in Matlab and still performed as well as quaternion interpolation in 6D seam tracking. The denotations in this subsection, generically labeled as $x'$ and $x''$ are local and not related to other denotations elsewhere.

In this method, linear interpolation was first performed between $P_{N-1}$ and $P_N$ including $o$, $a$ and $p$ for any sample $i$, resulting in $o'_i$, $a''_i$ and $p_i$. $o'$ was rescaled to the unit vector $o_i$, and $a''_i$ was projected on the plane perpendicular to $o_i$ by:

$$a'_i = a''_i - (a''_i \cdot o_i) o_i$$  \hspace{1cm} (18)

The dot product $a''_i \cdot o_i$ gives the length of the vector $a''_i$ orthogonally projected on $o_i$. Finally, $a_i$ was calculated by rescaling $a'_i$ to the unit vector $a_i$, and $n_i$ was calculated by $o_i \times a_i$. The weaving motion was added as a second layer of motion to $P_i$ in $n_i$ direction. In robot simulation the torch was allowed to freely rotate $\pm 180^\circ$ around $a$ as a third layer of motion, which showed to be important, but not entirely imperative for the robot, to be able to perform $360^\circ$ welding of pipe intersection joints, see Fig. 5. The angle around $a$ was determined by the position of the robot with respect to the position of the workpiece.

### 3.5 Code and pseudo code samples

The actual implementation of the 6D seam tracking system showed to be straightforward. Most of the development effort was spent on the design of the surrounding environment of the seam tracking system, including a semi-automatic testing and evaluation system. A pseudo code representation of the 6D seam tracking system in Fig. 6, follows below:

1. $p_N = p_{N-1} - r_w o + K_1 \cdot (\frac{1}{2} e_n n + e_a a)$ based on Eqs. 5, 7-8, 10-11 and the relation $K_1 = 2K_n = K_a$.

2. Calculation of $x_N$ by shifting $x_{N-1}$ one step to the left with $p_N$:

$$x_{N-1} = [p_0 \ p_1 \ ... \ p_{N-2} \ p_{N-1}] - x_N = [p_1 \ p_2 \ ... \ p_{N-1} \ p_N].$$

3. Estimation of the trajectory tangent vector $F$ by least square curve fitting of the parameterized 3D curve $X = C \cdot T$ with a parameter $t$ to three independent second-degree polynomials (for $x$, $y$ and $z$). $F = \frac{\partial X}{\partial t}|_{t=N}$ with $X = C \cdot T$, where $C = [c_0 \ c_1 \ c_2] = [c_x \ c_y \ c_z]^T$ and $T = [1 \ t \ t^2]^T$. (Note that $A^T$ is the transpose function for a vector or matrix $A$ and has no correlation with the vector $T$). This gives $F = C \cdot \frac{\partial T}{\partial t}|_{t=N} = C \cdot [0 \ 1 \ 2t]^T|_{t=N} = C \cdot [0 \ 1 \ 2N]^T$. $C$ is calculated by the estimation of the parameters $c_i$ for $i = \{x, y, z\}$, where $c_i = (H^T H)^{-1}(H^T p_i)$ and $p_i$ are
the row vectors of \( x_N \), which may also be written as \( x_N = [p_x \ p_y \ p_z]^T \).

The matrix \( H \) consists of \( N \) rows of \([1 \ t \ t^2]\) with \( t = 1\ldots N \), where \( t \) is the row number. It should be mentioned that \( C \) may be directly calculated by the expression \( C = ((H^T H)^{-1}(H^T x_N^i))^T \), but we have chosen to use the notation above for pedagogical reasons.

4. \( o_N = \text{normalize}(o_{N-1} - K_2 \cdot (\text{normalize}(F) + o_{N-1})) \). Normalization of a vector \( v \) denotes \( v \) divided by its length.

5. \( a'' = \text{normalize}(a_{N-1} - \text{dot}(o_N, a_{N-1}) \cdot o_N) \), where \( \text{dot}(o_N, a_{N-1}) \) is the dot product \( o_N^T \cdot a_{N-1} \).

6. \( P' = \begin{bmatrix} n_{N-1} & o_N & a'' & p_N \\ 0 & 0 & 0 & 1 \end{bmatrix} \)

7. For the estimation of \( k_1 \) and \( k_2 \), see previous text and Fig. 2. It should however be mentioned that the \( x \)-components in linear fitting depend on the weaving function between \(-A\) and \( A \). For a sinusoidal wave, this gives that the \( x \)-components in the negative \( n \) direction are \( A \sin(2\pi i/M) \) with \( i \) running from the position \(-A\) to 0 for \( k_1 \), and from the position 0 to \( A \) for \( k_2 \). \( M \) is here number of samples per weaving cycle. The \( y \)-components in linear fitting are the corresponding measured distances (subsets of \( s \) in Fig. 2). Linear fitting may be performed by:

\[
\begin{align*}
    k &= \frac{(n^sx_{xy} - s_x^sy)}{(n^sx_{xy} - s_x^sy^2)}; \\
    \text{where } n, s_x, s_y, s_{xx}, s_{xy} \text{ are defined below in the function pfit.}
\end{align*}
\]

8. \( P'' = P' \cdot \text{roty}(-K_3(k_1 + k_2)) \), where \( \text{roty}(\theta) \), with \( \theta \) counted in radians, is a \( 4 \times 4 \) transformation matrix with elements \( A_{mn} \) (rows \( m \), columns \( n \)), such that all elements are zero except for \( A_{11} = A_{33} = \cos(\theta), A_{13} = -A_{31} = \sin(\theta) \) and \( A_{22} = A_{44} = 1 \). This gives:

\[
\begin{bmatrix} n'' & o_N & a'' & p_N \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

9. \( P_N = \begin{bmatrix} o_N \times a_N & o_N & a_N & p_N \\ 0 & 0 & 0 & 1 \end{bmatrix} \).

The initial value of \( x \) is determined by linear extrapolation with respect to the estimated direction at the beginning of the seam. The elements of \( F(i) \) with \( i = \{1, 2, 3\} \) for \( \{x, y, z\} \) were here calculated by the Matlab function \text{pfit}((1:N)',x(i,:))' (the input parameters \( x \) and \( y \) in this function are \([1 \ 2 \ldots N]^T\) and \( b_1 \) as defined above). The pfit function was developed in this work using Maple (Waterloo Maple Inc.) and gives an analytical solution to 2nd degree polynomial curve fitting for this application, that works 8 times faster for \( N = 10 \) and 4 times faster for \( N = 100 \) than the standard Matlab function polyfit. The analytical function pfit was implemented according to below:
function k = pfit(x,y)

n = length(x); sx = sum(x); sy = sum(y); sxx = x' * x; sxy = x' * y;
sx3 = sum(x .^ 3); sx4 = sum(x .^ 4); sxyy = sum(x .^ 2 .* y);
t2 = sx * sx; t7 = sx3 * sx3; t9 = sx3 * sxx; t12 = sxx * sxx;
den = 1/(sx4 * sxx * n - sx4 * t2 - t7 * n + 2.0 * t9 * sx - t12 * sxx);
t21 = (sx3 * n - sxx * sx) * t15;
k = 2.0 * n * ((sx3 * n * t2) * t15 * sxyy - t21 * sy + (sx3 * sxx - t12) * t15 * sy) - ...
t21 * sxyy + (sx4 * n * t12) * t15 * sxy + (t9 - sx4 * sx) * t15 * sy;

The calculation speed was optimization by the application of Cramer’s rule by an analytical inversion of $H^T H$ in Maple and the elimination of $c_0$. In the function above, $x$ and $y$ are $N \times 1$ column vectors, $x'$ equals $x^T$, sum($x$) is the sum of the vector elements of $x$, and a point before an operator such as the exponential operator `^`, denotes elementwise operation. This implementation may be further optimized by reusing terms that only depend on $x$ in pfit (which is equal to $t$ outside pfit). Since such optimization is however outside the scope of this paper, we have chosen to keep this presentation as simple as possible.

### 3.6 Kinematics singularities

In sensor guided control, the manipulator may involuntarily move into areas in which no inverse kinematics solutions exist, generally called kinematics singularities. It is however possible to minimize inner singularity problems caused by robotic wrists. A novel method is suggested in [10] which was designed for the 6D seam tracking system, but works basically for any industrial robot with 6 degrees of freedom. In this method, the stronger motors, often at the base of the robot, assist the weaker wrist motors to compensate for any position error. This method allows also for smooth transitions between different configurations.

### 3.7 Initial and final conditions

In the experiments, seam tracking was initiated from a proper position already from start. The start position for seam tracking is in general found either by image analysis using a camera or by seam tracking itself. Since previous positions are unknown at start, an estimation is made of previous $N - 1$ positions, by extrapolation of the trajectory in the direction of $o$. This is the reason why in some experiments, higher orientation errors occurred at the beginning of seam tracking. The condition to stop seam tracking may be based on the position of the TCP. The stop signal may be triggered by definition of volumes outside which no seam tracking is allowed.
4 Experimental results

4.1 Workpiece samples

The position and orientation errors at seam tracking were calculated based on simulation experiments of basically 8 different workpieces, see Figs. 9-12. It should be mentioned that the experiment in the bottom of Figs. 10 is not actually possible in practice due to collision problems, but was performed for the analysis of the control system. The workpieces in Matlab were approximated by lines orthogonal to the direction of the weld and the resolution was defined as the nominal distance between these lines. These resolutions were set to 0.5 mm.

\[ r = 50 \text{ mm} \]
\[ \theta \]
\[ R = 2r \]
\[ r = 25 \text{ mm} \]
\[ \theta = 0 \]
\[ a \]
\[ \cdot \cdot \cdot \]
\[ \cdot \cdot \cdot a \]

Figure 9 Seam tracking, arc sensing, with addition of random noise to the measured distance. Pipe intersection (top) and workpiece for isolating control around a (bottom). \( K_1 = 1.0, K_2 = 0.5, K_3 = 0.25, N = 10 \). Error denotes the difference between desired and current pose.

The workpieces are categorized into two groups, continuous and discrete. In the experiments presented in this paper, the range was \( 0 - 180^\circ \) for continuous, and 120 mm for discrete workpieces. In seam tracking of a continuous workpiece, a sudden change is interpreted as a curve with a small radius of curvature, see Figs. 9-10, while in seam tracking of a discrete workpiece a sudden change is regarded as a disturbance, see Figs. 11-12. The settings of the control parameters decide alone the mode of the control system: continuous or discrete. For seam tracking of objects with a large radius of curvature, the discrete mode works well for both continuous and discrete workpieces.
Figure 10 Seam tracking, arc sensing, with addition of random noise to the measured distance. Isolating rotation around \( n \), following inner and outer curves. Note that in this figure, the case in the bottom is not very realistic, but only included for the completeness of the experiments. \( K_1 = 1.0, K_2 = 0.5, K_3 = 0.25, N = 10 \).

Figure 11 Seam tracking, arc sensing, with addition of random noise to the measured distance. The step is introduced at \( x = 0 \) mm. \( K_1 = 0.5, K_2 = 0.2, K_3 = 0.1, N = 20 \).
The control parameters \( K_1, K_2, K_3 \) and \( N \) used in the presented experiments showed to work well for each category, continuous and discrete. These settings were found by more than 100 arc sensing simulation experiments in Matlab, and large deviation from these parameters showed to create instability in the system, making the control components working against each other instead of in synergy, resulting in divergence from the seam and ultimately failure at seam tracking.

As mentioned previously, the seam tracking system is theoretically able to handle any angle \( \alpha \) (in Figs. 1 and 2) between 0 and 180°. In practice however, a small angle may cause collision between torch and workpiece. For simplicity, \( \alpha \) was set to 90° for all experiments except for pipe intersections. Further, \( d \) in Fig. 1 was set to 1 mm.

### 4.2 Arc sensing

According to [22], the accuracy of an arc sensor in seam tracking is 0.5 mm. In simulation, the workpiece line resolution was set to 0.5 mm. To increase the errors in the system, a random signal with a range of 1 mm was added to the measured distance. In addition, no integration of the current measurement values was made near the turning points at weaving. On the contrary, one measurement was considered as enough for each turning point. The signal
was prefiltered by an active 4th order Bessel low-pass filter in the ROWER-2 project. In the simulation experiments presented in this paper no filtering was performed. Thereby the accuracy had been decreased to less than half of the accuracy suggested in [22]. In practical applications however, the arc signal should be filtered by a method that does not add any phase shift to the signal, such as a proper digital filter.

The arc sensing simulations presented in this paper were performed with a constant welding speed of 10 mm/s, weaving amplitude and frequency of 5 mm and 3 Hz, and 64 current samplings per weaving cycle (or 192 samplings per second). The algorithm worked well also for lower sampling rates, such as 50 samples per second, which was used in the ROWER-2 project. In general, higher sampling rates showed to improve the control ability of the system, for rates higher than perhaps 10 samples per weaving cycle, though rather slightly than proportionally. The simulation models proved to predict the estimations derived from physical experiments in the ROWER-2 project for 6D motion. According to those experiments, it was possible to perform differential seam tracking in \( y \) and \( z \) directions in Fig. 1, up to 20 mm for a seam of the length 1 m. At a speed of 9 mm per second, weaving amplitude and frequency of 3 mm and 3 Hz, this gives a redirection capability of \( 1.15^\circ \) per weaving cycle which in the worst case is equivalent to a minimum radius of 150 mm. Using a power-source without any intrinsic current control would further give up to 3 times better results according to the experiments in the ROWER-2 project (current control was only applied by the power source to limit the current). Using a high performing industrial robot and a weaving amplitude of 5 mm instead of 3, this would finally bring down the curvature to 25-50 mm, which actually was the amount found by simulation.

5 Results and discussion

A 6D seam tracking system was designed and validated by simulation that is able to follow any continuous 3D seam with moderate curvature, using an arc sensor, and is able to continuously correct both position and orientation of the tool-tip of the welding torch along the seam. Thereby the needs for robot trajectory programming or geometrical databases were eliminated for workpieces with moderate radius of curvatures, increasing intelligence in robotic arc welding.

Simulations based on physical experiments in the ROWER-2 project showed however that arc sensing in real-time welding was theoretically possible for a radius of curvature below 50 mm. This is much less than the initial objective to design a 6D seam tracking system that could manage a radius of curvature below 200 mm, which is still considered as small for instance in shipbuilding.
6 Future work

Though the 6D seam tracking system showed to be very robust, optimization of the algorithm is still possible. Presently, pitch control in arc sensing is based on data measurements during a half weaving cycle. By extending the measurements to one full cycle, pitch control may be optimized. Further on, though the control system worked well using proportional constants alone, if needed, for instance PID or adaptive control may be included to enhance the performance. One of many examples is for instance to define $K_2$ as a function of the “radius of curvature” (or more specifically as a function of for example $c_2$ in Fig. 8) of the 2nd degree polynomial used for calculation of the trajectory tangent vector, thereby enable the system to automatically change between normal operation, used for maximum control stability, and extraordinary operation with fast response, used for management of small radius of curvatures.

References


Method eliminating position errors caused by kinematics singularities in robotic wrists for use in intelligent control and telerobotics applications

Mikael Fridenfalk
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Application No. SE0203180-5, October 2002
Method eliminating position errors caused by kinematics singularities in robotic wrists for use in intelligent control and telerobotics applications

Mikael Fridenfalk
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

1 Background of the invention

Forward kinematics in robotics denotes the calculation of the Cartesian position and orientation of the last link of the robot manipulator for a robot with \( n \) degrees of freedom as a function of its joint values, \( \theta_1 - \theta_n \), see Fig. 1. Inverse kinematics denotes the calculation of joint values as a function of the Cartesian position and orientation of the last link of the robot manipulator, and has often several solutions. These solutions are called kinematics configurations.

In robotics, the manipulator may move into areas in which no inverse kinematics solutions exist, generally called kinematics singularities. Kinematics singularities cause control disruption in robotics and has been the focus of a large number of scientific papers and a few patents (US4716350, EP0672507, US5159249). The method presented here minimizes the problems caused by kinematics singularities by a new approach.

2 Problem analysis

The Jacobian matrix that gives the relation between joint and Cartesian velocities is invalid near and on singularity points. Since joint, velocity, acceleration and torque constraints, to name some, are not included in the Jacobian, the Jacobian may at best be used to measure the amount of manipulability of the robot near singularity areas. The concept of a “Real Jacobian” is here introduced as a Jacobian that also takes consideration to the kinematics and dynamics constraints. Such Jacobian should for instance be non-singular near
and on the singularity point for Roll-Pitch-Yaw (RPY) wrists, \textit{i.e.} when $\theta_4$ and $\theta_6$ coincide in Fig. 1.

Motion near the singularity point of for instance a RPY-wrist may cause great position errors, which may cause damage to the robot equipment or the work-piece, or delays in a production line. As robotic systems become more intelligent by introduction of sensor guided robot control systems, the requirements on the ability to handle kinematics singularities is proportionally increased, since the precise motion of the robot is no longer known in advance. This applies also to telerobotics applications, where the motion of the robot manipulator most often is directly controlled in Cartesian space by a human operator.

3 New method

The new method is based of a different approach to this problem and offers a more general, yet direct solution. According to preliminary experiments based on theoretical simulation models, while position errors for the TCP in Fig. 1 up to several centimeters were detected in some cases caused by other methods, the position errors were entirely eliminated by the new method.

By $\theta_4 - \theta_6$ in Fig. 1, the wrist may orient the tool-tip in an arbitrary direction in space. In the example in Fig. 1, since a welding torch is mounted on the last link, a change of orientation effects also the position of the TCP. By modification of the inverse kinematics calculations according to Fig. 2, the position errors are however entirely eliminated. According to preliminary experiments mentioned above, while position errors up to several centimeters were eliminated by this method, the orientation errors remained basically unchanged.

The components of the scheme in Fig. 2 are labeled as $A$, $B$ and $C$, where

(A) is a limiter, that limits the motion for a robot with respect to kinematics and dynamics constraints such as maximum joint, velocity, acceleration and torque limits. The limiter may in addition be implemented to decrease or equalize constraint values in the neighborhood of singularities, for enhanced control stability.

(B) is a configuration manager that automatically changes kinematics configuration, if crossover to the new configuration, based on information from $A$, is unavoidable. The configuration of $B$ is linked to the inverse kinematics box in Fig. 2.

(C) is a compensator that on the basis of $A$ eliminates any TCP position errors, by the modification of the position of the wrist. The resulting transformation matrix $P_{out}$ is built by concatenation of the position part of $P_{in}$ and rotation part of $P_{lim}$.

One way to interpret the functionality of this new method is that since joints $\theta_1 - \theta_3$ are in general driven by much more powerful motors compared to the
wrist joints, position errors are eliminated by using the more powerful motors of the robot to compensate for the shortcoming of the less powerful ones.

**Brief description of the drawings**

1. A typical industrial robot, ABB IRB 2400.16, with a RPY-wrist and a welding torch. The Tool Center Point (TCP) is defined as the tool-tip of the robot. Joint 3 (θ₃) is here symbolically denoted according to this figure. In reality, for this specific robot, this joint is managed by a link near the base of the robot.

2. The flow scheme of the new method. The input to the system is $P_{in}$, which is a $4 \times 4$ transformation matrix often used in robotics, including both position and orientation of the TCP. In block $C$, the rotation part of the limited input $P_{lim}$ is concatenated with the position part of $P_{in}$, thereby eliminating the position errors of the TCP, while keeping the orientation errors basically unmodified. The reason why this is possible is because the motors of the first three joints are in general much more powerful than the wrist motors.
Abstract

The invention consists of a method that minimizes problems caused by kinematics singularities in robotic wrists. By elimination of position errors near and on inner singularity points, based on kinematics and dynamics constraints, fewer failures may occur, especially in intelligent robot control systems or telerobotics applications. One interpretation of the new method is that stronger motors are allowed to compensate the errors caused by weaker ones of a robotic manipulator.
Claims

1. Method for elimination of position errors near and on inner kinematics singularity points for robotic wrists, by using an inverse kinematics scheme according to Fig. 2, based on AC, where A is an inverse kinematics calculator followed by a limiter that limits the motion for a robot with respect to kinematics and dynamics constraints of the total robot system including its surrounding environment, incorporating properties such as maximum joint, velocity, acceleration and torque limits. This limiter may additionally decrease or equalize constraint values in the neighborhood of singularities, for enhanced control stability. C is a compensator that on the basis of A modifies the position of the TCP in Fig. 1, building the transformation matrix \( P_{out} \) in Fig. 2, by concatenation of the position part of \( P_{in} \) and the rotation part of \( P_{lim} \).

2. According to above, but based on ABC in Fig. 2, where B is a configuration manager that automatically changes kinematics configuration for A, if crossover to the new configuration, based on information from A, is unavoidable. The configuration to be used is selected by B and is communicated to the inverse kinematics box in Fig. 2, so that the same configuration is used throughout the scheme.
Fig. 1

TCP

Fig. 2

A or AB
Forward Kinematics

Configuration

Plim

Ppos

Pout

C

Inverse Kinematics

P_{in}

P_{end}

\theta_{in}

\theta_{end}

P_{pos}
Patent II

Intelligent system eliminating trajectory programming and geometrical databases in robotic welding

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

Application No. SE0203239-9, November 2002
Intelligent system eliminating trajectory programming and geometrical databases in robotic welding

Mikael Fridenfalk, Gunnar Bolmsjö
Division of Robotics, Department of Mechanical Engineering
Lund University, Box 118
SE-221 00 Lund, Sweden
www.robotics.lu.se

1 Introduction

1.1 Robot welding

Robot automation increases productivity, the product quality and frees man from involuntary, unhealthy and dangerous work. While computational power has increased exponentially during the last decades, since many new application areas require operation in relatively unstructured environments, the limitations in flexibility constitute today a bottleneck in the further evolution of robotic systems.

One application for intelligent systems is seam tracking in robotic welding. Seam tracking is among others used for the manufacturing of large ships such as cruisers and oil tankers, where high tolerances in the sheet material are allowed to minimize manufacturing costs [2, 11]. Seam tracking is today typically only performed in 2D, by a constant motion in \(x\) and compensation of the errors in \(y\) and \(z\) directions, see Fig. 1. There exist solutions for higher autonomy that include seam tracking in 3D where compensation is also performed in \(o\) direction or around an orientation axis. These limitations make however seam tracking only possible for workpieces with relatively simple geometrical shapes.

The next step in the evolution of intelligent systems in robotics welding is perhaps the introduction of a full 6D sensor guided control system for seam tracking that is able to correct the TCP in \(x\), \(y\), \(z\) and around roll, pitch and yaw. Such ultimate system is per definition able to follow any continuous 3D seam with moderate curvature. This system has many similarities with an airplane control system, where a change in either position or orientation effects all other five degrees of freedom.
Though seam tracking has been performed in 2D since at least 40 years ago, the hypothetical design of 6D seam tracking systems was probably first addressed in the beginning of the last decade by suggestions of models based on force sensors [4, 5] and laser scanners [15, 16]. It is however difficult to evaluate these systems, since no experimental results are presented, neither any explicit control systems for creating balance between the subsystems.

The major contribution of this patent is the invention of a universal 6D seam tracking system for robotic welding, validated by simulation experiments based on physical experiments, that proved that 6D seam tracking is possible and even very robust for laser scanning and arc sensing. On a more detailed level, the general contributions of this patent is the introduction of: (1) the concept of a 6D seam tracking based on arc sensing, (2) differential control, not only in arc sensing as performed in 2D seam tracking [6], but also in laser scanning, (3) a trajectory tangent vector using differential vector interpolation control in laser scanning, (4) the trajectory tangent vector in arc sensing, (5) pitch control for laser scanning by vector interpolation, (6) pitch control in arc sensing and (7) new method replacing quaternion interpolation and thereby increasing the computational efficiency significantly in arc sensing.

Though the development of the seam tracking system in this patent was performed entirely without any knowledge of previous work within 6D seam tracking, there is perhaps by coincidence one unique similarity to be found between the this system and previous in [15, 16], which is the introduction of a trajectory tangent vector by curve fitting. There exist however large differences between how such vector was used and implemented. The differences consist of (1) curve fitting by 2nd instead of 3rd degree polynomials, for faster computation and according to preliminary experiments, for higher control stability, (2) using an analytic solver for curve fitting of 2nd degree polynomials developed and implemented for this system, increasing the calculation speed further and (3) not using the trajectory tangent vector directly, but by differential vector interpolation between what could be called the current and the estimated trajectory tangent vectors, for essential control stability, and (4) the introduction of the trajectory tangent vector also in arc sensing.

The objective of this work was to increase the intelligence and flexibility in manufacturing systems based on robotic welding by the design and validation of a universal 6D seam tracking system based on laser scanning or arc sensing. This eliminates robot trajectory preprogramming and geometrical databases. The simulation results (based on physical experiments) showed that the initial objective to design a 6D seam tracking system that could manage a radius of curvature down to 200 mm was reached. Further on, according to the experiments presented in this paper, if low-speed seam tracking is performed without welding to find the geometry of the workpiece, there is basically no limit how small radius of curvature this system is able to handle, using laser scanning.
2 Materials and methods

2.1 Sensors guided robot control

Many systems have been developed during the last decades for seam tracking at arc welding. The patents and scientific publications within this application area show that laser scanners and arc sensors today replace older systems. Laser scanners have in general high accuracy but are expensive and have to be mounted on the torch, thereby decreasing the workspace of the robot. This problem is partly solved by introduction of an additional motion that rotates the scanner around the torch to keep its relative orientation constants with respect to the seam profile. Arc sensors are inexpensive but inaccurate due to the stochastic nature of arc welding. Since welding of large structures in general requires a relatively low accuracy however, arc sensors are competitive alternative to laser scanners. In addition, it should be mentioned that there exist solutions for digital image processing of the workpiece before welding. The problems are however that small changes in surrounding light may disturb analysis. Further on, 3D spline curve seams require the camera to follow the seam all the way to be able to capture segments that otherwise would have remained hidden. This applies also to very long seams. In practical applications today, solutions based on pure image processing by a CCD camera are successfully used to find the initial point where 2D seam tracking, based on laser scanning or arc sensing should start [11].

Laser scanning

In laser scanning, the seam profile is extracted by periodically scanning the laser beam across a plane perpendicular to the direction of the seam. The laser scanner is mounted on the torch in front of the direction of welding. As a general rule, the control system is based on full compensation of the position errors, which is equal to a constant value of $K_1 = 1$, as defined in Fig. 5.

Arc sensing

In “through-arc” sensing or simply arc sensing [6], the arc current is sampled while a weaving motion is added to the TCP in $n$ direction, see Fig. 1. Weaving is useful in arc welding due to process related factors and was used already before the introduction of arc sensing in the beginning of the 80s. In arc sensing, by keeping the voltage constant it is possible to calculate the distance from TCP to the nearest seam wall at each current sample, and thereby during a weaving cycle obtain the profile of the seam. According to experimental results [6] the approximate relationship between arc voltage $V$, arc current $I$ and
the nearest distance between the electrode and the workpiece \( l \) for an electrode extension ranging between 5-15 mm, may be expressed by the relation:

\[
V = \beta_1 I + \beta_2 + \beta_3/I + \beta_4 l
\]  

(1)

where the constants \( \beta_1 - \beta_4 \) are dependent on factors such as wire, shielding gas and power-source. In practice the voltage and current readings of the arc contain much noise, why the signal has to be low-pass filtered before it is used for feedback control. An example of such filtered signal, sampled in the ROWER-2 project is presented in Fig. 3. In automatic control, error is defined as the difference between desired (input) value \( u \) and current (output) value \( y \). The control algorithms in arc sensing are based on methods such as template matching or differential control \([6]\). In template matching the errors may for instance be calculated by integrating the difference of the template \( u \) and the arc signal \( y \), where \( A \) denotes the weaving amplitude:

\[
E_a = \epsilon_a = \int_{-A}^{A} (u(n) - y(n))dn
\]  

(2)

\[
E_n = \epsilon_n = \int_{-A}^{0} (u(n) - y(n))dn - \int_{0}^{A} (u(n) - y(n))dn
\]  

(3)

Another example includes using integrated difference squared errors. The equation above use \( n \) as variable to simplify the notation. Actually, more specifically, the integration between \(-A\) and \(A\) is performed in negative \( o \) direction, using an auxiliary variable \( \psi \) denoting the state of weaving in \( n \) direction, integrated over \([\psi_{-A}, \psi_A]\), where \( \psi_{-A} \) and \( \psi_A \) denote the turning points of weaving. In differential control, which has proven to be quite reliable for control in \( a \) and \( n \) directions, current sampling is performed at the turning points of the weaving. Here, the errors in \( a \) and \( n \) directions are obtained by:

\[
\epsilon_a = I_{ref} - \frac{i_{+A} + i_{-A}}{2}
\]  

(4)

\[
E_a = K_a \epsilon_a
\]  

(5)

\[
\epsilon_n = i_{+A} - i_{-A}
\]  

(6)
\[ E_n = K_n \epsilon_n \] (7)

where \( I_{ref} \) is a reference current value and \( i_{+A} \) and \( i_{-A} \) are the average measured current values at a pair of adjacent extreme points. The parameters \( K_a \) and \( K_n \) are dependent on the weld joint geometry and other process parameters such as shielding gas and wire feed rate. Since these parameters will be known in advance, \( K_a \) and \( K_n \) may be defined for any welding application. An example of how \( E_a \) and \( E_n \) are used for position control follows below:

\[ p_N = p_{N-1} + E_n n - r_w o + E_a a \] (8)

where \( p_{N-1} \) and \( p_N \) denote current and next positions and \( r_w \) is nominal displacement during a weaving cycle, also in Fig. 1, which is equal to welding speed (without consideration to the weaving motion) divided by weaving frequency.

Virtual sensors

The development of sensor guided control systems may be accelerated using virtual sensors in the simulation control loop. Since a system’s state is often easily obtained in simulation, assuming that the simulation model is sufficiently accurate compared to the real system, this provides for a better foundation for process analysis than in general is possible in physical experiments. Further on, insertion of a sensor in a control loop may cause damage to expensive equipment, unless the behavior of the entire sensor guided control system is precisely known, which is rarely the case at the development stage.

Virtual sensors are used in many application areas, such as robotics, aerospace and marine technologies [3, 8, 14, 17, 19]. Development of new robot systems, such as for seam tracking may be accelerated by application of simulation [12, 13]. In general, the design methodology of virtual sensors may vary due to their specific characteristics. If a sensor model is not analytically definable, artificial neural networks may be used to find appropriate models [1, 18].

2.2 Robot system

The design of the 6D seam tracking system was a spin-off from the European ROWER-2 project. The objective of this project was to automate the welding process in shipbuilding by the design of a robotic system, specifically for joining double-hull sections in super-cruisers and oil tankers. Figure 2 shows the
robot system developed in the ROWER-2 project and the corresponding simula-
tion model in FUSE (described later). The implementation of the 3D seam
tracking system was performed in C/C++, running on a QNX-based embedded
controller (QNX is an operating system for embedded controllers). The robot
was manufactured partly in aluminum alloy to decrease the weight for easy
transportation and mounted on a mobile platform with 1 degree of freedom
for increased workspace.

2.3 Simulation system

The development and implementation of the 6D seam tracking system was
initially performed in the Flexible Unified Simulation Environment (FUSE) [10].
FUSE is an integrated software system based on Envision (robot simulator,
Delmia Inc.) and Matlab (MathWorks Inc.), including Matlab Robot Toolbox [7].
The integration was performed by compiling these software applications and
their libraries into one single software application. The methodology de-
veloped for the design of the 6D seam tracking system was based on the
integration of mechanical virtual prototyping and software prototyping. The
6D seam tracking simulations in FUSE were performed either using the robot
developed in the ROWER-2 project or the ABB S4 IRB2400.16 robot, see Figs. 2
and 9. After the design and successful validation of the 6D seam tracking
system for arc sensing, the virtual prototyping and software models were en-
tirely translated to Matlab code. The Matlab models were modified to work
also for laser scanners by the addition of a virtual model of the Servo-Robot
M-SPOT laser scanner system. The translation was necessary for the accurate
calculation of position and orientation errors during simulation.

2.4 Calculation of position and orientation errors

Error is defined as the difference between desired and current pose $P_t$. The po-
sition errors were calculated by projection of $\epsilon_a$ and $\epsilon_n$ on a plane $\Omega_t$ perpen-
dicular to the desired trajectory while intersecting current TCP, $P_t$. Thereby,
per definition $\epsilon_o = 0$. The orientation error (the orientation part of $P_o$) is
calculated by:

$$ P_o = P_t^{-1} \cdot P_{\Omega_t} \quad (9) $$

where $P_t$ is the current TCP and $P_{\Omega_t}$ is the desired TCP for any sample point
$t$. The errors around $n$, $o$ and $a$ are calculated by the subsequent conversion
of the resulting transformation matrix to $roll$, $pitch$ and $yaw$, here defined as
rotation around, in turn, $a$, $o$ and $n$. A rotation in the positive direction in here
(and in general) defined as counterclockwise rotation from the perspective of
a viewer when the axis points towards the viewer.
3 Modeling

In this section the design of the 6D seam tracking system is described for laser scanning and arc sensing. The principal control scheme of this system is presented in Fig. 4. The control system is based on three main components, position, trajectory tangent vector and pitch control. Pitch denotes rotation around \( \omega \). The main input of this control system is the \( 4 \times 4 \) transformation matrix \( P_{N-1} \), denoting current TCP position and orientation. The output is next TCP, \( P_N \). \( N \) is the number of samples used for orientation control. The input parameters are \( N \), the proportional control constants \( K_1, K_2 \) and \( K_3 \), and the generic sensor input values denoting \( \xi = [\epsilon_a \, \epsilon_n] \) in Eqs. 10-11 and \( \zeta = [k_1 \, k_2] \) in Figs. 6-7 for arc sensing. The proportional constants \( K_1, K_2 \) and \( K_3 \) denote the control response to errors in position, trajectory tangent vector and pitch control. \( N \) is also used in trajectory tangent vector control and is a measure of the memory size of the algorithm.

The seam profile presented in Fig. 1 is called a fillet joint. Since all standard seam profiles are basically similar, the generic parameters \( \xi \) and \( \zeta \) may be estimated by the same profile extraction methods used for fillets joints in both laser scanning and arc sensing. An example of an alternative to fillet is the V-groove, which mainly contains a gap between the seam walls. In arc sensing, since electrical current always takes the shortest path between TCP and the seam walls, see Fig. 6, such gap has in general small influence on the measurements in arc sensing. In addition, since using other shapes than sinus, such as sawtooth and square wave, does not effect seam tracking as much as it effects the welding process in arc sensing, only sine waves were used in the simulation experiments.

3.1 Position control

The 6D seam tracking algorithm starts by the calculation of next position for the TCP. In laser scanning, next position \( p_N \) is calculated by vector interpolation based on \( K_1 \) in Fig. 5, between current position \( p_{N-1} \) and the position \( p' \) extracted from the laser scanner.

In arc sensing, the position is calculated by the differential method in Eqs. 10-11, rewritten as a function of distance instead of current (with \( d \) defined in Fig. 1), calculated by using Eq. 1:

\[
\epsilon_a = s(i_A) + s(i_{-A}) - 2d \tag{10}
\]

\[
\epsilon_n = s(i_{-A}) - s(i_A) \tag{11}
\]
Since the distance to the workpiece is approximately inverse proportional to the current, this may explain the main differences between Eqs. 4 and 6 and Eqs. 10-11. Also here, $E_a = K_a\epsilon_a$, $E_n = K_n\epsilon_n$ and $p_N$ is calculated by Eq. 8. In the ROWER-2 project, simulation and physical experiments showed that $K_n$ should be 50% of $K_a$ for maximum control stability, both for Eqs. 4-6 and Eqs. 10-11. Here, $K_a = K_1$ and $K_n = K_1/2$. Since the position is controlled by keeping the distance from TCP to the nearest seam wall constant at the turning points of the weaving, the desired distance from TCP to the intersection point of the walls in Fig. 1 was derived as:

$$D = A + d\sqrt{1 + \tan^2(\alpha/2)} \over \tan(\alpha/2)$$

(12)

This equation is valid for any angle $\alpha$ different to an integer multiple of $\pi$, where $D$ is instead equal to $d$. This relation was primarily used for the calculation of the position errors in evaluation of the 6D arc sensing system.

### 3.2 Trajectory tangent vector control

The positions $p_1 - p_N$ are used for trajectory estimation by curve fitting, see Fig. 8. To decrease calculation time, an analytical estimator was devised for 1st and 2nd degree polynomials that showed to work much faster than traditional matrix inversion techniques using Gauss elimination. The estimation of the trajectory at $N$ gives $o'$. Since an attempt to abruptly change the direction of $o$ from $o_{N-1}$ to $o'$ would most probably create instability in the system, the motion between $o_{N-1}$ and $o'$ is balanced by vector interpolation using $K_2$.

### 3.3 Pitch control

In laser scanning, the first step in pitch control is performed by vector interpolation between $a_{N-1}$ and $a'$, using $K_3$, see Fig. 5. In a second step $a_N$ is calculated by orthogonally projecting $a''$ on a plane perpendicular to $o_N$. Therefore, $n_N$ is calculated by the cross product $o_N \times a_N$.

In arc sensing, pitch control is performed in two steps. First, $a_{N-1}$ is orthogonalized with respect to $o_N$. This step could also, in similarity to laser scanning, be performed after the second step (which then becomes the first step). Empirical experiments showed however that the performance of the algorithm was slightly improved by pre-orthogonalization. In the second step the new coordinate system is rotated around $o$ by $\delta\phi$:

$$\delta\phi = -K_3 \cdot (k_1 + k_2)$$

(13)
Here, $\delta \phi$ is in radians, and $k_1$ and $k_2$ are estimated by the least square method, based on current measurements during a half weaving cycle, see Figs. 6-7. The index $i$ in these figures denote current samples. Since current always takes the shortest path between TCP and the nearest seam wall (homogeneously conducting surface) and the distance between the TCP and the seam walls may be calculated by Eq. 1, the full extraction of the seam profile is theoretically possible. With $\sin \gamma = \partial s/\partial n$ and $k = -\partial s/\partial n$ in Figs. 6-7, the following relation is deduced:

$$k = -\sin \gamma$$

(14)

The orientation of the walls projected on plane $\Omega$ is calculated from the vector components of $s_i$:

$$s_i = |s_i| (\cos \gamma \cdot a_i + \sin \gamma \cdot n_i)$$

(15)

where $|s_i|$ is the distance calculated from the current measurement, using Eq. 1. A combination of Eq. 14 and $\sin^2 \gamma + \cos^2 \gamma = 1$ gives:

$$\cos \gamma = \sqrt{1 - \sin^2 \gamma} = \sqrt{1 - k^2}$$

(16)

which finally yields the direction of the current, projected on plane $\Omega$:

$$s_i = |s_i| \left(\sqrt{1 - k^2} \cdot a_i - k \cdot n_i\right)$$

(17)

### 3.4 Code samples

The actual implementation of the 6D seam tracking system showed to be both simple and concise. In simulation, plane $\Omega$ is assumed to intersect the TCP point. In reality however, the laser scanner is placed on the torch, often several centimeters ahead of the direction of the seam. The transformation to such system is simply performed by storing the trajectory data, for use with a time delay. Most of the development effort was spent on the design of the surrounding environment, including a semi-automatic testing and evaluation system. The implementation of the arc sensing system contained much code, why it is not presented here. The Matlab implementation of the 6D seam tracking system for laser scanning, also presented in Fig. 4 follows below:

```matlab
function P = next_P(P,v,w);
```
p = P(1:3,4)-P(1:3,2)*w.r_w+w.K1*(v(:,1)-P(1:3,4)+v(:,2)*w.D);
global x; x = [x(:,2:end) p];
for i = 1:3, F(i) = pfit(1:w.N,x(i,:)); end
o = unit(P(1:3,2)-w.K2*(unit(F')+P(1:3,2)));
a = unit(P(1:3,3)-w.K3*(P(1:3,3)+v(:,2))); a = unit(a-o*a'*o);
P(1:3,:) = [cross(o,a) o a p];

The input of this function is current pose, here denoted simply as \(P\), readings from laser scanner \(v\) according to Fig. 5, and the parameter data structure \(w\). The output is next pose, also here denoted as \(P\) for code minimization. The parameter \(w.D\) is equal to \(D\) in the same figure, \(F\) is defined in Fig. 8 and \(w.r_w\) here is nominal displacement during a control sample. The matrix \(x\) denotes here the last \(N\) positions \(p_0\ldots p_{N-1}\). By calculation of \(p_N\) and the subsequent matrix shift, \(p_0\) is erased. The initial value of \(x\) is determined by linear extrapolation with respect to the estimated direction at the beginning of the seam. The function \(pfit(1:w.N,x(i,:))\) is equal to \(\text{polyfit}(1:w.N,x(i,:),2)*[2*w.N 1 0]'\). The \(pfit\) function was developed in this work using Maple (Waterloo Maple Inc.) and gives an analytical solution to 2nd degree polynomial fitting for this application, that for example works better than 8 times for \(N = 10\) and 4 times for \(N = 100\) than \(\text{polyfit}\). The normalization function \(\text{unit}(x)\) from Matlab Robot Toolbox [7] is for a \(n \times 1\) vector \(x\) equal to \(x/\sqrt{x'x}\). The analytical function \(pfit\) was implemented as below:

```matlab
function k = pfit(x,y)
    n = length(x); sx = sum(x); sy = sum(y); sxx = x*x'; sxy = x*y';
    sx3 = sum(x.^3); sx4 = sum(x.^4); sxxy = sum(x.^2.*y);
    t2 = sx*sx; t7 = sx3*sx3; t9 = sx3*sxx; t12 = sxx*sxx;
    den = (sx4*sxx*n-sx4*t2-t7*n+2.0*t9*sx-t12*sxx);
    if abs(den) < 1e-5, k = 0; else,
        t15 = 1/den; t21 = (sx3*n-sxx*sx)*t15;
        k = 2.0*n*(sxn-t2)*t15*sxxy-t21*sxy+(sx3*sx-t12)*t15*sy-...
            t21*sxxy+(sx4*n-t12)*t15*sxy+(t9-sx4*sx)*t15*sy;
        end
```

In linear fitting, used for the calculation of \(k_1\) and \(k_2\), \(k\) is calculated by:

\[
    k = (n*sxy-sx*sy)/(n*sxx-sx^2);
\]

### 3.5 Computational costs

The computational costs for the 6D seam tracking system based on laser scanning (without consideration to memory operations) was preliminary estimated
to less than $250 + 50N$ floating point operations, which for $N = 10$ gives 750 operations per control cycle. At a rate of maximum 30 scans per second, this gives 22,500 floating-point operations per second.

3.6 Arc sensing weaving interpolation

The interpolation between $P_{N-1}$ and $P_N$ is usually performed by vector interpolation of position $p$ and quaternion interpolation of the orientation $\{n,o,a\}$. Since most of the calculation time in arc sensing was spent on quaternion interpolation, a new method was adopted that empirically showed to work much faster and still performed as well as quaternion interpolation in 6D seam tracking. The denotations in this subsection, generically labeled as $x'$ and $x''$, are local and not related to other denotations elsewhere.

In this method, linear interpolation was first performed between $P_{N-1}$ and $P_N$ including $o$, $a$ and $p$ for any sample $i$, which resulted in $o'_i$, $a''_i$ and $p'_i$. $o'_i$ was rescaled to the unit vector $o_i$, and $a''_i$ was projected on the plane perpendicular to $o_i$ by:

$$a'_i = a''_i - (a''_i \cdot o_i) o_i$$  \hspace{1cm} (18)

The dot product $a''_i \cdot o_i$ gives the length of the vector $a''_i$ orthogonally projected on $o_i$. $a_i$ was finally calculated by rescaling $a'_i$ to the unit vector $a_i$, and $n_i$ was calculated by $o_i \times a_i$. The weaving motion was added as a second layer of motion to $P_i$ in $n_i$ direction. In robot simulation the torch was allowed to freely rotate $\pm 180^\circ$ around $a$ as a third layer of motion, which showed to be important, but not entirely imperative for the robot to be able to perform $360^\circ$ welding of pipe intersection joints, see Fig. 9. The angle around $a$ was determined by the position of the robot with respect to the position of the workpiece.

3.7 Kinematics singularities

In sensor guided control, the manipulator may involuntarily move into areas in which no inverse kinematics solutions exist, generally called kinematics singularities. It is however possible to minimize inner singularity problems caused by robotic wrists. A novel method is suggested in [9] which was designed for the 6D seam tracking system, but works basically for any industrial robot with 5-6 degrees of freedom. In this method, the stronger motors often at the base of the robot assist the weaker wrist motors to compensate for any position error. This method allows also for smooth transitions between different configurations.
In addition to the addition of this method, to ensure good welding quality, pre-scanning and trial run may be performed before welding in laser scanning. In the first step, low-speed seam tracking is performed to calculate the trajectory of the TCP, following the seam. In the second step, a trial run is performed in full speed to check the performance of the robot, and finally the actual welding is performed.

3.8 Initial and final conditions

In the experiments, seam tracking was initiated from a proper position already from start. The start position for seam tracking is in general found either by image analysis using a camera or by seam tracking itself. Since previous positions are unknown at start, an estimation is made of previous \( N - 1 \) positions, as previously mentioned, by extrapolation of the trajectory in the direction of \( o \). This is the reason why in some experiments, higher orientation errors occurred at the beginning of seam tracking. The condition to stop seam tracking may be based on the position of the TCP. The stop signal may be triggered by definition of volumes outside which no seam tracking is allowed.

4 Experimental results

4.1 Workpiece samples

The position and orientation errors at seam tracking were calculated based on simulation experiments of basically 8 different workpieces, see Figs. 10-18. The workpieces in Matlab were approximated by lines orthogonal to the direction of the weld and the resolution was defined as the nominal distance between these lines. These resolutions were set to 0.02 mm in laser scanning and 0.5 mm in arc sensing.

The workpieces are categorized into two groups, continuous and discrete. In the experiments presented in this patent, the range was 0 – 180° for continuous, and 10-100 mm for discrete workpieces. In seam tracking of a continuous workpiece, a sudden change is interpreted as a curve with a small radius of curvature, see Figs. 10-11 and 15-16, while in seam tracking of a discrete workpiece a sudden change is regarded as a disturbance, see Figs. 13-14 and 17-18. The settings of the control parameters decide alone the mode of the control system: continuous or discrete. For seam tracking of objects with a large radius of curvature, the discrete mode works well for both a continuous and a discrete workpiece, see Fig. 12.

The control parameters \( K_1, K_2, K_3 \) and \( N \) used in the presented experiments showed to work well for each category, continuous or discrete. These settings were found by more than 150 simulation experiments in Matlab, and
large deviation from these parameters showed to create instability in the system, making the control components working against each other instead of in synergy, resulting in divergence from the seam and ultimate failure at seam tracking.

As mentioned previously, the seam tracking system is theoretically able to handle any angle $\alpha$ (in Figs. 1, 5 and 6) between 0 and 180$^\circ$. In practice however, a small angle may cause collision between torch and workpiece. For simplicity, $\alpha$ was set to 90$^\circ$ for all experiments except for pipe intersections. Further, in arc sensing $d$ in Fig. 1 was set to 1 mm.

### 4.2 Laser scanning

M-SPOT has a sweeping angle of 28$^\circ$, an accuracy better than 0.02 mm at the distance of 100 mm, and 1.5 mm at a distance of 1 m. It is able to scan at a frequency of up to 40 Hz with a resolution of maximum 512 points per scan. The virtual model of the M-SPOT that was designed, were set to have an accuracy of 0.02 mm and was mounted on the torch at a distance of approximately 80 mm from the workpiece. The resolution was set to 11 points per scan for a sweeping frequency of 5$^\circ$ (the only exception was for $n$ direction steps, see Figs. 14 and 18, where 31 points per scan with a sweeping angle of 10$^\circ$ was used instead) and the scanning frequency to 30 Hz, which showed to be fully sufficient for successful seam tracking.

In fact, the simulation experiments proved that the accuracy of M-SPOT was higher than needed for seam tracking. The accuracy is so high in close range (100 mm) that only robots with very high performance are able to use the accuracy of this sensor to its full extent, when seam tracking is performed in real-time. The experiments presented in Figs. 10-14 show the limits of performance for the 6D seam tracking system (based on the M-SPOT), when it is at the verge of instability. For larger radius of curvatures and a lower value for $K_2$, i.e. for normal operation, Fig. 12 gives perhaps a better indication of the efficiency of the 6D seam tracking system. Since the performance of a robot is dependent on current joint values, it is further not possible to translate these values directly to joint position, velocity and acceleration. What these experiments proved however, was that performance of the robot is the only limitation in seam tracking based on laser scanning.

The laser scanning simulations presented in this patent were performed with a constant welding speed of 10 mm/s and a virtual TCP, 2 mm away from the intersection point of the seam walls. Empirical experiments showed that by moving the TCP from previously $D = 15$ mm in Fig. 5 to only 2 mm, the control ability of the 6D seam tracking was highly improved, especially in seam tracking of objects with a small radius of curvature. Though the parameter settings $K_1 = K_2 = K_3 = 1$ showed to be unstable in laser scanning, $K_1 = 1, K_2 = K_3 = 0.8$ showed to be relatively stable for continuous welds. For
4.3 Arc sensing

According to [20], the accuracy of an arc sensor in seam tracking is 0.5 mm. In simulation, the workpiece line resolution was set to 0.5 mm. To increase the errors in the system, a random signal with a range of 1 mm was added to the measured distance. In addition, no integration of the current measurement values was made near the turning points at weaving. On the contrary, one measurement was considered as enough for each turning point. The signal was prefiltered by an active 4th order Bessel low-pass filter in the ROWER-2 project. In the simulation experiments presented in this patent no filtering was performed. Thereby the accuracy had been decreased to less than half of [20]. In practical applications however, the arc signal should be filtered by a method that does not add any phase shift to the signal, such as a proper digital filter.

The arc sensing simulations presented in this patent were performed with a constant welding speed of 10 mm/s, weaving amplitude and frequency of 5 mm and 3 Hz, and 64 current samplings per weaving cycle (or 192 samplings per second). The algorithm worked well also for lower sampling rates, such as 50 samples per second, which was used in the ROWER-2 project. In general, higher sampling rates showed to improve the control ability of the system, for rates higher than perhaps 10 samples per weaving cycle, though rather slightly than proportionally. The simulation models proved to predict the estimations derived from physical experiments in the ROWER-2 project for 6D motion. According to those experiments, it was possible to perform differential seam tracking in $y$ and $z$ directions in Fig. 1, up to 20 mm for a seam of the length 1 m. At a speed of 9 mm per second, weaving amplitude and frequency of 3 mm and 3 Hz, this gives a redirection capability of $1.15^\circ$ per weaving cycle which in the worst case is equivalent to a minimum radius of 150 mm. Using a power-source without any intrinsic current control would further give up to 3 times better results according to the experiments in the ROWER-2 project (current control was only applied by the power source to limit the current). Using a high performing industrial robot and a weaving amplitude of 5 mm instead of 3, this would finally bring down the curvature to 25-50 mm, which actually was the amount found by simulation.

5 Results and discussion

A 6D seam tracking system was designed and validated by simulation that is able to follow any continuous 3D seam with moderate curvature, using a laser scanner or an arc sensor, and is able to continuously correct both position and
orientation of the tool-tip of the welding torch along the seam. Thereby the need for robot trajectory programming or geometrical databases were eliminated. This system was designed primarily to increase the intelligence in robotic arc welding, but is also applicable for instance in robotic laser welding.

Simulations based on the M-SPOT laser scanner showed that 6D seam tracking was theoretically possible, of objects down to a radius of curvatures of 2 mm. Seam tracking in real-time requires however a robot with very high performance. It is however possible to perform low-speed seam tracking without welding to find the geometry of the workpiece, and perhaps even checking that the trajectory is within the workspace of the robot, before actual welding is performed. Thereby it will be possible to perform welding of very small objects, theoretically also including sharp edges.

Simulations based on physical experiments in the ROWER-2 project showed that arc sensing was theoretically possible for a radius of curvature below 50 mm. This is however much less than the initial objective to design a 6D seam tracking system for laser scanners and arc sensors that could manage a radius of curvature down to 200 mm, which is still considered as small for instance in shipbuilding. By performance of “non-destructive” and low-speed seam tracking using low voltage and current in arc sensing, it is perhaps possible to seam track also workpieces with very small radius of curvatures, by calculation of the trajectory, before real welding is performed. Thereby, welding using weaving in arc sensing will no longer be imperative, but an option.

### 6 Future work

Though the 6D seam tracking system showed to be very robust, optimization of the algorithm is still possible. Presently, pitch control in arc sensing is based on data measurements during a half weaving cycle. By extending the measurements to one full cycle, pitch control may be optimized. Further on, though the control system worked well using proportional constants alone, if needed, PID or adaptive control may be included to enhance the performance. One of many examples is for instance to define $K_2$ as a function of the “radius of curvature” (or more specifically as a function of $c_2$ in Fig. 8) of the 2nd degree polynomial in trajectory tangent vector control, thereby enable the system to automatically change between normal operation, used for maximum control stability, and extraordinary operation, giving fast response which is essential for management of very small radius of curvatures.

### References


**Brief description of the drawings**

1. Left: Definition of Tool Center Point (TCP) and the orthonormal coordinate system \{n, o, a\}. Weaving, if any, is performed in n direction in arc sensing, o is opposite to the direction of welding and perpendicular to the plane \(\Omega\), and a is the direction of approach. \{x, y, z\} is a local coordinate system, defined for the workpiece. Right: The optimal position for seam tracking in arc sensing.
2. The design of the robot system for 3D seam tracking based on arc sensing in the ROWER-2 project (left) was based on simulations in FUSE (right). As a spin-off effect after the completion of the ROWER-2 project, the 6D seam tracking system presented in this patent was developed.

3. Example of a filtered arc signal at seam tracking, sampled in the ROWER-2 project. Though anomalies are common at the ignition of the arc, only minor deviations from this pattern showed to occur during continuous seam tracking. The unit of the horizontal axis is ticks, which equals 1/50 seconds. The current values were filtered by an analogue active 4th order Bessel low-pass filter with $f_0 = 10$ Hz before sampling.

4. The principal scheme of the 6D seam tracking control system. $P_{N-1}$ and $P_N$ denote current and next TCP. In laser scanning, $P_N$ may be calculated at every control sample. In arc sensing, next position is calculated at the beginning of each weaving cycle, and the motion between $P_{N-1}$ and $P_N$ is interpolated.

5. Cross section, plane $\Omega$. Vector interpolation used for position and pitch control at laser scanning. The input from the laser scanner system is the intersection point of the seam walls $v_0$ and the unit vector $v_1$ on $\Omega$, between the seam walls. $D$ denotes the distance between $v_0$ and $p'$.

6. Arc sensing. Left: Cross section, perpendicular to the approach vector $a$. The weaving is added to the interpolated motion between $P_{N-2}$ and $P_{N-1}$, including changes in position and orientation. Right: Cross section, plane $\Omega$. By the measurement of the current from start to end, the profile of the workpiece is approximately determined.

7. Cross section, plane $\Omega$, corresponding to the upper seam wall in Fig. 1. The measured distance from TCP to nearest wall is always perpendicular to the wall. The geometrical relation between the measured distance and the geometrical shape of the workpiece is calculated by the least-square estimate of the coefficients $k_1$ and $k_2$ in Fig. 6. Note that since both position and the orientation of TCP is interpolated between $P_{N-2}$ and $P_{N-1}$, this figure is only an approximation of the trajectory between $P_{i-}$ and $P_i$. Note also that $\gamma < 0$ for $k > 0$.

8. Trajectory tangent vector control is performed by the least square curve fitting of the last $N$ trajectory points to a 2nd degree polynomial curve for $x$, $y$ and $z$, followed by vector interpolation. Here $N = 5$. Note that $X, c_0, c_1$ and $c_2$ denote vector entities. The trajectory tangent vector $F$ is equal to $c_1 + 2Nc_2$ for an 2nd degree polynomial.

9. Examples of 6D seam tracking experiments performed in Matlab using laser scanner (left) and FUSE, Envision, using a model of the ROWER-2 robot (right).
10. Seam tracking, laser scanning. Pipe intersection (top) and workpiece for isolating control around a (bottom). $K_1 = 1.0, K_2 = K_3 = 0.8, N = 10$. Error denotes the difference between desired and current pose.

11. Seam tracking, laser scanning. Isolating rotation around $n$, following inner and outer curves. Note that in this figure, the case in the bottom is not very realistic, but only included for the completeness of the experiments. $K_1 = 1.0, K_2 = K_3 = 0.8, N = 10$.

12. Seam tracking, laser scanning. The same as in previous figure, except for $K_2 = 0.1$ and $r = 50$ mm. This shows that $K_2 = 0.1$ works fine as well for continuous seams with moderate radius of curvature. Further, the same experiment was also performed with $r = 20$ mm, which showed to be basically identical to this figure, but with saturation at 12 degrees instead of 4. Note that while other laser scanning experiments were performed on the verge of instability to find out the limits of the seam tracking system, using the M-SPOT laser scanner, the system is highly stable at moderate curvatures using low values for $K_2$.

13. Seam tracking, laser scanning. The step is introduced at $x = 0$ mm. $K_1 = 1.0, K_2 = 0.1, K_3 = 0.8, N = 10$.

14. Seam tracking, laser scanning. In this figure, the case in the bottom denotes a step of $-30^\circ$ performed around $a$. $K_1 = 1.0, K_2 = 0.1, K_3 = 0.8, N = 10$.

15. Seam tracking, arc sensing, with addition of random noise to the measured distance, as defined in the text. $K_1 = 1.0, K_2 = 0.5, K_3 = 0.25, N = 10$.

16. Seam tracking, arc sensing, with addition of random noise to the measured distance. $K_1 = 1.0, K_2 = 0.5, K_3 = 0.25, N = 10$.

17. Seam tracking, arc sensing, with addition of random noise to the measured distance. $K_1 = 0.5, K_2 = 0.2, K_3 = 0.1, N = 20$.

18. Seam tracking, arc sensing, with addition of random noise to the measured distance. $K_1 = 0.5, K_2 = 0.2, K_3 = 0.1, N = 20$. 
Abstract

Robot automation increases productivity and frees man from involuntary and dangerous work. Though computational power has increased exponentially during the last decades, today, limitations in intelligence and flexibility constitute a serious bottleneck in the further evolution of robotic systems.

This invention consists of a universal 6D seam tracking system that eliminates robot trajectory programming or geometrical databases in robotic welding, thereby increasing the intelligence and flexibility in manufacturing systems. The system is able to follow any 3D spline seam in space with relatively small radius of curvature by the real-time correction of the position and orientation of the welding torch, using a laser scanner in robotic welding including both laser and arc welding, or an arc sensor in robotic arc welding.
Claims

1. Method for 6D seam tracking of seams following 3D spline curves using laser scanning or arc sensing based on a system of three control components used in any arbitrary order consisting of (1) position control, limiting the motion between current and desired position by the control parameter $K_1$, (2) trajectory tangent control, by calculation of the desired trajectory tangent vector $F$ in Fig. 8 based on $N$ trajectory points and the limitation of the motion between current and desired trajectory tangent vector by the control parameter $K_2$ and (3) pitch control, limiting the motion between current and desired pitch by the control parameter $K_3$. The parameters $K_1$, $K_2$ and $K_3$ may be functions of any properties.

2. A rapid method replacing quaternion interpolation, distinguished by, in order, interpolation of any two coordinate axes, the orthogonalization of one of these axes in relation to the other and the calculation of the third axis by preferably the cross product.
\[ X = c_0 + c_1 t + c_2 t^2 \]

\[ F = \frac{\dot{X}}{t - N} \]

Figure 6

Figure 7

Figure 8
Figure 9

Figure 10
\[ \Delta a = 5 \text{ mm} \]

\[ \Delta a = -5 \text{ mm} \]

---

Figure 13

---

Figure 14
Figure 15

Figure 16
\[ \Delta a = 5 \text{ mm} \]

\[ \Delta a = -5 \text{ mm} \]

Figure 17

Figure 18
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programmer’s Interface</td>
</tr>
<tr>
<td>CAR</td>
<td>Computer Aided Robotics</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>FUSE</td>
<td>Flexible Unified Simulation Environment</td>
</tr>
<tr>
<td>MP</td>
<td>Envision Motion Pipeline</td>
</tr>
<tr>
<td>OOP</td>
<td>Object-Oriented Programming</td>
</tr>
<tr>
<td>RT</td>
<td>Real-Time</td>
</tr>
<tr>
<td>SGRC</td>
<td>Sensor Guided Robot Control</td>
</tr>
<tr>
<td>TCP</td>
<td>Tool Center Point</td>
</tr>
</tbody>
</table>