Gabor Sziebig

Man-machine and inter-machine interaction in flexible manufacturing systems

Thesis for the degree of Philosophiae Doctor

Narvik, August 2013

Norwegian University of Science and Technology
Faculty of Engineering Science and Technology
Department of Production and Quality Engineering

NTNU – Trondheim
Norwegian University of
Science and Technology

Narvik University College

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering Science and Technology
Department of Production and Quality Engineering

ISBN 978-82-471-4155-7 (printed ver.)
ISBN 978-82-471-4156-4 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2013:31

Printed by NTNU-trykk

Abstract

Manual work is the key source of skill and ingenuity in industrial manufacturing. At the same time, it is the most expensive resource factor. Often machines can replace humans for more effective manufacturing because machines outperform humans in terms of strength, precision and endurance. Humans, however, perform better than machines when flexibility and intelligence is required.

Flexible Manufacturing Systems are capable of manufacturing wide variety of products with low volume size and minimal lead time. These manufacturing cells are usually equipped with high level computer based control and include highly flexible industrial robots also. Robots can partially resolve contradiction of flexibility and intelligence due to their more humanlike structure and programmability. To a large extent, robots have already relieved workers of many of the tedious, hard and unhealthy parts of industrial work. But the universal application of robots in small-batch highly customized production is hindered by the time consuming robot programming.

This thesis shows efficient methods and processes for teaching and optimizing complex robot tasks by introducing cognitive robot programming, flexible robotics, and middleware. Intelligent user interfaces combining information from several sensors in the manufacturing system will provide the operator with direct knowledge on the state of the manufacturing operation. Thus, the operator will be able to determine the system state quicker. Sensory system calculates and proposes the optimal process settings. The key element is the new cognitive human-machine communication channels helping the operator to comprehend the information from the sensory systems. Novel middleware technology is facilitated for integration of elements from different system platforms into a coherent robot system and scaling it for different hardware complexity levels.

The contributions of the thesis could be summarized as follows:

1. Introduction of a new scientific concept of task dependent, software component based controller for flexible manufacturing cells.
2. Development of a new paradigm of robot teaching and supervising, which opens a new dimension of robotization in the area of small and medium sized enterprises.
3. Introduction of a new concept for Coginitve Telemanipualtion.

Preface

The work presented in this dissertation was carried out during the period October 2007 to December 2012 at Department of Industrial Engineering, Narvik University College. The work was supervised by Professor Bjørn Solvang (Narvik University College), Professor Terje K. Lien (Norwegian University of Science and Technology) and Professor Péter Korondi (Budapest University of Technology and Economics), without their support it would be impossible to complete my work.

The research on cognitive systems for industrial robots was carried out as a part of an international research project HUNOROB - Hungarian-Norwegian research based on innovation for the development of new, environmental friendly, competitive robot technology for selected target groups.

The research on integration of old and new industrial equipment was carried out as part of international research project called DIM – Digital Integrated Manufacturing.

The research on cognitive telemanipulation was carried out as part of national research project called OTKA K62836 – Fund of Hungarian National Development Agency and the National Science Research Fund.

This thesis results have significantly contributed to newly initiated international projects like BANOROB - Bosnian–Norwegian research based innovation for development of new, environmental friendly, competitive robot technology for selected target groups – and SMaE - Sustainable Manufacturing and Engineering.

Experiments and test were carried out in the facilities provided by Narvik University College (Norway) and online experiments were carried out using facilities from University of Tokyo (Japan), Budapest University of Technology and Economics (Hungary) and Hungarian Academy of Sciences Computer and Automation Research Institute (Hungary).

I would like express my greatest gratitude to my main supervisor Professor Bjørn Solvang for providing me this unique opportunity and his never ending support and guiding through the entire doctoral studies.

I would like to thank also Professor Péter Korondi for providing me a stable startup and continuous support, his initial contribution is inestimable.

I thank the Norwegian Research Foundation for the financial support.

Contents

Abstract i

Preface iii

Acronyms and Abbreviations vii

Included papers ix

Structure of the work x

1. Introduction 1

2. Inter-machine and machine-machine communication in Flexible Manufacturing Cells / Systems 5

   2.1. State-of-the-art in system integration (Middleware technology) 6

   2.2. State-of-the-art in supervisory system 7

   2.3. Summary of achievements in Inter-machine and machine-machine communication in Flexible Manufacturing Cells / Systems 9

3. Human-System Interaction in Flexible Manufacturing Cells / Systems 11

   3.1. Intelligent space as a new paradigm in human-system interaction 11

   3.2. Summary of achievements in Human-System Interaction in Flexible Manufacturing Cells / Systems 13

4. Telemanipulation as part of the Human-System Interaction 15

   4.1. State-of-the-art in cognitive info-communication channels for human-system interaction 16

   4.2. Summary of achievements in Telemanipulation as part of the Human-System Interaction 17

5. References 19

Papers 23

Paper 1: Multilevel Control of Flexible Manufacturing Systems 25

Paper 2: Robot Programming in Machining Operations 33

Paper 3: Vision Based Robot Programming 53

Paper 4: Image Processing for Next-Generation Robots 61

Paper 5: STEP-NC Based Industrial Robot CAM System 77

Paper 6: Achieving Total Immersion: Technology Trends behind Augmented Reality - A Survey 85

Paper 7: Vibro-tactile feedback for VR systems 93

Paper 8: Control of an Embedded System via Internet 99

Paper 9: Cog Framework - 3D Visualization for Mobile Robot Teleoperation 111

Paper 10: Shop-Floor Architecture for Effective Human-Machine and Inter-Machine Interaction 117

Paper 11: On Industrial Robots and Cognitive Infocommunication 139

Paper 12: Navigating in 3D Immersive Environments: A VirCa Usability Study 147

6. Additional papers 153

7. Conclusion.. 157

8. Future work 159

Acronyms and Abbreviations

AIST	Advanced Industrial Science and Technology
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAN	Controller Area Network
CNC	Computerized Numerical Control
CORBA	Common Object Request Broker Architecture
DNC	Distributed Numerical Control
EPFL	École Polytechnique Fédérale de Lausanne
FMS	Flexible Manufacturing System
FMC	Flexible Manufacturing Cell
HTTP	Hypertext Transfer Protocol
NC	Numerical Control
ICE	Internet Communications Engine
IDE	Integrated Development Environment
ISO	International Organization for Standardization
JARA	Japan Robot Association
KTH	Kungliga Tekniska Högskolan
NASA	National Aeronautics and Space Administration
SME	Small and Medium sized Enterprise
SOAP	Simple Object Access Protocol
STEP	International Standard for the computer-interpretable representation of product information and for the exchange of product data (ISO 10303)
STEP-NC	International Standard for STEP based data transfer between CAD/CAM systems and CNC machines (ISO 14649 family)
XML	Extensible Markup Language

Included papers

The thesis is based on the following papers:

Paper 1: Bjørn Solvang, Gabor Sziebig, Peter Korondi, Multilevel Control of Flexible Manufacturing Systems. In: Proc. IEEE Conference on Human System Interactions (HSI'08). Krakow, Poland, 25/05/2008-27/05/2008. (IEEE)pp. 785-790.(ISBN: 1-4244-1543-8)

Paper 2: Bjørn Solvang, Gabor Sziebig, Peter Korondi, Robot Programming in Machining Operations. In: Robot Manipulators. Vienna: I-Tech Education and Publishing, 2008. pp. 479-496. ISBN: 978-953-7619-06-0

Paper 3: Bjørn Solvang, Gabor Sziebig, Peter Korondi, Vision Based Robot Programming. In: Proc. IEEE International Conference on Networking, Sensing and Control (ICNSC'08). Sanya, China, 06/04/2008-08/04/2008. (IEEE)pp. 949-954.

Paper 4: Gabor Sziebig, Bjørn Solvang, Peter Korondi, Image Processing for Next-Generation Robots. In: Computer Vision. Vienna: I-Tech Education and Publishing, 2008. pp. 429-440. ISBN: 978-953-7619-21-3

Paper 5: Bjørn Solvang, Lars Kristian Refsahl, Gabor Sziebig, STEP-NC Based Industrial Robot CAM System. In: 9th IFAC Symposium on Robot Control (SYROCO2009). Gifu, Japan, 09/09/2009-12/09/2009. (IFAC) Gifu: IFAC by Pergamon Press, pp. 361-366. Paper 80.

Paper 6: Gabor Sziebig, Achieving Total Immersion: Technology Trends behind Augmented Reality - A Survey. In: SIMULATION, MODELLING AND OPTIMIZATION. Budapest, Hungary, 03/09/2009-05/09/2009. WSEAS Press, pp. 458-463.(ISBN: 978-960-474-113-7)

Paper 7: Gabor Sziebig, Bjørn Solvang, Csaba Kiss, Peter Korondi, Vibro-tactile feedback for VR systems. In: 2nd Conference on Human System Interactions (HSI '09). Catania, Italy, 21/05/2009-23/05/2009. (IEEE)pp. 406-410.

Paper 8: Gabor Sziebig, Bela Takarics, Peter Korondi, Control of an Embedded System via Internet. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS 57:(10) pp. 3324-3333. (2010) IF: 3.439

Paper 9: Gabor Sziebig, Peter Zanaty, Peter Korondi, Bjørn Solvang, Cog Framework - 3D Visualization for Mobile Robot Teleoperation. ADVANCED MATERIALS RESEARCH 222: pp. 357-361. (2011)

Paper 10: Bjorn Solvang, Gabor Sziebig, Peter Korondi, Shop_Floor Architecture for Effective Human-Machine and Inter-Machine Interaction. ACTA POLYTECHNICA HUNGARICA 9:(1) pp. 183-201. (2012) IF: 0.385

Paper 11: Bjørn Solvang, Gabor Sziebig, On Industrial Robots and Cognitive Infocommunication. In: 3rd IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2012. Kosice, Slovakia, 02/12/2012-05/12/2012. pp. 1-6

Paper 12: Gabor Sziebig, Øritsland Trond Are, Navigating in 3D Immersive Environments: A VirCa Usability Study. In: Syroco 2012. Dubrovnik, Croatia, 05/09/2012-07/09/2012. Dubrovnik: pp. 380-384. Paper 141.

Structure of the work

The dissertation is based on scientific articles and the correlation between these are visualized in Figure 1.

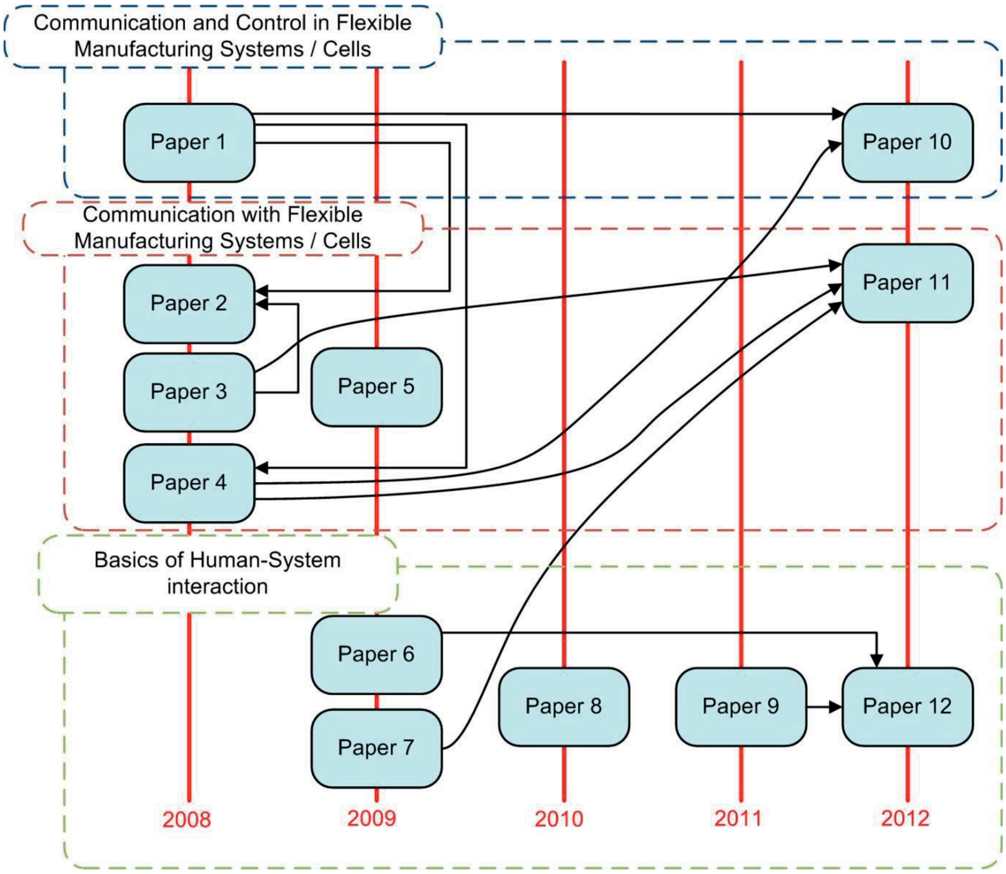


Fig. 1. Timeline and correlation between papers published in the PhD work period

Already in the first year of the PhD studies, research results were published in different areas:

Paper 1 shows a state-of-the art study of control and communication possibilities inside flexible manufacturing systems / cells. This paper also has development and testing results with showing one example of integration of outdated manufacturing equipment.

Paper 2 summarizes previously started research activities in the field of vision based robot programming.

Paper 3 describes a novel method for programing of industrial robots by visual information. This paper also has development and test results.

Paper 4 introduces an image processing toolkit using state-of-the-art middleware technology in order to be used with robotic applications or inside flexible manufacturing systems / cells.

In the first year of the work there has been several developments also in the field of human-system interaction including novel systems for telemanipulation and also interaction through internet, but these only shaped form in high quality papers in the second year of the work.

Paper 5 represents a milestone in the work. This paper is a result of a development toward unified programming languages for NC machines and industrial robots. The paper also includes test results along with methodology how to interpret and adapt the new machining standard for industrial robots.

Paper 6 is a review paper of the different technological solutions in interaction between human and system.

Paper 7 introduces a solution for force feedback in human-system interaction with a novel solution using cognitive infocommunication channels. This paper also has development and test results.

The third and fourth year of the PhD work was also active and resulted several publications and I have selected two of these, which are in the scope of the thesis.

Paper 8 shows a solution for controlling of a mechatronic system over internet. This should be interpreted as the basis of any kind of telemanipulation systems. The paper also shows solutions for educating of future engineers.

Paper 9 describes an in-house developed software framework, which is used in human-system interaction. This paper also has development and test results.

In the last year of the PhD work focus were put on refining the previous activities.

Paper 10 is a summery paper for interaction solutions inside Flexible Manufacturing Systems / Cells. This paper also includes a test case with a robotic arm, which is transformed to the newly proposed control architecture.

Paper 11 show the state-of-the-art in communicating with robots / manufacturing equipment.

Paper 12 is a test summary of achievements made with in-house developed software framework for human-system interaction.

1. Introduction

Flexible Manufacturing Systems are capable of manufacturing a wide variety of products with low volume size and minimal lead time. With these properties such a system is capable of responding to fast market and product changes. Also these manufacturing cells are usually equipped with high level computer based control and include highly flexible industrial robots [1]. For automated material handling an industrial robot is a very handy solution, however it needs special attention in programming of work tasks. Offline programming of industrial robot is difficult because it relies on very accurate system setups and the virtual programming environment must be carefully calibrated to capture the real-life setup and to avoid any changes of the robot program on the spot itself. These problems could be avoided by using online programming of principles, but during online programming the robot is unable to produce anything. This results a continuous demand for new and effective robot teaching methods. In the field of robotics the most challenging obstacle is that it takes approximately 400 times longer to program a robot in complex operations than to execute the actual task [2].

If we use the robot as a manipulator to extend our working capability we usually need complex and very expensive sensors or feedback devices, systems. Force or tactile sensors and feedback devices are particularly problematic and expensive. If we can develop a cheap and effective communication between the human operator and the robot, then we will be able to extend the profitable robotization fields. This requires new methodologies for programming and control of industrial robots and flexible manufacturing cells/systems.

The purpose of the work presented in this dissertation is to apply new cognitive info-communication channels for human-machine interaction to develop a new paradigm of robot teaching and supervising, which opens a new dimension of robotization. The robot is considered as an unskilled worker which is strong and capable for precise manufacturing. It has a special kind of intelligence but it is handicapped in some senses, hence it needs special treatment. We have to command it clearly in a special way and we have to supervise its work. If we can learn how to communicate with this "new worker" we can get a new capable "colleague". The long term goal is that the operator would be able to give the daily task to a robot in a similar way as he/she gives the jobs to the human workers for example, using CAD documentation and some verbal explanations.

The PhD work consists of three parts, the first deals with the problems of flexible manufacturing cells/systems (see Section 2). This part investigates todays existing solutions for controlling and supervising manufacturing cells/systems and proposes a solution for integrating outdated and new manufacturing components along with integration of any kind of real or virtual equipment in a standardized way. This part also features some readymade cell components that were developed in the duration of the work e.g. a vision system for old numerical controlled machines.

The second part deals with new possibilities for programming of industrial robots (see Section 3). This part introduces novel methods for programming robots e.g. vision systems combined with offline programming tools. This part along with the previous part introduces new standards (e.g. ISO 14646 STEP-NC) in the world of industrial robotics.

The third part shows the process of telemanipulation from a different viewpoint (see Section 4). This part introduces the so called cognitive commination channels in the telemanipulation process and also features state of the art review of human machine interaction in the scope of telemanipulation. For demonstration purposes different virtual environments were developed during the work and these are also discussed in this part. This part also includes some basic control theory solutions for telemanipulation and investigates solutions for education of this type of telemanipulation process.

In Figure 2 these processes and solutions are visualized in an overview picture. It is also visible from the figure that the PhD work deals with all aspects of the FMS.

- Communication between FMS components and communication inside components (Section 2)
- Production planning and automated production execution (Section 3)
- Task definition and human-system interaction (Section 4)

In the paper section of the thesis the contributing papers are presents in chronological order. The papers are referred from the overview sections (Section 2-4) and should only be read after these are referred to. All papers listed in the dissertation are published and could also be found online.

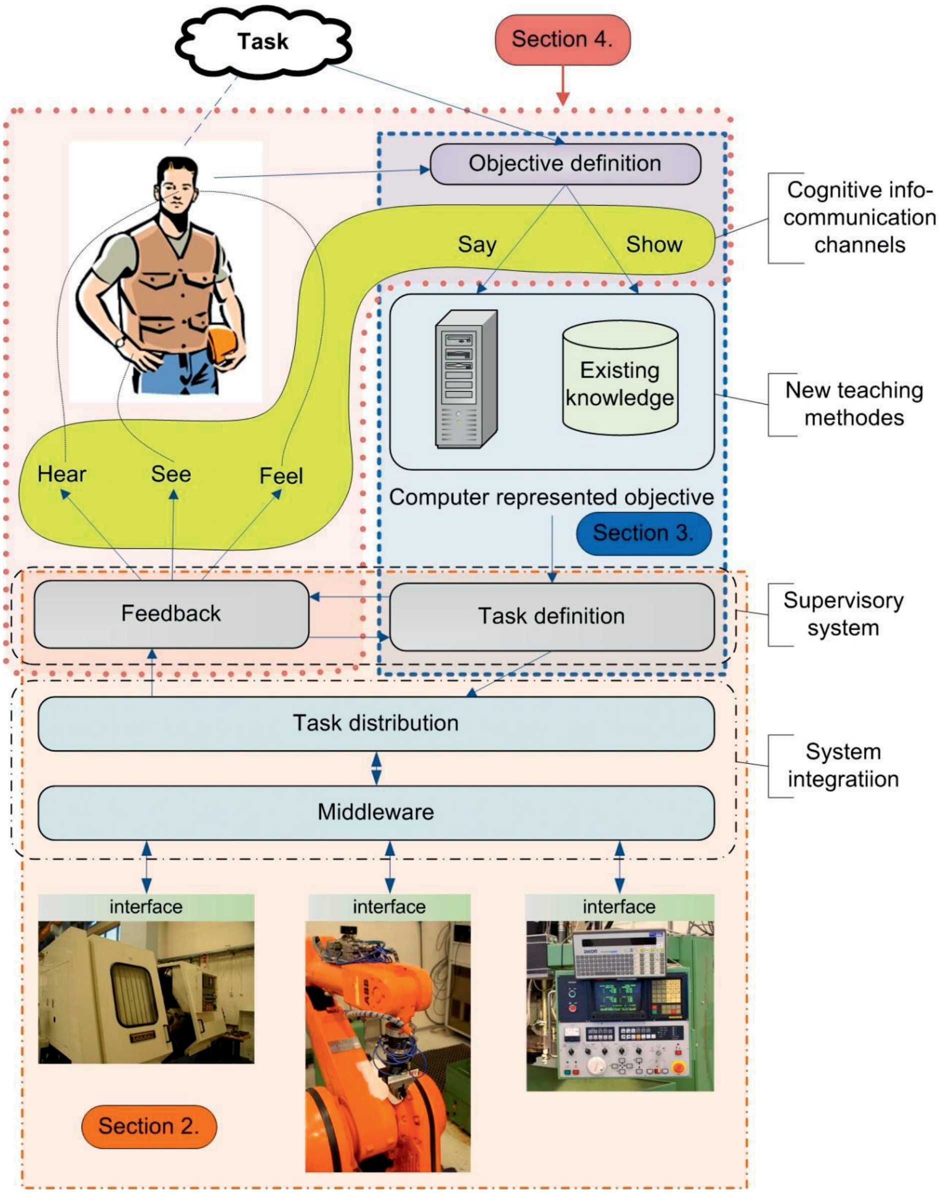


Fig. 2. Thesis structure and overview

2. Inter-machine and machine-machine communication in Flexible Manufacturing Cells / Systems

Parts of the material in this section are published in [3] and [4].

Today mass production would not exist without the usage of automated, flexible manufacturing cells. These cells usually consist of CNC machines, industrial robots, material storages, conveyor belts, etc. Each component in such cell has its own control architecture and communication link to other machines or humans.

Already in the 1960s, when the flexible manufacturing cells appeared, the autonomous operation of these cells was highly emphasized. However this was only achieved with careful planning of product, production and machine specific tasks by a highly qualified human, in other words by utilizing computer integrated manufacturing (CIM) [5].

In the 1970s the sustainability, fast response time to market changes and cost effectiveness made the utilization of CIM very hard [6]. This resulted that the concept of CIM reappeared from time to time in a new way: such as holonic manufacturing systems [7], agent-based manufacturing [8], virtual CIM [9], etc. The new CIM alternatives focus mainly on macro- and micro-process planning, without changing the common bottleneck of the whole manufacturing cell.

The key problem of most manufacturing cells is the lack of interoperability, program reuse and fast re-configurability. However there are standards to make the programming, planning of production coherent for any kind of machine, industrial robot or other equipment; these are outdated and not flexible at all. This problem statement will be investigated later on:

The standard for programming of CNC machines was originally developed in 1960s [10] and still used every day with all the bottlenecks of the standard [11]. To overcome these problems new standards are introduced [12] to replace and give a relief from the constraints. One of the most promising standards is the so called STEP (Standard for the Exchange of Product Model Data) or in other words ISO-10303 [13] and its combination with the STEP-NC (NC extension of original STEP) or in other words ISO-14649 [14]. These two standards are the largest and most widespread in the ISO group and are relatively new: STEP from 1994 and STEP-NC from 2003. Although these standards are already available for the manufactures more than 5 years ago, the industry is not seeming to change from the old standards. There were governmentally founded international projects [15]-[18] to promote these techniques among the manufacturers (e.g. Boeing, Daimler Chrysler, Volvo), but without much success. Also researchers from all over the world [11], [12], [19]-[24] are promoting and developing solutions for STEP and STEP-NC based machining.

But only dealing with one part of the whole manufacturing cell will not solve all the problems. Program portability from one vendor to other vendor's machine could be solved, but the inter-machine communication will be still limited to hard-wired Input/Output signaling.

To address the problems related to inter-machine communication and control a highly modularized architecture for shop floor controller is needed, where it is possible to "break down" the hardware/software components of the different machines and then carry out a task dependent assembly, to reach the goal of a specific work task. By using software based controller all components can be combined and reconfigured without any additional costs and this results utilization of components in a countless manner. As a result, increased system flexibility is achieved. The following sub sections deal with these kinds of communication possibilities.

Related papers: 5, 10

2.1. State-of-the-art in system integration (Middleware technology)

To manage a rapidly growing need for sensor communication in robotic applications several suitable architectures, commonly termed as middleware's, has been developed for easy system integration. Unfortunately, most of these middleware technologies are developed independently of each other and are often dedicated for specific user applications [25]. In Table 1. the most important participants of the robot middleware competition can be observed.

Table 1. Comparison of middleware technologies [4]

Name	Middleware technology	Open source?	Relevant contributors
ASEBA	CAN	Yes	EPFL
CLARAty	Multi-level mobility abstraction	Yes, but some restrictions	NASA
Microsoft RS	Web-Services	No	Microsoft
Miro	CORBA	Yes	University of California
Orca	ICE	Yes	KTH Stockholm
OrIN	DCON, SOAP, XML	Yes, but some restrictions	JARA
Open-R	Aperios OS	No	Sony
Orocos	RealTime Toolkit	Yes	Katholieke Universiteit Leuven
Player	Client / Server architecture	Yes	Multiple
RT-Middleware	CORBA	Yes	AIST
UPnP	HTTP, SOAP, XML	Yes	University of Coimbra
Urbi	Client / Server architecture	Yes	Gostai

Each of the above listed middleware solutions are composed from modularized components and have a hierarchical setup. The differences occur in the portability between different vendor's robots (how many robots are supported), the capability of adaptation to system changes (some are capable of plug-and-play discovery, others require system restart), the way of communication between system components and the programming environment (availability of IDE).

Almost every competitor is capable of hiding low level robot programming (motor drives, sensor value read out, camera image access, etc.) and provides standard interfaces for high level object oriented robot programming. Among its competitors, RT-Middleware is the only middleware solution that is under standardization and this solution has proved to be industry ready and used by many industrial partners (Toshiba (different system components), Honda (ASIMO humanoid robot), AIST (OpenHRP humanoid robot), etc.) and also many research institutes.

Based on the comparison of the different middleware technologies, the RT-Middleware platform was chosen as communication platform.

Related papers: 10

2.2. State-of-the-art in supervisory system

Manufacturing equipment undergoes continuous development as newer and better versions/solutions are pushed out into the market almost at a daily basis. To invest and keep pace with the newest technologies is only possible for a very few of the manufacturing companies. Small and medium sized enterprises (SME) cannot keep up with the investments, compared with their larger counterparts, and are facing special challenges.

During the last few decades much attention has been paid to offline programming (generation and transfer of the numerical control code (NC-code)) of CNC machines. There exist three older basic standards for the generation of the numerical control (NC) code namely: ISO 6983, DIN66025 and RS274D [26]. The NC data, from these standards, are often referred to as M and G codes. Reported weaknesses from the usage of M and G codes are, among others [27]:

1) Low level language merely presenting the cutter location (motion) without any reference to the work-piece geometry. Creates very long code sets where editing is almost impossible.
2) Vendor specific supplements to the M and G codes are often out of the limited scope of the standard and in such cases the NC code will not be inter-machine exchangeable.
3) One way data flow from design to manufacturing. Experiences from the shop-floor are difficult to push back to the design stage.

To overcome these shortcomings the ISO 14649 standard, often referred to as the STEP-NC standard, was derived from the initial work in the European Project OPTIMAL

(ESPRIT III 8643) by the WZL of Aachen University in the middle of the 1990s [23]. In general, the STEP-NC is a new object-oriented model for data transfer between computer aided design (CAD) and computer aided manufacturing (CAM) systems. STEP-NC data transfer specifies the steps of the machining process rather than the cutter location (as for ISO 6983 systems) [28]. These higher level commands, describing the actual machining process, are by far more comprehendible, informative and interchangeable than the G and M codes in use.

In order to transfer and/or distribute the generated machine code to the NC-machines a software solution named Direct Numerical Control or Distributed Numerical Control (DNC-software) has been developed by various manufactures. DNC programs are typically used to store larger NC program at separate computer, then the DNC will feed (upon request) the NC controller (which typically has a more limited memory capacity). Newer versions of DNC software features, among others, the possibility to edit the NC code and a simplified visualization of the manufacturing process.

Recent modern NC-machines have the possibility for inter machine (I/O) communication, typically targeted for communication with a shop-floor controller. However, this is a weaker feature in the history of development of NC-machines, compared to the development of offline programming tools and its appurtenant DNC software. Historically, in older machines there existed no prepared solution for inter-machine communication, so all the usual I/O signals (e.g. start spindle, stop spindle, open/close door/chuck) had to be captured individually from the controller by doing some-kind-of intervention on the NC controller itself. Also some NC codes (M-codes) were dedicated for I/O communication (e.g. wait until confirmation etc). In general, hardware intervention is a challenging task due to (very often) poor and limited documentation of the controllers. Re-engineering of a controller is time consuming and all hardware intervention tasks carry the possibility for damages. Delivery times/cost for damaged components leads also to the minimization of any hardware intervention tasks.

The industrial robot is another key component in a FMC/FMS setup. Typically, industrial robots are used for material handling, replacing the human operator from repetitive/hard work tasks. Normally, these robots have excellent capability for I/O communication and are especially adapted for integration with other cell members. Thereby, in many smaller FMC setups we will find that the industrial robot acts as a cell controller. However robots do not share the same language, even the same vendor often has different languages between different versions/types in their portfolio. In larger FMS system, with several robots doing different work tasks, the possibility that all robots are similar in kind and language will be very limited. In any case robots and NC-machines do not share the same language and software communication between these two groups of machines is often non-existent. Again, this leads to more specialized solutions to be derived from case to case and all communication is normally done on a hardware level with dedicated I/O ports where the typical NC-machine has limited functionality.

Related papers: 1, 11

2.3. Summary of achievements in Inter-machine and machine-machine communication in Flexible Manufacturing Cells / Systems

In a typical scenario at smaller production facilities, where old/outdate CNC machines are usually used along with newer machines there exists no standard no ready-to-use (standard) solution for communication between the members of the cell. Especially the outdated CNC machines are challenging as these can also lack any kind of communication interfaces. To solve this problem a vision system for rapid/simple/secure and low cost data retrieval from old CNC machines is presented. Along with this vision component also the possibilities for machine-machine communication in the shop-floor are investigated and a solution is proposed in **Paper 1**.

To address problems related to inter-machine communication, the STEP-NC standard is introduced to programming of industrial robots. A CAM system for industrial robot machining operations is introduced. Existing research related to STEP-NC development, as already showed in the previous sections; focus mainly on the typical implementation of the standard into traditional production equipment (e.g. CNC milling machines). However, industrial robots are getting more and more capable of taking onto machining operations and it is necessary to couple the industrial robot towards the new standard. Description of the CAM system and experimental results are discussed in **Paper 5**.

Dealing with machine-machine communication needs deeper investigation of relationship between manufacturing components in FMC/FMS. Normally, such cell members are from various manufactures and they have all their specific capabilities when it comes to man-machine and inter-machine interaction. Existing man-machine interfacing is quite simple and is typically done via a screen/keyboard interface while inter-machine communication is carried out through designated I/O lines. As already stated in the previous sections; there exists no ready-to-use solution for communication between the members of the cell, which fully utilizes the potential of the coordinated control of the complete cell. To deal with this problem a new architecture for man-machine and inter-machine communication and control is proposed. The shop-floor control architecture is based on the RT-middleware framework and the STEP-NC standard. The proposed architecture is general in its layout and emphasizes openness to the largest extent. Detailed description of the architecture, usability questions, experimental test with a robotic arm are presented in **Paper 10 and 11**.

3. Human-System Interaction in Flexible Manufacturing Cells / Systems

Parts of the material in this section are published in [29] and [30]

In manufacturing engineering, man-machine interaction has gone from typically online programming techniques into virtual reality based offline programming methodologies. Today, a wide range of offline software tools is used to imitate, simulate and control real manufacturing systems. However, also these new methodologies lack capability when it comes to human-machine communication. Today, communication with these programming environments is done via a keyboard/mouse interface while feedback to the operator is given through a computer screen. These desktop reality systems uses only a limited spectra of human senses and there is quite a low sensation of "being inside" the system. Programming is still done on the premises of the machines.

By introducing modern technologies, like motion capturing and augmented reality principles, with the goal of adding human representations into the programming environment it will be possible to create a new programming concept where the human operator can interfere with machines in a cognitive manner. Thus, the key-idea with the PhD work is to capture the knowledge of a skilled operator and make an automatic knowledge transfer to a manufacturing system. This allows a human operator to freely move around in an industrial environment (e.g. shop-floor) and identify the work-pieces which are necessary to modify via some interaction. Such a typical situation could be a robot grinding process following a molding process where the work-pieces suffer from some irregularities (burrs). By sight, the human operator can rather easily identify the area of problem, but cannot exactly quantify the error in term of necessary material removal to reach the ideal final geometry. In the following, these kinds of interaction possibilities will be presented.

Related papers: 2, 3, 6

3.1. Intelligent space as a new paradigm in human-system interaction

The Intelligent Space (iSpace) is an intelligent environment which provides both information and physical supports to humans and robots in order to enhance their performances [31]. In order to observe the dynamic environment, many intelligent devices, which are called Distributed Intelligent Network Devices, are installed into the room as shown in Fig. 3.

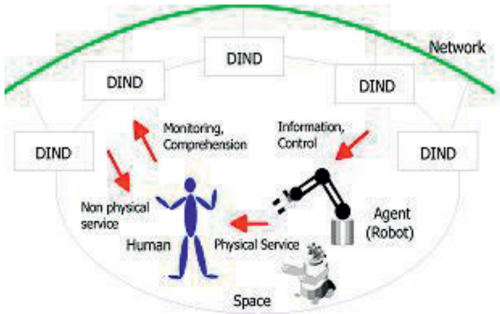


Fig. 3. Concept of the Intelligent Space [32]

The DIND is a fundamental element of the iSpace. It consists of three basic components including sensors, processors and network devices. By communicating with each DIND [33], iSpace can perceive and understand events in the whole space. In addition to observation, the iSpace actuates intelligent agents such as mobile robots, computer devices and digital equipment. Intelligent Space (iSpace) consists of three functions: "Observing", "Recognizing and Searching", and "Acting". Concretely, the iSpace offer appropriate service by understanding human intention and state in the space based on observation of space and human. In the Intelligent Space, the various sensor DINDs will be arranged such as cameras, laser range finders, microphones as well as these two sensing systems, and the information provided from them is vast. In order to enable humans to operate the agents in the iSpace, a suitable human interface is needed. For that reason, [34] has proposed the so-called "spatial memory" as an interface between human and the iSpace. The spatial memory system enables human users to store computerized information into the real world by assigning a three-dimensional position to the information, and to retrieve the information by directly indicating the point using their own hands. Therefore, by using the spatial memory, human users are able to smoothly implement and utilize information and services in the iSpace by themselves. On the other hand, when we consider a situation in which the iSpace try to provide suitable services for humans autonomously, observation of human activities is more important. Human activities have been observed and analyzed through usage history of the spatial memory because information spatially arranged can be regarded as description of human activities in a certain environment. The observation approach provides us to distinguish human activities even in a same area without any knowledge about human activities. However, the approach cannot take into account interaction between humans and physical objects except for iSpace agents in the observation, because the spatial memory hasn't had a function to associate the stored information to physical objects but to three-dimensional positions. Meanwhile, we can assume that physical objects, which are used for users' daily life and used to accomplish their activities, are important to describe or estimate human activities. Therefore, we focus on physical interaction between humans and physical objects in order to more precisely and widely describe human activities. There are several researches focusing on determination of physical objects. [35] estimate sequence of humans' cooking activities based on sequence of physical objects used. [36] have developed a language learning system which provides suitable information for users according to the position of physical objects. From these works, we can say that determination of physical objects and their position are useful to estimate human activities in the real world.

Related papers: 4

3.2. Summary of achievements in Human-System Interaction in Flexible Manufacturing Cells / Systems

Driven by the need for higher -flexibility and -speed during initial programming and path adjustments, new robot programming methodologies quickly arises. The traditional “Teach” and “Offline” programming methodologies have distinct weaknesses in machining applications. “Teach” programming is time consuming when used on freeform surfaces or in areas with limited visibility /accessibility. Also, “Teach” programming only relates to the real work-piece and there is no connection to the ideal CAD model of the work–piece. To overcome these problems a programming methodology especially targeted for machining applications is proposed. This methodology use a single camera combined with image processing, combines information of the real and ideal work-piece and represents a human friendly and effective approach for robot programming in machining operations. This methodology along with challenges and possible solutions for effective communication between the human and the robot system is introduced in **Paper 2 and 3**.

In conventional robot system development, the different robot parts (sensors, processing elements and actuators) are combined together in a compact, self-contained system. Both type of robot system (industrial or service type) faces challenges in the sense of flexibility, reusability and new part integration. Robot systems usually consist of sensors that provide information about the environment, computational units that process information about the environment and actuators that are controlled according to the decisions made by the computational units. As already shown in the previous section in robot systems one of the most important sensors are image sensors that provide visual information about the environment. For this reason an image processing framework (Distributed Image Analyzer) and its integration into RT-Middleware is introduced. Image processing modules can easily be loaded to RT-Middleware environment and provides an image processing and vision toolbox for building complex robot systems. If a simple image processing system is designed and implemented using the Distributed Image Analyzer toolbox in RT-Middleware, its advantages become apparent. Until now, vision components in robots were highly integrated to robot systems, without the possibility of reuse and easy adjustment. As a special kind of vision system, vision based observation of old NC machines is introduced in **Paper 4**.

Virtual Environments allow users to interact with virtual worlds, but usually these interactions are in front of a monitor or a projected wall. Augmented Reality brings the feeling of reality to these interactions. The user can combine the virtual world with the real world. Interact and feel the objects, together with artefacts that are generated by a computer. Problems of Augmented Reality systems, current technological developments and usability bottlenecks of existing systems and probable solutions are discussed in **Paper 6**.

4. Telemanipulation as part of the Human-System Interaction

Parts of the material in this section are published in [37].

In general telemanipulation refers to extending the human's (operator's) sensing and acting capabilities to a remote place, where some manipulation task is to be executed. The telemanipulation process so thus the telemanipulation system can be divided into three major parts (see in Figure 4): the master device, which creates the connection between man and machines, the slave device, which does the work at the remote environment and the information transmitter channel between the master and slave device [38].

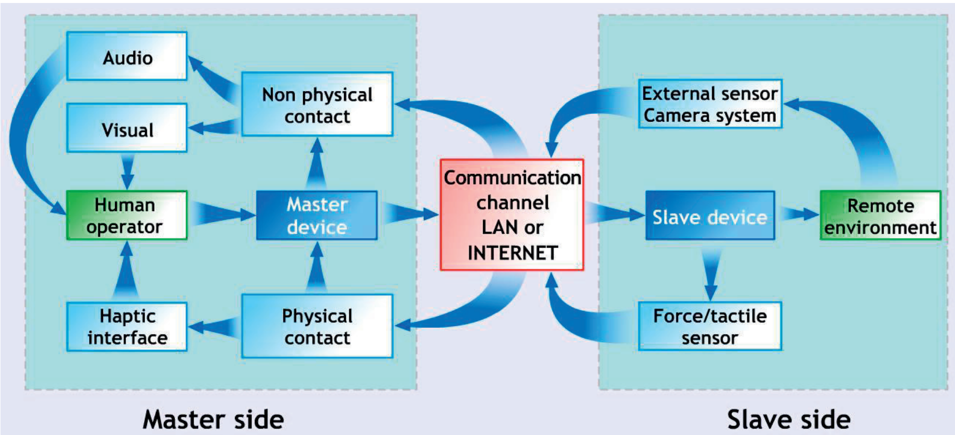


Fig. 4. Information streams of the Telemanipulation [37]

Today, in production engineering we have already taken into usage complex virtual representations of the manufacturing environment (telemanipulation systems) where we can program, simulate, analyze and optimize key performances. Such systems are mainly working through keyboard/ mouse input while user feedback normally are direct visual through computer screen and text files. Such communication relies on a limited spectra of our senses and the traditional keyboard/mouse/screen systems are not sufficiently human friendly, especially when the operator (master side) is expected to operate "out of office" in an unstructured, maybe dangerous or dirty, environment (slave side). In the near future, we expect to see more advanced input/output devices like motion capturing and speech input systems make their way into production environment while feedback will not be visual alone but include several other human senses. Combinatorial sensory information, interpreted by the human brain, becomes very handy when a standalone sense is not enough to interpret the actual situation.

Imitation of senses and feedback from virtual reality environments has always been a challenging topic. Realization of two (sight and hearing) out of the five human senses is indispensable. Simulation of these two senses is not complex issue; on the contrary, the imitation of the remaining three is quite challenging [39].

Related papers: 8, 9

4.1. State-of-the-art in cognitive info-communication channels for human-system interaction

Humans have various biological sensors that are capable of detecting direct physical information. However, human sensory mechanisms make use of a very complex combination of these biological sensors and cognitive processing mechanisms, which rely to a great extent on pre-learned skill and intelligence. We may define the output of this cognitive process as the understanding, rather than sensing, of the environment. Such complex cognitive sensory processes which are nearly impossible to break down into subcomponents, and attribute to different sensory organs, are often referred to as the 6^{th} or 7^{th} senses. These sensory behavioural types are the subject of very active research in the field of cognitive science. These researches recovered that when a human being talk to another human being less than 50 % of the total information is transmitted by the words. The majority of the information is transmitted by non-verbal meta communications like pronunciation, gesture, mimic, body language. This is the reason why smileys are so popular in the written communication [40].

When we look around at home, we can recognize the well-known environment immediately, but we need longer time to identify a strange object. This is in accordance with the claim, that only 10% of the total information are coming through our eyes in the recognition process. The 90% of the total information have already been in our brain as a result of a learning process.

To illustrate the problem statement, we define the human face as a special cognitive channel which is among the most complex channels with the highest dimensionality. Our brain has a special module and representation technique for the task of recognizing human faces and facial expressions. For instance, when we are walking on the street and someone accidentally hits us with his/her shoulder, we turn back and look into his/her face and we can quickly and easily decide whether or not he/she is angry. It is clear how dense and extremely complex this information is, and it is processed almost instantaneously. In accomplishing such tasks, we use our own sensors that have been trained ever since the very beginning of our life. This is the reason why we like to use cameras or facial pictograms in our communications (more than a 70% of our communication is supported by our face). An equally clear example is the use of facial expressions in computer games to depict the player’s status on the screen instead of providing a number of details [41].

Related papers: 7, 11 and 12

4.2. Summary of achievements in Telemanipulation as part of the Human-System Interaction

As already stated in the previous section, the imitation of senses and feedback from virtual reality environments always meant a great problem. It should highlighted that in many cases humans can learn and connect their understanding to the physical measurement system, however, to find the best channel for this communication is not easy. Such channels were already recognized; by using cognitive info-communication channels for these purposes. For instance, an operator who has a good sense of hearing can use his/her ears rather than eyes to understand distances. **Paper 7** describes the development of a vibro-tactile glove which can provide sensory feedback from a virtual environment, either as a stand-alone system but most important in combination with sight and audio feedback systems. Instead of implementing real force feedback, the focus is on tactile sensing, as an alternative way of achieving the same feedback.

Control of a mechatronic system through the internet is the basis of the telemanipulation. This involves interfaces for the human, the mechatronic system and description of communication rules between these. **Paper 8** describes an educational program of dc servo drives for distant learning. The program contains three parts: animation, simulation, and Internet-based measurement.

Telemanipulation of a mobile robot through the internet, which also includes simulation and visualization of the movement, needs coordination, synchronization and prediction on master and slave side. To solve this complex problem, a multi-layer mobile robot controller unit has been created. The system utilizes advanced human system interaction through handling a motion capturing suit and adapts it similarly as traditional peripheries. Robust posture recognition has been introduced on top of the motion suit adapter, which is used to instruct a mobile robot agent, while immerse stereographic feedback is provided to the human operator. This tool, which is an in extension of the traditional human-system interaction loop, is presented in **Paper 9**.

In order to utilize the full potential of industrial robots its human operators need to gain a deep technical understanding of rather complex mechatronic devices. However, such knowledge does not come easily, therefore, the goal is to develop various cognitive communication channels that can be easily combined under the RT-middleware platform and fit to the robot sensor system according to the special capabilities of the operator (as a matter of fact the operator should need practice in order to use the communication channel to perfection). Practical implementations of the main categories of interaction components; input systems, output systems and sensors are introduced in **Paper 11**.

Navigation in virtual, simulated environment encounters more challenges than in a planar, window based case. While there exists well established framework, tested solutions for window based environment these solution usually fail in simulated environment. For human beings the feeling of depth is a unique feature and simulated environments are the most appropriate way to take advantage out of it. In **Paper 12** the investigation of the problem of navigation in simulated immersive environment and usability study of an in-house developed software solution is described.

5. References

[1] P. Radhakrishnan, S Subramanyan and V. Raju, “CAD/CAM/CIM”, New Age International, New Delhi, 2008.

[2] T. Thomessen, P. K. Sannæs, and T. K. Lien, “Intuitive Robot Programming,” in Proc. 35th International Symposium on Robotics, 2004

[3] Bjorn Solvang, Gabor Sziebig, Peter Korondi, Shop_Floor Architecture for Effective Human-Machine and Inter-Machine Interaction. ACTA POLYTECHNICA HUNGARICA 9:(1) pp. 183-201, 2012

[4] Bjørn Solvang, Gabor Sziebig, Peter Korondi, Multilevel Control of Flexible Manufacturing Systems. In: Proc. IEEE Conference on Human System Interactions (HSI'08). Krakow, Poland, pp. 785-790., 2008

[5] J. Harrington, Computer Integrated Manufacturing, New York: Industrial Press Inc, 1973.

[6] J. Johansen, U.S. Karmarkar, D. Nanda and A. Seidmann, Business experience with computer integrated manufacturing. A survey of current strategy and practice, In Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, Vol. 4, pp. 970-979, 1995.

[7] J. Christensen, Holonic Manufacturing systems-initial architecture and standards directions, In Proceedings of the first European conference on holonic manufacturing systems, 1994.

[8] D. Ouelhadj, C. Hanachi and B. Bouzouia, Multi-agent system for dynamic scheduling and control in manufacturing cells, In Proc. IEEE International Conference on Robotics and Automation, pp. 2128–2133, 1998.

[9] D. Wang, S. V. Nagalingam and G. C. I. Lin, Development of an agent-based Virtual CIM architecture for small to medium manufacturers, Robotics and Computer-Integrated Manufacturing, Vol. 23, No. 1, pp. 1-16, 2007.

[10] International Organization for Standardization (ISO), ISO 6983-1, Numerical control of machine-program format and definition of address words Part-1: Data format for positioning, line and contouring control systems, First Edition, 1982.

[11] X. W. Xu and Q. He, Striving for a total integration of CAD, CAPP, CAM and CNC, Robotics and Computer-Integrated Manufacturing, Vol. 20, No. 2, pp. 101-109, 2004.

[12] S.T. Newman, A. Nassehi, X.W. Xu, R.S.U. Rosso Jr., L. Wang, Y. Yusof, L. Ali, R. Liu, L.Y. Zheng, S. Kumar, P. Vichare and V. Dhokia, Strategic advantages of interoperability for global manufacturing using CNC technology, Robotics and Computer-Integrated Manufacturing, Vol. 24, No. 6, pp. 699-708, 2008.

[13] International Organization for Standardization (ISO), ISO 10303-1, Industrial automation systems and integration - Product data representation and exchange Part 1: Overview and fundamental principles, 1994.

[14] International Organization for Standardization (ISO), ISO 14649-1, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 1: Overview and fundamental principles, 2003.

[15] ESPRIT Project 8643, Optimised preparation of manufacturing information with multi-level CAM-CNC coupling (OPTIMAL): final report, ESPRIT, 1997.

[16] ESPRIT Project 29708, STEP-compliant data interface for numeric controls, ESPRIT, 1999.
[17] IMS Project 97006, STEP-compliant data Interface for Numerical Controls, STEP-NC Consortium, 2003.
[18] National Institue of Standards and Technology (NIST) 99-01-4035, Model Driven Intelligent Control of Manufacturing, STEP Tools Inc., 1996.
[19] S.T. Newman and A. Nassehi, Universal Manufacturing Platform for CNC Machining, Annals of the CIRP, Vol. 56, No. 1, pp. 459-462, 2007.
[20] A. Nassehi, S.T. Newman and R.D. Allen, STEP-NC compliant process planning as an enabler for adaptive global manufacturing, Robotics and Computer-Integrated Manufacturing, Vol. 22, No. 5-6, pp. 456-467, 2006.
[21] A Nassehi, S T Newman, X W Xu, R D. Allen and R S U Rosso Jr., Adaptability and Interoperability in CNC Manufacturing, In Proceedings of the 3rd CIRP international conference on Digital Enterprise Technology, pp 1-8, 2006.
[22] Xun Xu, STEP into Distributed Manufacturing with STEP-NC, In book Process Planning and Scheduling for Distributed Manufacturing, pp. 393-421, Springer London, 2007.
[23] S.H. Suh, B.E. Lee, D.H. Chung and S.U. Cheon, Architecture and implementation of a shop-floor programming system for STEP-compliant CNC, Computer-Aided Design, Vol. 35, No. 12, pp. 1069-1083, 2003.
[24] S. M. Amaitik and S. E. Kilic, An intelligent process planning system for prismatic parts using STEP features, International Journal of Advanced Manufacturing Technology, Vol. 37, No. 16, pp. 978-993, 2007.
[25] Tetsuo Kotoku, "Robot Middleware and its Standardization in OMG" in International Conference on Intelligent Robots and Systems (IROS'06) Workshop on Robotic Standardization, Beijing, China, 11-13 Oct. 2006
[26] Y. Liu, X. Guo, W. Li, K. Yamazaki, K. Kashihara, M.Fujishima, "An intelligent NC program processor for CNC system of machine tool," in Robotics and Computer-Integrated Manufacturing, vol. 23, Elsevier, 2007, pp. 160–169
[27] X. Zhu, Y. Wang, H. Fu, "A 3-D Simulation System for Milling Machining Based STEP-NC" in Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, Dalian, China, 2006
[28] X.W. Wu, "Realisation of STEP-NC enabled machining" in Robotics and Computer-Integrated Manufacturing, vol. 22, Elsevier, 2006, pp. 144–153
[29] Peter Zanaty, Peter Korondi, Gabor Sziebig, Laszlo A Jeni, Image based automatic object localisation in iSpace environment. In: 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2009). Budapest, Hungary, Budapest: pp. 501-512. Paper 43., 2009
[30] Bela Takarics, Gabor Sziebig, Peter Korondi, Small Batch Size Robot Programming with Human in the Loop. JOURNAL OF AUTOMATION MOBILE ROBOTICS & INTELLIGENT SYSTEMS 3:(4) pp. 228-231., 2009
[31] J.-H. Lee and H.Hashimoto, "Intelligent Space - concept and contents", Advanced Robotics, Vol. 16, No. 3, pp. 265–280, 2002.
[32] K. Morioka, J-H Lee and H. Hashimoto, Mobile Robot Control for Human Following in Intelligent Space, Journal of the Robotics Society of Japan, Vol. 22, No. 1, pp. 103-111, 2004.

[33] Yoshihisa Toshima, Noriaki Ando and Hideki Hashimoto, "Information Display System using Active Projector in Intelligent Space -Integration of distributed devices based on RT-Middleware", The 12th International Symposium on Artificial Life and Robotics (AROB'07), pp.183-186, 2007
[34] Mihoko Niitsuma, Hiroshi Hashimoto, Hideki Hashimoto, "Spatial Memory as an Aid System for Human Activity in Intelligent Space", IEEE Transactions on Industrial Electronics, Vol.54, No.2, pp.1122-1131, 2007
[35] Y.Nakauchi, T.Fukuda, K.Noguchi, T.Matsubara, “Intelligent Kitchen: Cooking Support by LCD and Mobile Robot with IC-Labeled Objects”, Proc. of the 2005 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems, pp.2464-2469, 2005
[36] Y. Nishida, M. Hiramoto, F. Kusunoki, H. Mizoguchi, “Learning by Doing: Space-Associate Language Learning Using a Sensorized Environment”, in Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp. 1583-1588, 2005
[37] Gabor Sziebig, Bela Takarics, Viliam Fedak, Peter Korondi, Virtual Master Device. In: Proc. 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI'07). Poprad, Slovakia, 25/01/2007-26/01/2007. pp. 29-40. Paper 4.
[38] P. Korondi, P. Szemes, H.Hashimoto, ”Internet based Telemanipulation” Chapter in R. Zurawski (Ed.): "The Industrial Information Technology Handbook", pp. 60-1-60-25, CRC Press, ISBN 0-8493-1985-4, June 2004
[39] P. Galambos, A. Roka, P. Korondi, "Ergonomic Vibrotactile Feedback Design for HMI Applications." In Proc. of 11th International Carpathian Control Conference, pp. 31-34, 2010.
[40] P. Baranyi, G. Németh, P. Korondi, 3D Internet for Cognitive Infocommunication. INFORMATIKA - A GÁBOR DÉNES FŐISKOLA KÖZLEMÉNYEI 12:(2) pp. 3-6., 2010.
[41] P. Baranyi and A. Csapo, “Cognitive Infocommunications: CogInfoCom”, 11th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, pp. 141-146 2010.

Papers

From this point the included papers are presented. These papers are the most important part of the dissertation itself and should be considered as the main contributions. Each paper starts with a short description of the co-authorship in the given paper.

Paper 1: Multilevel Control of Flexible Manufacturing Systems

Bjørn Solvang, Gabor Sziebig, Peter Korondi, Multilevel Control of Flexible Manufacturing Systems. In: *Proc. IEEE Conference on Human System Interactions (HSI'08)*. Krakow, Poland, 25/05/2008-27/05/2008. (IEEE) pp. 785-790.(ISBN: 1-4244-1543-8)

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Gabor Sziebig and Peter Korondi.

The state-of-the-art survey and literature review on middleware technologies is module is entirely a contribution from Gabor Sziebig.

The development and testing of the supervision module is entirely a contribution from Gabor Sziebig. The tests were carried out at the production facilities at Narvik University College.

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing Section 2. A, 2 B, partly 2C and entirely 3 and produced the graphics and pictures through the paper. Gabor Sziebig's written contribution was thoroughly reviewed and revised by Peter Korondi.

Multilevel Control of Flexible Manufacturing Systems

Bjørn Solvang †, Gabor Sziebig †, and Peter Korondi ‡, *Members, IEEE*
† Narvik University College, Department of Industrial Engineering, Narvik, Norway (phone: +47 76 96 62 27; fax: +47 76 96 6815; e-mail: bjs@hin.no), ‡ Budapest University of Technology and Economics, Department of Automation and Applied Informatics, Budapest, Hungary (phone: +36 1 463 2184; fax: +36 1 463 3163; e-mail: korondi@elektro.get.bme.hu)

***Abstract* — Typical manufacturing equipment for small and medium scale production is arranged in a setup of a Flexible Manufacturing Cell (FMC). Normally, advanced manufacturing machinery in a FMC are individually well capable and equipped for offline programming. However, in terms of coordinated control between these machines there is a lack of capability. Especially in smaller production facilities, where the FMC components often are a mixture of older (poorly documented) and newer machines, there exist no ready-to-use (standard) solution for communication between the members of the cell.**

This paper presents an architecture for inter-machine communication and control based on an existing middleware framework. Further, and in detail, a vision system for rapid/simple/secure and low cost data retrieval from old computerized numerical control (CNC) machines is presented as a system key component. The architecture and vision system is general in its layout and can be utilized on a large majority of the components in flexible manufacturing systems.

***Keywords* — flexible manufacturing systems, inter-machine communication, machine vision, RT-Middleware, shop floor control**

I. Introduction

MANUFACTURING equipment undergoes continuous development as newer and better versions/solutions are pushed out into the market almost at a daily basis. To invest and keep pace with the newest technologies is a possibility of only a very few of the manufacturing companies. Small and medium sized enterprises (SMEs) cannot keep up with the investments, compared with their larger counterparts, and are facing special challenges.

During the last few decades much attention has been made to offline programming (generation and transfer of the numerical control code (NC-code)) of CNC machines. There exist three older basic standards for the generation of the numerical control (NC) code namely: ISO 6983, DIN66025 and RS274D [1]. The NC data, from these standards, are often referred to as M and G codes. Reported weaknesses from the usage of M and G codes are, among others [2]:

1) Low level language merely presenting the cutter location (motion) without any reference to the work-piece geometry. Creates very long code sets where editing is almost impossible.

2) Vendor specific supplements to the M and G codes are often out of the limited scope of the standard and in such cases the NC code will not be inter-machine exchangeable.

3) One way data flow from design to manufacturing. Experiences from the shop-floor are difficult to push back to the design stage.

As to overcome these shortcomings the ISO 14649 standard, often referred to as the STEP-NC standard, was derived from the initial work in the European Project OPTIMAL (ESPRIT III 8643) by the WZL of Aachen University in the middle of the 1990s [3]. In general, the STEP-NC is a new object-oriented model for data transfer between computer aided design (CAD) and computer aided manufacturing (CAM) systems. STEP-NC data transfer specifies the steps of the machining process rather than the cutter location (as for ISO 6983 systems) [4]. These higher level commands, describing the actual machining process, are by far more comprehendible, informative and interchangeable than the G and M- codes in use.

In order to transfer and/or distribute the generated machine code to the NC-machines a software solution named Direct Numerical Control or Distributed Numerical Control (DNC-software) has been developed by various manufactures. DNC programs are typically used to store larger NC program at separate computer, then the DNC will feed (upon request) the NC controller (which typically has a more limited memory capacity). Newer versions of DNC software features, among others, possibility to edit the NC code and a simplified visualization of the manufacturing process [5].

Recently modern NC-machines have the possibility for inter machine (I/O communication), typically targeted for communication with a shop-floor controller. However, this is a weaker feature in the history of development of NC-machines, compared to the development of offline

This work was supported by the Narvik University College and the Budapest University of Technology and economics. The authors also wish to thank the National Science Research Fund (OTKA K62836), Control Research Group and János Bolyai Research Scholarship of Hungarian Academy of Science for their financial support.

programming tools and its appurtenant DNC software. Historically, in older machines there existed no prepared solution for inter-machine communication, so all the usual I/O signals (e.g. start spindle, stop spindle, open/close door/chuck) had to be captured individually from the controller by doing some-kind-of intervention on the NC controller itself. Also some NC codes (M-codes) were dedicated for I/O communication (e.g. wait until confirmation etc). In general hardware intervention is a challenging task due to (very often) poor and limited documentation of the controllers. Re-engineering of a controller is time consuming and all hardware intervention tasks carry the possibility for damages. Delivery times/cost for damaged components leads also to the minimization of any hardware intervention tasks.

The industrial robot is another key component in a FMC/FMS setup. Typically, industrial robots are used for material handling, replacing the human operator from repetitive/hard work tasks. Normally, these robots have excellent capability for I/O communication and are especially adapted for integration with other cell members. Thereby, in many smaller FMC setups we will find that the industrial robot acts as a cell controller. However robots do not share the same language, even the same vendor often has different languages between different versions/types in their portfolio. In larger FMS system with several robots doing different work tasks the possibility that all robots are similar in kind and language will be very limited. Anyway robots and NC- machines do not share the same language and software communication between these two groups of machines is often non-existent. Again this leads to more specialized solutions to be derived from case to case and all communication is normally done on a hardware level with dedicated I/O ports where the typical NC-machine has limited functionality.

As mentioned above, there exist several problems of setting up a scheme for coordinated control between the members in a manufacturing system/cell due to the great diversity in software and hardware capabilities in the machining park of a normal SME. To meet these challenges, this paper presents a general architecture for inter-machine communication and control based on the existing RT(Robot Technology)-Middleware framework. Further and especially a vision system for easy data retrieval from old NC machines is focused in more detail. The proposed architecture and vision system is general in its layout and can be utilized on a large majority of the key components in manufacturing systems (mainly CNC machines and various robot systems).

The organization of the paper is as follows: Section II gives an overview and layout of the RT-Middleware based framework and its implementation in FMS while Section III presents details of some system components with focus onto the module for vision based digital control. Section IV concludes the paper.

II. System Overview

To address the problems related to inter-machine communication and control this section presents a highly modularized architecture for shop floor controller where it is possible to "break down" the hardware/software components of the different machines and then carry out a task dependant assembly, to reach the goal of a specific work task. By using software based controller all components can be combined and utilized in a countless manner. As a result increased system flexibility is achieved. After a comparison of different kind of middleware solutions, a detailed introduction to RT-Middleware is given, which is the chosen framework for the proposed architecture.

A. State-of-art in middleware techniques

To manage a rapidly growing need for sensor communication in robotic applications several suitable architectures, named middleware's, is being developed for easy system integration. Unfortunately, most of these middleware technologies are developed independently of each other and are often dedicated for specific user applications [6]. In Table 1. the most important participants of the robot middleware competition can be observed.

Table 1: Comparison of Middleware Technologies

Name	*Middleware technology*	*Open source?*	*Relevant contributors*
ASEBA [7]	CAN	Yes	EPFL
CLARAty [8]	Multi-level mobility abstraction	Yes, but some restrictions	NASA
Microsoft RS [9]	Web-Services	No	Microsoft
Miro [10]	CORBA	Yes	University of California
Orca [11]	ICE	Yes	KTH Stockholm
OrIN [12]	DCON, SOAP, XML	Yes, but some restrictions	JARA
Open-R [13]	Aperios OS	No	Sony
Orocos [14]	RealTime Toolkit	Yes	Katholieke Universiteit Leuven
Player [15]	Client / Server architecture	Yes	Multiple
RT-Middleware [16]	CORBA	Yes	AIST
UPnP [17]	HTTP, SOAP, XML	Yes	University of Coimbra
Urbi [18]	Client / Server architecture	Yes	Gostai

Each of the above listed middleware solutions are composed from modularized components and have a hierarchical setup. The differences occur in the portability between different vendor's robots (how many robots are supported), the capability of adaptation to system changes (some are capable of plug-and-play discovery other need system restart), the way of communication between system components and the programming environment (contains integrated development environment or not).

Almost every competitor is capable of hiding low level robot programming (motor drives, sensor value read out, camera image access, etc.) and provides standard interfaces for high level object oriented robot programming.

RT-Middleware was chosen as the platform in the proposed work, because it is the only middleware solution that is under standardization [19]. This solution has proved to be industry ready and used by many industrial partners (Toshiba (different system components), Honda (ASIMO humanoid robot), AIST (OpenHRP humanoid robot), etc.) and also many research institutes [16].

B. RT-Middleware

In 2002, the Japanese Ministry of Economy, Trade and Industry (METI), the Japan Robot Association (JARA) and National Institute of Advanced Industrial Science and Technology (AIST) started a project named "Consolidation of Software Infrastructure for Robot Development". With the intention of implementing robot systems to meet diversified users' needs by making robots and their functional parts in a modular structure, and further allow system designers or integrators building versatile robots or systems with relative ease by simply combining selected modular parts [20]. In Fig. 1 the conceptual idea of this middleware technology is introduced.

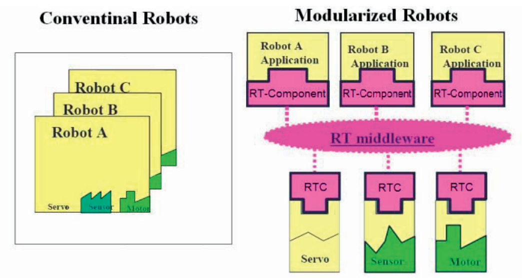


Fig. 1. The basic idea of RT-Middleware. Components are integrated through the RT-Middleware using a standard interface [20].

Originally RT-Middleware was designed for the development of complex robot systems and in particular humanoid robots. As can be seen in Fig. 1 the framework typically combines low level RT-Components (RTCs), like the motor drive elements (motor, servo and sensor), and advanced machines (like the humanoid) is made- up from an assembly of these middleware components. As a tool for RTC assembly a software solution named OpenRTM-aist is available. OpenRTM-aist includes a tool for making RTCs, a tool for combining existing RTCs into a new RTC, and a graphical user interface for linking and setting up the system functionality. The communication between RT-Components is realized with CORBA objects that are called InPort / OutPort. This hierarchical structure and the CORBA system provide the power of OpenRTM-aist. CORBA provides transparency of all levels of programming, from object location to protocol transparency. This means that the client doesn’t need to know where an object is located; he only knows that it is reachable for the program at any time. The usage of CORBA in shop floor control is found favorable by several other research studies, [21] represent as an example of those.

The scope of this paper is to narrow to introduce all details about the RT-Middleware framework. More can be found in references [16] [22].

C. Concept

The RT- Middleware framework could easily be adapted and extended to include the typical components of an FMC(S) system like industrial robots, CNC machines, automated guided vehicles (AGVs) etc. Preferably these machines should be “broken down” to the lowest system level (in general to the motor controller) and RT-Middleware (device drivers) developed, followed by a task dependant assembly and programming. This will create a truly flexible system where it is possible to control all the actions within machines and their interactions with the other members. It would be possible to merge two or more machines and achieve a full coordinated control. As an example a 6- axis robot could be merged with a 3- axis NC-machine into a 9 axis machine with the possibility of a synchronous machining/ handling of the work piece.

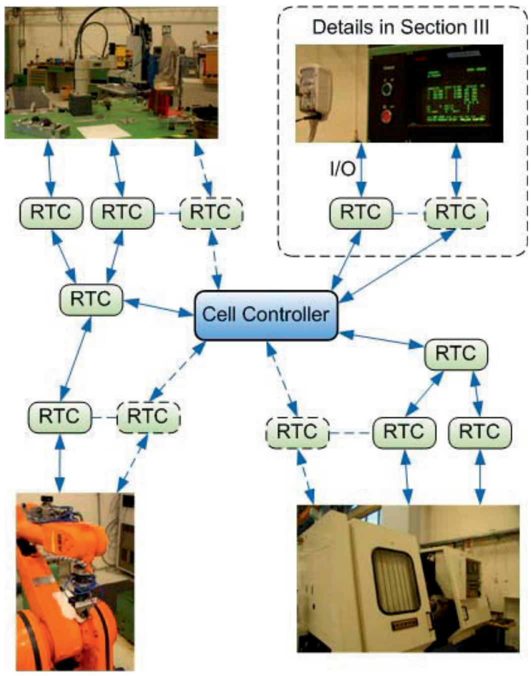


Fig. 2. Component based system setup

However a complete system “break down” will, in many cases, be very challenging to achieve because of week documentation in older machines. In such cases the system “break down” and the development of the device drivers should be stopped at a level where the necessary data can be retrieved. Fig. 2 shows the conceptual layout of a FMC system were the suggested RT-Middleware components exist in different levels. RTCs can be merged to become a higher level RTC or it can be used “as is” in the cell controller (OpenRTM-aist module). As the work task requirements changes the structure can be remodeled and if necessary a deeper system intervention and development of lower level RTCs could be carried out.

By using a CORBA based software controller (also introduced by J. Shin, S. Park, C. Ju, H. Cho [21]), and device drivers (also introduced in the comprehensive work of H. Van Brussel and P. Valckenaers [23]) the suggested architecture will be higly flexible and suitable for low series production in the SME sector.

III. Module for Supervision

In older NC machines there exist no prepared solution for inter-machine communication, so all the necessary I/O communication have to be captured individually based on hardware intervention . Even though some machines might be able to communicate with Personal Computers (PCs), it is typically a one way communication and only used for program uploading/downloading. To monitor the state of a manufacturing process an operator should always observe the screen (often low quality monitors) of the NC machine.

This operator can now be replaced with a vision system, which can supervise (monitor any changes on the screen of the machines) and also provide input for digital/analog communication with the other elements in flexible manufacturing systems. Fig. 3 show an NC machine with a camera mounted in front of the screen.

Fig. 3. Camera observing NC machine

Optical Character Recognition is used to detect state changes (position, distance to go, current line of program, warning messages, etc.) in NC machines. Because of low resolution of screens, the best solution for character recognition is based on artificial neural networks. Neural networks are capable of handling uncertainties occurring because of screen quality, lightning instability and other camera related problems.

In the following subsection the structure and realization of the module for supervision will be introduced.

A. Structure

The module structure can be observed in Fig. 4. an image from the digital camera is transferred to a PC, where the Optical Character Recognition is executed and the result from this stage can be transferred to any kind of equipment that can be connected to a PC. Before the module can be used a pre-setup must be done. This setup is only needed once by each FMC component. The setup phase consists of the following parts:

1) Training of Artificial Neural Network with learning data
2) Selection of a “Region of Interest” in the picture
3) Defining desired output of module

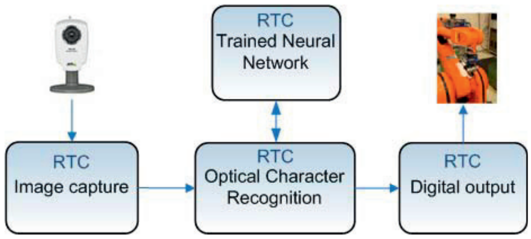


Fig. 4. Module structure

The structure is realized in a manner of robustness and flexibility. Every part of the structure can be replaced; even the neural network can be customized to the specific FMC component (number of layers, number of neurons, weights, learning rate, etc.). Also system outputs are configurable; both digital and analog values can be obtained and transferred for further treatment in the cell controller. All system components should be realized as RTC components as indicated in Fig. 4.

B. Realization

For training of the neural network a back propagation learning algorithm and for activation the bipolar sigmoid function (1) was used [24]:

$$f(x) = \frac{2}{1+e^{-\alpha * x}} - 1$$
$$f(x)' = \frac{2 * \alpha * e^{-\alpha * x}}{(1+e^{-\alpha * x})^2} = \alpha * \frac{1-f(x)^2}{2} \quad (1)$$

There are two inputs for training. One is the well formatted input image and the other is the desired outcome. Both inputs are in bit representations. The input is derived from the image captured by a digital camera. The captured image is processed the following way:

1) Extraction of screen letter color
2) Invert image
3) Threshold filtering

The resulting image only contains black or white pixels. A black pixel represents binary 1 and white pixel represents binary 0. If a “Region of Interest” is defined, the image is cropped to the specific size of the region. From the cropped image the detected characters are extracted and converted to a pixel matrix, where only the binary values of the character pixels are stored. This process is shown in Fig 5. The resulting matrix is the first input of the neural network.

Fig. 5. Processing of input image

The desired output is the second input for training. This is the 16 bit Unicode representation of the characters that can be seen on the picture.

The overall process can be seen in Fig 6, where a sample training is shown.

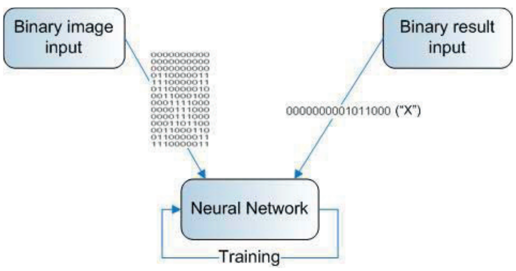


Fig. 6. Training of X to neural network

After the training is completed the module can be used as a decoder. The module input is the image stream from the digital camera and the output of the module is the desired digital or analog output.

IV. Conclusion

This paper has presented an architecture for inter-machine communication and control, based on the RT-Middleware framework. Further a vision system for data retrieval from FMC components has been developed. The proposed methodology and vision system are general in their layout. The vision system represents most of all a more secure solution than a solution based on hardware intervention.

References

[1] Y. Liu, X. Guo, W. Li, K. Yamazaki, K. Kashihara, M.Fujishima, “An intelligent NC program processor for CNC system of machine tool,” in Robotics and Computer-Integrated Manufacturing, vol. 23, Elsevier, 2007, pp. 160–169

[2] X. Zhu, Y. Wang, H. Fu, “A 3-D Simulation System for Milling Machining Based STEP-NC” in Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, Dalian, China, 2006

[3] S.H. Suh, B.E. Lee, D.H. Chung, S.U. Cheon, “Architecture and implementation of a shop –floor programming system for STEP-compliant CNC” in Computer–Aided Design, vol. 35, Elsevier, 2002, pp. 1069–1083

[4] X.W. Wu, “Realisation of STEP-NC enabled machining” in Robotics and Computer-Integrated Manufacturing, vol. 22, Elsevier, 2006, pp. 144–153

[5] B. Kelle, “Offline Programming of a Flexible Manufacturing System” Final project, Budapest University of Technology and Economics and Narvik University College, Dept. Integrated Engineering and Industrial Engineering, Hungary and Norway, 2007

[6] Tetsuo Kotoku, "Robot Middleware and its Standardization in OMG" in International Conference on Intelligent Robots and Systems (IROS'06) Workshop on Robotic Standardization, Beijing, China, 11-13 Oct. 2006

[7] S. Magnenat, V. Longchamp, F. Mondada,"ASEBA, an event-based middleware for distributed robot control" in Workshops of International Conference on Intelligent Robots and Systems (IROS), 2007

[8] H.D. Nayar, I.A. Nesnas, "Measures and Procedures: Lessons Learned from the CLARAty Development at NASA/JPL," International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, October 2007

[9] J. Jackson, "Microsoft robotics studio: A technical introduction" in IEEE Robotics & Automation Magazine, Volume: 14, pp. 82-87 ISSN: 1070-9932, Dec. 2007

[10] A. Weitzendfeld, S. Gutierrez-Nolasco and N. Venkatasubramanian, "MIRO: An Embedded Distributed Architecture for Biologically inspired Mobile Robots" in 11th IEEE International Conference on Advanced Robotics (ICAR'03), 2003

[11] Ozaki Fumio, Oaki Junji, Hashimoto Hideaki, Sato Hirokazu, "Open Robot Controller Architecture (ORCA)" in Journal: Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu, Vol. 2, 2003

[12] M. Mizukawa, H. Matsuka, T. Koyama, T. Inukai, A. Noda, H. Tezuka, Y. Noguchi, N. Otera, "ORiN: open robot interface for the network - the standard and unified network interface for industrial robot applications" in Proceedings of the 41st SICE Annual Conference (SICE 2002), Tokyo, Japan, Vol. 2, pp. 925- 928, ISBN: 0-7803-7631-55-7, Aug. 2002

[13] Nuno Lopes, Pedro Lima, "OpenSDK - An Open-source Implementation of OPEN-R" , Proc. Of 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008), Estoril, Portugal, 2008

[14] H. Bruyninckx, P. Soetens, B. Koninckx, "The real-time motion control core of the Orocos project" in Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03), Heverlee, Belgium, Vol. 2., pp. 2766- 2771, ISSN: 1050-4729, ISBN: 0-7803-7736-2, 14-19 Sept. 2003

[15] Matthias Kranz, Radu Bogdan Rusu, Alexis Maldonado, Michael Beetz, and Albrecht Schmidt, "A Player/Stage System for Context-Aware Intelligent Environments" in Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys 2006), at the 8th Annual Conference on Ubiquitous Computing (Ubicomp 2006), Orange County, California, September 2006

[16] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, Yoon Woo-Keun, “RT-middleware: distributed component middleware for RT (robot technology)” in Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 3933- 3938, ISBN: 0-7803-8912-3, 2-6 Aug. 2005

[17] G. Veiga, J. N. Pires,K. Nilsson “On the use of SOA platforms for industrial robotic cells” in Proceedings of Intelligent Manufacturing Systems (IMS2007), Spain, 2007

[18] J.-C. Baillie, "URBI: towards a universal robotic low-level programming language" in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 820- 825, ISBN: 0-7803-8912-3, 2-6 Aug. 2005
[19] Robotic Technology Component Specification [Online], Robotics DTF, Object Management Group (OMG), 1.0 beta 2, 2007. 08. 18., http://www.omg.org/docs/ptc/07-08-18.pdf
[20] G. Sziebig, A. Gaudia, P. Korondi, N. Ando, B. Solvang, "Robot Vision for RT middleware Frameworh" in Proceedings IEEE Instrumentation and Measurement Technology Conference, Warsaw, Poland, 2007
[21] J.Shin, S.Park, C.Ju, H.Cho, "CORBA-based integration framework for distributed shop floor control" in Computers & Industrial Engineering, vol. 45, Elsevier, 2003, pp. 457–474
[22] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.–K. Yoon. "RT-Middleware: Distributed component middleware for RT (robot technology)" in Proc. Computational Intelligence in Robotics and Automation, CIRA 2005, pp. 457-462, 2005
[23] H. Van Brussel, P. Valckenaers, " Robotics, FMS and CIM" in Handbook of Industrial Robotics, edited by S.Y. Nof, 2nd edition, John Wiley & Sons INC, New York, ISBN-0-471-17783-0, 1999, pp. 811-824
[24] H. Simon, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998, ISBN 0-1327-3350-1

Paper 2: Robot Programming in Machining Operations

Bjørn Solvang, Gabor Sziebig, Peter Korondi, Robot Programming in Machining Operations. In*: Robot Manipulators*. Vienna: I-Tech Education and Publishing, 2008. pp. 479-496. ISBN: 978-953-7619-06-0

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang and Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Gabor Sziebig and Peter Korondi.

The paper is an invited book chapter based on the previous work of the authors (e.g. Paper 1 and 3).

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing partly Section 2 and 3, entirely 4.1, 4.2 and produced the graphics and pictures through the paper. Gabor Sziebig's written contribution was thoroughly reviewed and revised by Peter Korondi.

27

Robot Programming in Machining Operations

Bjørn Solvang[1], Gabor Sziebig[1] and Peter Korondi[2]
[1]Narvik University College, [2]Budapest University of Technology and Economics
[1]Norway, [2]Hungary

1. Abstract

Driven by the need for higher -flexibility and -speed during initial programming and path adjustments, new robot programming methodologies quickly arises. The traditional "Teach" and "Offline" programming methodologies have distinct weaknesses in machining applications. "Teach" programming is time consuming when used on freeform surfaces or in areas with limited visibility /accessibility. Also, "Teach" programming only relates to the real work-piece and there is no connection to the ideal CAD model of the work–piece. Vice versa during offline programming there is no knowledge about the real work-piece, only the ideal CAD model is used. To be able to relate to both the real- and ideal- model of the work-piece is especially important in machining operations where the difference between the models often represents the necessary material removal (Solvang et al., 2008 a).
In this chapter an introduction to a programming methodology especially targeted for machining applications is given. This methodology use a single camera combined with image processing algorithms like edge- and colour- detection, combines information of the real and ideal work-piece and represents a human friendly and effective approach for robot programming in machining operations.
The presented methodology is a further development within the previous work of the authors as presented in Solvang et al. (2008 a), Solvang et al. (2007) and Sziebig (2007).

2. Introduction

Industrial robots are used as transporting devices (material handling of work pieces between machines) or in some kind of additive- (e.g. assembly, welding, gluing, painting etc.) or subtractive- manufacturing process (e.g. milling, cutting, grinding, de-burring, polishing etc.). Also the industrial robot controller has good capability of I/O communication and often acts as cell controller in a typical set-up of a flexible manufacturing cell or system (Solvang et al., 2008 b)
Until now, driven by the need from the car manufactures, material handling and welding has been the most focused area of industrial robot development. Zhang et al. (2005) reports that material handling and the additive processes of welding counted for 80% of the industrial robot based applications in 2003, but foresights an extension into subtractive manufacturing applications. Brodgård (2007) also predict that a lightweight

type robot, capable of both subtractive and additive manufacturing, will have an impact on the car industry and the small and medium sized enterprises (SMEs).

Large European research programs, like the framework programme 6 (FP 6) has put a major focus on the integration of the robot into the SME sector (SMErobot, 2005) and also in the newer research program FP 7: Cognitive Systems, Interaction, Robotics (CORDIS Information and Communication Technologies, 2008) there is focus onto the integration of the intelligent robot system with the human being.

In general there exist several strong incentives to open new areas for the usage of the industrial robot. However, many challenges in this development have been identified such as: lack of stiffness in the robot arm (Zhang et al., 2005) and difficulties in integration of various sensors due to vendor specific and their "normally closed" robot controllers.

In the case of subtractive manufacturing, contact forces must be taken into account. Forces must be measured and provided as feedback to the robot control system. Such force measurement usually involves sensors attached to the robot end-effector. At least older robot controllers do not support such possibility of sensor fusion within the control architecture. Several academic researches have been focusing in this field where some choose to entirely replace the original controller like Garcia et al. (2004) while others again include their own subsystem in-between the control loop, like Blomdell et al. (2005) and Thomessen & Lien (2000).

To address problems related to sensor integration in robotic applications several general architectures, named middleware's, are being developed for easy system integration. In Table 1. (based on Solvang et al. (2008 b) some developers of robot middlewares are identified. There are several differences between these units, among others; how many robots are supported; plug and play discovery; etc. Each of theses middleware solutions are composed from modularized components and have a hierarchical setup (Solvang et al., 2008 b). Brodgård (2007) states that one pre requisite for successful introduction of the industrial robot to the SMEs, and their rapidly changing demands, are modularity in the assembly of the robot arm (task dependant assembly of the mechanical arm) but also modularity of the software elements in the controller. As of today the robot manufactures seem to show more interest in sensor integration, perhaps driven by the competition from the middleware community as well as the possible new market opportunities represented by the very large SME sector.

Another challenge related to the introduction of the industrial robot into lighter subtractive machining operations is the human-robot interaction. Traditionally, man-machine interaction was based on online programming techniques with the classical teach pendant. Later we have seen a development into virtual reality based offline programming methodologies. Actually today, a wide range of offline software tools is used to imitate, simulate and control real manufacturing systems. However, also these new methodologies lack capability when it comes to effective human-machine communication. Thomessen et al. (2004) reports that the programming time of grinding robot is 400 times the program execution time. Kalpakjian & Scmid (2006) states that a manual de-burring operation can add up to 10% on the manufacturing cost. In general manual grinding, de-burring and polishing are heavy and in many cases unhealthy work-tasks where the workers need to wear protective equipment such as goggles, gloves and earmuffs (Thomessen et al., 1999).

Name	*Middleware technology*	*Relevant contributors*
ASEBA (Magnenat et al., 2007)	CAN	EPFL
CLARAty (Nayar & Nesnas, 2007)	Multi-level mobility abstraction	NASA
Microsoft RS (Jackson, 2007)	Web-Services	Microsoft
Miro (Weitzenfeld et al., 2003)	CORBA	University of California
Orca (Ozaki & Hashimoto, 2004)	ICE	KTH Stockholm
OrIN (Mizukawa et al., 2002)	DCON, SOAP, XML	JARA
Open-R (Lopes & Lima, 2008)	Aperios OS	Sony
Orocos (Bruyninkx et al., 2003)	RealTime Toolkit	Katholieke Universiteit Leuven
Player (Kranz et al., 2006)	Client / Server architecture	Multiple
RT-Middleware (Ando et al., 2005)	CORBA	AIST
YARP (Metta et al., 2006)	Client / Server architecture	MIT
UPnP (Veiga et al., 2007)	HTTP, SOAP, XML	University of Coimbra
Urbi (Baillie, 2005)	Client / Server architecture	Gostai

Table 1. Middleware architectures, based on Solvang et al. (2008 b)

Thus, there is a need to develop new methodologies for programming of industrial robots, especially associated to lighter machining operations like grinding/de-burring and polishing

In this chapter, focus will be kept on the human friendly and effective communication between the human operator and the robot system applied in lighter subtractive manufacturing processes like grinding, de-burring, and polishing.

The organization of this chapter is as follows: Section 3 gives an overview of the presented programming methodology while section 4 presents details of some system components. Section 5 gives initial experimental results while section 5 concludes and gives recommendations for further work.

3. Programming of the industrial robot in lighter machining operations: A conceptual methodology

Based on the challenges introduced above, the authors have developed a conceptual programming methodology, especially targeted for lighter machining operations.

The concept of the methodology is captured in Fig.1. The key-issue of the proposed methodology is to capture the knowledge of a skilled operator and make a semi- automatic knowledge transfer to a robot system. The expertise of the worker is still needed and appreciated. The man- machine interaction is carried out in a human friendly way with a minimum time spent on robot programming.

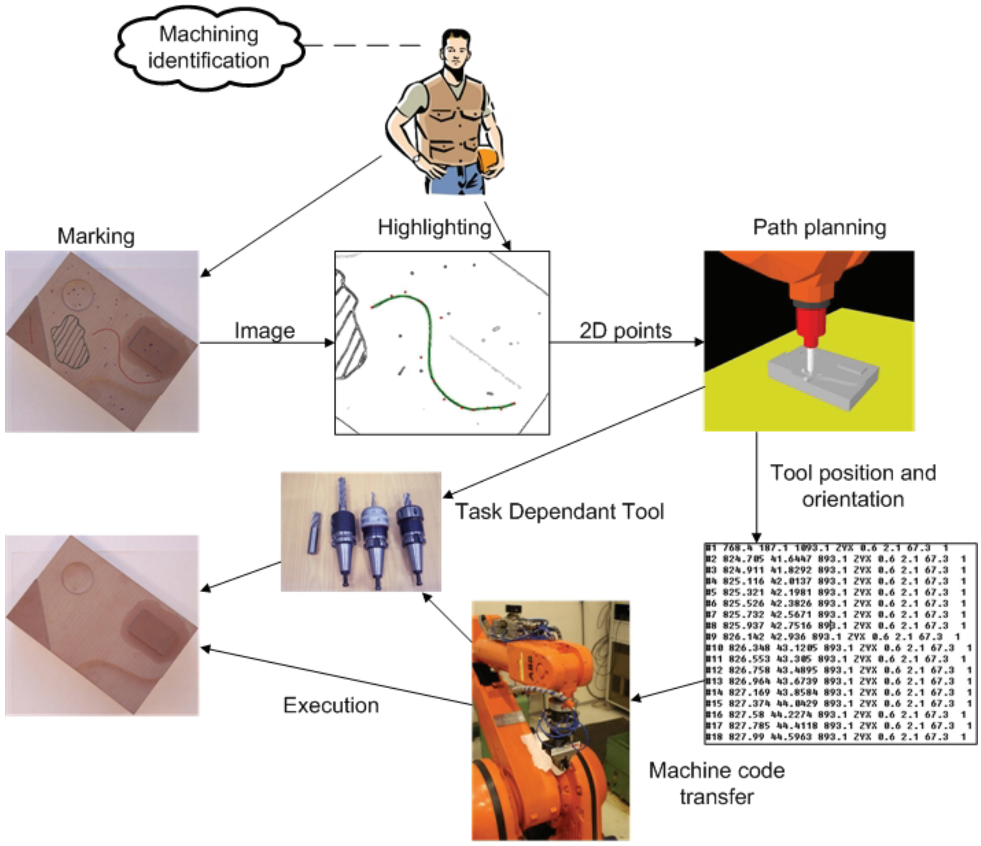


Figure 1. Conceptual overview

The concept is the following: a manufactured work-piece is inspected by an operator, who decides if there is any more machining necessary. The operator indicates the machining tasks by drawing (Marking) different shapes on the surface of the work-piece. Different colour mean different machining operation (e.g. green = polishing). The size of the machining is depending on the marking technique (e.g. curve = just one path if tool size is equals to the curve's width). After the marking a photo is taken of the work-piece. The marked surface is converted first to 2D points (with the help of operator and image processing techniques) and in a later stage to robot position and orientation data. The operator is involved only in the highlighting process from that point it is automated. Finally the "error-free", cleaned work-piece is manufactured by the robot.
The methodology steps are formalized in the following sequence:

1. Work-piece inspection

2. Machining selection
3. Image processing (see Section 4.1)
4. Image to cutting points conversion (see Section 4.2)
5. Tool orientation establishment (see Section 4.3)
6. Locating the work-piece (see Section 4.4)
7. Simulation
8. Execution of path

The steps in details are the following:

First of all the work-piece error identification should be carried out. This is done by a human operator which normally very easy can identify that there are irregularities at a certain area (but cannot at the same time state the magnitude and exact location of the error).

Next the operator should try to determine how such an error will impact on the functionality or the aesthetics of the work-piece and state when ether this is a critical error in such aspects. In fact, these operator messages can be written directly onto the work-piece by using standard marker pens. At a later stage, in the image processing module, colour differentiation is used to pick out messages from the operator.

Further the operator should determine if the error is of type point, line or region. Also in this case different colour pens can be used to draw lines, to mark regions, to scatter point clouds directly onto the surface.

After the error identification and classification the operator should select an appropriate machine tool for the material removal. This of course requires an experienced operator which is trained to evaluate error sources, its significance and the available machine tools to correct the error. Cutting depths are, at this stage, unknown to the operator and represents a challenge. Increasing depth means higher cutting forces and is a challenge for the less stiff industrial robot, compared with the conventional NC-machine. Zhang et al. (2005) states that the NC machines typically are 50 times stiffer than the industrial serial robots. Unfortunately, what is gained in flexibility and large working area is paid off by a decreased stiffness of the robot. However, in case of doubt verification of cutting depths can be undertaken, with high accuracy, by measuring the work-piece in a coordinate measuring machine (CMM). Also, at a later stage in the programming methodology and as indicated in Fig.1. there is a possibility to check for cutting depths by setting a few teach points on top of the machining allowance along the machining path. Anyway in lighter subtractive machine processes forces are small, and could as well be monitored by a force sensor attached to the robot end-effector.

Assuming selection of a robot as machine tool, the next step in the procedure is image retrieval and processing. A single camera system is used to capture the area of error and the possible written messages from the operator. A key issue is to capture some known geometrical object on the picture which can be used for sizing the picture and establishes a positional relationship between the error and the known geometry. The final goal of the image processing is to give positional coordinates of the error source related and stored in a work-piece reference frame. Details of the image processing module will be presented further in section 4.1

The image processing module present the machining path as 2D coordinates with reference to a work-piece coordinate system. In this next module this path must transferred into 3D coordinates by determination of the depth coordinate. In an offline programming environment the 2D path is transferred to the surface of a CAD model of the work-piece by an automatic "hit and withdrawal" procedure. After recovering the depth coordinate the 3D cutting points (CPs) are known. This procedure is further described in section 4.2

The best cutting conditions, in a given cutting point, is met when a selected tool is aligned with the surface at certain angles. To enable such tool alignment, information of the surface inclination must be determined. According to the step by step procedure in Fig.1 the next module is referred to as "Tool Orientation". Such procedure starts with finding the surface coordinate system in each cutting point by seeking the surface inclination in two perpendicular directions. Tool orientation is further described in chapter 4.3.

Out in the real manufacturing area, the relationship between the work-piece coordinate system and the industrial robot must be established. Here some already existing (vendor specific) methodology can be utilised. However, in section 4.4 a general approach can be found.

Before execution, path simulations could be undertaken in an offline programming environment in order to seek for singularities, out of reach problems or collisions.

The generated path previously stored in the work-piece coordinate system can be transferred to robot base coordinates and executed. As mentioned above, cutting depth calculations may be undertaken in order to verify that the operation is within certain limits of the machinery.

4. Some system components

In this paragraph several of the important modules of the programming methodology is presented in an overview manner, seeking to give valuable ideas and references for the reader's more than detailed technical information of each of the building blocks.

4.1 Vision and image processing module

In order to help the operator during path planning, different type of image processing techniques are used. After a short introduction of these techniques, the module structure will be shown.

Usually an image is represented as a matrix of pixels in the computer's memory.

$$I = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdot & \cdot & \cdot & \cdot & C_{1,n} \\ C_{2,1} & \cdot & & & & & \cdot \\ \cdot & & \cdot & & & & \cdot \\ \cdot & & & & & & \\ & & & & & & \\ \cdot & & & & \cdot & & \cdot \\ \cdot & & & & & \cdot & \cdot \\ C_{m,1} & \cdot & \cdot & & \cdot & \cdot & C_{m,n} \end{bmatrix} \quad (1)$$

where $C_{m,n} \in ((0\ldots255),(0\ldots255),(0\ldots255))$ represents the pixel's colour. The three values give the decomposition of the colour in the three primary colours: red, green and blue. Almost every colour that is visible to human can be represented like this. For example white is coded as (255, 255, 255) and black as (0, 0, 0). This representation allows an image to

contain 16.8 million of colours. With this decomposition a pixel can be stored in three bytes. This is also known as RGB encoding, which is common in image processing.
As the mathematical representation of an image is a matrix, matrix operations and functions are defined on an image.
Image processing can be viewed as a function, where the input image is I_1, and I_2 is the resulting image after processing.

$$I_2 = F(I_1) \tag{2}$$

F is a filter or transformation. Applying a filter changes the information content of the image.
Many types of filters exist. Some of them are linear and others are non-linear. Range is from basic filters (Colour extraction, Greyscale converter, Resize, Brightness, Rotation, Blending functions) to Matrix convolution filters (Edge detectors, Sharpen filters).
The basis of the convolution filters comes from signal processing (Smith, 2002). When you have a filter you can compute its response ($y(t)$) to an entering signal ($x(t)$), by convolving $x(t)$ and the response of the filter to a delta impulse ($h(t)$).
Continuous time (Smith, 2002):

$$y(t) = h(t) \times x(t) = \int_{-\infty}^{\infty} h(\alpha) \cdot x(t-\alpha) d\alpha = \int_{-\infty}^{\infty} h(t-\alpha) \cdot x(\alpha) d\alpha \tag{3}$$

Discrete time (Smith, 2002):

$$y[k] = h[k] \times x[k] = \sum_{i=-\infty}^{\infty} h[i] \cdot x[k-i] = \sum_{i=-\infty}^{\infty} h[k-i] \cdot x[i] \tag{4}$$

where $\times$ sign is the convolution integral. The same way that we can do for one-dimensional convolution, it can be easily adapted to image convolutions. To get the result image the original image has to be convolved with the image representing the impulse response of the image filter.
Two-dimensional formula (Smith, 2002):

$$y[r,c] = \frac{1}{\sum_{i,j} h[i,j]} \cdot \sum_{j=0}^{M-1} \sum_{i=0}^{M-1} h[j,i] \cdot x[r-j, c-i] \tag{5}$$

where y is the output image, x is the input image, h is the filter and width and height is M. For demonstration of the computation of (5) see Fig. 2., which shows an example of computing the colour value (0..255) of the output image's one pixel ($y[i,j]$).
So in general, the convolution filtering loops through all the pixel values of the input image and computes the new pixel value (output image) based on the matrix filter and the neighbouring pixels.
Let observe a sample work-piece in Fig. 3. The image processing will be executed on this picture, which shows a work-piece and on the surface of it, some operator instructions. The different colours and shapes (point, line, curve, and region) describe the machining type (e.g. green = polishing) and the shape defines the machining path.

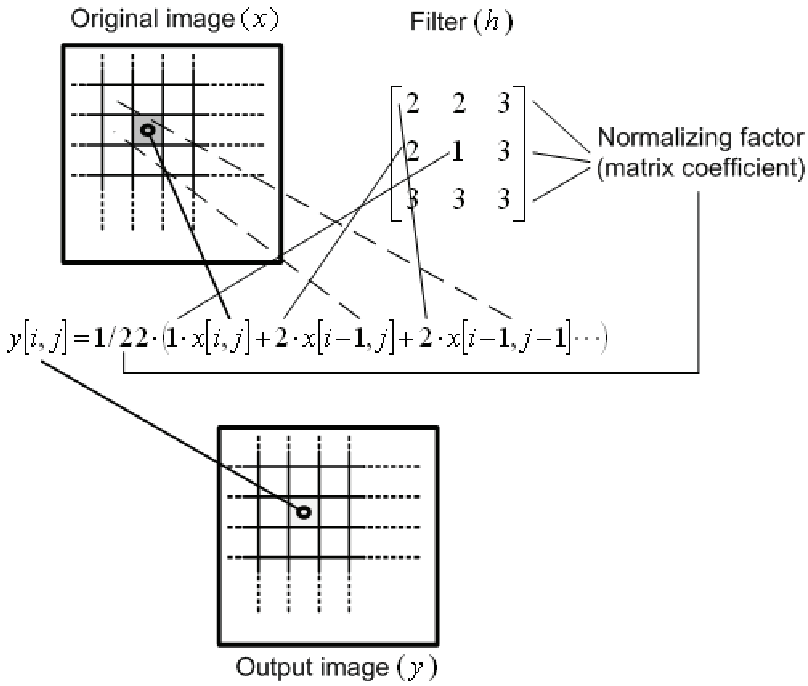


Figure 2. Demonstration of matrix convolution

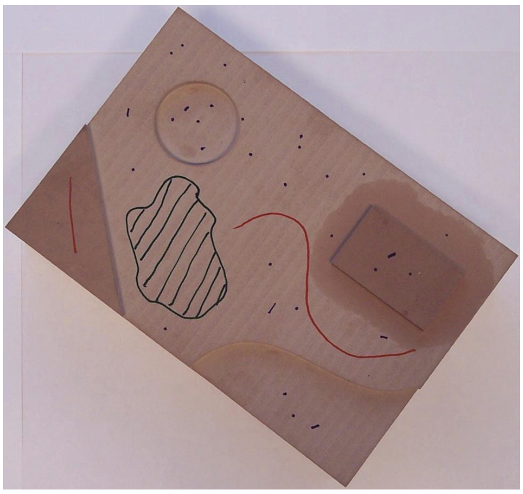
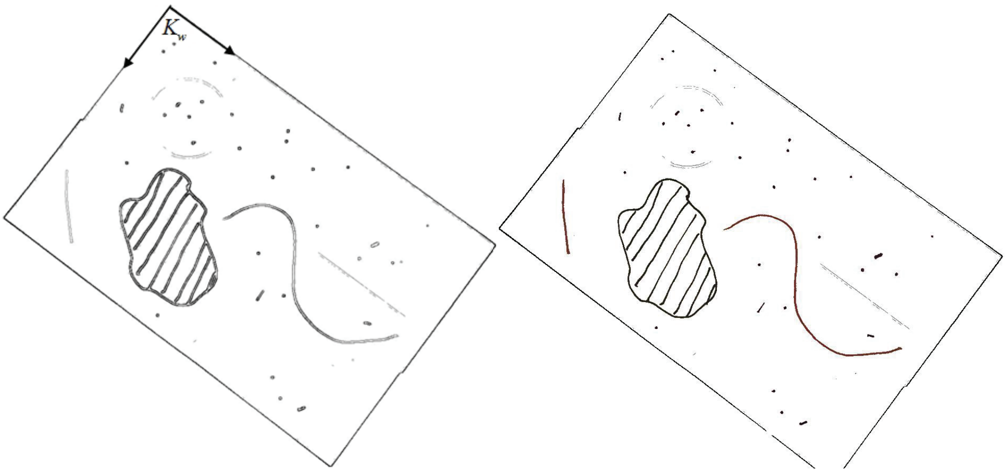


Figure 3. Work-piece example

As mentioned before the best result filters are the matrix convolution filters. Edge detection is used to make transitions more visible, which results high accuracy in work-piece coordinate system establishment (as seen in Fig. 4. (a)) and colour filtering is used to highlight the regions of the error (Fig. 4. (b)).



(a) (b)

Figure 4. (a) Work-piece coordinate system establishment. (b) Colour filtering

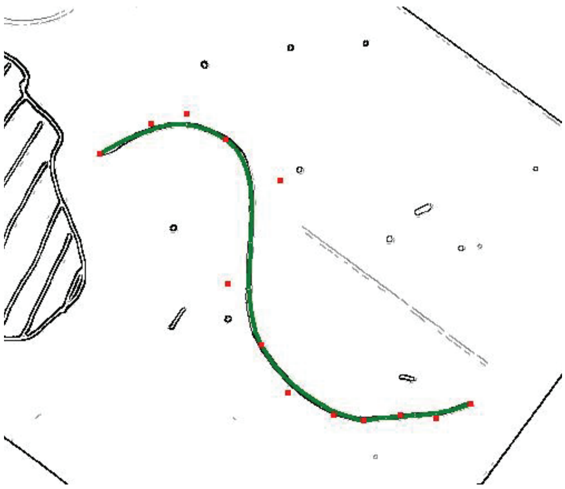


Figure 5. Curve highlighting example

The used Edge detector is the Canny edge detector. The canny edge detection is known as the optimal edge detector (Canny, 1986). With low error rate and low multiple responses (An edge is detected only once). Canny edge detection is built up from several steps for the best results. The steps contain smoothing filter, searching for edge strength (gradient of image), finding edge direction and eliminating streaks. The detailed step descriptions can be found in (Canny, 1986). Only the filter matrix and gradient formula is presented below.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad |G| = |G_x| + |G_y| \tag{6}$$

This performs a 2D spatial gradient measurement on the image. Two filter matrices are used for the function, one estimating the gradient in x direction and the other estimating it in the y direction (6).
The operator executes the following tasks to select machining path:

- Scale picture according to a known geometric object
- Establish a work-piece coordinate system K_w
- Select machining type, based on the colour
- Highlight the 2D path for machining
- Save the 2D path for further processing with reference to K_w

Path highlighting consists of very basic steps. The operator selects what type of shape she/he wants to create and points on the picture to the desired place. The shape will be visible just right after the creation. The points and curves can be modified by "dragging" the points (marked as red squares) on the picture. A case of curve can be observed in Fig 5.
From the highlighting the machining path is generated autonomously. In case of a line or a curve the path is constructed from points, which meets the pre-defined accuracy. The region is split up into lines in the same way as the computer aided manufacturing software's (CAM) do.
The result of the module is a text file with the 2D coordinates of the machining and the machining types. This file is processed further in the next sections.

4.2 From image to 3D cutting points

As the result of the previous section is only 2D (x and y) coordinate, the depth coordinate (z) must also be defined. This is achieved by a standard commercial available simulation program, where the industrial robot maps the surface of the work-piece. The mapping process (as indicated in Fig. 6.) is a "hit and withdrawal" procedure: the robot moves along the existing 2D path and in every point of the path the robotic tool tries to collide with the work-piece surface. If there is a collision the z coordinate is stored and a cutting position p_w^n (index w= work -piece reference coordinate system and index n= cutting point number) is established (Sziebig, 2007).

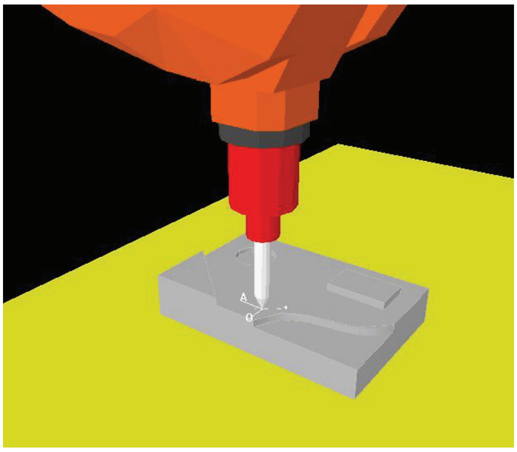


Figure 6. Demonstration of the "hit and withdrawal" procedure

4.3 Surface inclination and alignment of the cutting tool (3D to 6D)

From the previous paragraph, a set of cutting positions p_w^n were identified. However, a 3D positional description of the path is not enough.
To achieve the best and most effective cutting conditions, with a certain shaped and sized cutting tool, the tool must be aligned with the surface of the work piece at certain angles.
To enable such a tool alignment a surface coordinate system must be determined in each cutting point. The authors have previously developed an automatic procedure for finding the surface coordinate system (Solvang et al., 2008 a). According to this procedure the feed direction is determined as the normalized line X_n between two consecutive cutting points p_w^n and p_w^{n+1}.

$$X_n = \frac{p_w^{n+1} - p_w^n}{\left|p_w^{n+1} - p_w^n\right|} \tag{7}$$

Surface inclination Y_n in a direction perpendicular to the feed direction is also determined by an automatic collision detection procedure using a robot simulation tool to determine two points p_w^{n1} and p_w^{n2} perpendicular to the X_n line. This procedure consists of 6 steps:

(0. Robot tool-axis Z_v ($|Z_v| = 1$) already aligned with current work- piece reference axis, Z_w ($|Z_w| = 1$)

1. Rotate robot tool axis Z_v around $Z_v \times X_n$ so that $Z_v \perp X_n$
2. Step along the robot tool- axis Z_v, away from the cutting point (p_w^n)
3. Step aside in a direction $Z_v \times X_n$
4. Move to collide with the surface and store position as p_w^{n1} ;

Relocate above the cutting point (p_w^n); according to step 2.

5. Step aside in a direction $-(Z_v \times X_n)$
6. Move to collide with the surface and store position as p_w^{n2} .

$$Y_n = \frac{p_w^{n2} - p_w^{n1}}{\left|p_w^{n2} - p_w^{n1}\right|} \tag{8}$$

The surface normal Z_n in the cutting point is found as

$$Z_n = X_n \times Y_n \tag{9}$$

These steps are shown in Fig. 7.
In each cutting point p_w^n the directional cosines X_n , Y_n , Z_n forms a surface coordinate system K_n. To collect all parameters a (4×4) transformation matrix is created. The matrix (10) represents the transformation between the cutting point coordinate system K_n and the work piece current reference coordinate system K_w.

$$T_w^n = \begin{bmatrix} X_n & 0 \\ Y_n & 0 \\ Z_n & 0 \\ p_w^n & 1 \end{bmatrix} \tag{10}$$

Tool alignment angles are often referred to as the "lead" and "tilt" angles. The lead angle (β) is the angle between the surface normal Z_n and the tool axis Z_v in the feeding direction while the tilt angle (α) is the angle between the surface normal Z_n and the tool axis Z_v in a direction perpendicular to the feeding direction (Köwerich, 2002). Also a third tool orientation angle (γ), around the tool axis itself, enables usage of a certain side of the cutting tool, or to collect and direct cutting sparks in a given direction.
In case of the existence of a lead, tilt or a tool orientation angle the transformation matrix in (10) is modified according to:

$$T_w^n = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot T_w^n \tag{11}$$

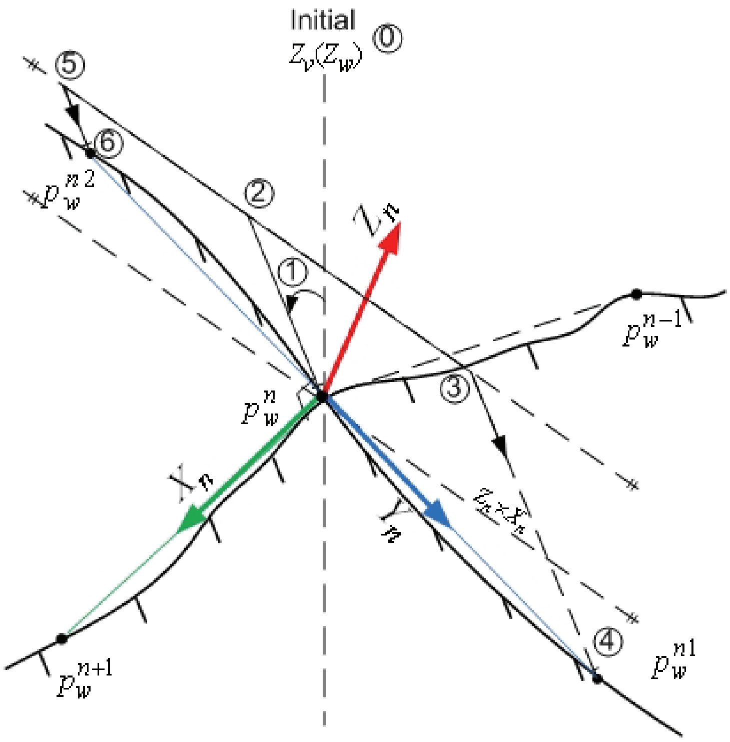


Figure 7. Steps in procedure to determine surface inclination

Fig. 8. shows the angular relationships described above.

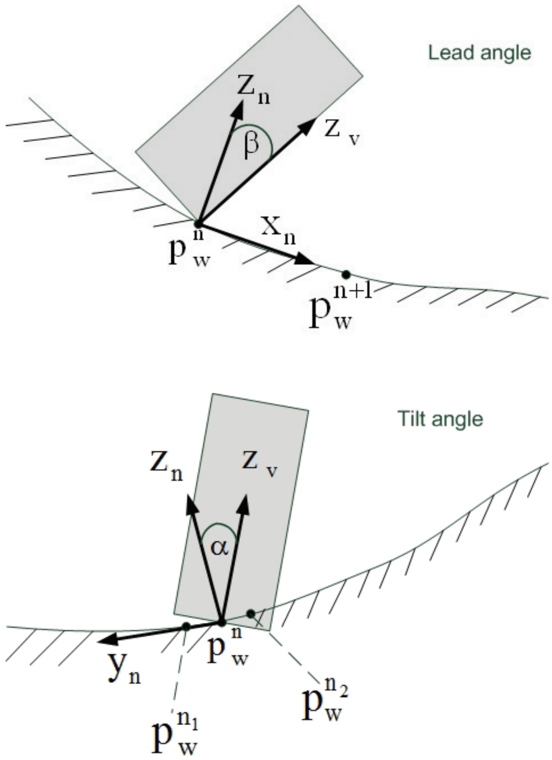


Figure 8. Angular relationships (based on Solvang et al., 2008 a)

4.4 Locating the work-piece

The robot path generated in the previous section is stored with reference to the work-piece coordinate system K_w. By such an arrangement the generated path is portable together with the work-piece. Next, before machining can start the path must be re-stored with reference to the robot base coordinate system K_0.

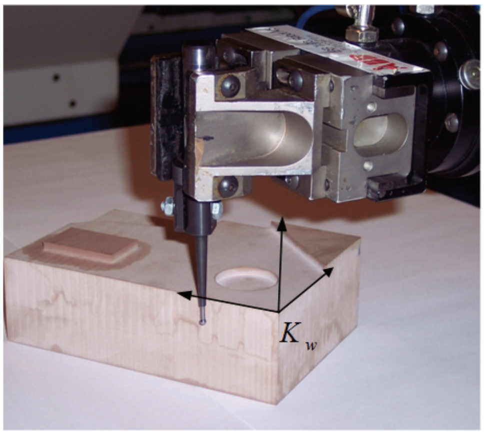
Typically, industrial robot systems have their own set-up procedure to determine such coordinate system relations but in many cases these are based on a minimum number of measurements and focus on a rapid simplicity more than accuracy. However, in robot machining operations the accuracy in reproducing the path heavily depends on the set-up procedure of these coordinate relationships. By adapting the typical coordinate set-up procedures found in coordinate measurement machines (CMMs) accuracy issues are well undertaken.

For the most significant (largest geometries) substitute (ideal) geometries are created based on robot point measurements.

The creation of substitute geometries is done by letting an operator identify what kind of ideal geometry should be created (plane, cylinder, sphere, cone etc.) Then, based on a minimum number of measurement points, such geometry is created so that the squared sum of the distance l_i to each measurement point is minimised according to the Gaussian Least Square Methodology (Martinsen,1992).

$$\sum_{i=1}^{m} l_i^2 \rightarrow \min \tag{12}$$

These substitute geometries are given a vector description with a position and a directional (if applicable) vector.



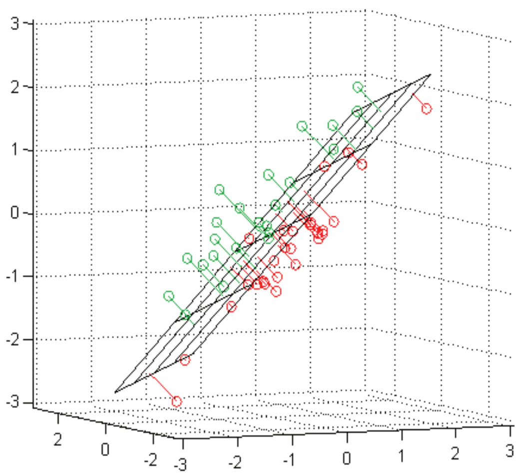


Figure 9. Probing and defining a plane

When establishing K_w, typically a directional vector from the most widespread geometry is selected as one of the directional cosines X_w, Y_w, or Z_w. The next axis could be defined by the intersection line from the crossing of two substitute geometries while the final third is found with the cross product of the two first directional cosines. The origin of K_w is typically located in the intersection of three substitute geometries. Fig. 9. shows this methodology for a given work-piece.

The directional cosines X_w, Y_w, or Z_w and the origin p_0^w is collected in the transformation matrix T_0^w, as introduced in the previous paragraph. Finally, the T_0^w transformation matrix is multiplied with the T_w^n transformation to create the total transformation matrix T_0^n :

$$T_0^n = T_w^n \cdot T_0^w \tag{13}$$

(13) hold information on the position and orientation of the machining path in each cutting point, related to robot base coordinate system K_0. According to (10) position data is extracted from the matrix as the last row in the matrix and orientation is given by the by the directional cosines. The nine parameter directional cosines can be reduced to a minimum of three angular parameters dependant on the choice of the orientation convention. In Craig, (2005) details of most used orientation conventions and transformation are found.

Finally, the robot position and orientation in each cutting point are saved as a robot coordinate file (vendor dependant).

Before execution of the machining process, simulations can be undertaken to search for singularities, out of reach or collisions.

5. Conclusion and future work

In this chapter a conceptual methodology for robot programming in lighter subtractive machining operations have been presented, seeking to give the reader an overview of the challenges and possible solutions for effective communication between the human and the robot system. Some key-modules have been elaborated, in order to let the readers in on more technical details necessary when developing their own systems.

Some important features of the proposed methodology are:

- User communication is human friendly and rapid
- Information of the real work-piece is combined with the ideal CAD-model
- Lines, regions ,dots and messages can be drawn directly onto the work-piece
- Uses only one camera
- Semi –automatic robot path generation

The authors have carried out some initial laboratory tests with the various modules of the proposed methodology. The results are very promising, first of all operator communication is improved and the operator is presented only those tasks he can master intuitively. When it comes to accuracy measurements results shows that the selection of camera is important. A high resolution camera (2856*2142) produced twice as good results as a low resolution web camera (640*480). To measure the complete error chain, a set of experiments was carried out with an ABB irb 2000 robot. For these tests the low resolution web camera were used to capture a hand drawn line on a sculptured surface After the image processing the robot was instructed to follow the generated path.. The deviation between the hand drawn and the robot drawn answer was approximate 1 mm, the same accuracy as for the web camera (Solvang et al, 2008 a) and (Sziebig, 2007).

The initial test shows promising results for the methodology. Tests were carried as path generations only, without any machining action undertaken. The next step would be to look at accuracy tests under stress from the machining process. For these test we need to integrate our equipment with a robot system, preferably equipped with a force sensor at the end-effector.

8. References

Ando, N.; Suehiro, T.; Kitagaki, K. & Kotoku, T. (2005). RT-middleware: distributed component middleware for RT (robot technology), *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3933- 3938, ISBN 0-7803-8912-3, Aug. 2005.

Baillie, J.C. (2005). URBI: towards a universal robotic low-level programming language, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 820-825, ISBN 0-7803-8912-3, Aug. 2005.

Blomdell, A.; Bolmsjo, G.; Brogårdh, T.; Cederberg, P.; Isaksson, M.; Johansson, R.; Haage, M.; Nilsson, K.; Olsson, M.; Olsson, T.; Robertsson, A. & Wang, J. (2005). Extending an industrial robot controller: implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, Vol. 12, No. 3, Sep. 2005, pp. 85-94, ISSN 1070-9932

Brogårdh T. (2007). Present and future robot control development—An industrial perspective. *Annual Reviews in Control*, Vol. 31, No. 1, 2007, pp. 69-79, ISSN 1367-5788

Bruyninckx, H.; Soetens, P. & Koninckx B. (2003). The real-time motion control core of the Orocos project, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03)*, pp. 2766-2771, ISBN: 0-7803-7736-2, Sep. 2003.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, Nov. 1986, pp. 679–698, ISSN 0162-8828

CORDIS (2008). *European Union Seventh Framework Program (FP7) to improve the competitiveness of European industry,* http://cordis.europa.eu/fp7/ict/, 2008-2012.

Craig, J.J. (2005). *Introduction to robotics, Mechanics and control*, Pearson Education, ISBN 0-13-123629-6, USA

Garcia, J.G.; Robertsson, A.; Ortega, J.G. & Johansson, R. (2004) Sensor fusion of force and acceleration for robot force control, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, pp. 3009-3014, ISBN 0-7803-8463-6, Sep. 2004.

Jackson, J. (2007). Microsoft robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine*, Vol. 14, No. 4, Dec. 2007, pp. 82-87 ISSN: 1070-9932

Köwerich, S. (2002). *5 akse maskinering: En state-of-the-art studie*, SINTEF report STF 38 A97247, ISBN 82-14-00695-3, Norway

Kranz, M.; Rusu, R.B.; Maldonado, A.; Beetz, M. & Schmidt, A. (2006). A Player/Stage System for Context-Aware Intelligent Environments, *Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys 2006) at 8th International Conference of Ubiquitous Computing (UbiComp 2006),* pp. 17-21, ISBN 3-5403-9634-9, Sep. 2006.

Lopes, N. & Lima P. (2008). OpenSDK - An Open-source Implementation of OPEN-R, *Proceedings of 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*, ISBN 978-0-9817-3813-0, May. 2008.

Magnenat, S.; Longchamp, V. & Mondada, F. (2007). ASEBA, an event-based middleware for distributed robot control, *Proceedings of Workshops DVD of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, ISBN 978-1-4244-0912-9, Oct. 2007.

Martinsen, K. (1992). *Kompendium i måleteknikk og vektorielle toleranser*, Norges Tekniske Høgskole, Norway

Metta, G.; Fitzpatrick, P. & Natale, L. (2006). YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems*, Vol. 3, No. 1, Mar. 2006, pp. 43-48, ISSN 1729-8806

Mizukawa, M.; Matsuka, H.; Koyama, T.; Inukai, T.; Noda, A.; Tezuka, H.; Noguchi, Y. & Otera, N. (2002). ORiN: open robot interface for the network - the standard and unified network interface for industrial robot applications, *Proceedings of the 41st SICE Annual Conference (SICE 2002)*, pp. 925- 928, ISBN 0-7803-7631-55-7, Aug. 2002.

Nayar, H. D. & Nesnas, I. A. (2007). Measures and Procedures: Lessons Learned from the CLARAty Development at NASA/JPL, *Proceedings of Workshops DVD of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, ISBN 978-1-4244-0912-9, Oct. 2007.

Ozaki, F. & Hashimoto, H. (2004). Open Robot Controller Architecture (ORCA), the Robot Platform. *Toshiba Review*, Vol. 59, No. 9, 2004, pp. 20-24, ISSN:0372-0462

SMErobot (2005). *European Union Sixth Framework Program (FP6) proposal number 011838*, http://www.smerobot.org/, 2005-2009.

Smith, S. (2002). *Digital Signal Processing: A Practical Guide for Engineers and Scientists*, Elsevier, ISBN 0-7506-7444-X, USA

Solvang, B.; Korondi, P.; Sziebig, G. & Ando, N. (2007). SAPIR: Supervised and Adaptive Programming of Industrial Robots, *Proceedings of 11th International Conference on Intelligent Engineering Systems (INES 2007)*, pp. 281-286, ISBN 1-4244-1147-5, Jun. 2007.

Solvang, B.; Sziebig, G. & Korondi, P. (2008 a). Vision Based Robot Programming, *Proceedings of IEEE International Conference on Networking, Sensing and Control (ICNSC 2008)*, pp. 949-954, ISBN 978-1-4244-1685-1, Apr. 2008.

Solvang, B.; Sziebig, G. & Korondi, P. (2008 b). Multilevel Control of Flexible Manufacturing Systems, *Proceedings of IEEE Conference on Human System Interactions (HSI'08)*, accepted for presentation, May. 2008.

Sziebig, G. (2007). Interactive vision-based robot path planning, *Master of Science thesis*, Budapest University of Technology and Economics, 82 pp., May 2007.

Thomessen, T.; Lien, T. K. & Solvang, B. (1999). Robot control system for heavy grinding applications, *Proceedings of 30th International Symposium on Robotics*, pp. 33–38, Oct. 1999.

Thomessen, T. & Lien T. K. (2000). Robot control system for safe and rapid programming of grinding applications. *Industrial Robot: An International Journal*, Vol. 27, No. 6, 2002, pp. 437-444, ISSN 0143-991X

Thomessen, T.; Sannæs, P. K. & Lien, T. K. (2004). Intuitive Robot Programming, *Proceedings of 35th International Symposium on Robotics*, ISBN 0-7695-0751-6, Mar. 2004.

Veiga, G.; Pires J.N. & Nilsson, K. (2007). On the use of service oriented software platforms for industrial robotic cells, *Proceedings of Intelligent Manufacturing Systems (IMS2007)*, May. 2007.

Weitzenfeld, A.; Gutierrez-Nolasco, S. & Venkatasubramanian, N. (2003). MIRO: An Embedded Distributed Architecture for Biologically inspired Mobile Robots, *Proceedings of 11th IEEE International Conference on Advanced Robotics (ICAR'03)*, ISBN 972-96889-9-0, Jun. 2003.

Zhang, H.; Wang, J.; Zhang, G.; Gan, Z.; Pan, Z.; Cui, H. & Zhu, Z. (2005). Machining with flexible manipulator: toward improving robotic machining performance, *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1127-1132, ISBN 0-7803-9047-4, July 2005.

Paper 3: Vision Based Robot Programming

Bjørn Solvang, Gabor Sziebig, Peter Korondi, Vision Based Robot Programming. In: *Proc. IEEE International Conference on Networking, Sensing and Control (ICNSC'08)*. Sanya, China, 06/04/2008-08/04/2008. (IEEE)pp. 949-954.(ISBN: 978-1-4244-1685-1)

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang and Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Gabor Sziebig and Peter Korondi.

The development and testing of the vision based robot programming module is entirely a contribution from Gabor Sziebig. The tests were carried out at the production facilities at Narvik University College.

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing partly Section 1, entirely 2, partly 3.1, entirely 3.1.1 and 4 and produced the graphics and pictures through the paper. Gabor Sziebig's written contribution was thoroughly reviewed and revised by Peter Korondi.

Vision Based Robot Programming

Bjørn Solvang, Gabor Sziebig, and Peter Korondi

***Abstract*—Programming of industrial robots is normally carried out via so-called "Teach" or "Offline" programming methodologies. Both methodologies have associated weaknesses like a high time consumption for programming on sculptured surfaces (especially "Teach"), and both methodologies cannot, at the same time, carry information of both the real work piece and the ideal work piece (CAD-model). To be able to see both the real and ideal world is especially important when the industrial robot is used in manufacturing processes like machining, grinding, deburring etc. where the difference between the real and ideal world represents the material removal.**

This paper presents a new vision based programming methodology which combines information of the real and ideal world, especially adapted for robot grinding and deburring operations. Further, the presented methodology is especially developed with simplicity in mind when it comes to man-machine communication. Thus, a standard marker pen can be used by the operator to draw the robot path directly on the work piece. This path is captured by the vision system and finally leads to the generation of the robot path.

The presented methodology is a further development of the previous work of the authors as presented in [1, 2], especially for the robot tool orientation with respect the work piece surface.

I. INTRODUCTION

Many manufacturing processes leaves irregularities (burrs) on the surface of a work piece. Burrs are often triangular in shape and is found after casting, forging, welding, and shearing of sheet metals [4]. These irregularities are often identified and removed by human operators using by manual grinding. Manual grinding means hard and monotonous work and the workers need to protect themselves by wearing protecting equipment such as goggles, gloves and earmuffs. [3].

Grinding is mostly carried out at an ending stage in the manufacturing process where the work piece gets its final geometry. Any manufacturing error in this stage could be very costly and even lead to dismissal of the entire work piece. Not accounting for possible operator errors the deburring process itself is said to add up to 10% of the total manufacturing cost [4].

Manuscript received October 15, 2007. This work was supported in part by the Narvik University College and the Budapest University of Technology and Economics

Bjørn Solvang is an associate professor at Industrial Engineering department at the Narvik University College, Narvik, Norway (phone: +47 76 96 62 27; fax: +47 76 96 6815; e-mail: bjs@hin.no.).

Gabor Sziebig is a Ph.D. scholar at the Industrial Engineering department at the Narvik University College. Norway (e-mail: gabszi@hin.no).

Peter Korondi is an associate professor at the Department of Automation and Applied Informatics at Budapest University of Technology and Economics, Hungary (e-mail: korondi@elektro.get.bme.hu).

Due to the health risk associated with grinding and the possible added cost from operator errors there is a strong incentive to explore new automated solutions. The industrial robot is viewed as a strong component for this job.

Programming of industrial robots is traditionally done by the "Teach" or the "Offline" programming methodologies. However both methods encounter problems in grinding/deburring operations. The "Teach" method is time-consuming since it requires the operator to manually define all poses (position and orientation) of the tool- centre- point (TCP) on the work piece. Also, the "Teach" methodology carry no information on the ideal work piece, or say the expected end result after the grinding process. In traditional "Offline" programming the robot path is generated from a CAD model of the work piece. This CAD model holds no information on the irregularities (burrs) and then the necessary path cannot be created.

Thus, there is a need to develop new methodologies for programming of industrial robots, especially associated to manufacturing operations like grinding/deburring.

Typically, manual grinding operations is carried out based on individual observations on each work piece. The human operator is very well able to identify the location of these "problem areas", but cannot clearly state what is the necessary cutting depth and how much material is to be removed to reach the ideal final geometry.

In this paper a modified and improved methodology for vision based robot programming is presented, especially adapted for grinding/deburring operations.

This methodology:

- allows the user to draw the grinding path directly onto the work piece
- uses only one camera system to catch the operator generated path
- calculates automatically the cutting depth (material removal)
- calculates automatically the robot grinding/machining 6- dimensional path

By purpose the methodology is not 100% automatic. The expertise of the worker is still needed and used to identify the locations of areas to be modified. However, the man-machine communication is greatly simplified since the programming is done by moving a standard marker pen. Fig. 1 summarizes the concept of the new programming methodology.

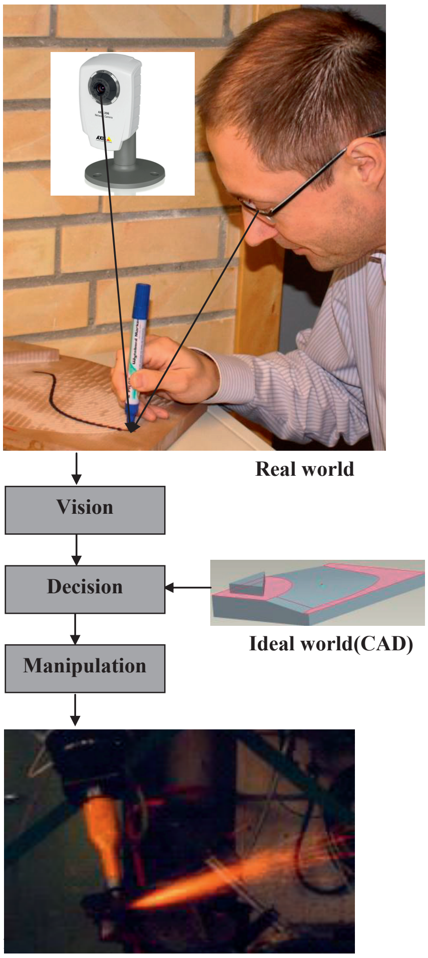


Fig. 1. Conceptual overview

The organization of the paper is as follows: Section II gives an overview of the programming methodology while section III presents details of system components. Section IV presents experimental results while section V concludes the paper.

II. System Overview

Fig. 2 summarizes the proposed methodology. According to figure 2 the image of the real work piece is captured and transferred to the 2D track generator. In this module several image processing tools are available but especially contour lines are developed. Further, the operator should interact with the image to define a work piece coordinate system and then select/define the robot 2D path. This selection is done via a drawing module which allows the operator to generate curvatures onto the existing picture wherever necessary. The generated robot path could be as simple as a straight line, described by a start and an end point, but it can also be a Bezier curve described by 500 or more points. It is the operator's role to make this decision.

Finally, all paths are stored with reference to the work piece coordinate system.

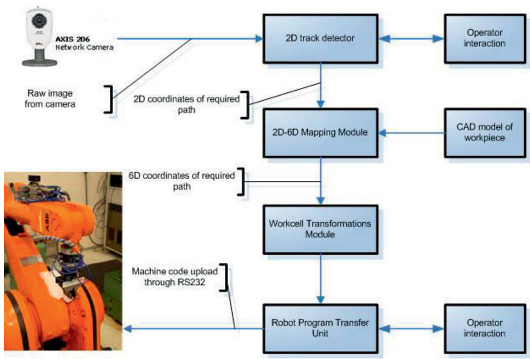


Fig. 2. System overview

In the next step the two dimensional path is transferred into six dimensional (6D) robot coordinates. This is done in an offline programming tool where the CAD model of the work piece is introduced. Since the 2D track generator holds no information on the depth coordinate (z) this is established using an automatic collision (hit and fallback) procedure. Also, in each cutting point, the surface inclination is automatically found and a cutting point coordinate system is generated. Based on the size and geometry of the machining tool the most effective orientation of this tool with respect to the cutting surface is established. At this point, all of the six coordinates necessary to describe a robot path is stored with reference to the work piece coordinate system.

Next step is to establish the relationship between the robot base coordinate system and the work piece coordinate system, out in the machining area. To describe the work piece position and orientation with respect to robot base coordinates a high accuracy procedure is introduced. When the relationship between these two coordinate systems is found the robot path is transferred into robot base coordinates.

The final step in the procedure is to compile the robot program into machine code and execute.

In section III details of selected system components are given.

III. System Components

3.1 The 2D Track Detector

The goal of the application was already presented in the previous section, now the details and the transformation steps (from raw image to 2D work piece coordinates) will be presented.

In Fig. 3 the application use-case diagram can be seen. In this picture the transformation steps are clearly identified. The program starts with an initial screen, a typical sample as seen in Fig. 4.

After choosing the input image source, the image is shown in the left part of the application, in the Image from camera box. If it is a live video or an IP camera the live video is shown. We can stop the video whenever we want, and process the actual image shown in the Input image box. The processed image is shown in the right part of application in the Processed image box. The process means that the picture is sent through image filters, and the resulting image is shown in the Processed image box. The image processing steps, algorithms and filters are presented later in this section.

If we are satisfied with the image processing, we can step forward with pushing View with the Window button, or we can start the whole process from the beginning to get a better result with tuning some parameters of image processing. A sample screen can be seen in Figure 5.

The View Window shows the same picture as in the Processed image box, but in full size.

In the View Window the operator can identify the errors of the work piece and can create the robot path by creating lines curves and regions. The line, curve and region functions can be selected from the menu on the bottom of the window. A status box instructs the operator what to do and how many points are needed for one line or curve or region. The following geometrical figures can be created on the surface of the work piece:

- Line: contains one start and one end point
- Curve: contains four points, one start and one end point, and two control points (Represented as a Bezier curve)
- Curves: contains connected curves, connection on end and start point
- Region: contains at least three points

In the View environment the selected grinding/deburring paths are stored as 2D data points with reference to a camera coordinate system K_c. However, to make these points portable they should be transferred to a coordinate system K_w on the work piece, preferable the same coordinate system used by the CAD system. Prior to set-up of this work piece coordinate system the pixels positions are scaled according to the distance from the camera lens. The work piece coordinate system K_w is then set-up and defined from a selection of well known basic geometric elements found at the work piece. A unit vector z_w normal to the picture plane is selected as one of the coordinate axis, while typically, a line on the surface form basis for the second unit vector direction x_w. The third axis y_w , in this right hand coordinate system, is then given as the cross product between the two other unit vectors.

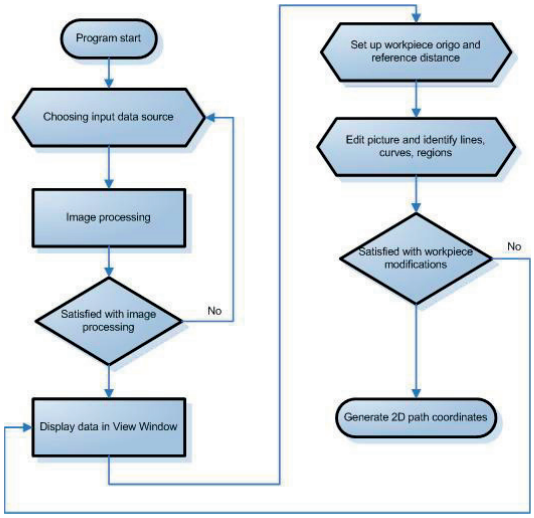


Fig. 3. 2D track detector use-case diagram

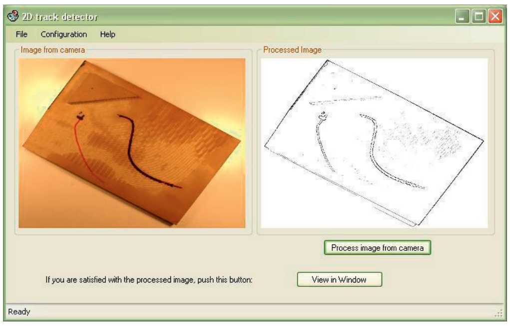


Fig. 4. Sample screen of 2D track detector

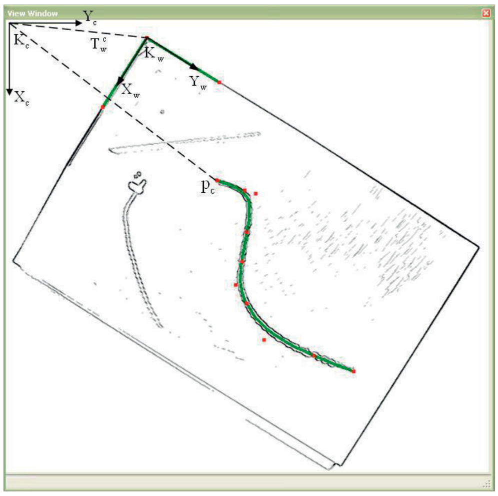


Fig. 5. Sample screen of View Window

$$y_w = z_w \times x_w \qquad (1)$$

The origin coordinates of K_w is selected, typically, as the intersection of two non- parallel lines with the z coordinate set to 0. Pixel position p_c, originally stored in K_c, can now be transferred into K_w coordinate system by the transformation matrix T_W^c.

$$[p_w \quad 1] = [p_c \quad 1]\cdot T_w^c = [p_c \quad 1]\cdot \begin{bmatrix} c\phi & s\phi & 0 & 0 \\ -s\phi & c\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x & y & 0 & 1 \end{bmatrix} \qquad (2)$$

Where c=cosines and s=sinus. Angle ϕ is the angle between x_w and x_c axis while the coordinates x and y is the distance between the two coordinate system origins.

The View application does not know anything of the CAD model of the work piece; it is mainly for identifying errors. However it is possible to match the work piece with the original CAD model. The opacity of View Window can be modified from a menu, and the window can be moved over a CAD modeler.

Finally, by pushing the save button, the 2D coordinate path will be stored with reference to the work piece coordinate system.

3.1.1 About the image processing

The goal of the image processing, in the 2D track detector, is to reveal the errors. This can be achieved by using a sequence of image filters. The sequence steps can be seen in Fig. 6.

The sequence starts with a grayscale conversation, this step cannot be left out, because it is faster and easier to apply filters to a grayscale image. A grayscale image only contains one byte information per pixel, what is great reduction compared to three byte per pixel.

The next step is the sharpening filter. This is a pre-processing step for the edge detection. Sharpening is also a convolution filter, and can be given with its filter matrix.

With a sharpening filter, the contour of the objects in the image is accentuated. Other type of pre-processing filters could also be applied, but the sharpen filter is the most commonly used.

Next a Canny edge detector is used. There exists much kind of edge detectors, but our experiments shows that, this type is the best for error detection.

The last step in our sequence is the inverting. This step could be left out, if the black based edge detected image is better for the operator.

Previous work of authors (DIMAN: Distributed Image Analyzer) [5], made a stable background of the implementation of the image processing steps.

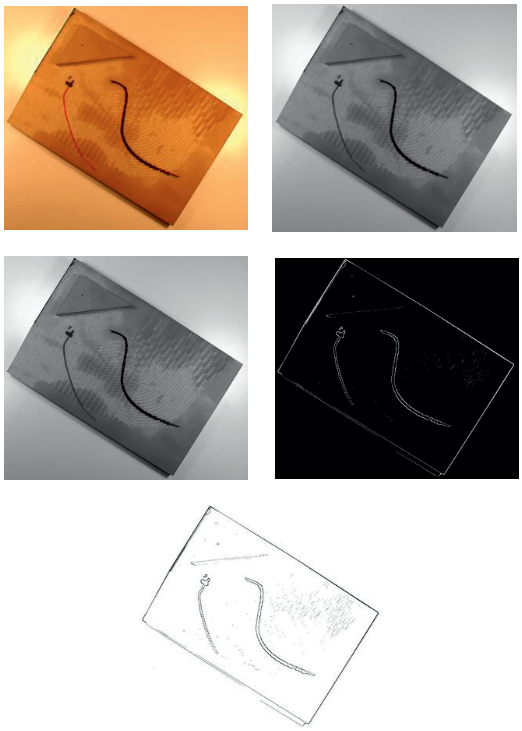


Fig. 6. Image processing sequence

3.2 Mapping from 2D to 6D

The result of the previous section was 2D process coordinates. In this section the 2D coordinates will be transformed into 6D robot pose coordinates. This 2D-6D mapping is done within a standard offline programming tool where the CAD model of the work piece is introduced together with the robot. Fig. 7 shows the offline environment, programmed in the IGRIP® simulation tool. IGRIP® is a powerful robot simulation tool, where complete robot manufacturing cell can be constructed and controlled in virtual reality.

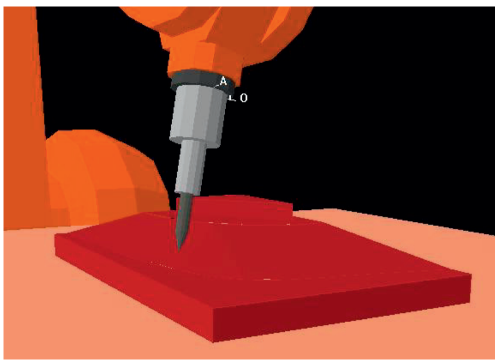


Fig. 7. Offline programming environment

Since the 2D track generator gives no information on the depth coordinate (z) this is firstly established using an automatic collision (hit and fallback) procedure. A pen is attached to the robot arm, and the robot tries to reach the surface of the work piece from the predefined 2D coordinate. If the robot hits the surface, the z position is stored. These 3D positions are known as the cutting points p_w^n (index w= work piece reference coordinate system).

In each cutting point p_w^n we determine the surface inclination in the feed direction x_n as

$$x_n = \frac{p_w^{n+1} - p_w^n}{\left|p_w^{n+1} - p_w^n\right|} \tag{3}$$

Surface inclination y_n in a direction perpendicular to the feed direction is determined by

$$y_n = \frac{p_w^{n2} - p_w^{n1}}{\left|p_w^{n2} - p_w^{n1}\right|} \tag{4}$$

Then the surface normal z_n in the cutting point is found as

$$z_n = x_n \times y_n \tag{5}$$

In each cutting point p_w^n the directional cosines x_n, y_n, z_n forms a surface coordinate system K_n. To collect all parameters a (4×4) transformation matrix is created

$$T_w^n = \begin{bmatrix} x_n & 0 \\ y_n & 0 \\ z_n & 0 \\ p_w^n & 1 \end{bmatrix} \tag{6}$$

The matrix (4) represents the transformation between the cutting point coordinate system K_n and the work piece reference coordinate system K_w.

Dependant on the shape and size of the cutting tool, as mounted in the robot hand and represented by the tool centre point coordinate system K_v, the most effective cutting conditions is achieved when the tool is aligned at certain angles to the surface of the work piece [6]. These angles often are often referred to as the "lead" and "tilt" angles.

The lead angle (β) is the angle between the surface normal z_n and the tool axis z_v in the feeding direction while the tilt angle (α) is the angle between the surface normal z_n and the tool axis z_v in a direction perpendicular to the feeding direction Fig. 8 shows the lead and tilt angles.

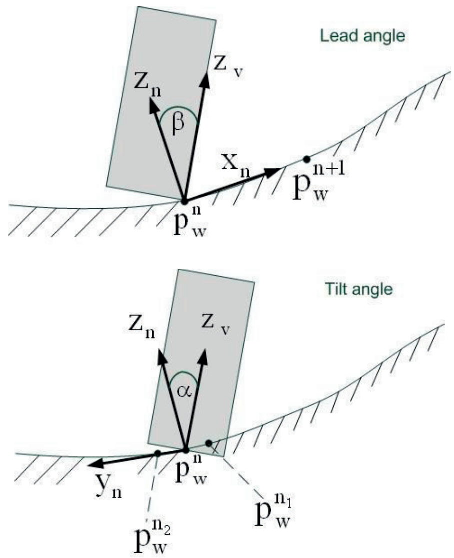


Fig. 8. Lead and tilt angles

In each cutting point the existence of a lead and/or tilt angle modifies the cutting point coordinate system K_n and the belonging transformation matrix (6) according to the following

$$T_w^n = \begin{bmatrix} c\beta & 0 & -s\beta & 0 \\ 0 & 1 & 0 & 0 \\ s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & s\alpha & 0 \\ 0 & -s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot T_w^n \tag{7}$$

Where c=cosines and s=sinus.

A tool orientation angle γ, defined as a rotation around the z_v axis, can in some cases come handy when it is necessary to direct the cutting sparks away from the surface of the work piece or collect cutting chips in a suitable container mounted on the robot wrist. The γ angle adds to the transformation in (7)

$$T_w^n = \begin{bmatrix} c\gamma & s\gamma & 0 & 0 \\ -s\gamma & c\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot T_w^n \tag{8}$$

The transformation matrix (8) holds information on the position and orientation of the robot tool centre coordinate system K_v in each cutting point. According to (6) position data is extracted from the matrix as the last row in the matrix

The orientation is given by the directional cosines represented by the upper left (3×3) corner matrix of (9). This nine parameter matrix can be reduced to a minimum of three angular parameters dependant on the choice of the orientation convention. Most industrial robot systems operate with less than nine parameter description of orientation and a full conversion from nine to three parameters reduces the amount of necessary data input. In [7] details of most used orientation conventions and transformation are found.

After the above calculations the 6D coordinates are saved as robot (vendor dependant) coordinate file, still with reference to the work piece coordinate system.

3.3 Work cell transformation module

Out in the machining cell it is necessary to establish the relationship between the robot base and work piece coordinate system. The accuracy of the robot path will greatly depend on how well this work is carried out. The work cell transformation module undertakes these calculations.

As mentioned earlier the work piece coordinate system is set-up and defined from a selection of well known basic geometric elements (in the ideal work piece) and these elements is re-found on the actual work piece by using the robot as the measuring device. In general, the same methodology used for coordinate measuring machines should be used for the re-establishment of the work piece coordinate system. In [8] extensive recommendations are found.

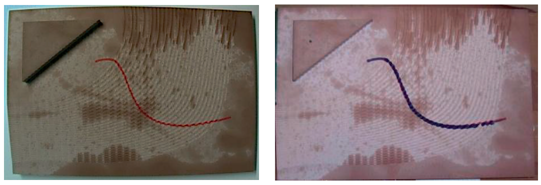
After the work piece coordinate system is found the generated robot path is transformed into robot base coordinates and the program executed by the robot.

IV. Experimental results

Some initial experimental test has been carried out on the proposed system.

Firstly, to measure the effect of the barrel/pincushion distortion and chromatic aberration of the cameras, photos were taken of three elementary geometrical objects (drawn onto a "millimeter" paper). Two different kinds of cameras were used for this test. The first camera was a web camera (640*480 resolution) and the second camera was a typical compact digital camera (2856*2142 resolution). Pictures were taken from the same height and enlarged to the same resolution. In the 2D track detector, after the operator had identified the three geometrical objects, the generated 2D path was matched with the original geometrical objects. The mean deviations between the original and captured images were approximately 1 mm for the web camera and 0.5 mm for the compact digital camera. As expected, the accuracy is better for the higher resolution camera.

Secondly, to measure the complete error chain, a set of experiments have been carried out with an ABB irb 2000 robot. For these tests the low resolution web camera were used to capture a hand drawn line on a sculptured surface (work piece). After the image processing the robot was instructed to follow the generated path. Fig. 9. shows the ideal hand-drawn path (a.) and the resulting path (b.) executed by the robot.



(a) (b)

Fig. 9. System test.

The deviation between the hand drawn and the robot generated answer was approximate 1 mm, the same accuracy as for the web camera. A machining accuracy of less than a millimeter is acceptable in grinding and deburring operations.

V. Conclusion

In this paper, a further development, of our vision based robot programming methodology has been introduced. The proposed system makes robot programming easy, rapid and flexible. The man-machine interaction is very human friendly and the system captures the expertise of an experienced operator in grinding/machining.

The proposed system is accurate enough for most grinding and deburring operations.

References

[1] B. Solvang, P, Korondi, G. Sziebig and N. Ando.,(2007) "SAPIR: Supervised and Adaptive Programming of Industrial Robots," in Proc. 11th International Conference on Intelligent Engineering Systems (INES'07), pp. 281-286 ISBN: 1-4244-1147-5.

[2] G. Sziebig, (2007) "Interactive vision-based robot path planning" Final project, Budapest University of Technology and Economics and Narvik University College, Hungary/Norway

[3] T. Thomessen, T. K. Lien, and B. Solvang, (1999), "Robot control system for heavy grinding applications," in Proc. 30th International Symposium on Robotics, pp. 33–38

[4] S. Kalpakjian , S. R. Schmid,(2006) "Manufacturing Engineering and Technology," Fifth Edition. Pearson Education, Inc. Pearson Prentice Hall. Upper Saddle River, NJ 07458 ISBN 0-13-148965-8

[5] G. Sziebig, A. Gaudia, P. Korondi, and N. Ando, (2006),"Video image processing system for rt-middleware," in Proc. 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI'06), pp. 461–472.

[6] S. Köwerich, (2002) "5 akse maskinering, En state-of-the-art- studie", SINTEF report STF 38 A97247, Norway, ISBN 82-14-00695-3

[7] J.J. Craig, (2005), "Introduction to robotics, Mechanics and control" third ed. Pearson Education, Inc. Pearson Prentice Hall. Upper Saddle River, NJ 07458, ISBN 0-13-123629-6

[8] K.Martinsen, (1992), "Kompendium i måleteknikk og vektorielle toleranser" Institutt for vekstedteknikk, Norges Tekniske Høgskole, Norway.

Paper 4: Image Processing for Next-Generation Robots

Gabor Sziebig, Bjørn Solvang, Peter Korondi, Image Processing for Next-Generation Robots. In: *Computer Vision*. Vienna: I-Tech Education and Publishing, 2008. pp. 429-440. ISBN: 978-953-7619-21-3

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Gabor Sziebig and Peter Korondi.

The paper is an invited book chapter based on the previous work of the authors (e.g. Paper 1). The development and testing of the image processing module is entirely a contribution from Gabor Sziebig. The tests were carried out at the production facilities at Narvik University College.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing almost all the sections and produced the graphics and pictures through the paper (if not referred to external sources, permission for usage of external sources is acquired and could be proofed on request). The entire paper was thoroughly discussed and revised by Bjørn Solvang and Peter Korondi.

Chapter Number

Image processing for Next-Generation Robots

Gabor Sziebig[1], Bjørn Solvang[1], Peter Korondi[2]
[1]*Narvik University College, Norway*
[2]*Budapest University of Technology and Economics, Hungary*

1. Introduction

During the 21st century rapid progress in computer communication and technologies influences robot systems: becoming larger and more complicated than ever previously. To manage the rapidly growing sensor data the single robot systems transformed to networked systems, facing new challenges. The demand of low-cost mass production of robots (industrial or service type), the rising number of elderly people, who needs support of their everyday life, called forth of middleware technology in robot technologies (Brugali & Reggiani, 2005). The middleware technologies provide the heterogenic environment that hides low level functions (hardware specific implementations) and provides the required flexibility for robot system developers. In the beginning of the middleware developments it was doubtful, that the technology causes retardation in speed, but after benchmarks of Christopher D. Gill and William D. Smart (Gill & Smart, 2002) it showed that it has more advantages then disadvantages. They strongly believe that Common Object Request Broker Architecture (CORBA)–based middleware offers great advantages in robotics and embed sensor applications. The urgent need of middleware forced the robot programmers create their own middleware's that meets best their need. Unfortunately, most of the pioneering initiatives are developed independently of the others, driven by specific applications and objectives (Kotoku & Mizukawa, 2006).

In conventional robot system development, the different robot parts (sensors, processing elements and actuators) are combined together in a compact, self contained system. Both type of robot system (industrial or service type) faces challenges in the sense of flexibility, reusability and new part integration. In case of industrial robots, the end-user has very limited intervention to the control system itself (Brogårdh, 2007) and in case of the service robots, the human – robot co-existence demands an extreme flexibility to care take all of human ever changing needs. Robot systems usually consist of sensors that provide information about the environment, computational units that process information about the environment and actuators that are controlled according to the decisions made by the computational units. In recent robot systems one of the most important sensors are image sensors that provide visual information about the environment. Effective robot system design requires that image sensors and image processing functionalities can as easily be integrated with robot systems as any other component. The vision related components of a robot system should thus be integrated using the same middleware as the robot system itself. Thus, development of new robot systems (industrial and service type) should be addressed with great concern of the user demand for flexible solutions.

In Japan the basics of Next-Generation Robots are being developed with the purpose of improving the efficiency of robot design. The development was started in 2004 at the International Robot Fair, Fukuoka, Japan. The following three expectations were defined for the Next-Generation Robots (IRF, 2004):

1. Be partners that coexist with human beings.
2. Assist human beings both physically and psychologically.
3. Contribute to the realization of a safe and peaceful society.

To achieve these expectations new technologies are being promoted and spread in wider areas. To satisfy every user's individual needs, robot systems must be constructed more flexibly. Creation of new functions is made easy by using RT (Robot Technology) - Middleware (Ando et al., 2005 b), which is a modularized software supporting robot program development for Next-Generation Robots (Ikezoe et al., 2006). RT-Middleware was chosen as the platform also in the proposed work, because it is the only middleware solution that is under standardization (Object Management Group, 2007). This solution has proved to be industry ready and used by many industrial partners (Toshiba (different system components), Honda (ASIMO humanoid robot), AIST (OpenHRP humanoid robot), etc.) and also many research institutes (Ando et al., 2005 a).

In this chapter the Distributed Image Analyzer and one example application of the framework will be introduced. Distributed Image Analyzer is a distributed image processing framework for RT-Middleware. Originally it was designed to be grid computing like distributed image processing framework (with own communication interfaces), which was developed to increase processing speed by applying distributed computational resources. The modules created for the grid type version can be utilized without any change in the RT-Middleware based. As the system is modular, the high computational costs of image processing can be shared with different components on board the robot or the environment can be utilized to execute the computation.

The rest of this chapter is organized as follows: in Section 2 the idea and basic structure of the RT-Middleware is presented. In Section 3 the Distributed Image Analyzer framework is presented. In Section 4 example application of vision system in industrial environment is introduced.

2. RT-Middleware

This section is intended to give an overview about the RT-Middleware. The proposed distributed image processing framework is built upon this middleware and will be introduced in the next section.

In 2002 the Japanese Ministry of Economy, Trade and Industry (METI) in collaboration with the Japan Robot Association (JARA) and National Institute of Advanced Industrial Science and Technology (AIST) started a 3 year-national project "Consolidation of Software Infrastructure for Robot Development". With the purpose of implementing robot systems to meet diversified users' needs, this project has pursued R&D of technologies to make up robots and their functional parts in modular structure at the software level and to allow system designers or integrators building versatile robots or systems with relative ease by simply combining selected modular parts (Ando et al., 2005 b).

To realize the proposed robot architecture a robot technology middleware was developed, named "OpenRTM-aist", where OpenRTM stands for Open Robot Technology Middleware.

The concept of RT-middleware can be seen in Fig. 1. Not only thousands of hours of robot programming could be saved, but even more the interoperability between simulation and real applications in robots is solved.

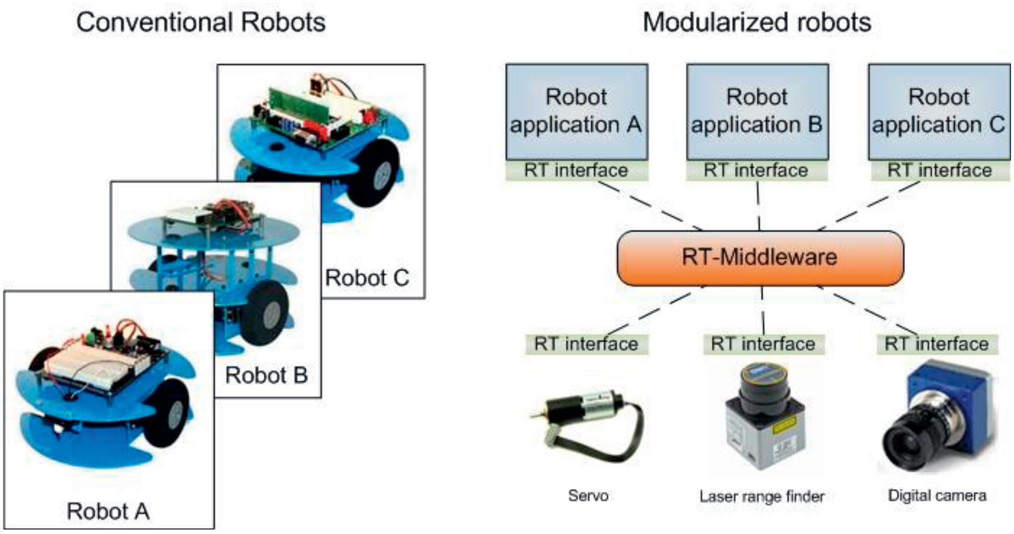


Fig. 1. Difference between the conventional and modularized robot concepts

Two prototype systems have been made to ascertain the effectiveness of the developed RT-middleware (Ando et al., 2005 a).

1. A robot arm control system based on real time control.
2. A life supporting robot system (also known as iSpace (Lee & Hashimoto, 2002)), one of the promising applications.

2.1 Architecture

The framework's basic functional unit is the RT-Component. Modularization is achieved by utilizing the RT-Components. The necessary functions, structure and a realization method based on distributed objects are defined within this component. Fig. 2. shows the architecture block diagram of the RT-Component.

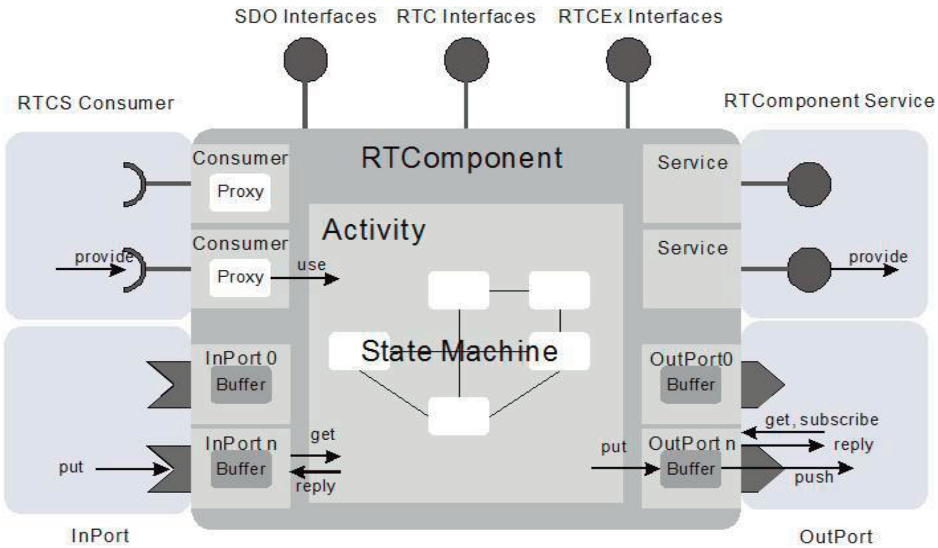


Fig. 2. RT-Component architecture (OpenRTM-aist, 2007)

For reasons of platform independency, the RT-Components are modelled using CORBA as a distributed object middleware. An RT-Component consists of the following objects and interfaces (items not listed here are mainly for administrative functions and description can be found at (Ando et al., 2005 a)):

- Component object (Describes the component itself (e.g. name))
- Activity (Core logic, this must be implemented separately in every component)
- InPort as input port object (Data is received here)
- OutPort as output port object (Data is sent from here)
- Command interfaces (Management of Component object)

In general the distributed object model can be described as some interfaces that contain operations with parameters and a return value. Every single component has the same structure (contains the same interfaces) and the only difference is inside the core logic in the activity and the number of the InPort and OutPort. This allows system transparency, creating a "black-box" of all components. Every RT-Component has an activity which is responsible for data processing with the purpose of device control, such as controlling a motor drive, speech recognition, video processing, etc. The life-cycle of an RT-Component can be observer in Fig. 3.

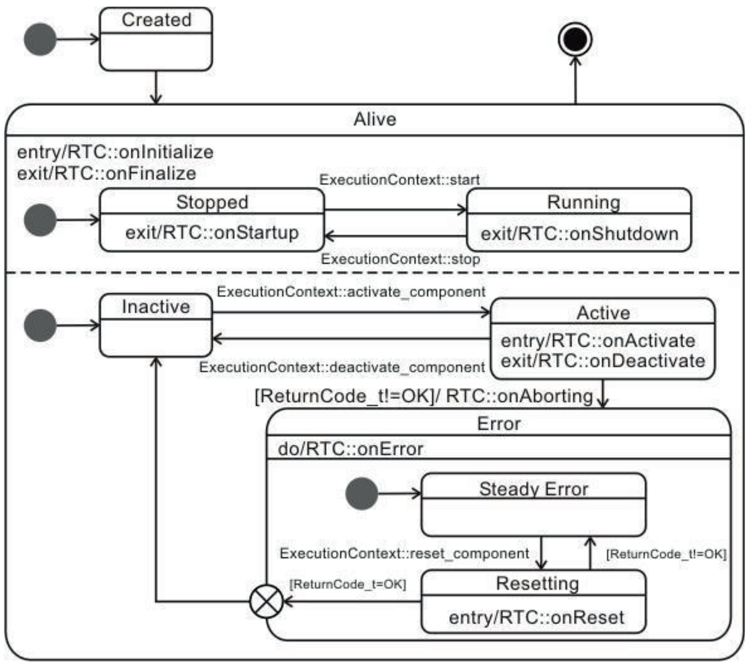


Fig. 3. RT-Component life-cycle (OpenRTM-aist, 2007)

Once the RT-Components are filled up with content and ready to be used, they have to be integrated to form a robot system. The assembly of a system is helped by a Graphical User Interface (GUI) tool that manages a connection of InPort/OutPort between RT-Components like a control block diagram and performs activation/deactivation of an RT-Component. This GUI can be seen in Fig. 4.

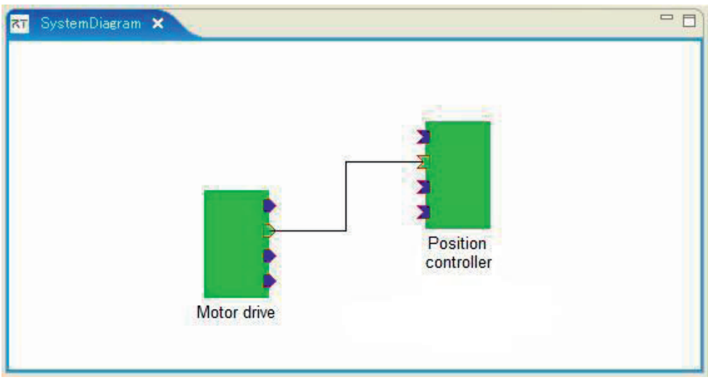


Fig. 4. User interface for system construction

The scope of this chapter is narrow to involve further details about RT-Middleware and its evaluation. More can be found in references (Ando et al., 2005 a), (Ando et al., 2005 b).

3. Distributed Image Analyzer

This section presents the Distributed Image Analyzer, which is a framework for distributed information processing, with a special respect on image processing. It was designed to overcome the heavy computational load of image processing and vision related operations. Image processing tasks are packed in modules and are distributed on several computers, forming a grid computing system. Also it provides a high level of modularity so that modules can easily be connected with each other. Another consideration is that the display of visual data can be done on any of the participating computers. This allows simultaneous supervision of many steps of an image processing algorithm on many screens connected to the computers participating in the framework. This feature is usually unavailable with most of the grid computing systems. The framework can accommodate several modules that are processing nodes of a dataflow-like module graph.

New modules can easily be constructed and added to the framework; only the information processing function has to be implemented. The modules represent the operators for information processing in the framework. They are designed to cooperate as a distributed network of modules, allowing a higher level of complexity. A module provides standard interfaces for communication through a container. Every module can be treated the same way, as a "black-box". The development of a new module is very efficient. A new module is derived from the same class (*CoreModule*). This holds the communication interfaces and the Application Programming Interface (API), only the data processing part of the module has to be implemented.

The modularized system architecture is not only achieved by standardized communication, in Distributed Image Analyzer framework the Dynamically-Loadable Library (DLL) technology is also used, which is a basic technology in Microsoft Windows environment. Every module is isolated in one DLL file and can easily be distributed over the network. There is no need for setup or installation, the DLL containing a certain module is copied to a specified directory and can be immediately used. This becomes possible by introducing a parent class (*CoreModule*) for the modules, where the basic interfaces are specified as virtual functions. When a new module is created, it is derived from *CoreModule*, and the virtual functions are implemented in order to perform the necessary data processing functions. The framework handles the modules through the interfaces defined in *CoreModule*, and has no specific information about the inside of any module. A DLL containing a module is loaded into memory and treated as a *CoreModule* type. The communication between modules is based on events: there is an event for data receiving (called *IncomingDataEvent*) and data sending (called *OutgoingDataEvent*). The management of each module and connection establishment between modules are done by the framework. This architecture can be seen in Fig. 5.

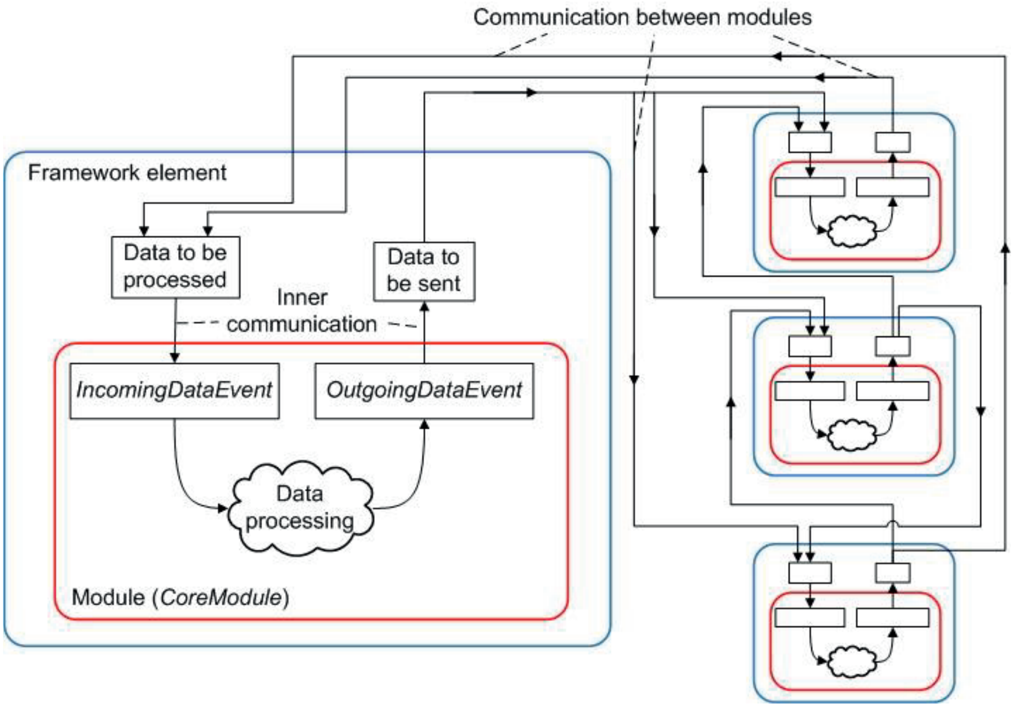


Fig. 5. Architecture of Distributed Image Analyzer

As a reference the following modules are implemented:

- Camera Module, to simulate an eye of a robot
- Edge Detector Module, to detect edges
- Motion Detector Module, to demonstrate an application of image processing
- Colour Mode Converter Module, conversion between colour spaces
- 2D-3D Converter Module, 3D stereo calculations
- Display Module, to display the result of image processing

The similarity between the architecture of the Distributed Image Analyzer framework and the RT-Middleware, which can be noticed from the previous paragraph, makes the module integration to RT-Middleware easy. Only an interface conversion is needed between the modules used in Distributed Image Analyzer framework and RT-Middleware. This allows new image processing algorithms to be tested without involving a robot and after successful test, seamless integration to real environment is possible. This is demonstrated in Fig. 6.

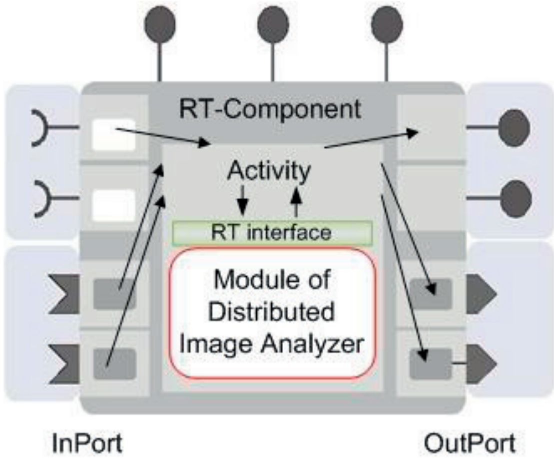


Fig. 6. Integration of an image processing module to RT-Component

After the introduction of the basics of a vision system for robots, in the next section one of its applications is shown.

4. Vision system for old Numerical Control (NC) machines

One example of vision systems in industrial environment is a supervisory system of old NC machines; these machines are usually manufactured before 1980s and lacks capability of Input/Output communication. In that time, these were produced to be as user friendly as possible and only had output through low quality monitors. Even though some might be able to communicate with Personal Computers (PC), it was only one way communication and can only be used from program uploading/downloading. In order to monitor the state of manufacturing process an operator should always observe the screen of NC machine and decisions made by the operator are mainly based on the information shown on the screen.

Based on this idea the operator can be replaced with a vision system, which can detect changes in the screen of machines and can also provide input for other elements in flexible manufacturing systems, also an industrial robot that is used for feeding the NC machine with raw material can be utilized to "operate" the machine by pushing buttons on the operator console. Full remote control of old NC machines can be achieved by this solution. Fig. 7. shows a flexible manufacturing system with a robot and an NC machines.

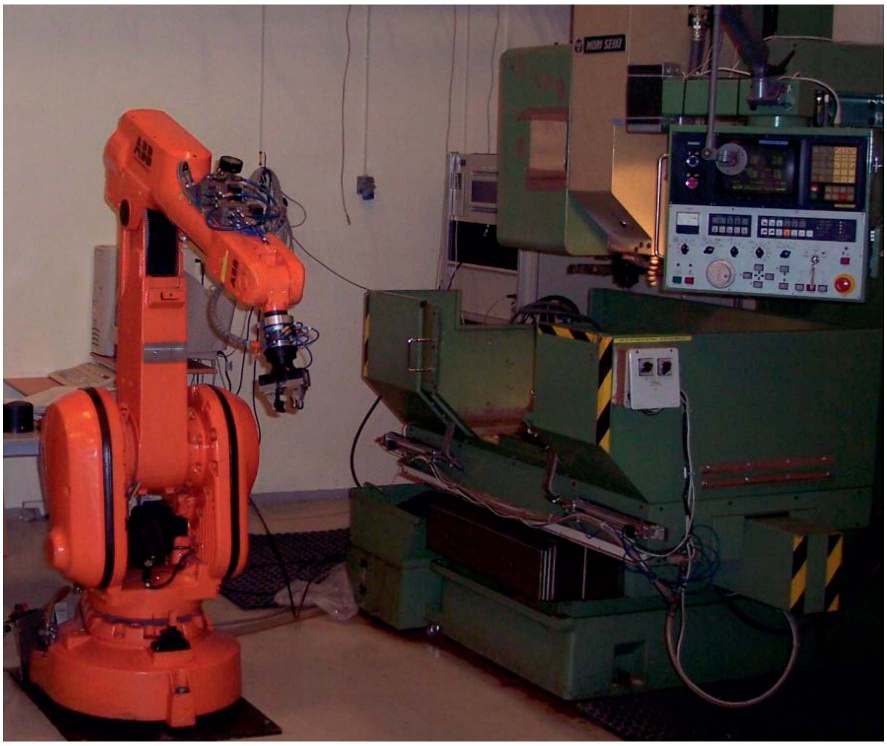


Fig. 7. Example setup of flexible manufacturing system, where the operator is replaced by a camera system and an industrial robot

In this case the vision system is constructed of a digital camera and a PC, which uses image processing tools for Optical Character Recognition (OCR). From the image acquired by the camera, numerous data can be extracted (position, distance to go, current line of the running program, operator messages, etc.). These data describes the current state of the NC machine. For OCR artificial neural networks are trained and used for character recognition. This makes the system robust and self learning. The novelty of the system is not the technology behind (OCR is used for over 20 years now), but the application of it in such an industrial environment. OCR is mainly used in industry for tracking objects (serial number recognition, container localization (Elovic, 2003)), but in this case utilized as a supervisory system.

In the following section after a small introduction of neural networks, the structure and realization of the vision system will be introduced.

4.1 Artificial Neural Network

Modeling functions and systems with neural networks (Gurney, 2003) is a developing science in computer technologies. This special field derives basis from biology, especially from brain of humans. It is well known, that human brain is constructed from neurons and interconnections among neurons, which are called synapses. One neuron can be connected to thousands of other neurons, which results a complex system and provides high level of parallelism. So thus, makes the human brain capable of recognition, perception of objects, human beings, animals, speech, etc. Also this is the explanation for the damage recovery capabilities of the brain.

On the other hand the computers used in everyday life are working in serial mode, which means that only one instruction is processed at a time. Measuring the reaction time of a human brain would result millisecond response, and would result nanoseconds in case of an average computer. Despite of this speed difference, until now, computer technology has not reached the level of the understanding of human brain. This inspired researchers to use the speed of modern computers and the parallel model of the human brain to create artificial intelligence. The combination of the speed and parallel computation model resulted mega leap in solving complex problems, which were previously believed to be impossible to solve.

Power of the neural networks is the fact, that they are nonlinear, which make them capable of solving nonlinear problems, while observing the limits in dimensions. It is easy to apply to any kind of problem and there is no need to know the exact data representation, unlike traditional statistical nonlinear methods. A neural network learns by examples, the more provided the better network is created, which is the same as how children learn.

4.2 Multi-Layer Perceptron Neural Network Model

Nonlinear functions, such as Optical Character Recognition (OCR) (Bhagat, 2005), cannot be learned by single-layer neural networks. Hence use of multi-layer neural network is a must.

A multi-layer neural network is constructed from one input layer, one or more hidden layers and one output layer. In Fig. 8 this structure is introduced. Each layer has a predefined amount of artificial neurons (perceptons). The Multi-Layer Perceptron Neural Network (Haykin, 1998) is a feed-forward network, where the layers are in distinct topology. Every artificial neuron is connected to each of artificial neuron in preceding layer.

The artificial neuron receives number of inputs. Input can be either original data (in case of input layer) or from the output of other neurons in the network. Each connection has strength, called weight, which corresponds to synaptic efficiency of biological neuron. Every artificial neuron also has a threshold value, which is subtracted from the sum of incoming weights. This forms the activation of neuron. Activation signal is passed through the activation (transfer) function, called sigmoid function. The result of the activation function is the output of the artificial neuron.

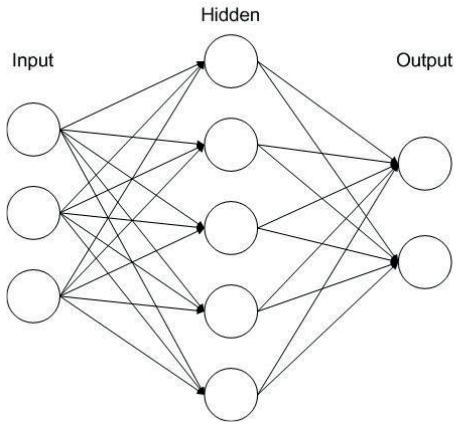


Fig. 8. Example of Multi-Layer Perceptron Neural Network

4.3 Structure

The vision system structure can be observed in Fig. 9. The image acquired by a digital camera is transferred to a PC, where the OCR is executed. The result of the OCR is the information of the operator screen. Based on this, different tasks can be carried out: instruct industrial robot to push button, change I/O levels, stop NC machine program upload, etc.

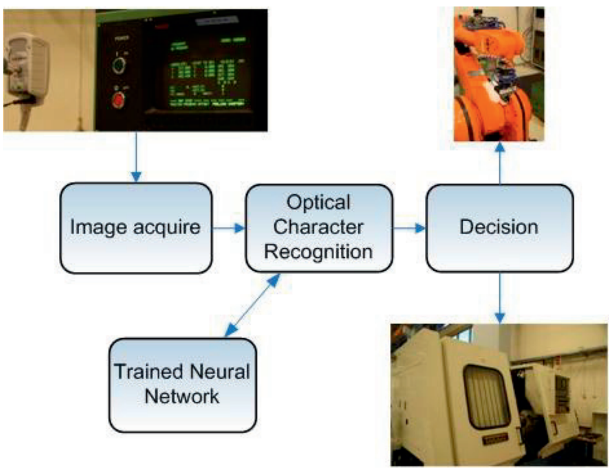


Fig. 9. Vision system structure

The structure has a modular system. Every component can be replaced to machine (where is the needed information on the screen) and camera (connection type) specific module. In the setup phase also the neural network (layer and neuron number, weights and learning rate) can be customized.

4.4 Realization

The Multi-Layer Perceptron Neural Network is trained with back propagation learning algorithm (Rojas & Feldman, 1996) and for the activation function bipolar sigmoid function is used (1):

$$
\begin{aligned}
f(x) &= \frac{2}{1+e^{-\alpha * x}} - 1 \\
f(x)' &= \frac{2 * \alpha * e^{-\alpha * x}}{(1+e^{-\alpha * x})^2} = \alpha * \frac{1-f(x)^2}{2}
\end{aligned}
\tag{1}
$$

where α is the parameter that decides the gradient of the function and x is the sum of the outputs of the previous layer.
The training uses two inputs: the well formatted input image, captured by the digital camera, and the desired output. Both inputs are represented as binary values (0 or 1) and the process is shown in Fig. 10.

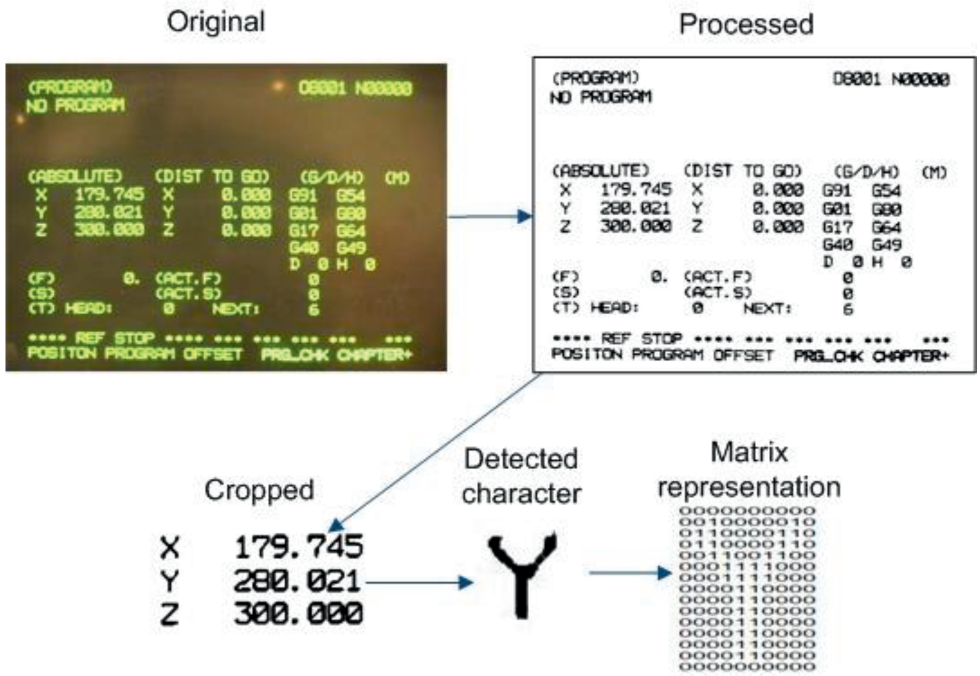


Fig. 10. Processing of input image

In order to get a well formatted image the following steps are executed:

1. Screen colour extraction (usually the old NC machines use one colour), the goal is to achieve higher contrast
2. Threshold filtering

With these steps an image will only black and white pixels. The black pixels are representing the binary value 1 and the white pixels represent 0. When only a specific region (Region of Interest) is needed, the image is cropped after this stage to specific dimension.

3. Characters are detected
4. Characters are converted to matrix representation (binary values of pixels)

These matrixes form the first input of the training.
The second input is the desired output: the 16bit Unicode representation of the detectable character (e.g. X = 0000000001011000).

The training is using character sets, which are machine specific (images and binary value of the characters).

4.5 Results

The neural networks weakest point is the training. Experiments were carried out to define the layer number, neuron numbers in each layer, best learning rate and sigmoid function alpha value. The goal with the experiments was to achieve 100% character detection in a 30 minute video stream of the operator screen. The character is stored as a 10*15 matrix, which results 150 neurons in the input layer and the output layer is composed of 16, because of the Unicode characters. Table 1. concludes the experimental results.

Parameter name	Parameter value
Layer number	3
Neurons in 1. layer	150
Neurons in 2. layer	300
Neurons in 3. layer	16
Maximum learning iteration	300
Maximum average error	0.0002
Initializing weight	30
Learning rate	120
Sigmoid function alpha value	0.014

Table 1. Results of experimental trials for neural network

7. Conclusion

In this chapter an image processing framework (Distributed Image Analyzer) and its integration into RT-Middleware as RT-Components is introduced. The integration becomes possible by introducing a simple conversion between interfaces. By this interface, image processing modules can easily be loaded to RT-Middleware and provides an image processing and vision toolbox for building complex robot systems.
If a simple image processing system is designed and implemented using the Distributed Image Analyzer toolbox in RT-Middleware, its advantages become apparent. Until now, vision components in robots were highly integrated to robot systems, without the possibility of reuse and easy adjustment.
As a special kind of vision system, vision based observation of old NC machines is introduced. This supervisory system replaces an operator with a camera and an industrial robot.
In the future the proposed image processing framework will be implemented and tested on a humanoid robot.

8. References

Ando, N.; Suehiro, T.; Kitagaki, K. & Kotoku, T. (2005 a). RT-middleware: distributed component middleware for RT (robot technology), *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pp. 3933- 3938, ISBN 0-7803-8912-3, Aug. 2005.

Ando, N.; Suehiro, T.; Kitagaki, K.; Kotoku, T. & Yoon, W.-K. (2005 b). RT-Component Object Model in RT-Middleware: Distributed Component Middleware for RT (Robot Technology), *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2005)*, pp. 457-462, ISBN 0-7803-9355-4, Jun. 2005.

Bhagat, P. (2005). *Pattern Recognition in Industry*, Elsevier Science, ISBN 0-0804-4538-1, USA

Brogårdh T. (2007). Present and future robot control development—An industrial perspective. Annual Reviews in Control, Vol. 31, No. 1, 2007, pp. 69-79, ISSN 1367-5788

Brugali, D. & Reggiani, M. (2005). A Roadmap to crafting modular and interoperable robotic software systems, *Proceedings of IEEE ICRA 2005 Workshop on Software Development and Integration in Robotics*, pp. 1-4, Apr. 2005, IEEE Robotics and Automation Society, Barcelona

Elovic, P. (2003). Implementation of Gate and Crane OCR Systems for Container Terminal Automation and Security, *Proceedings of Terminal Operations Conference ASIA (TOC ASIA 2003)*, pp. 1-7, Feb. 2003.

Gill, C.D. & Smart, W.D. (2002). Middleware for Robots?, *Proceedings of AAAI Spring Symposium on Intelligent Distributed and Embedded Systems*, pp. 1-5, 2002.

Gurney, K. (2003). *An Introduction to Neural Networks*, CRC Press, ISBN 1-8572-8503-4, United Kingdom

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, ISBN 0-1390-8385-5, USA

Ikezoe, A.; Nakamoto, H. & Nagase, M. (2006). Development of RT-Middleware for Image Recognition Module, *Proceedings of SICE-ICASE International Joint Conference, pp. 2036-2041*, ISBN 89-950038-5-5, Oct. 2006.

International Robot Fair (IRF) Organizing Office (2004). *World Robot Declaration*, press release, 25. Feb. 2004.

Kotoku, T. & Mizukawa, M. (2006). Robot Middleware and its Standardization in OMG - Report on OMG Technical Meetings in St. Louis and Boston, *Proceedings of SICE-ICASE International Joint Conference*, pp. 2028-2031, ISBN 89-950038-5-5, Oct. 2006.

Lee, J. H. & Hashimoto, H. (2002). Intelligent Space—concept and contents. *Advanced Robotics*, Vol. 16, No. 3, 2002, pp 265-280, ISSN 0169-1864

Object Management Group (OMG) Robotics DTF (2007). *Robotic Technology Component Specification [Online]*, http://www.omg.org/docs/ptc/07-08-18.pdf, 1.0 beta 2, Aug. 2007.

OpenRTM-aist project team (2007). *OpenRTM-aist [Online]*, http://www.openrtm.org/, 2007.

Rojas, R. & Feldman, J. (1996). *Neural Networks: A Systematic Introduction*, Springer, ISBN 3-5406-0505-3, Germany

Paper 5: STEP-NC Based Industrial Robot CAM System

Bjørn Solvang, Lars Kristian Refsahl, Gabor Sziebig, STEP-NC Based Industrial Robot CAM System. In: *9th IFAC Symposium on Robot Control (SYROCO2009)*. Gifu, Japan, 09/09/2009-12/09/2009. (IFAC) Gifu: IFAC by Pergamon Press, pp. 361-366. Paper 80.

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Lars Kristian Refsahl (at that time master student at NUC) and Gabor Sziebig.

The design of the CAM system is a cooperative effort from Lars Kristian Refsahl and Gabor Sziebig. The PC based implementation of the design is entirely a contribution of Lars Kristian Refsahl. Lars Kristian Refsahl and Gabor Sziebig performed the overall system tests in close collaboration at the production facilities at Narvik University College.

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing partly Section 1 and 2 and produced the graphics and pictures in these sections. Gabor Sziebig's written contribution was thoroughly reviewed and revised by Bjørn Solvang.

STEP-NC Based Industrial Robot CAM System

B. Solvang*, L. K. Refsahl, G. Sziebig*****

**Narvik University College, Narvik NO-8514 NORWAY (Tel: +4776966227; e-mail: bjs@hin.no).*
***Narvik University College, Narvik NO-8514 NORWAY (Tel: +4795867555; e-mail: lakrre@hotmail.com).*
****Narvik University College, Narvik NO-8514 NORWAY (Tel: +4776966323; e-mail: gabszi@hin.no).*

Abstract: In this paper a computer aided manufacturing (CAM) system for industrial robot machining operations is introduced. The system is based on the new machining standard STEP-NC which is rapidly making its way into the world of numerical controlled (NC-) machines (e.g. milling and turning machines). Existing research, related to STEP-NC development, focus mainly on the typical implementation of the standard into these traditional production equipments. However, industrial robots are getting more and more capable of taking onto machining operations and it is necessary to couple the industrial robot towards the new standard.

Keywords: Industrial robot, STEP-NC standard, offline programming, flexible manufacturing systems

1. INTRODUCTION

In manufacturing systems the key role of an industrial robot has been material handling and welding. A survey in Guizzo, (2008) shows that handling covers 35,4 % of all applications while welding covers 28.9 %. At the other end of the scale we find cutting and milling applications with only 2,5% of all the robot installations.

Subtractive machining (e.g. polishing, grinding, milling, turning etc.) has been challenging for the industrial robot, mainly due to the lack of stiffness in robotic arms (Zhang et al., 2005), but also because the industry did not focus much onto such application areas. Today, stiffer and stronger structures are available, mainly through the parallel structured manipulators (e.g. hexapods). However, the serial linked manipulator is as well improving, via stiffer structures, but also through more advanced process control. In example force sensing and control can be utilised to keep the overall stress on the robot manipulator within certain limits.

Initially, machining operation starts with a design stage where computer aided design and computer aided manufacturing (CAD/CAM) software is used to generate a computer model and the necessary cutter movements to produce this model, from a given starting geometry (raw material). The generated cutter location file (CL-file) needs then to be translated (post-processed) to machine specific language. In the case of robot machining this is a challenge because robots do not share a common language. Robot languages are vendor specific and often even model specific. As a result, CAD/CAM software, generally, do not support post-processors for robots. Production companies with several kinds of robots are having a hard time to handle all these languages and self-development of post-processors is time consuming. Program exchange from one robot to another is often impossible and seen from the small and medium sized enterprises (SMEs) the whole idea of using robots in machining applications is quite challenging.

For the typical NC-machine the development has been a bit different. From the very beginning a common language standard ISO 6983, (1982) (often referred to as G-code) has been available. CAD/CAM software has as well been building around this ISO standard and the necessary connection between software and the different machines has been commonly available.

However, the ISO 6983 standard has met criticism related to the fact that it is a low level language which outputs a long code list, hardly understandable and difficult to edit. G-codes represents simple movements of machine axes and do not relate these movements to the work piece geometry. In other words, there is no clear connection between the generated code and its impact on the item to be machined (Zhu et al., 2006). To deal with this, a new standard ISO 14649 (ISO 14649-1, 2003), referred to as STEP-NC, was introduced in 2003 and represents a new paradigm in programming of machining operations. STEP-NC is a high level object oriented programming language where the work-piece geometry and the machining task are strongly connected. STEP-NC based files specify the steps of the machining process rather than information on the cutter location movements alone (Wu, 2006). Around the world various NC-machine and software manufactures are working to implement the STEP-NC in their products. Large companies and many research units are working with implementations and experimental issues related to the new standard: Xu et al., (2004), Newman et al., (2008), Xu, (2007), Suh et al., (2003).

It seems that most development and research related to the STEP-NC standard focus onto the classical NC-machines (milling-, turning machines etc.) and there is a lack of initiatives towards the industrial robot.

So, this paper seeks to couple the new standard towards the industrial robot and the article is organised as follows: In paragraph 2 an overview of a framework for STEP-NC/Robot path generation is laid out. In paragraph 3 follows some system specifics. Experimental results are given in section 4 while paragraph 5 concludes the paper.

2. STEP-NC BASED ROBOT PATH PLANNING

In this section the robot path generation based on STEP-NC standard is laid out. The methodology is adapted to certain software solutions but the concept may be transferred to other environments as well.

A small introduction to the STEP-NC standard is given before the robot CAM system is introduced.

2.1 Introduction to STEP-standards

As already mentioned in the introduction the bottleneck of the older programming standard is the low level language, due to the old architecture, which is still limited in computational capabilities. The existing standard requires machine specific post-processors to translate the high level language to a lower level machine code. This one-way transformation converts the work-piece geometry to machine axis movements. The old programming approach specifies how (axes movement only) instead of what (work-piece geometry related) to do (Xu et al., 2004). The problem with this approach is that the geometry information is lost and from the tool axis movement it is nearly impossible to recreate the work-piece geometry. This effects program portability from one machine to another machine (axis movement in one machine do not correspond to movements at another) and also the change tracking possibility (any change on the final, translated machine code cannot be converted back to work-piece geometry).

The ISO 10303 (ISO 10303-1, 1994), introduced in 1994, is meant to replace the variety of geometry description languages (e.g. IGES, SET, VDA, VRML, PRO) and serves as a standard to describe any geometrical object. ISO 10303 standard, which is built on an object oriented language (so called EXPRESS (ISO 10303-11)), then eliminates the need for different geometry converters. The full name of the standard refers to this: Standard for the exchange of product model data (STEP). By using a general description language for geometrical modelling, the generalisation of the connected machining operations was expected next.

So, first a sub-standard, named as ISO 10303-238 (ISO 10303-238, 2004), was established and was followed by a standalone, but compatible standard, named ISO 14649. This new standard carries both the information of work-piece geometry and the planned machining operation. All data can be stored in files following a predefined structure (normal ASCII text file (ISO 10303-21, 1994) or XML file (ISO 10303-28, 2003)). It can also be directly stored in a database. Every file can be opened and edited by suitable CAD/CAM/CAPP software to make the necessary changes, or it can even be edited directly on the manufacturing machine (if capable). Edited files/changes can be fed back to the designer at any stage. This possibility is the most important change since the appearance of the ISO 6983. As mentioned, editing of G-codes on the shop-floor has unpredictable changes to the original design. The new STEP family standards result in a machine independent language.

However this makes the NC machine vendor’s task somewhat harder. These vendors have to provide a STEP-NC interface for their machines since the planning of the tool-path is normally expected to be done internally, resulting in a more complex controller. As already mentioned, initiatives for such implementations can be found around the world, but applications are generally for classical subtractive manufacturing machines and not targeting the industrial robot.

2.2 STEP-NC based robot CAM system

Fig. 1. summarises the layout of the proposed Robot CAM system. The main contribution and focus in this publication is the Robot CAM module (C) itself, but for the sake of completeness information on the other modules are given as well.

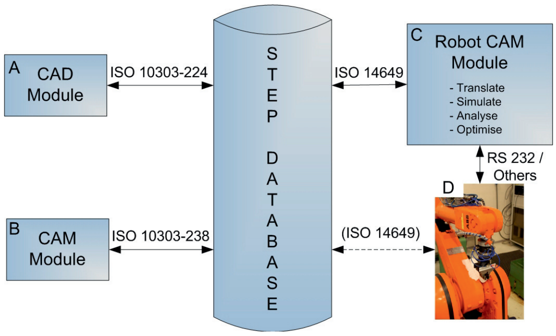


Fig. 1. From design to robot manufacturing, through STEP-NC standard.

First of all, it will be necessary to have a fully operating CAD Module (A) capable of geometry handling according to the STEP standard. In this module, the design of the work-piece, to be manufactured, is carried out. Next, a STEP based CAM Module (B) capable of generating the “cutting orders” and add it to the generated STEP geometry is necessary. Several manufactures of CAD/CAM software are following the STEP-NC development closely and are likely to implement STEP-NC file generation in their CAM-modules. Input to the Robot CAM module (C) is the STEP-NC based manufacturing orders. In our implementation we have selected to develop the CAM module within a standard offline programming and simulation software. In general such software has a common language for programming and visualisation of all the supported machinery. Machine specific translators (post-processors) are available for generation of the necessary machine code. In specific, we have developed our system in the DELMIA Digital Manufacturing & Production software IGRIP® and VNC®. In this environment, a graphic simulation language (GSL) is commonly used to visualise the movements of the programmed machinery. Machine code translators are available. The goal of Robot CAM module is to translate the STEP-NC code into commonly understandable language (GSL code) in order to simulate, analyse and optimise the generated robot machining trajectory. Finally, the real robot can execute the machining orders, as indicated in Fig. 1. (D).

3. SYSTEM COMPONENTS

This section will present some system details of the suggested Robot CAM module; from reading and parsing a STEP-NC file, running a simulation and finally carry out a machining process with an industrial robot.

Fig. 2. clearly indicates the overall processing procedure. The process contains of these four main steps; Read and parse file; Generate tool path with reference to work-piece coordinate system K_W; Path verification and optimisation through simulations, with reference to the virtual world coordinate system K_V; Path execution, with reference to real world coordinate system K_R. As indicated in the figure the resemblance between the virtual- and the real world should be within accepted tolerances, $K_V = K_R \pm TOL$, otherwise calibration procedures must be undertaken.

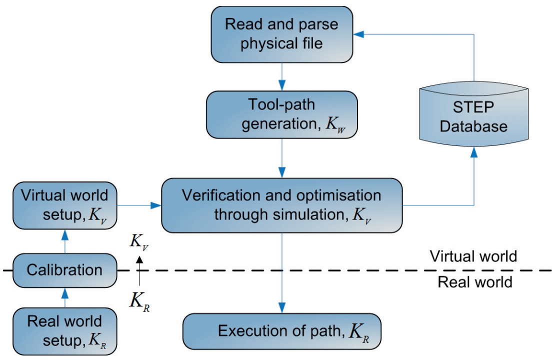


Fig. 2. Robot CAM structure

3.1 Reading and parsing the STEP-NC file

The process starts with a STEP-NC file which contains information about a stock piece to be milled. The STEP-NC file is an ISO 10303-21 file and contains information about tools to be used, geometric features, tolerances, material properties and other information such as “author” and production dates. In this STEP-NC file each line is a command. Each command can contain references to other lines in the file-structure, or it can contain information which controls the milling of the work piece. The structure of the STEP-NC file can be represented like a tree which gives an easy visual representation of the structure, as seen in Fig. 3.

The STEP- NC file is read from a text file and loaded line for line into the program internal memory. Here it is stored to give an easy access for further interpretation. When the program has uploaded the file in its memory, it evaluates each line and its underlying structure or say branch, seeking to determine what kind of information is presented and what to do with this information. The program starts at the top and works it way up and down until it has evaluated each command in the STEP- NC file. In such a way, information about each feature that describes the final geometry of the work- piece is determined.

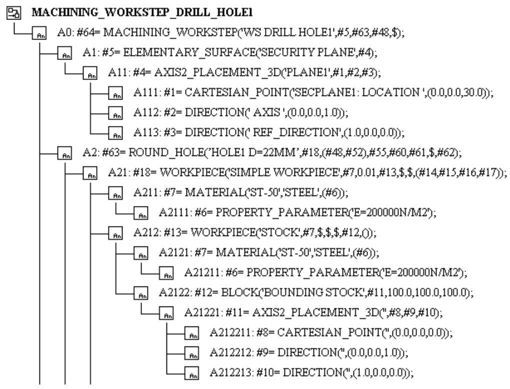


Fig. 3. Tree structure of STEP-NC file

3.2 Tool-path generation

The STEP-NC file does not contain any tool paths; therefore the GSL program has to generate such paths in syntax understandable for the industrial robot. In order to have a fully functional Robot CAM module, translators between STEP-NC and the GSL language should be developed for all STEP-NC commands. Currently we are working with this implementation of the complete standard, and as a typical example the algorithm for milling of a closed pocket will be described in some detail.

A Closed Pocket, as shown in Fig. 4, is a typical geometric feature in a STEP-NC file. Milling of such pocket is to be carried out via two machine operations, named Contour- and Bidirectional- movement. Contour is the movement between the four corner points P1 to P4. Dependant on the tool diameter, Contour operations will leave some material in the middle. Such material is then often removed by the Bidirectional operation (ISO 14649-11, 2004). In Fig. 4. the Bidirectional movement is shown as a line which is indicated with START and STOP.

To be able to mill with Bidirectional movement a tailor made algorithm had to be created in the Robot CAM module. The four corner points P1 to P4, the tool diameter, the pocket depth and tool cut depth are general input parameters to the algorithm. The Bidirectional movement, as described in the ISO 14649-11 standard, consists of a zigzag motion with four directional changes.

The algorithm to calculate these directional changes contains three major steps, first calculating the four corner points in the pocket, then calculating number of sideways steps necessary to do the milling and at last calculating the specific points in the tool path. The number of sideways steps is used to decide one of two cases of formulas to calculate the necessary direction changes and the resulting points. The two cases have the same general structure. Some details about this general structure is given below.

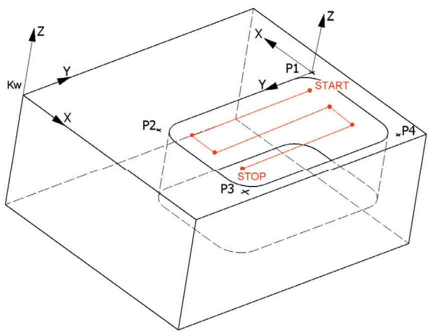


Fig. 4. Geometry of Closed pocket and Bidirectional tool movement

The four direction changes appear in a cyclic order, e.g. repeating each other every fourth time. A method for getting these four cyclic directional changes was created using two integer counters (T and G) and its combinations of odd and even. In general, the algorithm works by adding 1 to the integer G at each direction change, and adding 1 to the integer T at every second direction change, and then examining the combination of odd and even. By following this method, the next direction at any point can be decided by using Tab. 1. Although case Odd-Odd and Even-Odd both give movement in –X direction, the algorithm use a different value to calculate the Y coordinates. This method of deciding the direction of the next movement, together with the other general input parameters is used to calculate the next point in the tool path. Tab.1. shows the correlation between odd and even and the direction in which to place the next point, using the local coordinate system of the pocket.

Table 1. Relationship of cutting direction for bidirectional movement

T	G	Direction
0	0	-
Odd	Odd	-X
Odd	Even	-Y
Even	Odd	-X
Even	Even	Y

Fig.5. shows the integers used and the resulting coordinates to each point in the tool path. Positions P1" to P4" is at the same positions as P1 to P4, only adjusted into the pocket by half the diameter of the tool in use.

The movement between the two first points in the tool path is treated as special cases in the algorithm. To stop the algorithm, the next calculated point is compared with the geometry of the Closed pocket and if it is outside the geometry the algorithm is stopped and the tool path is complete. This algorithm only calculates the X and Y coordinate. The Z position is decided by the cutting depth of the tool in use. The algorithm is run a specific number of times, according to the possible cutting depth of the tool and the desired depth of the pocket. At each point in the tool path, the GSL program sets up a coordinate system (tag-point). These tag points are then stored in the GSL programs internal memory, in the order of which the robot has to move.

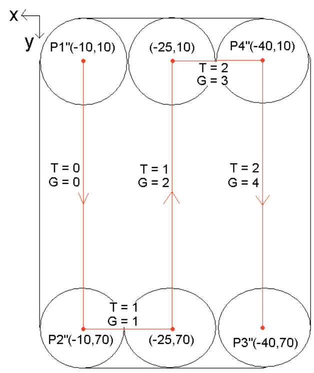


Fig. 5. Tool path, coordinates and correlating integers

3.3 Verification and optimisation through simulations

In the previous section the generated tool- path is stored with reference to the work-piece coordinate system K_w. By such an arrangement the generated path is portable together with the work-piece.

In this paragraph we suggest to bring the work-piece and the generated tool path into a standard visualisation and optimisation module (typical robot offline programming software). In such 3D environment, a replica of a real manufacturing cell is build up and virtual robots are commanded to follow the generated tool-path. In this environment we can check the robot movements for singularities, out of reach, or collisions. Lead times can be measured and the result of the machining operation thoroughly inspected. Fig. 6. shows a typical virtual world robot manufacturing environment.

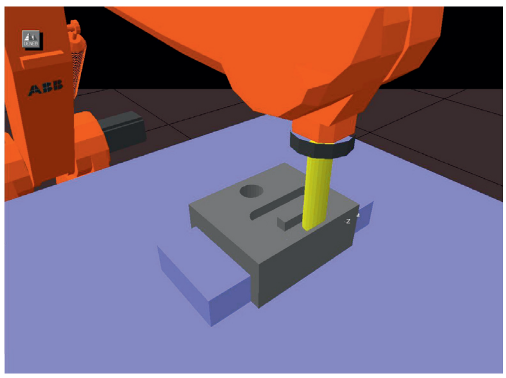


Fig. 6. Robot milling operation

During simulations it may occur situations were the operator wants to modify the generated machining path. However, it is not enough to simply update the robot movements, changes must be reflected in the STEP-NC file as well. Updating the STEP-NC file at any stage of the production cycle is a very important feature of the STEP concept. Modifications made on the shop-floor should be made visible for all participants in the manufacturing process. Our CAM module has implemented such STEP-NC editing and updating possibilities for the selected geometrical features: Closed pocket and Hole. Change of diameter, depth, location and geometry for such objects is possible and the STEP-NC file is updated accordingly.

In the virtual environment, the tool-path coordinates are presented with reference to the virtual world coordinate system K_V. As mentioned, it is of outmost importance that the virtual world coordinates K_V and the real world tool coordinates K_R are within accepted tolerances $K_V = K_R \pm TOL$, otherwise calibration procedures must be undertaken. Solvang et al., (2008) gives recommendations on how to locate the work-piece coordinate system K_w with respect to the real robot base coordinate system K_R.

Finally, after simulations and code optimisation the tool-movements can be translated into machine specific language and transferred to the real robot manufacturing cell.

4. EXPERIMENTS

An STEP-NC file has been used to carry out a simulation experiment with a virtual industrial robot acting as a milling machine.

In the selected work-piece and corresponding STEP-NC file (collected from ISO 14649-11, Annex F (2004)), milling of the experimental work piece is made up by five working steps. These working operations are Planar Face, Drill Hole, Ream Hole, Rough Pocket and Finish Pocket, see Fig. 7 a). Our simulations are concentrated on the last four working steps, since these gives the most visible results. There are used three milling tools, all with the same dimensions as the tools described in the STEP-NC file in the ISO standard. These tools are Endmill-20mm, Drill-20mm and Reamer-22mm.

Fig. 7 b). and c). shows the finished result after the four steps have been carried out on a stock work piece. The virtual model represents an identical geometry as found in the STEP-NC input file.

Second, the generated robot tool-path was downloaded to a NACHI SA-130F robot which was set-up as a XY-plotter (Fig. 8.).

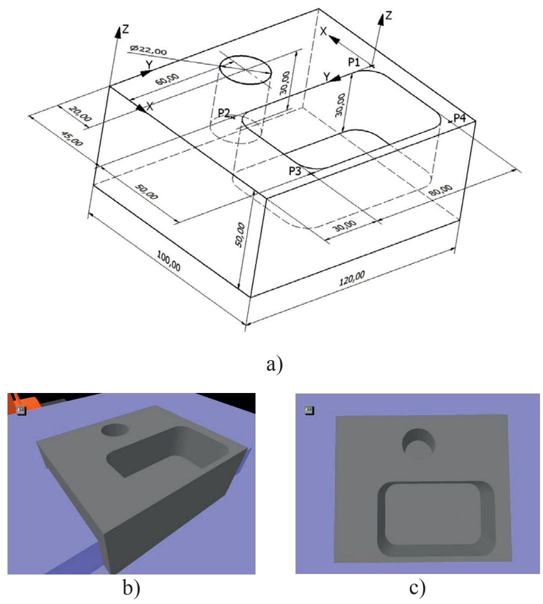


Fig. 7. a) 3D design model (based on ISO 14649-11 Annex F Example 1), b) and c) after virtual machining

The generated tool-path was drawn by the robot on a millimetre-scale paper. The resulting tool-path is shown in Fig. 9. Line lengths were measured and compared with the theoretical lengths found in the STEP-NC input file. All measurements were found identical to their theoretical counterpart. Accuracy of these measurements is dictated by the accuracy of the millimetre-scale paper.

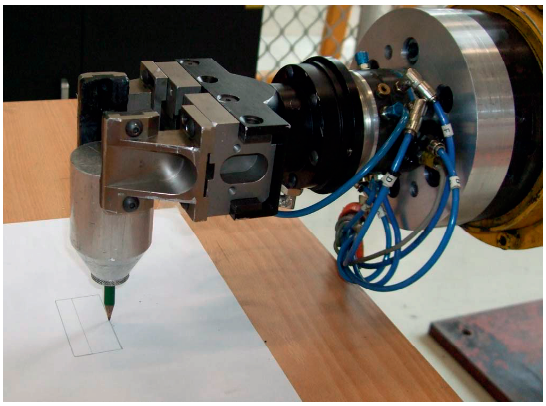


Fig. 8. Experimental setup

Experiments were carried out without any material cutting actions. However, influence on accuracy from the machining process is heavily dependent on the: 1) selected process parameters and 2) the capability (stiffness) of the chosen robot system. Although, these are very important parameters on the overall machining accuracy they do not influence on the accuracy of the Robot CAM system itself.

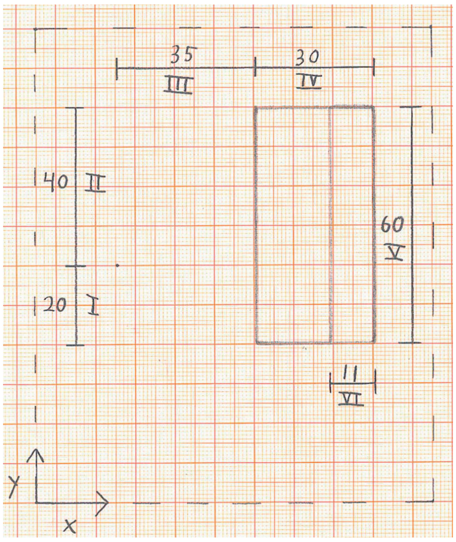


Fig. 9. Resulting robot tool-path

5. CONCLUSIONS

In this paper a Robot CAM module for industrial robot machining operations is introduced. The system is working towards the new machining standard referred to as STEP-NC, which is rapidly making its way into the world of numerical controlled (NC-) machines. By conducting research on the merging of robot systems towards the new standard we may expect to see more robots in machining operations in a near future. Experiments shows that the suggested Robot CAM module can, based on a STEP-NC input file, interpret, translate and transfer the manufacturing orders to an industrial robot.

ACKNOWLEDGEMENT

The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

REFERENCES

Guizzo E. (2008). The Rise of the Machines. *IEEE Spectrum*, Vol. 45, No. 12, pp. 60, ISSN 0018-9235.

International Organization for Standardization (1982). ISO 6983-1, Numerical control of machine-program format and definition of address words Part-1: Data format for positioning, line and contouring control systems, First Edition.

International Organization for Standardization (1994), ISO 10303-1, Industrial automation systems and integration - Product data representation and exchange Part 1: Overview and fundamental principles.

International Organization for Standardization (1994), ISO 10303-11, Industrial automation systems and integration - Product data representation and exchange Part 11: the EXPRESS language reference manual.

International Organization for Standardization (1994), ISO 10303-21, Industrial automation systems and integration - Product data representation and exchange Part 21: implementation methods: clear text encoding of the exchange structure.

International Organization for Standardization (2003), ISO 10303-28, Industrial automation systems and integration - Product data representation and exchange Part 28: implementation methods: XML representations of EXPRESS schemas and data.

International Organization for Standardization (2004), ISO 10303-238, Industrial automation systems and integration - Product data representation and exchange Part 238: Application protocol: Application interpreted model for computerized numerical controllers.

International Organization for Standardization (2003), ISO 14649-1, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 1: Overview and fundamental principles.

International Organization for Standardization (2004). ISO 14649-11, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 11: Process data for milling.

Newman, S.T., Nassehi, A., Xu, X.W., Rosso, Jr. R.S.U., Wang, L., Yusof, Y., Ali, L., Liu, R., Zheng, L.Y., Kumar, S., Vichare, P. and Dhokia, V. (2008). Strategic advantages of interoperability for global manufacturing using CNC technology. *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 6, pp. 699-708.

Solvang, B., Sziebig, G., and Korondi, P. (2008). Robot Programming in Machining Operations. *ch. Robot Manipulators*, Vienna, Austria: I-Tech, pp. 479–496.

Suh, S.H., Lee, B.E., Chung, D.H. and Cheon, S.U. (2003). Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Computer-Aided Design*, Vol. 35, No. 12, pp. 1069-1083.

Wu, X.W. (2006). Realisation of STEP-NC enabled machining. *Robotics and Computer-Integrated Manufacturing*, vol. 22, Elsevier, pp. 144–153.

Xu, X. W. and He, Q. (2004). Striving for a total integration of CAD, CAPP, CAM and CNC. *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 2, pp. 101-109.

Xu Xun (2007). STEP into Distributed Manufacturing with STEP-NC. *Process Planning and Scheduling for Distributed Manufacturing*, pp. 393-421, Springer London.

Zhang, H., Wang, J., Zhang, G., Gan, Z., Pan, Z., Cui, H. and Zhu, Z. (2005). Machining with flexible manipulator: toward improving robotic machining performance. *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1127-1132.

Zhu, X., Wang, Y. and Fu, H. (2006). A 3-D Simulation System for Milling Machining Based STEP-NC. *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, Dalian, China.

Paper 6: Achieving Total Immersion: Technology Trends behind Augmented Reality - A Survey

Gabor Sziebig, Achieving Total Immersion: Technology Trends behind Augmented Reality - A Survey. In: *SIMULATION, MODELLING AND OPTIMIZATION*. Budapest, Hungary, 03/09/2009-05/09/2009. WSEAS Press, pp. 458-463.(ISBN: 978-960-474-113-7)

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig.

The paper is a state-of-the-art review of current technological trends in augmented reality. From the time of publishing the paper already 11 independent citing articles were published.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing all the sections. The entire paper was thoroughly reviewed and revised by Øritsland Trond Are (Lecturer in PD8401 Interaction Design course).

Achieving total immersion:
Technology trends behind Augmented Reality - A survey

GABOR SZIEBIG
Narvik University College
Department of Industrial Engineering
Lodve Langes gate 2., 8514 Narvik
NORWAY
gabor.sziebig@hin.no

Abstract: Virtual Environments allow users to interact with virtual worlds, but usually these interactions are in front of a monitor or a projected wall. Augmented Reality brings the feeling of reality to these interactions. The user can combine the virtual world with the virtual world. Interact and feel the objects, together with artefacts that are generated by a computer. This article provides an overview and introduction to the problems of Augmented Reality systems and shows the state of the art in current technological developments. The usability bottlenecks of existing systems and probable solutions are also discussed.

Key–Words: Augmented Reality, Usability, Head-mounted displays, Trackers, Immersion

1 Introduction

Augmented Reality (AR) is a special type of Virtual Reality. It combines the real-world with computer generated virtual data [1]. This enables the programmers to create such a mixed environment, where virtual objects can be attached to real things or placed into an otherwise real environment. It is a special type of human-machine interaction, which utilizes the user's perceptual-motor skills in the real world. This is due to the tangible interface, which can be virtual or real. Simulation, training, assistance, telemanipulation and communication can be performed in a whole new way [2]. AR systems exist in many sizes: from a simple handheld device, that can show virtual objects superposed on real objects using a camera and the display of the device, up to world sized environments, where the user is tracked by global positioning system (GPS) and the real world and virtual data is shown overlaid on a head mounted display (HMD) based on the position of the user and the scene that can be seen from the HMD [3]. Each size has different level of reality feeling and immersion level. The scope of this paper is limited to technologies that let humans enter a virtual world without mayor restrictions and tabletop [4], handheld [5] or desktop [6] based AR systems are left out.

Until now true AR systems (that utilizes each feature of AR) are not available for commercial use. This is mainly, because of the different needs (system flexibility, type of user-system interaction, level of immersion, hardware and software elements) of each AR application and the lack of general framework and the developed AR systems were mainly a result of a research projects which focused on specified problems. AR is mainly a gathering name for the technologies that involves the following techniques: haptics, stereoscopic visualization, human-machine interaction, motion tracking, virtual reality. This already shows how widespread the field of AR systems is. Although there are successful demonstrations in different application fields and a huge interest in research fields, the big commercial breakthrough of AR systems is still waiting. Some of the successful applications are in the following domains: entertainment [7], maintenance [8], [9], manufacturing [10]-[14], medical [15], [16], military training [17], [18] and telerobotics [19].

In this article a typical setup of AR systems is discussed. In general, such a system consists of two layers: hardware and software. These will be introduced in the following subsections: in Section 2.1 the needed hardware elements (HMD, trackers, haptics) of an AR system are introduced, which is followed by the necessary software solutions in Section 2.2. In Section 3, these hardware and software trends and developments are summarized.

2 System components

As already mentioned in the introduction, the AR field stands on widespread technological foundation. It utilizes different hardware components (haptic devices, HMDs, monitors, motion capture equipment, etc.) with software components (image processing, object recognition, virtual word rendering, etc.). In the past ten years research focused on the general bottlenecks of AR systems: tracking of human position, tracking of real-world objects, synchronizing the virtual- and real-world, elimination of lags, jitters and drift caused by the hardware elements. Not only the technology advanced, but also the AR software interfaces [20]. The most widespread AR software platform is the ARToolkit [21], which will be introduced in Section 2.2, and the common hardware elements are cameras connected to video displays (see Section 2.1.1). In the next subsections the AR systems different components will be introduced.

2.1 Hardware

2.1.1 Head-mounted displays

There are two basic types of HMDs: optical- and video- see-through. In case of the optical see-through system the real-world and the computer generated image are mixed in an optical combiner (transparent display) in front of the user eyes. In the other case (video see-though) the real-world is captured by digital video cameras and the camera images are overlapped with computer generated images and displayed on monitors in front of the user's eyes.

Both technologies suffer from drawbacks and have a lot of technological difficulties. It is much easier to create AR system with a video see-through system, because the user only see, what the system generates based on the video cameras images, but it much harder to achieve the feeling of reality than in the optical see-through system. Also registration of the head position in both cases highly defines the overall performance of the system (this problem is also referred as Tracking of head motion and will be introduced in Section 2.1.2).

10 years ago, when augmented reality was just starting to spread around the world, there were only a few commercially available systems (MicroOptical, Minolta, Sony, MicroVision, etc.) and these systems have mainly disappeared by now [1]. As technology advanced, HMDs got more and more functions and better and better displays.Now the market consist of more than 38 different products, starting from entry level (low resolution, cheap) to experimental (high FOV, expensive) high-end systems. In general there are more video see-through (over 80%) than optical see-through systems, this is because of the expensive and technologically challenging optical combiner. The average resolution is 800*600 and the FOV is around 35 degrees, which is far less than is required for full immersion. Natural human vision has a field of view of up to 80 degrees without eye movement, although only 1 degree provides sharp sight. Humans have an oculomotor horizontal range of 55 degrees without moving the head [22].

Manufacturers are continually trying to make their equipment more user friendly, but not with much success. The reasons of this are the following:

- The average weight of high-end HMD is 700 grams, while normal reading glasses weight around 100 grams.
- The HMD needs fixed and known position relative to head, which can be only achieved through a helmet or a belt, this is usually uncomfortable.
- There are cables that run to the HMD, is usually not wireless, so the movement of the user is limited within the range of the wires.
- The optical see-though HMDs are more convenient to use and give higher feeling of immersion, but they do not provide the same FOV as the video see-through system.
- Display resolution is not high enough.

Until the above mentioned problems are not solved, the commercialization of the HMDs will be limited only to researchers or task specific applications.

2.1.2 Trackers

In order to match the virtual world with the real-world, the movement of the objects in both worlds must be tracked. In the virtual case everything is created in an artificial way, where positions are readily defined. It is much harder to follow changes in the real-world. For this reason different type of trackers are used: electromagnetic, mechanical, inertial, vision-based, ultrasound and hybrid systems (combination of two or more). Each type of tracker has different operation conditions and can be used for different type of tasks. The overall goal of each tracker is to provide high accuracy, low latency, low jitter and robustness. These are crucial for a successful AR application, as it depends on these hardware elements performances in calculating position and orientation for virtual objects and for the HMD virtual scene. Sometimes it is needed to track the whole human body motion, even including the fingers. As this type of tracking may involve several types of tracking technique it is not included in the comparison. Also the mechanical types are left out, as these are detailed in the next subsection.

Inertia tracker systems are typically compact systems where rate gyroscopes and accelerometers are mounted on an electrical circuit. Given the initial position (calibration position), velocity and orientation, the rate gyroscopes measure the angular velocities, from which the mathematical integral gives the angular displacements or orientation. The only problem is that, it can only serve with 3 degree of freedom information and does not react to very slow movement. Acoustic tracker, on the other hand, uses the time of flight of ultrasonic pulses for tracking. It uses the echo scheme or direct transmission and typically requires multiple sensors to triangulate the 3D position of the tracked object. The time of flight is very effective solution in case of no disturbing electrical equipment and no blocking objects. Electromagnetic tracker includes using infrared, visible (optical) and radio waves. Two configurations are possible, outside-looking-in, where the sensors are mounted in the environment tracking emitters on the user and inside-looking-out, where sensors are mounted on the user. The former scheme is typically used for motion capture in the entertainment industry while the latter is used in mobile applications including AR and machine vision in robotics. Magnetic tracker uses magnetic sensors to determine the intensity and direction of the magnetic field. Optical tracker uses one or more cameras and the computer vision technology to detect targets in the images and compute their position and orientation based on this information [3].

The trackers are usually mounted inside the HMD or glued on surfaces of objects or belted to the user. The ultimate solution would be some kind of hybrid system, but until now there is no general solution for tracking. Each AR application requires specific trackers and creates new solutions for this problem.

2.1.3 Haptics

AR systems without feedback are worth nothing. HMDs serve as visualization for the virtual world, but without having the capability of touching and feeling the weight of these virtual objects, there will never be realistic environment. For this purpose haptic devices serve as key elements in the feedback process. These haptics can feedback: force, heat, vibration. The 3D virtual objects will also have weight and it is transferred to the user through some kind of mechanical solution. It is known that our skin hosts four types of tactile receptors (mechanoreceptors). When these receptors are activated they emit small electrical discharges, which are detected by the brain. Two of the mechanoreceptors are slow-adapting and two are fast-adapting. The slow-adapting receptors respond to static force applied on the skin. The fast-adapting ones respond to vibrations and accelerations. Density of the receptors determines the spatial resolution of the skin. The fingertips, where the density of receptors is the highest, can discriminate two contacts much closer than, for example, the palm can. The minimum distance value for the fingertips is 2.5 millimetre apart, for the palm 11 millimetre apart. Contacts points closer than these values are perceived as a single contact point. Spatial resolution is complemented by temporal resolution when two contacts occur on the skin close in time. To be sensed as two successive ones, the contacts must not occur closer in time than 5 milliseconds. Heat sensing is realized by thermoreceptors sensitive to cold and warm. Their spatial resolution is less than the mechanoreceptors [23].

Force feedback is usually achieved with an arm stage, that also records the movement of arm and if needed can apply inverse forces, for slowing or acceleration of the arm. Heat is transferred to the skin with small heat pads, which can cool and warm the pads to a specified temperature. It is quite convenient to use these pads, as it has small size factor and easy integration to human skin. To use vibration (or tactile sensing) as a feedback, is a little bit more challenging task. It is mainly an alternative to force feedback, as it is much easier to wear a vibration glove that have small vibrating motors on each finger, than to connect the arm to a big mechanical stage. The hard part is to teach the brain that specific levels of vibration intensity refer to specific weights. As the weight in a normal environment is detected by different types of receptors, this results in a long learning period.

The usability of the arm stages in mobile AR systems (which are fixed to specific positions) is questionable and there is no mobile, lightweight and commercially available force feedback system, which can be mounted on humans. The only alternative is heat or vibration (tactile) feedback systems. There are already successful experiments, where small vibration motors are used as guidance for navigation [24] or for mapping different colours to specific vibrations [25] and these experiments show that humans can learn basic interactions very fast and efficiently, but the effectiveness of these vibration gloves is not promising [26]. One possible explanation for the failure of the haptics in AR could be that usually AR system developers use a top-down approach in application development: they look for efficient interaction interfaces for the given application rather than matching the physical and mental capabilities of the given interaction (e.g. touch) to the application. Also objects in Virtual Reality are usually too simplified and don't have enough properties for accurate haptic interaction and the lack of resistance in tactile feedback makes things feel unreal [27].

2.2 Software

As already seen at the hardware level of an AR system, there is no general solution for any kind of problem. Luckily on the software level, generalization is available. However, until now programs are mainly created from scratch and for specific applications. There are some pioneering activities that tried to generalize parts of an AR system. In order to use any kind of tracker, an open-source software library was developed called OpenTracker [28]. SenseGraphics, a haptic interface reseller, created an open-source haptic library, called H3D API [29]. The API is more than a simple haptic interface reader and writer library, it already includes some low level visualization commands.

Also several research groups have developed well-known AR software platforms that allow AR researchers to further develop specific-purpose AR applications. These solutions mainly build on existing open-source libraries (as mentioned above), or creates a brand new solution for the specific problem. The main contributors are the following: ARToolkit [21], Designer's Augmented Reality Toolkit (DART) [30] and Studierstube [31].

ARToolkit is a marker-based platform and it is the most well-known tool that has been extensively used for AR applications. It is an open-source and free platform containing video tracking libraries, which calculate the real camera position and orientation relative to physical markers in real time. This enables the easy development of a wide range of AR applications [21].

DART is based on Macromedia Director [32] and provides low-level support for the management of trackers, sensors, and cameras. DART allows designers to specify complex relationships between the physical and virtual worlds, and supports 3D animatic actors (informal, sketch-based content) in addition to more polished content. Designers can capture and replay synchronized video and sensor data, allowing them to work off-site and to test specific parts of their experience more effectively [30].

Studierstube is based on the OpenInventor [33] real-time rendering framework and OpenTracker is used to support a wide range of software and hardware, including various HMDs and trackers. This system allows the user to combine multiple approaches - augmented reality, projection displays, and ubiquitous computing - to the interface as needed. The environment is controlled by the Personal Interaction Panel, a two handed, pen-and-pad interface that has versatile uses for interacting with the virtual environment. Studierstube also borrows elements from the desktop, such as multitasking and multi-windowing. The resulting software architecture is a user interface management system for complex augmented reality applications [31].

Beside the standalone applications there are also mobile frameworks. One of these frameworks is the Distributed Wearable Augmented Reality Framework (DWARF). The application is shielded from the low-level services, such as for user interface or tracking hardware, and accesses these at a higher level of abstraction using the various DWARF services. It includes a special service which provides bootstrapping functionality and ”glue logic”. This provides the other services with models of the world and of tasks the user wishes to perform [34].

One another mobile framework is the Morgan AR/VR Framework. This supports the development and the usage of distributed multi-modal multi-user AR/VR applications and their customization to the individual requirements of the users and application scenarios. It was developed to provide full support for heterogeneous distributed environments and a large variety of individual input and output devices [35].

The above mentioned frameworks fill in the gap between the hardware and software level, making AR application development easy, but these are only visualization tools for creating an AR environment and beside a few initiatives (e.g. Studierstube Personal Interaction Panel [31]) not much focus is given on the interaction (system control) with the AR environment. Techniques which work well in 2D environments (e.g. menus, windows, text fields) mainly fail in direct 3D implementation [36]. The main reason for failure is that the spatial input is more complex (physical constraints, lack of feedback) than using a pointing device in 2D. Such a simple task, as selecting a menu item from a menu list, is not easy in 3D [37].

3 Conclusion

There is a huge potential in Augmented Reality systems, as the already successful demonstrations show, but the everyday use of this type of systems is far away. Within years the display technologies (like OLED) will become so advanced that a HMD will look like normal reading glasses and will have built in batteries and trackers. The communication with the virtual world will be wireless and the environment surrounding the human will carry the intelligence. Joining an AR world will be as easy as putting on glasses and interact with the virtual world with the person's own hands without any trackers on the body / hands. The technology development in the last 10 years already shows that this is achievable in the near future.

Due to advances in personal computing and rapid growth of high bandwidth internet, virtual worlds will reach everyone [38]. As already seen in 2003, the virtual world of Second Life has achieved remarkable success. Multinational enterprises and companies have invested millions of US dollars to participate in a virtual galaxy [39]. Until now Second Life is only a 3D application running on normal PCs using keyboard and mouse as interaction interface, with a few initiatives to extend its functionalities with haptics [40]. Next breakthrough could be in Augmented Reality. Already in Japan AR is promoted in different TV shows (e.g. Dennou Coil (Cyber Coil: Circle of Children)). The effect on everyday life is unimaginable: old, disabled people can live without constraints or there is no need to travel, just meet in the AR world, in your own living room. The impact of this type of virtual worlds in education is already under investigation [41], [42].

The first step toward this world is a standardized interface to AR worlds, a general approach to interaction with virtual objects. Usability of AR systems will highly depend on this issue.

Acknowledgements: The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

References:

[1] R. Azuma, A Survey of Augmented Reality, *Presence: Teleoperators and Virtual Environments* Vol. 6, No. 4, pp. 355–385, 1997.

[2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, Recent advances in augmented reality, *IEEE Computer Graphics and Applications*, Vol. 21, No. 6, pp. 34–37, 2001.

[3] S.K. Ong, M.L. Yuan and A.Y.C. Nee, Augmented reality applications in manufacturing: a survey, *International Journal of Production Research*, Vol. 46, No. 10, pp. 2707–2742, 2008.

[4] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto and K. Tachibana, Virtual Object Manipulation on a TableTop AR Environment, *In Proceedings of the International Symposium on Augmented Reality*, pp. 111–119, 2000.

[5] D. Schmalstieg and D. Wagner, Experiences with Handheld Augmented Reality, *In Proceedings of 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 3–15, 2007.

[6] S. Di Verdi, D. Nurmi and T. Hollerer, ARWin a desktop augmented reality Window Manager, *In Proceedings of Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 298–299, 2003.

[7] B. MacIntyre, M. Lohse, J.D. Bolter and E. Moreno, Integrating 2D video actors into 2D augmented-reality systems. *Presence: Teleoperators and Virtual Environments*, Vol. 11, No. 2, pp. 189–202, 2002.

[8] S. Bernd and L. de Blandine, An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context, *Journal of Winter School of Computer Graphics*, Vol. 11, No. 1, pp. 101–110, 2003.

[9] S. Goose, S. Sudarsky, X. Zhang and N. Navab, SpeechEnabled Augmented Reality Supporting Mobile Industrial Maintenance, *IEEE Pervasive Computing*, Vol. 2, No. 1, pp. 65–70, 2003.

[10] W. Barfield, K. Baird, J. Shewchuk and G. Ioannou, Applications of wearable computers and augmented reality to manufacturing, *In Proceedings of Fundamentals of Wearable Computers and Augmented Reality*, pp. 695-724, 2001.

[11] S.K. Ong and A.Y.C. Nee, *Virtual and Augmented Reality Applications in Manufacturing*, Springer, p. 388, 2004.

[12] H. Regenbrecht, G. Baratoff and W. Wilke, Augmented reality projects in the automotive and aerospace industries, *IEEE Computer Graphics and Applications*, Vol. 25, No. 6, pp. 48–56, 2005.

[13] F. Doil, W. Schreiber, T. Alt and C. Patron, Augmented reality for manufacturing planning, *In Proceedings of the workshop on Virtual environments*, pp. 71–76, 2003.

[14] W. Dangelmaier, M. Fischer, J. Gausemeier, M. Grafe, C. Matysczok and B. Mueck, Virtual and augmented reality support for discrete manufacturing system simulation, *Computers in Industry*, Vol. 56, No. 4, pp. 371–383, 2005.

[15] M. Rosenthal, A. State, J. Lee, G. Hirota, J. Ackerman, K. Keller, E.D. Pisano, M. Jiroutek, K. Muller and H. Fuchs, Augmented Reality Guidance for Needle Biopsies: A Randomized, Controlled Trial in Phantoms, *Medical Image Analysis*, Vol. 6, No. 3, pp. 313–320, 2002.

[16] F.K. Wacke, S. Vogt, A. Khamene, J.A. Jesberger, S.G. Nour, D.R. Elgort, F. Sauer, J.L. Duerk and J.S. Lewin, An Augmented Reality System for MR Image-guided Needle Biopsy: Initial Results in a Swine Model, *Radiology*, Vol. 238, pp. 497–504, 2006.

[17] S. Julier, M. Lanzagorta, Y. Baillot, L. Rosenblum and S. Feiner, Information filtering for mobile augmented reality, *In Proceedings of IEEE*

and ACM International Symposium on Augmented Reality (ISAR 2000), pp. 3–11, 2000.

[18] M.A. Livingston, L.J. Rosenblum, S.J. Julier, D. Brown, Y. Baillot, J.E. Swan II, J.L. Gabbard and D. Hix, An Augmented Reality System for Military Operations in Urban Terrain, *In Proceedings of the Interservice / Industry Training, Simulation, & Education Conference (I/ITSEC '02)*, pp. 89–96, 2002.

[19] M. Utsumi, T. Hirabayashi and M. Yoshie, Development for teleoperation underwater grasping system in unclear environment, *In Proceedings of International Symposium on Underwater Technology*, pp. 349–353, 2002.

[20] J.E. Swan and J.L. Gabbard, Survey of User-Based Experimentation in Augmented Reality, *In Proceedings of 1st International Conference on Virtual Reality*, pp. 1–9, 2005.

[21] M. Billinghurst and H. Kato, Collaborative Augmented Reality, *Communications of the ACM*, Vol. 45, No. 7, pp. 64–70, 2002.

[22] D. Guitton and M. Volle, Gaze control in humans: eyehead coordination during orienting movements to targets within and beyond the oculomotor range, *Journal of Neurophysiology*, Vol. 58, No. 3, pp. 427–459, 1987.

[23] G.C. Burdea and P. Coiffet, *Virtual Reality Technology*, WileyIEEE, p. 464 2003.

[24] H. Uchiyama, W.D. Potter, M.A. Covington, J. Tarver and R. Eunice, Perceptual Navigation for SemiAutonomous Wheelchair Operations, *In book Service Robot Applications*, Ed. Yoshihiko Takahashi, InTech Education and Publishing, pp. 71–94, 2008.

[25] J. Tapson, J. Diaz, D. Sander, N. Gurari, E. Chicca, P. Pouliquen and R. Etienne-Cummings, The Feeling of Color: A Haptic Feedback Device for the Visually Disabled, *In Proceedings of IEEE Biomedical Circuits and Systems Conference*, pp. 381–384, 2008.

[26] D. Clark and M. Bailey, VirtualVirtual Haptic Feedback, and Why it Wasn't Enough, *In Proceedings of SPIE Visualization and Data Analysis Conference*, SPIE Volume 4665, pp. 308–318, 2002.

[27] K.E. MacLean, Haptic Interaction Design for Everyday Interfaces, *Reviews of Human Factors and Ergonomics*, Vol. 4, pp. 149–194, 2008.

[28] G. Reitmayr and D. Schmalstieg, OpenTracker: A flexible software design for threedimensional interaction, *Virtual Reality*, Vol. 9, No. 1, pp. 79–92, 2005.

[29] H3D API, `http://www.h3dapi.org/`, *Online*.

[30] B. MacIntyre, M. Gandy, S. Dow and J.D. Bolter, DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences, *In Proceedings of Conference of User Interface Software and Technology (UIST'04)*, pp. 1–10, 2004.

[31] D. Schmalstieg, A. Fuhrmann, G. Hesina, Zs. Szalavari, L.M. Encarnacao, M. Gervautz and W. Purgathofer, The Studierstube Augmented Reality Project, *Presence*, Vol. 11, No. 1, pp. 33–54, 2002.

[32] Macromedia Director, `http://www.adobe.com/products/director/`, *Online*.

[33] Open Inventor, `http://oss.sgi.com/projects/inventor/`, *Online*.

[34] M. Bauer, K. Bruegge, G. Klinker, A. MacWilliams, T. Reicher, S. Riss, C. Sandor and M. Wagner, Design of a ComponentBased Augmented Reality Framework, *In Proceedings of the International Symposium on Augmented Reality (ISAR)*, pp. 1–10, 2001.

[35] J. Ohlenburg, W. Broll and A.K. Braun, MORGAN: A Framework for Realizing Interactive RealTime AR and VR Applications, *In Proceedings of IEEE VR 2008 Workshop on Software Engineering and Architecture for Realitme Interacitve Systems*, pp. 27–30, 2008.

[36] C. Hand, A Survey of 3D Interaction Techniques, *Computer Graphics Forum*, Vol. 16, No. 5, pp. 269–281, 1997.

[37] R. Dachselt and A. Hubner, Threedimensional menus: A survey and taxonomy, *Computers & Graphics*, Vol. 31, No. 1, pp. 53–65, 2007.

[38] J. Wu, P. Li, S. Rao, Why they enjoy virtual game worlds? An empirical investigation, *Journal of Electronic Commerce Research*, Vol. 9, No. 3, pp. 219–230, 2008.

[39] S. Kish, Second Life: Virtual Worlds and the Enterprise, *an essay*, pp. 1–13, 2007.

[40] M. de Pascale, S. Mulatto, and D. Prattichizzo, Bringing Haptics to Second Life for Visually Impaired People, *Eurohaptics 2008, Volume 5024 of lecture Notes in Computer Science*, pp 896–905. Springer Verlag, 2008.

[41] J. Kirriemuir, Measuring the Impact of Second Life for Educational Purposes, *Responses and Second Life meeting transcript*, pp. 1–32, 2008.

[42] V. Fedák, P. Bauer, V. Hájek, H. Weiss, B. Davat, S. Manias, I. Nagy, P. Korondi, R. Miksiewicz, P. van Duijsen, P. Smékal, Interactive E-Learning in Electrical Engineering, *In Proceedings of International Conference on Electrical Drives and Power Electronics (EDPE'03)*, pp. 368–373, 2003.

Paper 7: Vibro-tactile feedback for VR systems

Gabor Sziebig, Bjørn Solvang, Csaba Kiss, Peter Korondi, Vibro-tactile feedback for VR systems. In: *2nd Conference on Human System Interactions (HSI '09)*. Catania, Italy, 21/05/2009-23/05/2009. (IEEE)pp. 406-410.(ISBN: 978-1-4244-3959-1)

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Gabor Sziebig, Bjørn Solvang, Csaba Kiss (at that time master student at NUC) and Peter Korondi.

The mechanical and functional design of the vibration glove is a cooperative effort from Csaba Kiss and Gabor Sziebig. The PC based implementation of the design is entirely a contribution of Csaba Kiss. The physical implementation of the vibration glove is a cooperative effort from Csaba Kiss and Gabor Sziebig. Csaba Kiss and Gabor Sziebig performed the overall system tests in close collaboration.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing almost all the sections in close collaboration with Csaba Kiss. Csaba Kiss produced the graphics and pictures through the paper (except figure 1, which is done by Gabor Sziebig). The entire paper was thoroughly reviewed and revised by Bjørn Solvang and Peter Korondi.

Vibro-tactile feedback for VR systems

Gabor Sziebig†, Bjorn Solvang†, Csaba Kiss‡, and Peter Korondi‡
†Department of Industrial Engineering, Narvik University College, Narvik, Norway
‡Department of Automation and Applied Informatics, Budapest University of Technology and Economics, Budapest, Hungary

Abstract— **Today, in production engineering we have already taken into usage complex virtual representations of the manufacturing environment were we can program, simulate, analyse and optimise key performances. Such systems are mainly working through keyboard/ mouse input while user feedback normally are direct visual through computer screen and text files. Such communication only relays on a limited spectre of our senses and the keyboard/mouse/screen systems cannot be said to be very human friendly, especially when the operator is expected to operate "out of office" in a unstructured, maybe dangerous or dirty, environment. In the near future we expect to see more advanced input/output devices like motion capturing and speech input systems while feedback will not be visual alone but include several other human senses. Combinatorial sensory information, interpreted by the human brain, becomes very handy when a stand alone sense is not enough to interpret the actual situation.**

Imitation of senses and feedback from virtual reality environments always meant a great problem. Realization of two (sight and hearing) out of the five human senses is indispensable. Simulation of these two senses is not complex issue; on the contrary the other three are quite challenging. This paper describes the development of a vibro-tactile glove which can provide sensory feedback from a virtual environment, either as a stand alone system but most important in combination with sight and audio feedback systems. Instead of implementing real force feedback, the focus is on tactile sensing, as an alternative way of achieving the same feedback. The glove contains six vibration motors on different locations on the hand. These locations include all five fingers, and the palm. Communication with the glove is wireless, enabling free movement for the user. The system is low cost and small sized which allows for combining it with advanced input devices like a motion capturing suit.

***Index Terms*—Virtual reality, Feedback, Sensing**

I. Introduction

THE term "virtual reality" officially appeared for the first time in the Oxford English Dictionary in 1987 [1]. Basically it is a technology that enables the user to enter a virtual environment generated and simulated by a computer. This environment can be anything that is imaginable by the human fantasy. Nowadays, technology is orientated around the visualization of a virtual world [2]. Usually it is done by a computer screen, or a head-mounted stereoscopic display. Stereoscopic view means that the left, and the right eyes receive different images from the display, corresponding to the distance between them. With utilizing this technology the perception of depth is achieved [3]. Based on these images the human brain is able to build up an authentic 3D world. Displays built into the head-mounted helmets will soon have such quality that makes differentiating the virtual world from the real one nearly impossible. Some simulators are already capable of transmitting sounds as additional sensory information. A technology called holophony is being developed to generate real like sound effects [4]. Holophony uses the spatial audio rendering technique called wave field synthesis, which produces artificial wave fronts by an array of speakers. Hereby sounds can be generated to seem to originate from any direction.

The technology of virtual reality is mostly used in military training. It gives the opportunity to place trainees into real-like combat simulations, which were impossible before [5]. The more realistic the simulation is, the more effective the training is.

Another important application field of virtual reality is healthcare [6]. Medical students can perform virtual operations, without any risk to the patient. VR can be a valuable asset in exposure therapy treatments. Recently it has been used successfully to cure various forms of phobias and posttraumatic stress disorder [7].

As real 3D (or even the so-called 4D) cinemas are spreading worldwide, the role of virtual reality in entertainment is more and more increasing. Tracked 3D joysticks and head mounted displays are available on the market for almost all game systems (e. g.: Nintendo, PlayStation) to make the gamers' experience more realistic.

Also, as the technology mature we expect to see more industrial application related to advanced virtual reality environments. At the manufacturing scene we expect that virtual reality systems will play a more vital role. Rapid set-up and initial programming of manufacturing cells as well as advanced process feedback can be achieved through an advanced virtual environment.

In this paper focus is put on process feedback through the development of a vibro-tactile glove. At the present time gloves with force feedback are already available on the market [8]-[11], but these are expensive and sometimes uncomfortable to use, because of their big size factor. These devices mainly contain large mechanic or hydraulic components to achieve force feedback effect. These components are heavy, thus the free movement of the user is

Corresponding author: Bjorn Solvang, Department of Industrial Engineering, Narvik University College, Lodve Langes gate 2., 8514. Narvik, Norway; phone: +47-7696-6227; fax: +47-7696-6814; e-mail: bjs@hin.no

limited. Some products need extra staging to operate [10].

The organization of the paper is as follows: Section II gives an overview on related research while section III presents details of system components. Section IV discusses some initial experiments. Section V concludes the paper while VI gives some recommendations for future work.

II. Related Research

Basically, there are three types of receptors in the skin responsible for transferring mechanical solicitation [12]:

- Merkel's receptors: they have high spatial resolution, but the adopting rate is slow.
- Pacini's corpuscles: they can detect very rapid vibrations, and their adopting rate is quick.
- Meissner's corpuscles: they are quickly adopting receptors located in the glabrous skin (fingers, lips) with high spatial resolution.

The information gathered by the receptors is transferred to the brain via the fastest communication channel in the human body: the dorsal-lateral column way, where the information travels at the speed of 100 m/s.

The sensing resolution of the skin is variable on the human body, it has been examined since the year of 1826 [13]. The most precise points are on the fingers, where the spatial resolution is of the order of millimetres.

Vibro-tactile devices have been used to help blind or deaf people by providing an extra sensing channel [14] - [15]. The sensitivity of stimulation depends on the following [16]:

- The age of the subject.
- The body position of the subject.
- The frequency of the vibration. The most effective frequencies are from 50 to 300 Hz.
- The sensation is also modified by the tissue beneath the stimulated point (fat, bone, muscle) and body temperature.

III. System Components

The tactile feedback system is built up from the following four components:

- Gloves (vibration motors are placed in this)
- Controller board (Microcontroller driven)
- Wireless communication (In this case Bluetooth)
- Remote device (e.g.: desktop PC, mobile phone, PDA)
- Software running on remote devices (For instructions)

An overview of these components can be seen in Fig. 1. In the following, these will be introduced.

A. Glove

The glove was designed in such a way, that it is compatible with a motion capture system (i.e. Measurand ShapeHand [17]). This makes the system to be able to receive data about the user's hands and to transmit feedback at the same time. To achieve this, the feedback glove is a modified plastic glove, which must be worn on the top of the ShapeHand motion capture device (as seen in Fig. 2. (a)). Fig. 2. (b) shows the motor emplacements on the glove, while Fig. 2. (c) shows the wiring.

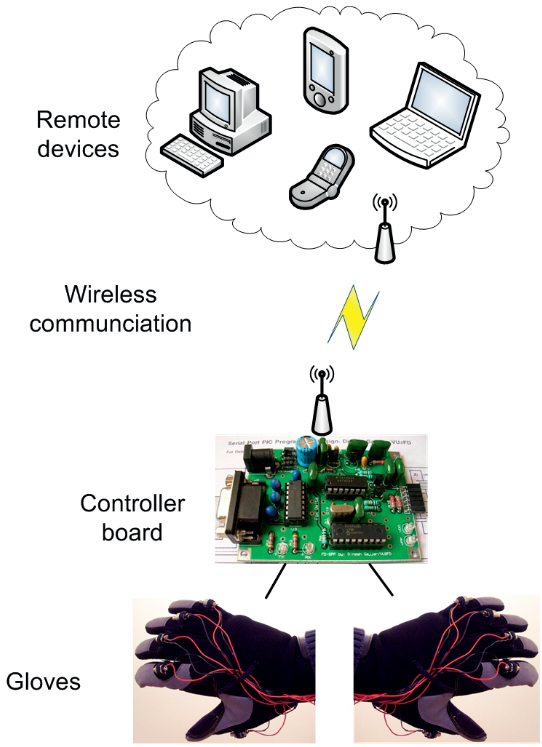
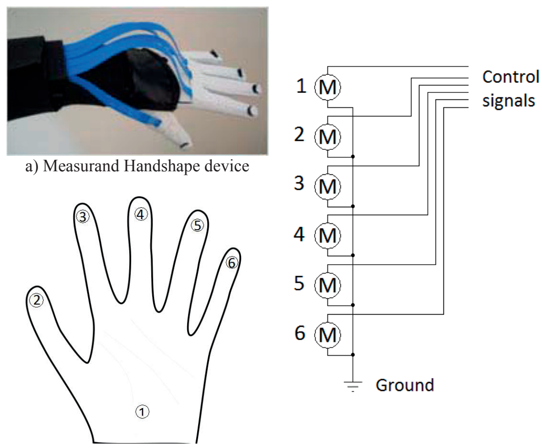


Fig. 1. Tactile feedback system overview

B. Controller board

The designed printed circuit board serves as the mobile controller for the gloves. It communicates with the remote



a) Measurand Handshape device

b) Motor emplacement in the glove

c) Wiring of the glove

Fig. 2. Glove setup

devices via a wireless Bluetooth module.

The motors are controlled by the digital outputs of the microcontroller with the help of transistors. The microcontroller transmits the driving waveform (pulses) to the base of the transmitter. At high values the collector-emitter circuit becomes open. The motors are connected into the transistors' collector-emitter circuit.

When stopping or starting the motors can easily generate some inverse spikes. To avoid damage caused by these, extra diodes are placed into the circuit.

Both the microcontroller and the Bluetooth module have Universal Asynchronous Receiver Transmitter devices (UART) integrated.

The main purpose of an UART is to translate between serial and parallel forms of data. It is usually a separable integrated circuit, but recently it is common to include it in microcontrollers. On both sides of the communication the devices have to produce the same baud (signals per one second) to their UARTs to be capable of transmitting and receiving.

The microcontroller unit communicates with the Bluetooth module with the aid of RS-232 standard. To make this connection possible an extra chip had to be placed in. It provides the necessary voltage conversion between the standard RS-232 levels and the TTL voltages used in the microcontroller.

C. *Wireless communication*

The stationary controller communicates with the remote device using Bluetooth's Serial Port Profile (SPP). Basically there are two types of messages used by the system: one is for controlling the gloves, and one is for handling connection.

Messages received from the controller are interpreted by the micro controller unit, and based on these messages it drives the motors on the gloves. In addition, there are messages transmitted by the Bluetooth module to the micro controller unit. These contain information on the radio connection.

Connection messages

The list of the connection messages and their purposes is shown in the Table 1.

TABLE 1
BLUETOOTH CONNECTION MESSAGES

Message	Description
RING 123456789012	When the lower layer detects an incoming call this message is sent to the host once. 123456789012 is the Bluetooth address of the remote device.
CONNECT 123456789012	When the connection is accepted, this message is sent to the host.
NO CARRIER	This message is sent when the connection is terminated.

Control messages

Control messages are sent from the stationary controller to microcontroller via the Bluetooth module. The first part of the message selects the motor to drive; the second part contains the desired power. For proper synchronizing every frame starts with a start indicator.

The build-up of a frame is shown in the Table 2. and 3.

TABLE 2
CONTROL MESSAGE FRAME

0.bit	7.	8.	15.	16.	23.
Start indicator		Motor number		Desired power level	

TABLE 3
CONTROL MESSAGE INFORMATION

Part	Value or range	Description
Start indicator	224 (Alpha in Greek alphabet)	Indicates the start of the frame for synchronization purposes
Motor number	0-12	Selects a motor, whose power will be set
Power	0-100	The power of the selected motor in percentages

Each motor has a number assigned to it. The special number 0 selects all motors on both gloves. This makes the communication more compact in case of swift hand movements, or when turning off the device.

The assigned numbers are listed in Table 4.

TABLE 4
MOTOR NUMBERS

Motor number	Glove	Position
0	Left and Right	Select all twelve motors
1	Right	Palm
2	Right	Thumb
3	Right	Index finger
4	Right	Middle finger
5	Right	Ring finger
6	Right	Little finger
7	Left	Palm
8	Left	Thumb
9	Left	Index finger
10	Left	Middle finger
11	Left	Ring finger
12	Left	Little finger

D. *Software environment*

Software components are essential for proper operation of the system. Basically two programs are needed to run the system:

- Microcontroller's program
- Program on the remote device

The software in the microcontroller is responsible for receiving Bluetooth messages and generating PWM output for the motors.

There are no restrictions on controller software (assumed that the communication algorithm is implemented): it can be written in any programming language, it can run on any Bluetooth capable device. For easier future integration a Microsoft .NET dynamic link library and a Java class is available. These classes provide methods to open and close Bluetooth communication channels towards the glove device, and to set power levels on each and every motor.

Microcontroller's program

The program used to operate the MCU is written in C language and compiled with MPLAB C18 official compiler.

Initialization

The initialization section is responsible for setting up the unit for proper operation. First of all, the internal oscillator's frequency is set to 8MHz. Then the serial port and interrupts (both global and receive) are enabled and the USART is opened using the previously described communication parameters. After that, all ports are set to output, and values are pulled down to zero.

Receive interrupt handler routine

The handler routine is called every time a character is received by the USART component. It stores the necessary information in proper global variables.

The main loop

The main loop handles the generation of the PWM waveform to drive the motors, and it is responsible for blinking the indicator light emitting diodes.

The driving signal is produced by a cycle that runs one hundred times. While the value of the cycle counter is less than the power level of the engine, the according output is raised to logical high, thus enabling to control the power using percentages. Since the motors don't start spinning under 25%, values are increased to make controllability smoother.

Program on the remote device

The software that is running on the controller is responsible for transmitting the desired power levels to the control board. The main requirements for being able to achieve this are the capability of transmitting on Bluetooth channels, and the implementation of the used communication algorithm.

E. Demo Applications

Demo applications for desktop PC (.NET based) and mobile phones (JAVA based) are available. In these programs the power level of each motor (for one glove) can be adjusted between 0% and 100%. Both applications use the same Application Protocol Interface (API) libraries to open, maintain and close connections between the remote devices and the controller board and to set power levels on the motors.

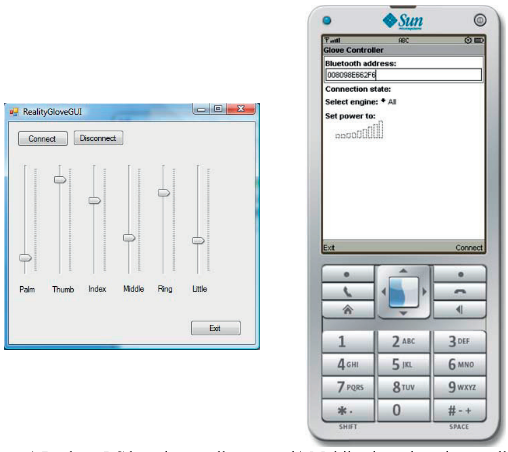
The user interfaces can be seen in Fig. 3.

F. Validation

This section includes calculations and justifications on the used clock frequencies. In general a graphical application provides 30 to 60 frames per second. The glove is controlled by the physical engine of the graphical application, which runs at least 10 times in a second, depending on the complexity of the program. Between two runs of the physical engine positions are linearly interpolated for proper display.

G. Clock frequency

The microcontroller unit's internal oscillator's frequency is set to 8MHz. On cycle (The microcontroller is capable of completing one basic instruction in every cycle) is done in $\frac{1}{8} * 10^{-6}$ seconds. The cycle for generating PWM output for the motors contains no more than 50 instructions. This cycle runs



a) Desktop PC based controller b) Mobile phone based controller

Fig. 3. Demo application user interfaces

100 times in one cycle of the main loop. All the necessary data is available to compute the duration of generation of the one value in the driving waveform t_{PWM}:

$$t_{PWM} = \frac{1}{8} * 10^{-6} * 50 * 100 = 0.625\,ns \quad (1)$$

Using the selected clock frequency of 8MHz the microcontroller is capable of transmitting 1600 changes in one second – excluding interrupt requests.

H. Communication speed

The baud rate for the wireless communication is set 9600 bits per second (bps).

Communication speed: 9600 bps = 1200 byte per second.

The size of one frame: 3 byte.

Frames transmitted in one second: 1200 / 3 = 400.

This means that 400 changes can be made in one second for all the motors.

For one motor: 400 / 12 = 33.33 changes.

This means that the system is capable of transmitting at least 30 changes to one finger of the user.

IV. Experimental Results

A. Sensing of power

Experiments show that a small change in percentage (3-4 %) is noticeable by the user when only one motor is operating. Above 75% (60% means normal operation at 3 Volts) power level the vibration feels uncomfortable, which means intense effects are possible to create.

Even working on the lowest power level, when the motor is operating creates a sensible vibration.

B. Reaction time

There is no noticeable delay between setting the desired power on the user interface and feeling the vibration. Further testing with visualization software is necessary for being able to measure reaction time with higher precision.

C. *Distinction of vibrating motors*

It is not easy to differentiate which motor is vibrating on neighbouring fingers. Small differences between the power levels of neighbouring motors are hard to detect. Although, it is assumed, that with further and more frequent usage of the glove, the vibro-tactile sensation can be improved greatly.

V. Conclusion

The development cost of the glove has been very low, in fact we will be able to produce it for less than 100$ even on such an early stage of development.

The glove operates with a wireless connection and allows the user to move freely around in a possible unstructured working environment.

The glove can be combined with a motion capture suit to form a very compact input/output device for effective human machine interaction. Even not focused in this publication a fingertip version of the glove is being developed to provide even lesser size and easy integration with possible input devices.

Some initial experiments shows promising results related to the sensibility of vibration signals.

The combinatorial feedback from sight, hearing, audio provide us a promising environment for testing different application for this vibro-tactile system.

VI. Future Work

The combination of the glove with a motion capture device would open new possibilities in usability and the glove would be able to serve as a major feedback device. The input for the glove would be a virtual word, where software has to possess a physical engine to calculate the pressure on several points of the hands and has to communicate with the remote device. The power levels for the motors can be linearly interpolated between two physical calculation steps as positions used for rendering.

For example, the user will be able to bounce a ball and feel it every time it hits back. Another great experience would be to hold a bucket of water in the hands and make circles with it. The user could be able to feel the flow of to water as it makes rounds on the inner wall of the bucket. Of course the physical modelling of water is a very complex task.

In the future, laboratories could be established far away from each other and people could meet in any kind of virtual environment. Persons in these remote rooms could interact with each other, and equipments far away from their positions. Also a handshake between two people in the virtual world could be felt.

Multinational companies and corporate enterprises would be able to organize meetings in a virtual environment without having the need of personal presence, but maintaining personal contact with the use of this technology. Thus travelling cost, and what is more important, the time used on travelling, could be easily saved.

Also the developed system enables the possibility to perform a research about human reaction to vibro-tactile simulation. This could investigate the speed of reaction as well as how close the real life tactile sensing and vibro-tactile stimulated sensing are to each other (how effective the stimulation is) and how fast the user can adapt to the stimulation.

Acknowledgement

The authors wish to thank Jostein Øysæd for his work on the initial design of the vibro-tactile glove during his M.Sc. study at Narvik University College.

The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

References

[1] F. P. Brooks, “What’s real about virtual reality?” *IEEE Comput. Graph. Appl.*, vol. 19, no. 6, pp. 16–27, 1999.

[2] T. Kanade, P. Rander, and P. Narayanan, “Virtualized reality: Constructing virtual worlds from real scenes,” *IEEE Multimedia*, vol. 4, no. 1, pp. 34–47, 1997.

[3] P. Narayanan, P. Rander, and T. Kanade, “Constructing virtual worlds using dense stereo,” in *Proc. Sixth International Conference on Computer Vision, 1998, pp. 3–10.*

[4] S. Uosukainen, “Active sound scatterers based on the jmc method,” *Journal of Sound and Vibration*, vol. 267, no. 5, pp. 979–1005, 2003.

[5] E. Ritter and M. Bowyer, “Simulation for trauma and combat casualty care,” *Minimally Invasive Therapy & Allied Technologies*, vol. 14, no. 4-5, pp. 224–234, 2005.

[6] K. R. Rosen, “The history of medical simulation,” *Journal of Critical Care*, vol. 23, no. 2, pp. 157–166, 2008.

[7] M. Krijn, P. M. G. Emmelkamp, R. P. Olafsson, M. Bouwman, L. J. Van Gerwen, P. Spinhoven, M. J. Schuemie, and C. A. P. G. Van der Mast, “Fear of flying treatment methods: Virtual reality exposure vs. cognitive behavioral therapy,” *Aviation Space and Environmental Medicine, vol. 78, no. 2, pp. 121–128, 2007.*

[8] *CyberTouch Tactile Feedback for the CyberGlove System*, 1st ed., Immersion Corporation, 2007.

[9] *CyberGlove II Wireless Glove*, 1st ed., Immersion Corporation, 2008.

[10] *CyberForce Force Feedback System*, 2nd ed., Immersion Corporation, 2007.

[11] *Shadow Dextrous Hand*, 1st ed., Shadow Robot Company, 2003.

[12] E. Weber, *Tactile Senses*. Psychology Press (UK), 1996.

[13] ——, *The Sense of Touch*. Academic Press for Experimental Psychology Society, 1978.

[14] B. Richardson and M. Symmons, “Vibrotactile devices for the deaf: are they out of touch?” *Ann Otol Rhinol Laryngol Suppl*, vol. 166, pp. 458–61, 1995.

[15] R. Kuc, “Binaural sonar electronic travel aid provides vibrotactile cues for landmark, reflector motion and surface texture classification,” *IEEE Trans. Biomed. Eng.*, vol. 49, no. 10, pp. 1173–1180, 2002.

[16] R. Cholewiak and A. Collins, “Vibrotactile localization on the arm: Effects of place, space, and age,” *Perception & Psychophysics*, vol. 65, no. 7, pp. 1058–1077, 2003.

[17] ShapeHand motion capture glove, Measurand Inc. [Online], http://www.motion-capture-system.com/, Available: 2009.01.15.

Paper 8: Control of an Embedded System via Internet

Gabor Sziebig, Bela Takarics, Peter Korondi, Control of an Embedded System via Internet. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS* 57:(10) pp. 3324-3333. (2010) IF: 3.439

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Gabor Sziebig, Bela Takarics (PhD candidate at Budapest University of Technology and Economics, Hungary) and Peter Korondi.

The paper is an overview paper of solutions for control theory education over internet. The paper also presents an e-learning material for home and distance learning which is entirely a contribution from Bela Takarics and Peter Korondi. The design of the internet based measurement laboratory is a cooperative effort from Gabor Sziebig and Peter Korondi.

The PC based implementation and hardware setup of the internet based measurement laboratory is entirely a contribution from Gabor Sziebig. Tests were carried out through the internet, while the physical measurement device is situated at the Department of Mechatronics, Optics and Applied Informatics, Budapest University of Technology and Economics, Hungary.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing partly Section 1, and 2, entirely Section 3 and 4 and produced the graphics and pictures through the paper (except Section 2, which are produced by Bela Takarics). The entire paper was thoroughly reviewed and revised by Peter Korondi.

Is not included due to copyright

Paper 9: Cog Framework - 3D Visualization for Mobile Robot Teleoperation

Gabor Sziebig, Peter Zanaty, Peter Korondi, Bjørn Solvang, Cog Framework - 3D Visualization for Mobile Robot Teleoperation. *ADVANCED MATERIALS RESEARCH* 222: pp. 357-361. (2011)

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig. The working principles and conceptual design was a cooperative effort from Gabor Sziebig, Peter Zanaty (PhD student at NUC), Peter Korondi and Bjørn Solvang.

The functional design of the cog framework is a cooperative effort from Peter Zanaty and Gabor Sziebig. The concept of mobile robot teleoperation is entirely a contribution from Peter Zanaty and it is continuation of his Master of Science thesis (carried out under supervision of Gabor Sziebig). The PC based implementation of the design is entirely a contribution of Peter Zanaty.

The testing of the cog framework is cooperative effort from Gabor Sziebig and Peter Zanaty. The tests were carried out at the integrated contusions facilities at Narvik University College.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing almost all the sections in close collaboration with Peter Zanaty. Peter Zanaty produced the graphics and pictures through the paper. The entire paper was thoroughly reviewed and revised by Bjørn Solvang and Peter Korondi.

Cog Framework - 3D Visualization for Mobile Robot Teleoperation

Gábor Sziebig[1, a], Péter Zanaty[2,b], Péter Korondi[3,4,c] and Bjørn Solvang[1,d]

[1]Narvik University College, Department of Industrial Engineering, Lodve Langes gate 2, 8514 Narvik, Norway

[2]Narvik University College, Department of Scientific Computing, Electrical Engineering and Space technology, Lodve Langes gate 2, 8514 Narvik, Norway

[3]Budapest University of Technology and Economics, Department of Mechatronics, Optics and Applied Informatics, Po. Box 91., 1521 Budapest, Hungary

[4]Computer and Automation Research Institute, 1111 Budapest, Hungary

[a]gabszi@hin.no, [b]pza@hin.no, [c]korondi@sztaki.hu, [d]bjs@hin.no

Keywords: 3D stereoscopic visualization, motion capture, mobile robot control, teleoperation

Abstract. The paper presents a modular extensible system which is relying on top of modern open source libraries. A multi-layer mobile robot controller unit has been created and tested successfully to be able to work with different types of mobile robot agents. The system handles a motion capturing suit and adapts it similarly as traditional peripheries. Robust posture recognition has been introduced on top of the motion suit adapter, which is used to instruct a mobile robot agent, while immerse stereographic feedback is provided to the human operator.

Introduction

The main goal of this paper is to create a framework focused on extending the traditional human-system interaction by applying motion capturing as an input source and stereographic visuals as a feedback to the human operator. The paper focuses on exploring new possibilities through the non-conventional channels, keeping reasonability as a main concern. Whether we consider mobile robots, mobile or smartphones, personal computers, cars, the surrounding interactive and sociable technology is advancing rapidly. The long term goal is to establish a robotized environment and to help the integration and communication between these tools in order to catalyze the renaissance of motorized/computerized environments shifting them towards being more human centric.

Nonverbal communication

Nonverbal communication is the process of communication through sending and receiving messages without using verbal signs such as words. Humans use their nonverbal channel more dominantly than one would naturally notice it. The earliest research on this field was done in [1], which concluded that "It is suggested that the combined effect of simultaneous verbal, vocal, and facial attitude communications is a weighted sum of their independent effects - with coefficients of .07, .38, and .55, respectively.", which means that nonverbal communication is found out to be very important even between humans.

These types of communications are often divided into different types, namely:

Posture refers the intentional or habitual position of the human body, can also refer to the proximity of other objects and in this way it is related to proxemics, which deals with the comfort zones of individuals.

Gesture is a specific movement of human body with contextual meanings. Gestures can express a wide variety of things; an interesting theory is the existence of universal microexpressions [2].

Haptics is dealing with the study of touching as form of nonverbal communication.

Oculesis is investigating the role of eyes in nonverbal communication, such as eye gaze, pupil dilation and blink rate.

Paralinguistics deals with the non-verbal part of human speech, which includes the pitch, and volume, as well as prosodic features such as rhythm, intonation and stress. It is also examining the details of human sounds which are not considered to belong to the human speech.

Nonverbal communication can be transmitted through object communication such as clothing, hairstyles or even architecture, symbols, infographics and much more.

In human-system interaction, where the communicating parties are humans and robots, we should take these means of nonverbal communication into serious consideration. Especially, when the objective is to found the base communication channels between humans and robots.

Motion capturing

Motion capturing is the process of recording human movement, and mapping this data to an anatomical model of the human. Hand gesture recognition, as one of pattern recognition and analysis problems, is so important that the motion of human hands can provide abundant information of human intention and implicit meaning to the machines in real world. Many reports on intelligent human machine interaction using hand gesture recognition have already been presented in [3-5], which can be mainly divided into Data Glove-based and Vision-based approaches.

Posture versus gesture recognition

Postures can be recognized to examine only spatial information, as opposed to being able to cope with gestures one should also take time into consideration. Most of the studies in posture and gesture recognition are targeting hand recognition, for more detailed information about recognizing facial expressions see [6].

Haptic feedback

The word haptic means: regarding to the sense of touch. Haptic technology interfaces to the user by applying forces and/or vibrations back. This mechanical stimulation may be used to provide feedback through conventional channels, which can enhance the remote control of systems.

Virtual reality

It is a generally accepted fact, that human vision is an extremely powerful information processing system, and that humans receive most of their information from their sight. Most of the systems use visual output as the main way to give information back to the human user. The classic way is to use two dimensional graphical user interfaces with labels, buttons, and other standard controls. In this setup the user can issue precise commands to the system, but the visual feedback is limited. However, in some situations, such as working with multidimensional data or when the operator needs to manipulate in space, the control via the usual 2D graphical user interface becomes a complex task. With the rapid evolution of computer hardware, software visualization started to explore and apply new 3D representations for various tasks successfully. For an overview of the current state of 3D visualization, see [7].

The cog Framework

The purpose of the work is to create a framework, which mimics the real environment and maps the objects into the virtual space. The second critical mission is to recognize human commands, and to create a framework to control the robotized environment (the actuators) according to these.

The main structural entities of the presented system are the following:

Visualization part is capable of rendering the virtual image of the objects and of utilizing different 3D stereographic visualization systems.

Physics backend is designed to amend the virtual objects with physical properties in the virtual space, and to simulate their behaviour.

Suit module is responsible for handling the data originating from a motion recording suit directly, or indirectly; this module has the facilities of recognizing human commands and propagating them among the interested parties.

Robot controller is in charge of providing transparent control over different mobile robot agents. The general scheme of the module can be seen in Fig. 1.

Layer 1 Using the lowest level layer allows to use robot specific commands. It is also possible to access these controllers via the *IRobotControl* interface.

Layer 2 Using the second layer enables to use the robots transparently. Apart from that, this layer provides outer circle controlling.

Layer 3 Using the third layer enables remote control of the robot controller via special network packets.

Layer 4 Provides different stubs using the lower layers.

Scene module is responsible for managing objects, and manages the connection of the other modules. It connects the visualization, the physical emulated virtual world and the peripheries. It describes what is present in the current scene and what the parameters of the virtual scene are. On Fig. 2, the scene is visible - where a human controls the robot using hand postures.

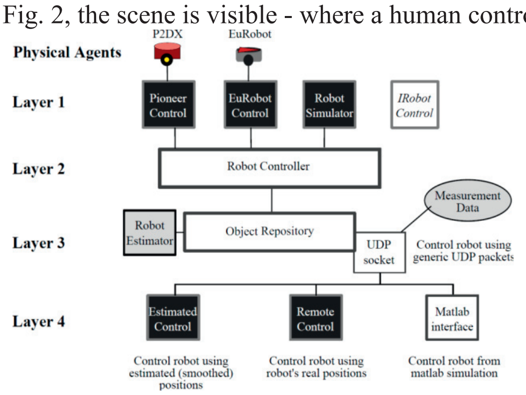


Fig. 1. Structural overview of Robot control components

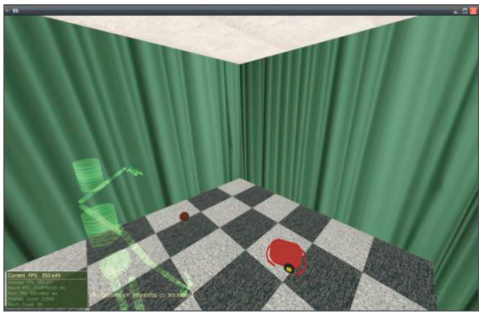


Fig. 2. Screenshot from the virtual scene: the human user commands the robot to go near to the ball (pointing on the floor)

Because of page length constraints not all the components could be introduced in the necessary detail level.

Experiments

The main aim of this type of experiment is to demonstrate the capabilities of the cog framework. Fig. 3. introduces the structural entities which are used in the experimental setup. There is a human operator, who wears a motion detection suit. In the room there is a mobile robot. The human user commands the robot with pointing finger.

During the operation the coordinate system of robot and the observation camera is synchronized regularly. If the camera can identify the robot the controller updates its inner coordinate system. The error of the positioning which is present during the synchronization is coming from the following factors:

Camera measurement error is varying in the room as the camera covers the area of the room differently, but the mean measured error is around 3.3 centimeters with a maximal error of 12 cm. The fault of the camera is mostly consistent in a meaning that the mapping of coordinates is almost continuous, which makes it possible to navigate to recognized objects.

Camera delay is varying and hard to measure or predict. Experiments showed it to be around 300 msec.

Odometry error (e.g. wheel slippage) is not taken into account. The nature of this error is insidious, because it is additive.

Robot controller experiment

In a separate experiment, the control of the robot, including the camera system has been tested. The robot was placed in the central part of room, and was ordered to traverse through the following coordinates (0.5, 0.5), (4.2, 0.5), (0.5, 1.5), (4.2, 1.5), (0.5, 2.5), (4, 2.5). Results are shown in Fig.4.

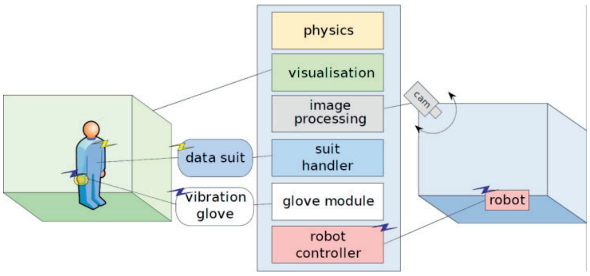


Fig. 3. Structural overview of experimental setup

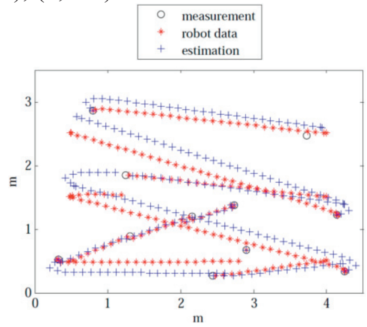


Fig. 4. Test results of robot control: real and estimated robot positions following a zigzag command

Summary

The paper presented the Cog Framework, which can be a useful tool in extension of the traditional human-system interaction loop. A framework has been described to integrate the human operator and mobile robots in one virtual space.

Acknowledgment

This work has been by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point and the Control Research Group of Hungarian Academy of Science.

References

[1] A. Mehrabian, “Attitudes inferred from neutral verbal communications,” Journal of Consulting Psychology, vol. 31, no. 4, pp. 414–417, Aug. 1967.

[2] P. Ekman and H. Oster, “Facial expressions of emotion,” Annual Review of Psychology, vol. 30, pp. 527–554, Jan. 1979.

[3] Y. Wu and T. S. Huang, “Vision-based gesture recognition: A review,” vol. 1739, pp. 103–115, Jan. 1999.

[4] Klara Wenzel, Akos Antal, Jozsef Molnar, Bertalan Toth, Peter Tamas, NEW OPTICAL EQUIPMENT IN 3D SURFACE MEASURING, Journal of Automation, Mobile Robotics & Intelligent Systems, Vol. 3, No. 4, pp. 29-32, 2009.

[5] Tamás P.; et al.: 3D Measuring of the Human Body by Robots, 5th International Conference Innovation and Modelling of Clothing Engineering Processes - IMCEP 2007, pp 109-115; ISBN 978-961-248-047-9, Moravske Toplice, Slovenia, October 2007, University of Maribor, Maribor (2007).

[6] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. S. Huang, “Facial expression recognition from video sequences: Temporal and static modelling,” Computer Vision and Image Understanding, vol. 91, no. 1-2, pp. 160–187, Jul. 2003.

[7] A. Teyseyre and M. Campo, “An overview of 3d software visualization,” IEEE Trans. Vis. Comput. Graphics, vol. 15, no. 1, pp. 87–105, Jan. 2009.

Paper 10: Shop-Floor Architecture for Effective Human-Machine and Inter-Machine Interaction

Bjorn Solvang, Gabor Sziebig, Peter Korondi, Shop_Floor Architecture for Effective Human-Machine and Inter-Machine Interaction. *ACTA POLYTECHNICA HUNGARICA* 9:(1) pp. 183-201. (2012) IF: 0.385

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang. The working principles and conceptual design was a cooperative effort from Bjørn Solvang, Gabor Sziebig and Peter Korondi.

The state-of-the-art survey and literature review on STEP-NC standard and other production related technologies is entirely a contribution from Gabor Sziebig.

The functional design of the RT-Middleware based SCARA robot is entirely a contribution from Gabor Sziebig. The physical implementation of electronic circuits of the SCARA robot involves many master students from NUC (Karoly Szell, Bence Kovacs, Geza Szayer, Balazs Varga, Ferenc Tajti). Testing of the SCARA robot is a cooperative effort from Gabor Sziebig and the above mentioned master students. The tests were carried out at the production facilities at Narvik University College.

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing partly Section 1, entirely 2 and 3, partially section 4 and produced the graphics and pictures through the paper (if not referred to external sources, permission for usage of external sources is acquired and could be proofed on request). Gabor Sziebig's written contribution was thoroughly reviewed and revised by Peter Korondi.

Shop-Floor Architecture for Effective Human-Machine and Inter-Machine Interaction

Bjørn Solvang[1], Gábor Sziebig[2], Péter Korondi[3]

[1,2]Narvik University College, PO Box 385, 8505 Narvik, Norway

[3]Budapest University of Technology and Economics, PO Box 91, 1521 Budapest, Hungary

[1]Bjorn.Solvang@hin.no, [2]Gabor.Sziebig@hin.no, [3]korondi@mogi.bme.hu

Abstract: Manufacturing equipment for small and medium scale production is often arranged in a setup of a Flexible Manufacturing Cell (FMC). Normally, such cell members are from various manufactures and they have all their specific capabilities when it comes to man-machine and inter-machine interaction. Existing man-machine interfacing is quite simple and is typically done via a screen/keyboard interface while inter-machine communication is carried out through designated I/O lines. Although such communication channels provide us a certain possibility to program and control the cell members, it is not enough to fully utilize the potential of the coordinated control of the complete unit. Especially in smaller production facilities, where often the FMC components are a mixture of older (poorly documented) and newer machines, there exists no ready-to-use solution for communication between the members of the cell. This article discusses and presents a new architecture for man-machine and inter-machine communication and control. This new architecture allows for the integration of both new and old equipment and is especially targeted for efficient man-machine interaction.

Keywords: Shop- floor architecture; flexible manufacturing cells/systems; human-machine interaction

1 Introduction

The development of novel technologies often stems from experience acquired during the production of a previous series by the same manufacturer. This allows for compatibility with older, in-house, versions of software/hardware solutions. Such a strategy is understandable from a vendor's point of view, but it is not enough from the end-user perspective. The typical end-user has many different machines/software and basically wants everything to be compatible, independent of who produced what.

During the last few decades much research has been dedicated towards general manufacturing equipment such as turning- and milling- machines (NC-machines), industrial robots (IRBs) and automated guided vehicles (AGVs). In the case of NC machines, researchers have focused on establishing a common language that can be shared by all machines [1]. On the other hand, language has not been the main focus among researchers in the field of robotics; their main concern has been to allow users to create robot programs more intuitively and rapidly (e.g. [2, 3]). AGV research has in many senses been related to path-planning and the automatic guidance in a semi-unstructured environment. At the same time we have available new, reasonably-priced ICT technologies; TV/screen solutions, mobile and wireless equipment, 3D technologies, scanners, virtual reality and haptic interfaces. The upcoming challenge is how these classical and new technologies cooperate and be combined in an industrial environment.

For a typical small- and medium-sized enterprise (SME), flexibility is the main concern. SME flexibility means most of all: a rapid setup and initial programming phase for a new production series, programming interoperability between different machines, the tracking of changes at any production step, and the possible incorporation of both older and newer equipment into the manufacturing cell. While there are many promising developments related to various equipments, there are still many challenges to be addressed in the case of SMEs:

a) *Outdated equipment*: the typical SME doesn't have a new and updated machine park, and their equipment ranges from very old to more up-to-date solutions.

b) *No inter-machine communication*: machines are operated in standalone mode, without any structured communication between them.

c) *Manual programming*: machine codes are mainly produced and tested on the shop-floor, without any feedback to production engineers and other decision makers.

d) *Manual labour*: monotonous and repetitive work-tasks are executed by humans instead of industrial robots, often because of the lack of fast and effective programming methods for smaller production volumes.

Flexibility calls for an open shop-floor architecture which allows for efficient man-machine and inter-machine interaction, and, as pointed out in the introduction, an architecture that allows for the incorporation of both typical manufacturing machines with new 3D/mobile/virtual technologies.

Thus, the key point of this paper is the lay-out of a new shop-floor control architecture which allows for the use of a combination of both older and newer technologies. The following criteria are set against the new architecture:

i) Flexibility through on-line system reconfigurability and the implementation of the plug'n'produce paradigm.

ii) Machine invariance, so that old and new equipment, including a wide variety of different sensors, may be used.

iii) The architecture should enable effective man-machine communication.

iv) The architecture should allow for the use of a heterogeneous programming language.

v) The architecture should be capable of tracking changes during manufacturing processes and should provide feedback to previous stages.

The proposed architecture incorporates the future trends in machine development, especially through the incorporation of the new programming standards and the introduction of a software-based middleware technology for inter-machine communication. The following technologies and techniques serve as the basis in our work:

1) low-cost, accurate and efficient electronics (e.g. microcontrollers) [5]

2) high-speed communication links for inter-machine communication [6]

3) machine programming and development software based on open source solutions (e.g. [7])

4) standards for efficient machine communications and control [8]

The remainder of this article is organized as follows; in Section 2, the incorporated standards are shown along with their main characteristics and benefits. This section also describes the importance of using standards, instead of case-by-case specific solutions as building blocks. In Section 3, the proposed shop-floor architecture is introduced in detail. The architecture has a general layout based on existing standards and can be utilized on a large majority of the key elements in flexible manufacturing systems. The practical implementation of the new architecture is discussed in section 4. Finally, the article is concluded in Section 5.

2 Related Work

In order to establish a highly flexible shop-floor architecture, we believe that a modularized software, component-based solution gives the best result. In this section the key elements of such a controller solution will be introduced.

The software of the controller should hide the details of hardware layers and provide a platform for standardized communication. Further, it should allow for the reconfiguration of manufacturing cells without hardware changes (within certain limits) and should serve as a platform for human-machine interaction. Every member of the shop-floor will typically have a software-component that will manage its life-cycle and provide services on different levels. The layered structure of such an ideal software component is shown in Fig. 1.

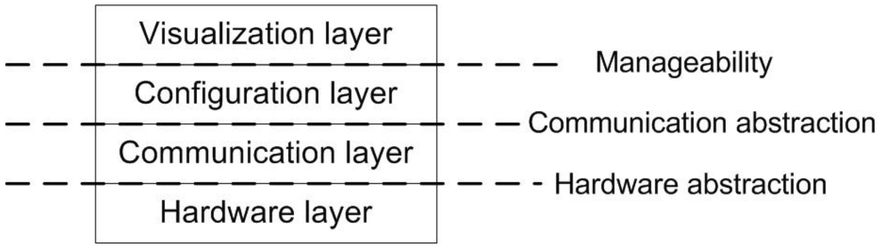


Figure 1
Manufacturing cell member's software layer-structure. Each layer hides the details of its' content. Connection can be established with any of the layers. Command complexity grows from bottom to top.

A member can be constructed from more than one component and, as it is software based, can be assembled on-line for every different task. Note that the definition of tasks within such flexible shop-floor control architectures relies mostly on detailed descriptions of manufacturing processes.

As an example, a 6-axis industrial robot and a 3-axis milling machine can be simultaneously controlled to obtain the functionality of a 9-axis machine. The robot can be used to support and re-orientate the work piece during some machining actions (typically light operations) carried out by the NC-machine.

Until now this would be solved using either I/O signalling and pre-programmed position and orientation information or through manual labour.

In our case, based on the work piece and manufacturing cell information, the shop-floor controller would be set up intuitively and automatically (with or without human-machine interaction) for the given task. To achieve this, two things are needed: 1) a uniform description language for manufacturing processes and 2) a uniform shop-floor/machine/component control language. These two will be introduced in the next subsection.

2.1 STEP-NC Standard

Today, medium and small-scale production is centred around automated and flexible manufacturing cells. These cells usually consist of computer numeric controlled (CNC- or just NC-) machines, industrial robots, material storages, conveyor belts, etc. Each component in the cell has its own control architecture and communication link to other machines or humans. Even in recent NC machines, the communication is limited to Input/Output (I/O) level changes, which are used for signalling from one machine to another or signalling to the shop floor controller. This allows for only severely limited communication and limits flexibility. For example, in many cases the automatic opening and closing of a door clutch still relies on manual interventions to the machine's control system.

As early as in the 1960s, when flexible manufacturing cells appeared, considerable emphasis was laid on their autonomous operation. However, this was only achieved through the careful planning of the product itself, as well as production and machine-specific tasks by a highly qualified human, using the concept of computer integrated manufacturing (CIM) [9].

In the 1970s issues such as sustainability, fast response time to market changes and cost-effectiveness made the utilization of CIM unsustainable [10]. As a result, the concept of CIM reappeared from time to time in new forms, such as holonic manufacturing systems [11], agent-based manufacturing [12], virtual CIM [13], etc. The bottlenecks of most manufacturing cells are the lack of interoperability, program reusability and fast re-configurability. However, the new CIM alternatives focus mainly on macro- and micro-process planning without addressing these important bottlenecks.

The standard for the programming of CNC machines was originally developed in the 1960s (ISO-6983 [14]) and is still used every day with all of its original weaknesses [15]. As pointed out by [16]:

1) It is a low-level language that describes mainly the cutter location of the spindle. Modification of geometrical properties on such a low level is almost impossible. File sizes grow very fast with the level of complexity.

2) It contains several machine-specific language elements (e.g. waiting condition, loops, etc.) which result in a lack of interoperability between machines.

3) The resulting file (to be used uniquely by the machine) is a result of a one-way transformation. It is not possible to feed back any kind of change from the shop-floor to the design stage.

To overcome these problems, new standards were introduced [17] to replace the old one and relieve the user from its constraints. One of the most promising standards is STEP (Standard for the Exchange of Product Model Data - ISO-10303 [18]) and its combination with the STEP-NC - ISO-14649 [8] – (the NC extension of the original STEP).

The benefits of using the STEP-NC standard can be summarized as follows:

a) Shorter time-to-market interval due to self-contained work piece and process planning information in one place

b) Changes can be tracked at any stage even during manufacturing

c) Machines are more efficient and adaptive and hybrid control theories can be applied to manufacturing processes

d) Machine vendor independence

These two standards are the largest and most widespread in the ISO group and are relatively new: STEP was created in 1994 and STEP-NC was created in 2003. Although these standards have been available to manufacturers for over 8 years, machine manufacturers are seemingly not willing to switch from the older standard. Our guess, as mentioned in the introduction, is that manufactures build on previous versions of their products in order to keep the compatibility. On the other hand, there are several government-funded international projects [19]-[22] that promote standards among large manufacturers (e.g. Boeing, Daimler Chrysler, Volvo). At the same time, many researchers from all over the world [15], [17], [23]-[28] are promoting and developing solutions for STEP and STEP-NC based machining.

It is important to note, however, that dealing with only one aspect of the whole manufacturing cell will not solve every problem. Program portability from one vendor's machine to another's can be solved using newer standards, but the inter-machine communication will still be limited to hard-wired I/O signalling.

2.2 RT-Middleware Framework

Industrial robots have had a slightly different history of development than the typical NC-machine. More specifically, there are two major differences between the two: 1) industrial robots do not share any common language and from the outset, industrial robots were well suited for I/O communication with other machining cell components (this has led to their widespread use as system integrators and shop-floor controllers), and 2) industrial robots use vendor-specific and, in many cases, even model-specific languages.

Typically, NC-machines and industrial robots have proven to be mechanically very long-lasting, while the electrical control systems have become more rapidly outdated. Lately, there has also been a growing quest for advanced robot-sensor integration, at a completely different level than before.

Due to such issues, many researchers have turned to new and open control architectures for robot systems (so-called middleware systems) [7]. The overall goal is to provide a common language and control structure that would allow for user/task-specific sensor integration. The most important middleware solutions can be found in [29]-[40] and a comparison can be found in [41]. [41] also reports that these systems in general provide the user with high-level object-oriented robot programming and control environments. Issues related to specifics, like the number of supported robots, may vary. It is also reported that only one of these middleware technologies - called RT (Robot Technology)-middleware - is under standardization [42].

RT-middleware has proven to be well-adapted to industrial projects and is in use by many different industrial companies (Toshiba, Honda, AIST and also other research institutes) [38].

In 2002, the Japanese Ministry of Economy, Trade and Industry (METI), the Japan Robot Association (JARA) and National Institute of Advanced Industrial Science and Technology (AIST) started a project named "Consolidation of Software Infrastructure for Robot Development". The goal of the project was to implement humanoid robot systems in a modular fashion so that they could meet the diverse needs of users. Further, the project aims to allow system designers or integrators to build versatile robots or systems with relative ease by simply combining selected modular parts [43]. RT-middleware was originally developed for humanoid robots. However, humanoids, industrial robots, and NC-machines are built up from the same building blocks, at a component level (motor/axis control logic form the basis for all of these machines).

The RT-middleware framework combines low-level components (named RTCs, RT-Components) into larger operating units (e.g. like an industrial robot). If the robot needs more sensors or an extra drive axis, software drivers (RTCs) can be written, allowing the sensor/axis to be introduced in the existing robot control architecture. The RT-middleware environment offers various tools for including and even merging together the different RTC units. RTCs are realized as platform independent, CORBA objects. The use of CORBA in shop-floor control architectures has been found to be favourable in other research areas as well [44].

RTCs are the basic, modular units for achieving distributed computing. These elements can be observed as black-boxes. Each black-box has predefined interfaces for communication and the data manipulation and calculation is hidden from the external parties. This results in a highly modular framework, which was already shown in Fig. 1.

More details about the RT-middleware can be found in references [17, 38].

3 Shop-Floor Control Architecture Description

Until now the control of flexible manufacturing systems was limited to I/O level signalling, using an industrial robot or a PLC as an overall controller. The proposed architecture is based on software components (RT-Middleware) and on standards (STEP family). The shop-floor controller is a normal PC connected to the high-speed communication network of the shop-floor. This PC serves as an interface for system assembly and task specifications.

Fig. 2. shows the proposed shop-floor control architecture. At the bottom level (*Cell components*) we have all the cell members represented by NC-machines,

industrial robots, cameras, sensors, conveyor belts, feeders, etc.. In general, all cell members that have the ability to communicate are defined as active cell components. Cell members, if possible or needed, are built up from smaller components, forming controllable objects where the communication drivers between the software components are based on the RT-middleware framework. These components represent building blocks for the manufacturing system and are called RT-Components or RTCs. Depending on the specified task, these RTCs can be used as fully standalone cell members or can be assembled in a way that will satisfy the needs posed by a given task. For better understanding, the following example is given (see Figs. 3 and 4): a typical SCARA robot is built up from 4 drive units. 3 drive units are used for positioning and the 4th is used for orientation. The 4 drive units can be represented by 4 standalone RTC components or can be attributed to 1 RTC that has all the necessary functions (kinematics, dynamics, interpolation) included and is capable of the execution of high-level commands (e.g. go to x, y, z position), while the first option uses 4 independent, low-level axis drives that can only understand commands such as turn to the given angle.

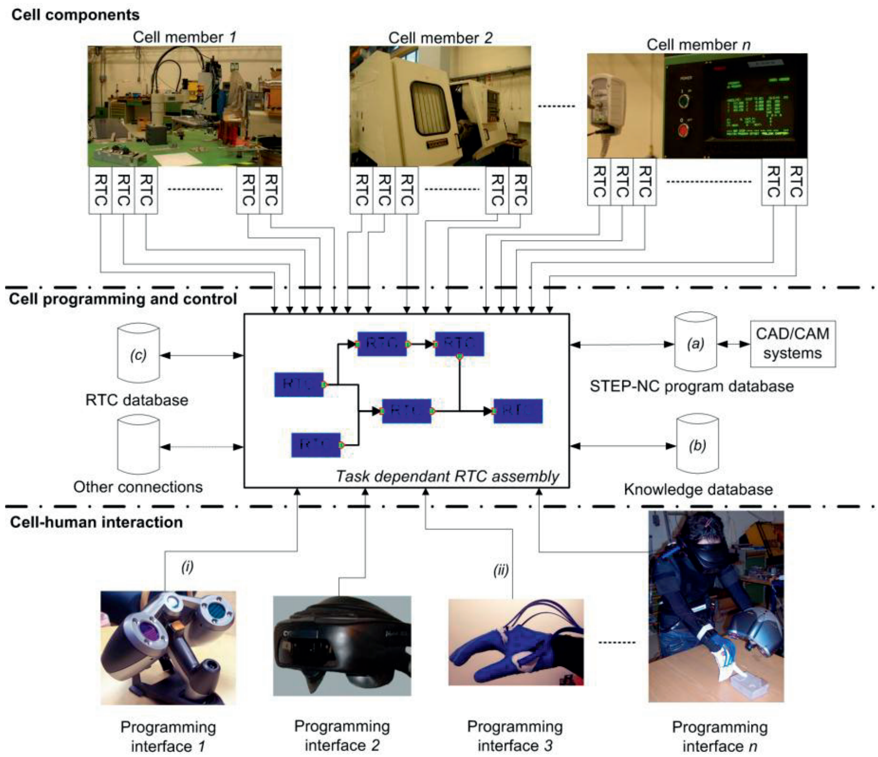


Figure 2

Shop-floor control architecture. The architecture is assembled for every given task, based on the databases contents (STEP-NC work-piece and process description (a), previous knowledge of task executions (b), available cell members and tools (c), etc.) and on the used programming interface (3D scanner (i), motion capture (ii), etc.). New cell members can be created on-the-fly (within certain limits).

If sensors are attached to the SCARA robot's end-effector (e.g. force sensor, which is also an RTC), the measurement values of the sensors can be directly fed to the axis drive RTCs, resulting in a closed-loop controller as low as the drive level for a specific drive or can be fed to the whole SCARA robot, affecting positioning or other parameters in the control.

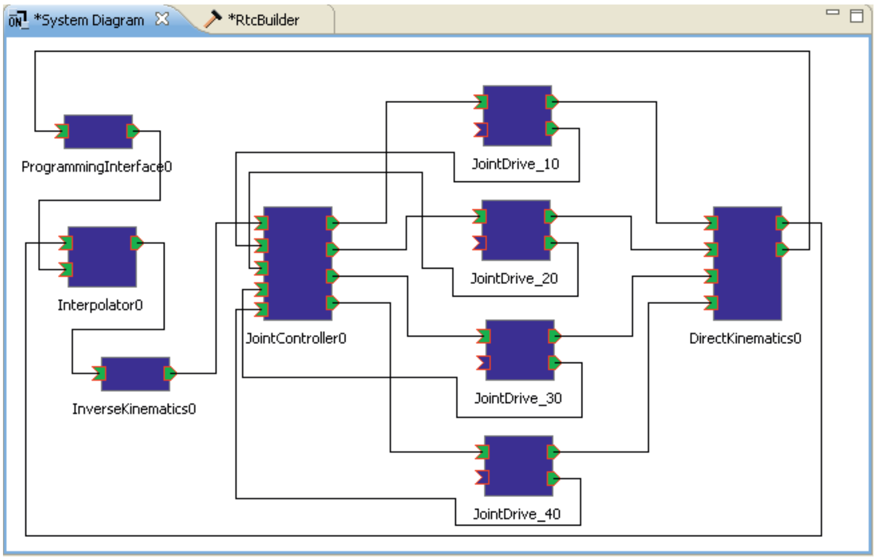


Figure 3
Low level control software component diagram. Every blue box is a standalone RTC. A SCARA robot is built up from 4 independent drive units (JointDrive). Each drive unit gets the reference torque from the JointController, which calculates the desired rotation based on the data received from the InverseKinematics component and the positioning feedback. The movement is split up by the Interpolator, using the feedback data from the DirectKinematics and the instruction from the ProgrammingInterface. The robot can be instructed by the ProgrammingInterface component, which can also represent a joystick or a mouse.

When the necessary RTCs are developed and implemented they are all registered in to the software component database (*RTC database*). From this database, the shop-floor controller can select and combine RTCs according to the specific task-dependent needs. This selection can be automatic or can be based on human-machine interaction using one of the *Programming interfaces*. The decision is also supported from the *Knowledge database*, where the previous task executions are stored with the corresponding RTC assemblies and also from the *STEP-NC program database*, where the actual manufacturing processes and work piece information are defined. A further strength of the software based RTCs is that they can be situated anywhere, as long as they are reachable by the controller (anywhere on the network connected to the controller) and it is possible to receive control messages from other sub-manufacturers (suppliers) or distributors in the supply chain.

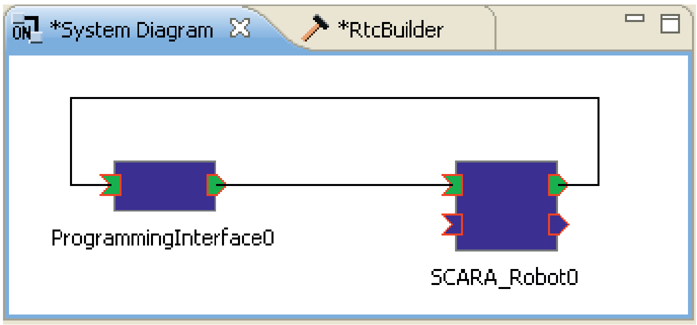


Figure 4

High level control software component diagram. Every blue box is a standalone RTC. With this configuration the SCARA robot is instructed by high level commands (e.g go to given position with linear interpolation).

Combining RTCs into new and higher level RTCs provides us with the opportunity to increase the level and complexity of the commands. This is very important in order to be able to incorporate the STEP-NC standard. The main idea behind the STEP-NC standard is to give the machine the task and let it execute and interpret it independently, based on the local logics and control, which is a much better approach than the previous one, where the predefined movement of each axis was given to the machine. All actions in the manufacturing cell, and all modifications to the geometry of work pieces, are done according to the STEP-NC standard, thus allowing the user to track changes through the manufacturing chain. For these reasons, a STEP-NC interpreter as well as logger RTCs are created, which can be used to convert the high-level commands to joint-level ones and to report changes made to the manufacturing process.

In existing manufacturing equipment, man-machine interaction is often carried out via a screen/keyboard/mouse interface. These communication links will also be important in future manufacturing systems, although more advanced and human friendly interaction systems will most likely be created as well.

The simplest *Programming interface* is the PC's screen, where a special graphical user interface is used to assemble the shop-floor controller (a part of this interface was already shown in Figs. 3 and 4), named RTSystemEditor [38]. More advanced interaction modes can be achieved using a 3D scanner during the inspection procedure or by instructing an industrial robot through human gestures [45]. Distant programming of shop-floor controllers is possible by introducing Virtual Reality techniques [41].

The above described shop-floor control architecture is under implementation at our laboratory. No shop-floors are identical but normally similar components are present (NC machines, industrial robots, conveyor belts, coordinate measurement machines, material feeders, digital cameras, PCs, etc.) and in the next section we would like to share our experiences when it comes to practical implementation towards some typical components.

4 Practical Implementation at a Typical Shop-Floor

In general every member of a flexible manufacturing cell has its own controller (NC machines, industrial robots) or supervisory system (PLC or PC). The proposed shop-floor controller replaces these with software components which can be configured and assembled for every given task. In order to achieve the goals stated in the introduction the cell members can and actually should be constructed from low and high-level software components (e.g. axis drives, sensors). This can be done in two ways: 1) by implementing a wrapper for the given cell member, as one standalone unit (in the beginning this is the most convenient) [4] or 2) by re-engineering the controllers from low-cost highly modularized electrical components [5].

The first solution's biggest drawback is that it is limited to the actual capabilities of the already existing controller. In the simplest case, a PC is connected to the given cell member (e.g. serial link, Ethernet or LPT) and the software component (RTC) is implemented on the PC. The RTC is simply converting the instructions from RT-Middleware to member-specific instructions (e.g. STEP-NC instructions to G code in case of an NC machine [46]).

In the second case, re-engineering the controller, everything has to be developed and implemented from scratch, but as a result we obtain a high level of flexibility and a truly software-based controller.

In our experimental system, we chose this second option to exemplify the proposed shop-floor controller. The architecture was validated through several experiments.

The experimental shop-floor controller is now limited to one cell member, an Adept 604-S SCARA robot, whose control system is replaced by an RT-Middleware based controller. The mechanical drawing of this SCARA robot is shown in Fig. 5.

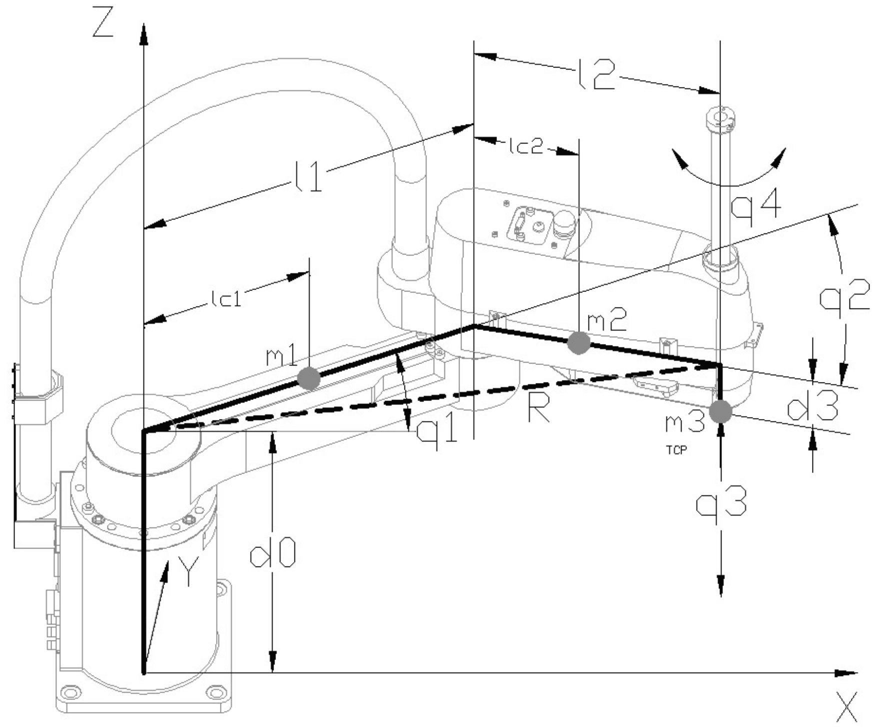


Figure 5
Mechanical drawing of the Adept 604-S SCARA robot. m1, m2, m3 are the masses, l1,l2,d0,d3 are the length, q1,q2,q3,q4 are the angles of the corresponding joints. lc1 and lc2 are the masses position on joint 1 and joint 2, respectively. These data were used in calculations of robot dynamics [47]

4.1 Software and Hardware Layout

To match the mechanical setup of the robot, the following standalone hardware and software components were created (RTCs):

a) *Programming interface*: this is a PC-based graphical user interface, in which the user can enter coordinates for the robot to go to. It also displays the current position of the robot. The input of the component is the actual position of the robot in Cartesian coordinate system. The output is the desired position of the robot.

b) *Interpolator*: based on the actual and desired positions, it plans the trajectory of the robot. The inputs of the component are the desired and the actual position of robot in Cartesian coordinate system. The output is the steps position through the trajectory in Cartesian coordinate system.

c) *Inverse Kinematics*: the component calculates the motor angles for the desired movement based on the actual position. The input of the component is a position on Cartesian coordinate system. The output is the angles of the robot axis for the given position.

d) *Direct Kinematics*: based on the motor angles it calculates the actual position of the robot. The inputs for the component are the angles from the axis drives. The output is the robot's actual position in Cartesian coordinate system.

e) *Joint Controller*: the component uses the desired motor angles to give torque references for the axis drives. The inputs for the component are the angles for rotation of the motors. The outputs are the torque references for every axis drive.

f) *Axis drive for every joint*: the 4 motor drives are used independently. Each joint has a current control loop inside. The input for the component is the torque reference signal. The output is the angle of the motor, which is calculated from encoder position.

The block diagram based on the above mentioned components is shown in Fig. 6.

This modularized structure of the controller gives the possibility of including sensor data in the control loop on different control levels. The sensor data can be introduced as low as at the *Axis drive* level or at *Joint Controller* level or even higher. The synchronization of the data and the real-time execution of the control loop are guaranteed by the RT-Middleware framework.

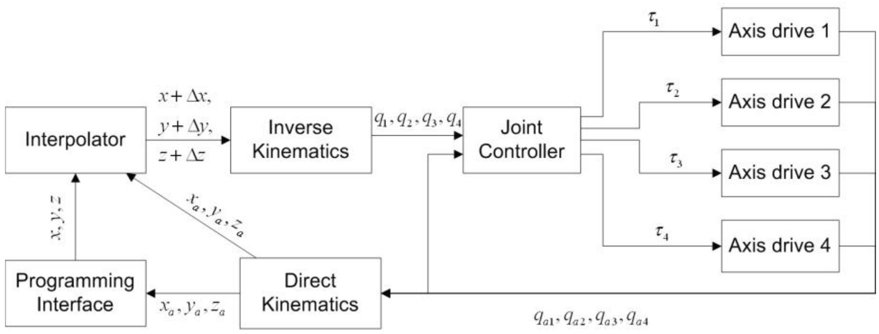


Figure 6
Block diagram of the SCARA robot's controller. x, y, z are the desired point's Cartesian coordinates. x_a, y_a, z_a are the robot actual position's Cartesian coordinates. $x+\Delta x, y+\Delta y, z+\Delta z$ are the robot's next position's Cartesian coordinates. q_1, q_2, q_3, q_4 are rotation angles. $\tau_1, \tau_2, \tau_3, \tau_4$ are torque reference signals. $q_{a1}, q_{a2}, q_{a3}, q_{a4}$ are angles derived from the encoders.

4.2 Experimental Results

In order to validate the control architecture, experiments were carried out on the major joints (Joint 1 and 2) of the SCARA. The software component based controller's main benefit is that each joint is driven independently, but this is also its biggest drawback when designing the controller for the given system.

A robust, decentralized PID control was implemented in the *Joint Controller*. Two different experiments were executed and were repeated two times to ensure the correct measurement. The feedback system's sampling time was 1 *kHz*.

The first experiment tested the step response of the first joint. The results can be seen in Fig. 7. The feedback shows a negligible overshot. Stabilisation time is approximately 200-300 ms, which is to be considered very good for an industrial robot.

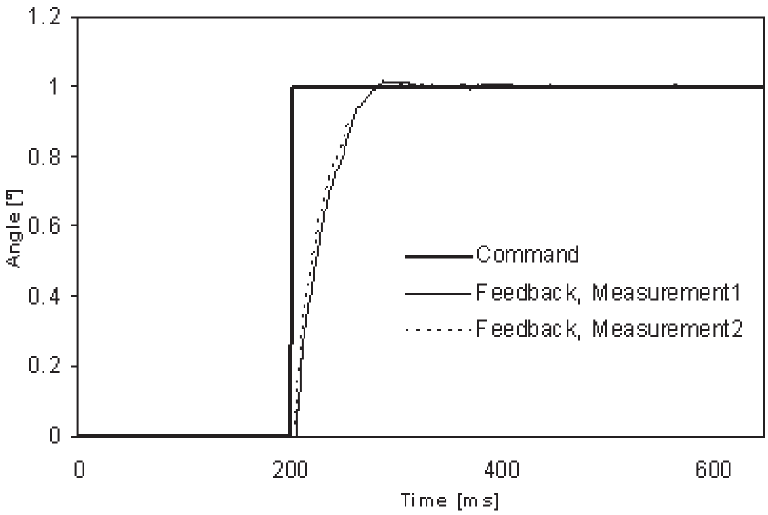


Figure 7
Step response of the first joint. Result of two measurements [48]

The second experiment tested the step response of the second joint in two different arm positions. The first measurement was carried out with straightened arms and the second measurement with the second arm perpendicular to the first. The results can be seen in Fig. 8. The feedback shows a more significant overshoot compared to the first experiment and a longer stabilisation time of 400-500 ms. Such results were expected since this is a SCARA (selective compliance articulated robot arm), especially developed for assembly operations. The compliant structure of joint 2 plays an important role in assembly operations, and the horizontal flexibility of the robot arm is favourable during mounting operations, allowing for initial misalignment between the mating parts.

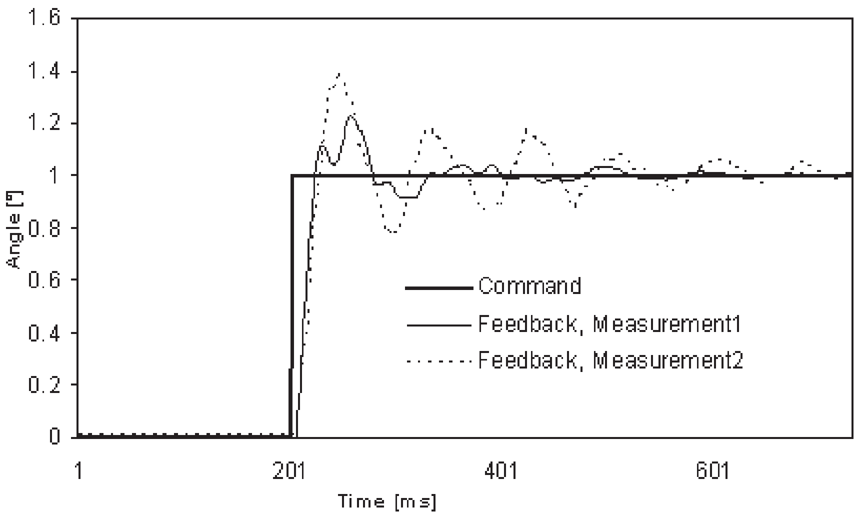


Figure 8

Step response of the second joint, with decentralized PID control. The Measurement 1 is with straightened arms, while the Measurement 2 is with the second joint perpendicular to the first one [48]

Conclusions

This paper presented a shop-floor control architecture for inter-machine communication and control, based on the RT-middleware framework and the STEP-NC standard. The great variety, diversity and complexity of equipment used in manufacturing cells call for an open control architecture capable of integrating all of its members. The proposed methodology is general in its layout and emphasizes openness to the largest extent. The implementation of this new architecture was exampled on a typical shop-floor member (a SCARA robot), which is driven by low and high level commands.

Acknowledgement

This work has been supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

The authors would like to thank Károly Szell, Bence Kovács, Géza Szayer, Balázs Varga, Ferenc Tajti for their contribution to the development process.

References

[1] S.T. Newman, A. Nassehi, X.W. Xu, R.S.U. Rosso Jr., L. Wang, Y. Yusof, L. Ali, R. Liu, L.Y. Zheng, S. Kumar, P. Vichare and V. Dhokia, Strategic advantages of interoperability for global manufacturing using CNC technology, Robotics and Computer-Integrated Manufacturing, Vol. 24, No. 6, pp. 699-708, Dec. 2008.

[2] J.N. Pires, G. Veiga and R. Araujo, Programming-by-demonstration in the coworker scenario for SMEs, Industrial Robot - An International Journal, Vol. 36, No. 1, pp. 73-83, 2009.

[3] S. Ekvall and D. Kragic, Robot Learning from Demonstration: A Task-level Planning Approach, International Journal of Advanced Robotics, Vol. 5, No. 3, pp. 223-234, Sep. 2008.

[5] J.U. Cho, Q.N. Le and J.W. Jeon, An FPGA-Based Multiple-Axis Motion Control Chip, IEEE Trans. Ind. Electron., Vol. 56, No. 3, pp. 856-870, Mar. 2009.

[6] T. Li and Y. Fujimoto, Control system with high-speed and real-time communication links, IEEE Trans. Ind. Electron., Vol. 55, No. 4, pp. 1548-1557, Apr. 2008.

[7] J.G. Garcia, J.G. Ortega, A.S. Garcia and S.S. Martinez, Robotic Software Architecture for Multisensor Fusion System, IEEE Trans. Ind. Electron., Vol. 56, No. 3, pp. 766-777, Mar. 2009.

[8] International Organization for Standardization (ISO), ISO 14649-1, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 1: Overview and fundamental principles, 2003.

[9] J. Harrington, Computer Integrated Manufacturing, New York: Industrial Press Inc, 1973.

[10] J. Johansen, U.S. Karmarkar, D. Nanda and A. Seidmann, Business experience with computer integrated manufacturing. A survey of current strategy and practice, In Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, Vol. 4, pp. 970-979, 1995.

[11] J. Christensen, Holonic Manufacturing systems-initial architecture and standards directions, In Proceedings of the first European conference on holonic manufacturing systems, 1994.

[12] D. Ouelhadj, C. Hanachi and B. Bouzouia, Multi-agent system for dynamic scheduling and control in manufacturing cells, In Proc. IEEE International Conference on Robotics and Automation, pp. 2128–2133, 1998.

[13] D. Wang, S. V. Nagalingam and G. C. I. Lin, Development of an agent-based Virtual CIM architecture for small to medium manufacturers, Robotics and Computer-Integrated Manufacturing, Vol. 23, No. 1, pp. 1-16, 2007.

[14] International Organization for Standardization (ISO), ISO 6983-1, Numerical control of machine-program format and definition of address words Part-1: Data format for positioning, line and contouring control systems, First Edition, 1982.

[15] X. W. Xu and Q. He, Striving for a total integration of CAD, CAPP, CAM and CNC, Robotics and Computer-Integrated Manufacturing, Vol. 20, No. 2, pp. 101-109, 2004.

[16] X. Zhu, Y. Wang, H. Fu, A 3-D Simulation System for Milling Machining Based STEP-NC, in Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, Dalian, China, 2006.

[17] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.–K. Yoon. RT-Middleware: Distributed component middleware for RT (robot technology), in Proc. Computational Intelligence in Robotics and Automation, CIRA 2005, pp. 457-462, 2005.

[18] International Organization for Standardization (ISO), ISO 10303-1, Industrial automation systems and integration - Product data representation and exchange Part 1: Overview and fundamental principles, 1994.

[19] ESPRIT Project 8643, Optimised preparation of manufacturing information with multi-level CAM-CNC coupling (OPTIMAL): final report, ESPRIT, 1997.

[20] ESPRIT Project 29708, STEP-compliant data interface for numeric controls, ESPRIT, 1999.

[21] IMS Project 97006, STEP-compliant data Interface for Numerical Controls, STEP-NC Consortium, 2003.

[22] National Institue of Standards and Technology (NIST) 99-01-4035, Model Driven Intelligent Control of Manufacturing, STEP Tools Inc., 1996.

[23] S.T. Newman and A. Nassehi, Universal Manufacturing Platform for CNC Machining, Annals of the CIRP, Vol. 56, No. 1, pp. 459-462, 2007.

[24] A. Nassehi, S.T. Newman and R.D. Allen, STEP-NC compliant process planning as an enabler for adaptive global manufacturing, Robotics and Computer-Integrated Manufacturing, Vol. 22, No. 5-6, pp. 456-467, 2006.

[25] A Nassehi, S T Newman, X W Xu, R D. Allen and R S U Rosso Jr., Adaptability and Interoperability in CNC Manufacturing, In Proceedings of the 3rd CIRP international conference on Digital Enterprise Technology, pp 1-8, 2006.

[26] Xun Xu, STEP into Distributed Manufacturing with STEP-NC, In book Process Planning and Scheduling for Distributed Manufacturing, pp. 393-421, Springer London, 2007.

[27] S.H. Suh, B.E. Lee, D.H. Chung and S.U. Cheon, Architecture and implementation of a shop-floor programming system for STEP-compliant CNC, Computer-Aided Design, Vol. 35, No. 12, pp. 1069-1083, 2003.

[28] S. M. Amaitik and S. E. Kilic, An intelligent process planning system for prismatic parts using STEP features, International Journal of Advanced Manufacturing Technology, Vol. 37, No. 16, pp. 978-993, 2007.

[29] S. Magnenat, V. Longchamp, F. Mondada,"ASEBA, an event-based middleware for distributed robot control" in Workshops of International Conference on Intelligent Robots and Systems (IROS), 2007.

[30] H.D. Nayar, I.A. Nesnas, "Measures and Procedures: Lessons Learned from the CLARAty Development at NASA/JPL," International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, October 2007.

[31] J. Jackson, "Microsoft robotics studio: A technical introduction" in IEEE Robotics & Automation Magazine, Volume: 14, pp. 82-87 ISSN: 1070-9932, Dec. 2007

[32] A. Weitzendfeld, S. Gutierrez-Nolasco and N. Venkatasubramanian, "MIRO: An Embedded Distributed Architecture for Biologically inspired Mobile Robots" in 11th IEEE International Conference on Advanced Robotics (ICAR'03), 2003.

[33] Ozaki Fumio, Oaki Junji, Hashimoto Hideaki, Sato Hirokazu, "Open Robot Controller Architecture (ORCA)" in Journal: Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu, Vol. 2, 2003.

[34] M. Mizukawa, H. Matsuka, T. Koyama, T. Inukai, A. Noda, H. Tezuka, Y. Noguchi, N. Otera, "ORiN: open robot interface for the network - the standard and unified network interface for industrial robot applications" in Proceedings of the 41st SICE Annual Conference (SICE 2002), Tokyo, Japan, Vol. 2, pp. 925- 928, ISBN: 0-7803-7631-55-7, Aug. 2002.

[35] Nuno Lopes, Pedro Lima, "OpenSDK - An Open-source Implementation of OPEN-R" , Proc. Of 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008), Estoril, Portugal, 2008.

[36] H. Bruyninckx, P. Soetens, B. Koninckx, "The real-time motion control core of the Orocos project" in Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03), Heverlee, Belgium,

Vol. 2., pp. 2766- 2771, ISSN: 1050-4729, ISBN: 0-7803-7736-2, 14-19 Sept. 2003.

[37] Matthias Kranz, Radu Bogdan Rusu, Alexis Maldonado, Michael Beetz, and Albrecht Schmidt, "A Player/Stage System for Context-Aware Intelligent Environments" in Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys 2006), at the 8th Annual Conference on Ubiquitous Computing (Ubicomp 2006), Orange County, California, September 2006.

[38] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, Yoon Woo-Keun, "RT-middleware: distributed component middleware for RT (robot technology)" in Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 3933- 3938, ISBN: 0-7803-8912-3, 2-6 Aug. 2005.

[39] G. Veiga, J. N. Pires,K. Nilsson "On the use of SOA platforms for industrial robotic cells" in Proceedings of Intelligent Manufacturing Systems (IMS2007), Spain, 2007.

[40] J.-C. Baillie, "URBI: towards a universal robotic low-level programming language" in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 820- 825, ISBN: 0-7803-8912-3, 2-6 Aug. 2005.

[41] B. Solvang, G. Sziebig, and P. Korondi, "Multilevel control of flexible manufacturing systems," in Proc. IEEE Conference on Human System Interactions (HSI'08), pp. 785–790, 2008.

[42] Object Management Group (OMG), Robotics DTF, "Robotic Technology Component Specification," 1.0 beta 2, 2007. 08. 18.

[43] G. Sziebig, A. Gaudia, P. Korondi, N. Ando, B. Solvang, "Robot Vision for RT middleware Framework," in Proc. IEEE Instrumentation and Measurement Technology Conference, pp. 1-6, 2007.

[44] J.Shin, S.Park, C.Ju, H.Cho, "CORBA-based integration framework for distributed shop floor control," Computers & Industrial Engineering, vol. 45, Elsevier, pp. 457–474, 2003.

[45] G. Soros, B. Resko, B. Solvang and P. Baranyi, "A cognitive robot supervision system," In Proceedings of IEEE 7th International Symposium on Applied Machine Intelligence and Informatics, pp.51-55, Jan. 2009.

[46] M. Hardwick and D. Loffredo, Lessons learned implementing STEP-NC AP-238, International Journal of Computer Integrated Manufacturing, Vol. 19, No. 6, pp. 523-532, Sep. 2006

[47] B. Kovacs, Development of a universal robot controller: Development of FPGA hardware and software, Modelling of a Scara robot, B.Sc Thesis, p. 35, Budapest University of Technology and Economics and Narvik University College, 2009

[48] G. Szayer, Development of a universal robot controller: Hardware development, designing and implementing central motion control, B.Sc Thesis, p. 38, Budapest University of Technology and Economics and Narvik University College, 2009

Paper 11: On Industrial Robots and Cognitive Infocommunication

Bjørn Solvang, Gabor Sziebig, On Industrial Robots and Cognitive Infocommunication. In: *3rd IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2012*. Kosice, Slovakia, 02/12/2012-05/12/2012. pp. 1-6

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Bjørn Solvang. The working principles and conceptual design was a cooperative effort from Bjørn Solvang and Gabor Sziebig.

This paper is a plenary lecture, invited paper on the conference on Cognitive Infocommunications. The paper is based on the previous work of the authors (e.g. Paper 3, 4 and 7) and is more like a summary of all the activities that were carried out in the work period of the PhD studies.

Bjørn Solvang is the main author and writer of the paper. Gabor Sziebig has contributed with writing almost all the sections in close collaboration with Bjørn Solvang and produced the graphics and pictures through the paper. Gabor Sziebig's written contribution was thoroughly reviewed and revised by Bjørn Solvang.

On Industrial Robots and Cognitive Info-Communication

Bjørn Solvang, Gabor Sziebig
Department of Industrial Engineering, Narvik University College, Narvik, Norway

Abstract— **In order to utilize the full potential of industrial robots its human operators need to gain a deep technical understanding of rather complex mechatronic devices. However, such knowledge does not come easily, its building blocks is university level mathematics, physics, mechanical and electrical engineering, control theory and computer science.**

In this paper we will shift focus, by asking the question; could we expect that the robot system should understand us humans, rather requiring the opposite?

A selection of our initiatives and recent developments in programming and control of industrial robots will be presented to promote a development towards a more human friendly future at the shop-floor.

***Index Terms*—Robot programming, human friendly communication, cognitive info-communication**

I. Introduction

Generally, programming of industrial robots are done in two different ways, either online or offline. Online (or teach) programming is done standing in front of the robot, carefully guiding the robot through its operations using a pendant (typically joystick and some pushbuttons). The operator "teaches" the robot the work to be undertaken. In offline programming the operator works disconnected from the robot itself, using designated software where he/she manipulates a virtual representation of the robot and work-piece(s). The robot path is stored in a file and transferred to the physical robot. Both offline and online programming solutions comes in a great variety, e.g. different pendant solutions and various software packages and a widely implemented in industrial applications. However there are challenges with these programming principles:

- First of all they can be quite complex and operators should be very well educated in general robotics [1].
- Second they are (very) vendor specific and operators have to be trained towards each different system.
- Third, many applications are not best solved using either offline or online programming, rather a combination of those.
- Forth, existing system do not particularly support open access to the system and thereby user driven developments.

Corresponding author: Bjorn Solvang, Department of Industrial Engineering, Narvik University College, Lodve Langes gate 2., 8514. Narvik, Norway; phone: +47-7696-6227; fax: +47-7696-6814; e-mail: bjs@hin.no

All above leads to the conclusion; Interaction and programming of industrial robots are undertaken on the premises of the machine. Basically, human operators should learn and understand all details of the robot system.

In this paper we will shift focus, by asking the question; could we expect that robots should understand human natural communications? Asking the robot systems to understand us humans, more than the opposite. A set of our initiatives and recent developments in human friendly programming principles will be presented to promote a development towards a more human friendly future. Solutions building on existing manipulators (as typical add-ons) are discussed, but also solutions which require deeper system integration or even a new robot control architecture are introduced. Addressing typical tasks in specific and as well general solutions.

The organization of the paper is as follows: Section II is devoted to input, output, and sensor solutions, especially targeted towards existing manipulator systems. Section III presents a deeper architectural approach, where we discuss a general interaction layout which can be utilized for any advanced application. Both in sections II and III we will focus on real applications which may serve as inspiration towards other developers, fronting new challenges and applications. Section IV concludes the paper.

II. Human friendly input and output systems

A. Input devices

By using some kind of motion capture system we can record all movements of a human operator. This can again be used as an input device in the programming of industrial robots.

The result of the motion capture will be a 3D curve, represented as points, in the work-piece's coordinate system. However a few pre-processing steps are necessary to gain access to this information. First of all the relationship between the work-piece coordinate K_w system and the motion capture system has to be established. This can be done in a systematic way. Preferably by adapting the coordinate set-up procedures found in coordinate measurement machines (CMMs).The creation of a set of substitute geometries are essential in the

CMM methodology. For example the operator uses his index finger to identify and define three planes, P1, P2, and P3 as shown in Fig. 1. (a) and (b). The normal unit vector of the largest surface is typically selected as the Z axis in the coordinate system. The intersection of two planes (e.g. P1 ∩ P2) may form the second unit vector X. Origo in the coordinate system may be defined as the crossing point of all three planes (e.g. P1 ∩ P2 ∩ P3). The final unit Y vector is found perpendicular to Z and X, defined by the cross product Z x X. [2].

Because of the inaccuracies in motion capture systems including the elastic properties of human's fingers the recorded finger point measurement data typically need further processing. Some kind of filtering is needed to minimize the error of the transformation. A minimum number of measurement points are undertaken to form each substitute geometry as shown in Fig 1.c. The filtering takes place in this step, where the squared sum of the distance l_i of each measurement point is minimised by using the Gaussian Least Square Methodology, as described by [2].

$$\sum_{i=1}^{m} l_i^2 \rightarrow \min \qquad (1)$$

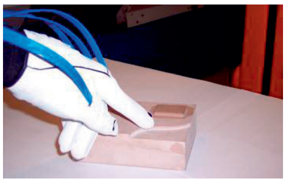


Fig. 1a. Identification of the work-piece coordinate system

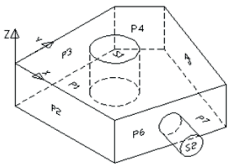


Fig. 1b. Substitute geometries

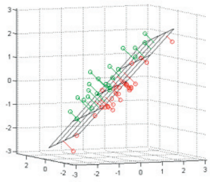


Fig. 1c. Defining a plane

When K_w is readily established the operator can turn to defining the path where the machining action should take part. All the finger movements are calculated and stored with reference to the work- piece coordinate system. The relationship between the robot base coordinate system K_B and K_w is easily found by applying similar methodologies, using the robot itself as a pointing device. Typically, robot manufacturers provide such procedures, as part of their user documentation. Finally, the path described in robot coordinates allows us to start-up the robot and carry out the designated task.

B. Output devices

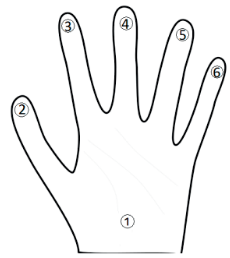
Implementing imitation of senses and feedback from virtual reality always meant a great problem. Realization of two (sight and audition) out of the five human senses is indispensable. Simulation of these two senses is not a complex issue; on the contrary the other three are nearly impossible [3]. Instead of implementing the, often very expensive and complex, force feed-back effect, our focus is on tactile sensing (suitable for Cognitive Info- Communication, according to [4]). To achieve this, a glove containing of six vibration motors on different locations on the hand has been developed. These locations include all five fingers, and the palm. The motors are controlled by a microcontroller unit using PWM (Pulse-Width Modulation) technology. Communication with the microcontroller unit uses the wireless Bluetooth protocol enabling free movement for the user. Since the Bluetooth protocol is industry standard, the mobile unit can be controlled by any Bluetooth-capable device (for example: PC, cellular phone, PDA, etc.) using the developed software libraries. A tactile feedback system was built up from the following components:

- Gloves
- Controller board
- Wireless communication
- Remote device
- Software running on remote devices

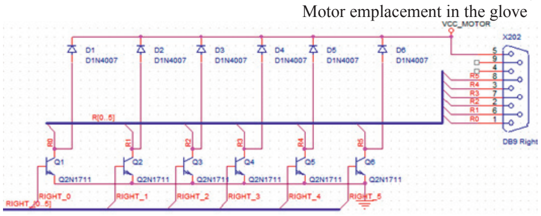
An overview of these components can be seen in Fig. 3.

The glove was designed in such a way, that it is compatible with the motion capture system mentioned in the previous chapter (i.e. Measurand ShapeHand™ [5]). This makes the system able to receive data about the user's hands and to transmit feedback at the same time. To achieve this, the feedback glove is a modified plastic glove, which must be worn on the top of the ShapeHand™ motion capture device (as seen in Fig. 2. (a)). Fig. 2. (b) shows the motor emplacements on the glove, while Fig. 2. (c) shows the wiring. Our initial

a) Measurand ShapeHand™ device



b) Motor emplacement in the glove



c) Wiring of the glove

Fig. 2. Glove setup

experimental results could be found in [6].

The vibration glove is an inexpensive and simple solution compared to most classical force- feedback systems. On the other hand, by using vibrations to describe physical parameters , e.g. weight or distance, calls for some operator training. Luckily the human shows a remarkable capability of re-mapping "unexpected" sensory input towards the "expected" physical parameter.

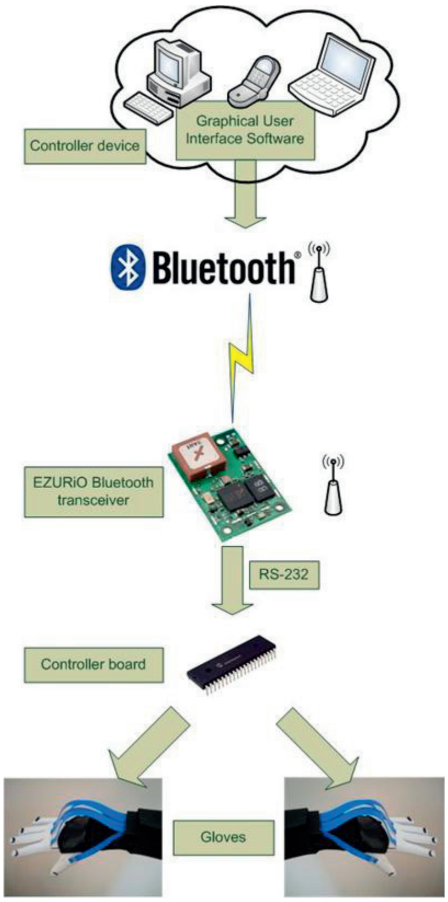


Fig. 3. Tactile feedback system overview

C. *Sensors*

In this paragraph an example of an external sensor is demonstrated. The novelty of this vision based sensor is that it allows for direct communication with any kind of machine that has an output screen. This visual information from the screen is processed (digitalised) by the camera system and could be converted to any kind of information. It could even be connected to the tactile system demonstrated in the previous chapter.

One example of the usage of such vision system, in an industrial application, is a supervisory system of old NC-machines. These machines are usually manufactured in the early 1980s and lacks capability of Input/Output communication. Man-machine communication is normally through low quality monitors. Even though some machines might be able to communicate with Personal Computers (PCs), this is typical one way communication and can only be used for program uploading/downloading. In order to monitor the state of a manufacturing process an operator should always observe the screen of NC machine and decisions made by the operator are mainly based on the information shown on the screen.

We suggest the operator to be partially replaced by a vision system. The vision system should automatically detect changes in the NC-screen. Digitalisation of the screen gives us access to the actual state of the NC machine. By transferring this data to an industrial robot, typically used for feeding the NC machine, we can instruct the robot to take the operators position, monitoring and even interact with the push-button panel of the NC-machine. Full remote control of old NC-machines can be achieved with a simple camera solution.

In this case the vision system is constructed of a digital camera and a PC, which uses image processing tools for Optical Character Recognition (OCR). From the image acquired by the camera, numerous data can be extracted (position, distance to go, current line of the running program, operator messages, etc.). These data describes the current state of the NC machine.

The most convenient way to have a robust OCR is to use artificial neural networks. The novelty of the system is not the technology behind (OCR is used for over 20 years now), but the application of it in such an industrial environment. OCR is mainly used in industry for tracking objects (serial number recognition, container localization [7]), but in this case utilized as a supervisory system.

The Multi-Layer Perceptron Neural Network is trained with back propagation learning algorithm [8] and for the activation function bipolar sigmoid function is used (2):

$$
\begin{aligned}
f(x) &= \frac{2}{1+e^{-\alpha * x}} - 1 \\
f(x)' &= \frac{2 * \alpha * e^{-\alpha * x}}{(1+e^{-\alpha * x})^2} = \alpha * \frac{1 - f(x)^2}{2}
\end{aligned}
\qquad (2)
$$

where α is the parameter that decides the gradient of the function and x is the sum of the outputs of the previous layer.

Two inputs are used for training: an image captured by a digital camera (this image must undergo a few pre-processing steps, which will be introduced in the following) and a desired output. The desired output is composed from binary values (0 or 1).

In order to get a well formatted image the following steps are executed:

1. Screen colour extraction (usually the old NC machines use one colour), the goal is to achieve higher contrast
2. Threshold filtering

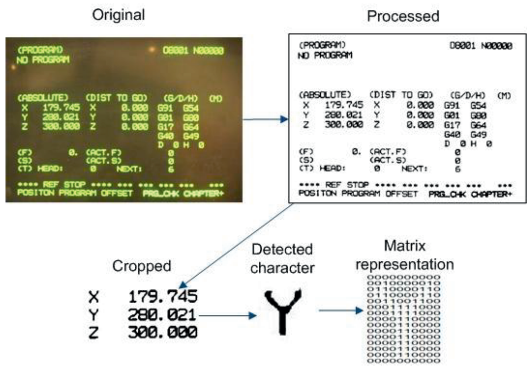


Fig. 4. Processing of input image [9]

After this point the image will only contain black or white pixels. The black pixels are logically 1 and white pixels are logically 0. Fig. 4. Summarizes the process itself.

When only a specific region (Region of Interest) is needed, the image is cropped after this stage to specific dimension.

3. Characters are detected

4. Characters are converted to matrix representation (binary values of pixels)

These matrixes form the first input of the training.

The second input is the desired output: the 16bit Unicode representation of the detectable character (e.g. X = 0000000001011000).

We use machine specific training sets in order to achieve higher success rate.

Further details on this topic can be found in [9].

III. Towards a deeper system Integration

Modern shop-floor architecture for human- machine and inter-machine communication and control should aim to be:

1) As flexible as possible, enabling the user community to develop according to their ever changing needs
2) Able to include old and new machinery
3) Capable of incorporating equipment for effective man-machine interaction
4) Preferably, speak only one language
5) Provide interoperability, as in easy exchange of programs between different users
6) Capable of tracking changes to the work-piece at any stage of the manufacturing process

Unfortunately, existing architectures cannot meet these requirements. No single standard is available for the user community for the development of such systems. The on-going development of the STEP-NC [10] standard and RT-middleware [11] are very promising, but they cannot alone cover the necessary level of system integration, providing effective shop floor integration and control.

However, by combining the capabilities of the STEP-NC standard and RT-middleware and urge for the further development of these into specific application areas (not covered by the standard today) a possible solution arise.

Fig. 5. represent a conceptual control architecture where RT-middleware and STEP-NC is combined, seeking to reach the above mentioned requirements.

At the bottom level, we have all of the cell members, represented by all the NC-machines, industrial robots, cameras, all other sensors, conveyor belts, and feeders etc. In general, all cell members that have the ability for communication is defined as an active cell component.

Cell members are “broken down” into controllable objects where a communication driver from the RT-middleware framework is “wrapped around” forming a so-called RTC component. How deep RTC “breakdown”, of a cell member, depends on several factors. Especially in older machines hardware intervention is challenging due to often poor and limited documentation. Re-engineering of a controller is time consuming and opens for possible severe circuit damages. Original replacement components could turn out to be impossible to find. However, the deeper hardware intervention goes (by design of low level RTCs) higher flexibility will be achieved. Also, electronics are available, for less cost than ever, and developing new controllers based on microcontrollers represent a powerful way of system improvement. Building a completely new controller structure is challenging, but such a solution provides us with a completely open structure with almost no restrictions compared to the massive machine-controllers from the last few decades.

For better understanding, the following example is given (see Figs. 6): a typical SCARA robot is built up from 4 drive units. 3 drive units are used for positioning and the 4th is used for orientation. The 4 drive units can be represented by 4 standalone RTC components or can be attributed to 1 RTC that has all the necessary functions (kinematics, dynamics, interpolation) included and is capable of the execution of high-level commands (e.g. goto: x, y, z position), while the first option uses 4 independent, low-level axis drives that can only understand commands such as turn to the given angle.

If sensors are attached to the SCARA robot’s end-effector (e.g. force sensor, which is also an RTC), the measurement values of the sensors can be directly fed to the axis drive RTCs, resulting in a closed-loop controller as low as the drive level for a specific drive or can be fed to the whole SCARA robot, affecting positioning or other parameters in the control.

Finally, it is up to the system designer to decide the necessary level of system breakdown and building up of its belonging RTCs. The RT-middleware framework itself do not request any specific depth, a driver can be developed for any component level as long as it can receive and send the expected data types.

When the necessary RTCs are developed these are all put in a component database. From this database the system designer can pick and combine together RTCs according to his/her specific task dependant needs. From the knowledge database he/she can even select and modify previously developed RTC assemblies. A further strength of the CORBA based RTCs are that they can be situated anywhere, as long as it is reachable for the controller. For this we can even think of receiving

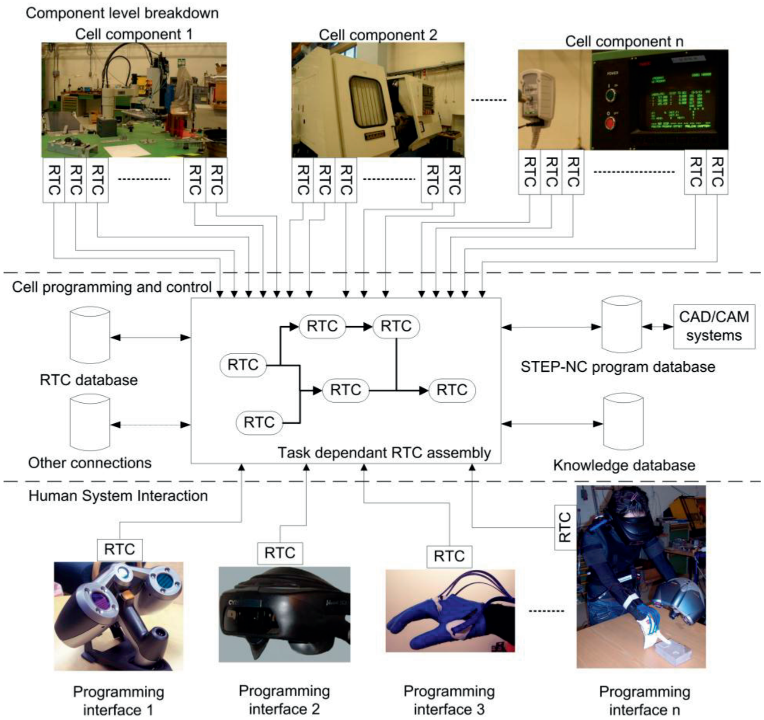


Figure 5
Shop-floor control architecture. The architecture is assembled for every given task, based on the databases contents (STEP-NC work-piece and process description. New cell members can be created on-the-fly (within certain limits).

control signals from other sub-manufactures (suppliers) or distributers in the supply chain.

As a tool for RTC set-up and assembly a software named OpenRTM-aist is available. OpenRTM-aist is also capable of combining existing RTCs into a new single RTC. The software act as a graphical user interface for linking and setting up the system [11].

Combining RTCs into new and higher level RTCs provides us with the opportunity to increase the level and complexity of the language (what is sent through its inport /outport). This is very important to be able to incorporate the STEP-NC standard. All actions in the manufacturing cell, modifying the geometry of the work-piece, should be done according to the STEP-NC standard. Originally, the idea was that any NC-machine should be able to understand and speak (answer) the STEP-NC language. Unfortunately, not too many NC-manufactures support STEP-NC, yet. Instead of waiting for vendor acceptance of the standard we suggest that the RTCs are build up according to the communication standard of the STEP-NC standard. This implies that the RTCs that are capable of interacting with geometry, must be able to read and speak back of its modifications. All such communications is done towards the STEP-NC database.

There are several other connections to the STEP- NC database as well, of very high importance is the CAD/CAM software (computer aided design-/computer aided manufacturing). Luckily, software developers seem to be implementing the STEP standard more rapidly than the machine-manufactures.

Tracking changes to the work-piece (back to the work-piece model database) is one of the major strengths of the STEP-NC concept. STEP-NC provides information from the designer down to the shop floor and it reports back to the designer any changes occurred on the shop floor.Introducing the STEP-NC standard to the RTC components that directly interfere with work-piece geometry is truly a leap towards a unified cell programming language.

In existing manufacturing equipment, man-machine interaction is often carried out via a screen/keyboard/mouse interface. These communication links will also be important in future manufacturing systems, although more advanced and human friendly interaction systems will most likely be created as well.

The simplest Programming interface is the PC's screen, where a special graphical user interface is used to assemble the shop-floor controller. More advanced interaction modes can be achieved using a 3D scanner during the inspection procedure or by instructing an industrial robot through human gestures [12]. Distant programming of shop-floor controllers is possible by introducing Virtual Reality techniques [13]. More detailed description of different components could be found in [14].

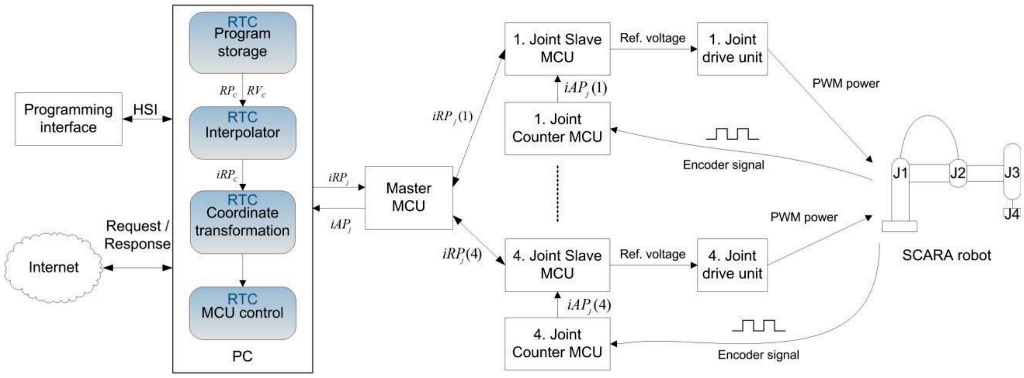


Fig. 6. RT-middleware based SCARA robot

Fig. 5. also shows an example of a "future" operator wearing a see-trough head-mounted display capable of mixing computer generated images with real sight. Further, a motion capture suit allows for the operator to be present in a 3D virtual environment interacting with other computer generated objects. A handheld 3D scanner can be used to capture unknown geometry and visualize it in the same computer generated environment. These equipment only represent as examples of what can be expected to arise in the near future. Today, existing manufacturing components such as robots and NC- machines cannot directly interface with advanced interaction equipment. However, through RT-middleware we can see a possible solution. By defining the interaction equipment as standard RTCs (as seen in the bottom of Fig. 5.) and by developing those necessary drivers, advanced human-system-interaction can be utilised.

IV. Conclusion

In this paper we have focused on showing how to shift towards a more human friendly interaction at the shop floor. We have been focusing on practical implementations of the main categories of interaction components; input systems, output systems and sensors. The introduced components in the chapters above open possibilities for Cognitive Info-communication. Finally, we have laid out an overall shop-floor architecture, building on and combining two existing standards, which we believe could serve as general solution for the future. All components, systems and the proposed architecture are under implementation in our laboratory.

Acknowledgement

The authors would like to thank Károly Szell, Bence Kovács, Géza Szayer, Balázs Varga, Ferenc Tajti, Csaba Kiss and Péter Korondi for their contribution to the development processes.

The research was partly supported by SMaE project, a grant from EUs interregional program INTERREG IVa NORTH, Nordland and Troms county.

References

[1] Sz. Kovacs, D. Vincze, M. Gacsi, A. Miklosi, P. Korondi, "Interpolation based Fuzzy Automaton for Human-Robot Interaction." In Proc. IFAC Syroco 2009, pp. 451-456. Paper CR2-3, 2009.

[2] K.Martinsen, (1992), "Kompendium i måleteknikk og vektorielle toleranser" Institutt for vekstedteknikk, Norges Tekniske Høgskole, Norway.

[3] P. Galambos, A. Roka, P. Korondi, "Ergonomic Vibrotactile Feedback Design for HMI Applications." In Proc. of 11th International Carpathian Control Conference, pp. 31-34, 2010.

[4] P. Baranyi, G. Németh, P. Korondi, *3D Internet for Cognitive Infocommunication*. INFORMATIKA - A GÁBOR DÉNES FŐISKOLA KÖZLEMÉNYEI 12:(2) pp. 3-6., 2010.

[5] ShapeHand^{TM} motion capture glove, Measurand Inc. [Online], http://www.motion-capture-system.com/

[6] G. Sziebig, B. Solvang, Cs. Kiss, P. Korondi, "Vibro-tactile feedback for VR systems.", In Proc. 2nd Conference on Human System Interactions, pp. 406-410., 2009.

[7] Elovic, P. (2003). Implementation of Gate and Crane OCR Systems for Container Terminal Automation and Security, Proceedings of Terminal Operations Conference ASIA (TOC ASIA 2003), pp. 1-7, Feb. 2003.

[8] Rojas, R. & Feldman, J. (1996). Neural Networks: A Systematic Introduction, Springer, ISBN 3-5406-0505-3, Germany

[9] G. Sziebig, B. Solvang,P. Korondi, Image Processing for Next-Generation Robots. In: Computer Vision., I-Tech Education and Publishing, pp. 429-440., 2008.

[10] S.T. Newman, A. Nassehi, X.W. Xu, R.S.U. Rosso Jr., L. Wang, Y. Yusof, L. Ali, R. Liu, L.Y. Zheng, S. Kumar, P. Vichare and V. Dhokia, Strategic advantages of interoperability for global manufacturing using CNC technology, *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 6, pp. 699-708, Dec. 2008.

[11] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, Yoon Woo-Keun, "RT-middleware: distributed component middleware for RT (robot technology)" in Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), pp. 3933-3938, ISBN: 0-7803-8912-3, 2-6 Aug. 2005.

[12] G. Soros, B. Resko, B. Solvang and P. Baranyi, "A cognitive robot supervision system," In Proceedings of IEEE 7th International Symposium on Applied Machine Intelligence and Informatics, pp.51-55, Jan. 2009, ISBN: 0-7803-8912-3, 2-6 Aug. 2005.

[13] B. Solvang, G. Sziebig, and P. Korondi, "Multilevel control of flexible manufacturing systems," in Proc. IEEE Conference on Human System Interactions (HSI'08), pp. 785–790, 2008.09, ISBN: 0-7803-8912-3, 2-6 Aug. 2005.

[14] B. Solvang, G. Sziebig, P. Korondi, *Shop_Floor Architecture for Effective Human-Machine and Inter-Machine Interaction*., ACTA POLYTECHNICA HUNGARICA , 9:(1) pp. 183-201., 2012.

Paper 12: Navigating in 3D Immersive Environments: A VirCa Usability Study

Gabor Sziebig, Øritsland Trond Are, Navigating in 3D Immersive Environments: A VirCa Usability Study. In: Syroco 2012. Dubrovnik, Croatia, 05/09/2012-07/09/2012. Dubrovnik: pp. 380-384. Paper 141.

Declaration of co-authorship

The idea and concept of the paper is entirely contribution from Gabor Sziebig.

The paper investigates usability questions in a 3D simulation environment. The usability testing, questionnaire and execution of the test is entirely contribution from Gabor Sziebig.

Gabor Sziebig is the main author and writer of the paper. Gabor Sziebig has contributed with writing all the sections and produced the graphics and pictures through the paper. The entire paper was thoroughly reviewed and revised by Øritsland Trond Are (Lecturer in PD8401 Interaction Design course).

Navigating in 3D Immersive Environments: a VirCa usability study

Gabor Sziebig *, Trond Are Øritsland **

**Narvik University College, Department of Industrial Engineering, Norway*
(e-mail: gabor.sziebig@hin.no).
***Norwegian University of Science and Technology, Department of Production Design, Norway*
(e-mail: trond.are.oritsland@ntnu.no)

Abstract: Navigation in 3D environment encounters more challenges than in 2D case. While there exist well established framework, tested solutions for 2D environment these solution usually fail in 3D. In the recent years immersive 3D environments are becoming more common in research and also in education. This tool helps to visualize information simpler and more understandable way as any other solution. For human beings the feeling of depth is a unique feature and this is the most appropriate way to take advantage out of it.

The goal of this study is to investigate the problem of navigation in 3D immersive environment and propose solution to it with carrying out usability study in a software solution called VirCa (Virtual Collaboration Arena). Total 6 Master level students have participated in the test. They have carried out simple navigation and manipulation tasks in the immersive environment and had to fill in different questionnaires. Methodology and results are also presented in the paper.

Keywords: User interfaces, Cognitive systems, Computer simulation, Stereo vision, Navigation

1. INTRODUCTION

Human vision is an extremely powerful information processing system. Humans receive most of their information from their sight. This results that, most of the systems use visual output as the main way to give information back to the human operator.

The common way is to use two dimensional (2D) graphical user interfaces with labels, buttons, and other standard controls. With these tools, the user can issue precise commands to the system, but the visual feedback is limited. This standard approach works efficiently for numerous scenarios. However, in some situations, such as working with multidimensional data or when the operator needs to manipulate in three dimensional (3D) space, the control via the usual 2D graphical user interfaces become a complex task. With the rapid evolution of computer hardware, software visualization started to explore and apply new 3D representations for various tasks successfully. For an overview of the current state of 3D visualization, see (Bowman, 2005 and Vincze, 2012).

One of the purposes of the human vision is to deliver spatial information to assist the control of our actions. In nature the extent of depth perception is variable among different species. Humans have their eyes positioned on the front of their heads, thereby reducing field of view in favour of precise depth perception. The objects which are visible with both eyes have different images, which enable the brain to calculate distances. The difference of the same object in the same environment viewed along two different lines of sight is called parallax (see Fig. 1). The human nervous system is responsible for mapping the parallax into spatial information.

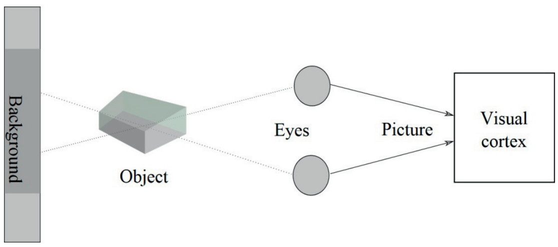


Fig. 1. Parallax effect

The aim of stereoscopic systems is to create a real 3D sense of space in a limited environment. This task can be decomposed into two parts. First, generate the two pictures separately with the proper settings. This is supported by specialized hardware, which uses four image buffers instead of the usual two and generates the images into the dedicated image buffers continuously. Second, the images have to reach the eyes separately using some form of physical separation: the images must only be seen by the intended eye.

Already from the above mentioned properties, the change from 2D navigation to 3D navigation encounters lot of difficulties, usually due to the fact that humans are unique and does not react the same way to visual feedbacks. The goal of this study was to investigate the problem of navigation in 3D immersive environment and propose solution to it with carrying out usability study in a software solution called VirCa (Virtual Collaboration Arena) (Galambos, 2011).

VirCa is a 3D simulation tool supporting immersive environment. On overview picture can be seen on Fig. 2. The basic concept of VirCa is that it is composed from distributed components. Such components can be registered into a global naming service (central controller), which in turn allows the user to connect the components the way it is necessary for his or her purpose (task driven controller assembly).

Fig. 2. A sample screen from VirCa 3D simulator including an industrial robot and showing a menu for interaction

2. RELATED RESEARCH

As a general rule it can be stated, that what proved to be well working in 2D will fail 90% in 3D situation (Bowman, 2008 and Dang, 2008). The basic manipulation tasks (selection, translation, scaling, and rotation) all facing problems in 3D and not even mentioning navigation. This is due to the fact that most of input devices are designed to have two degree of freedom (such as a conventional PC mouse). Users who have experience in different type of first-person shooter (FPS) games, knows that in order to have higher degree of freedom, input devices has to be mixed, e.g. combination of keyboard and mouse at the same time.

In immersive environments such simple case as showing a "menu" is a complex task. As the menu structure and position is usually well defined for 2D interaction, this is not fast-forward in 3D case.

Challenges for menus in 3D

- Overlay: to always show a menu ribbon in the field of view for the user, is reducing the immersive feeling of the simulator.
- Positioning: usually there is no frame for simulation window to attach menu to it.
- Orientation: when you "invoke" the menu command, where should the menu look? Usually menus are 2D, paper like objects. The menu should face the viewpoint when invoking it.
- Selecting menu items: how do you click on the item? You have to highlight selection, move forward and backward in the menu tree.

Just only looking as simple case as menus in 3D the differences are highly visible compared to 2D.

The most common task in immersive environments is the navigation. This usually refers to modification of the position of the viewpoint with the help of some kind of input device (mouse, tracker, keyboard, etc.). With the wide spreading of low-cost input devices (such as Microsoft Kinect, Nintendo Wii, Smartphones) the testing and development of new 3D interaction techniques have got more attention in research in the last years (Gallo, 2009 and Korondi, 2010).

3. USABILITY TESTING

Participants

We have totally 6 participants from Narvik University College. From these six participants we had 3 male and 3 female with ages ranging from 21 to 25 resulting an average of 23 years old. All participants have a good understanding of English language; they characterized their level to be moderate (4 people) or advanced (2 people). All the participants are master level students, studying Industrial Engineering. All of the participants have met some kind of computer simulator previously. 3 of them were active players in FPS type games. However none of them played console games, only on PCs.

Environment

The testing was carried out in the Visualization Center at Narvik University College. This room is equipped with a screen size of 6.5 meters by 1.5 meters (width x height). The room is equipped with active stereo 3D image generation technology which requires the participants to wear special glasses while looking on the screen. Currently only keyboard and PC mouse was available for the participants as interaction device. We would like to repeat the test with using other input devices also in the near future. The general view for the participants, when they entered the room can be seen in Fig. 3. When the tests were started the VirCa was aligned to full screen to cover the whole area of the screens. The participants were sitting in the first row, closest to the screen. The keyboard and the mouse were placed in front of them on a table.

Fig. 3. Overview of the experimental room in Narvik University College

Procedure

As the system could only be instructed by one mouse and keyboard, all the participants have been executing the test one by one. They were not allowed to discuss experiences, while waiting in a separate room.

When a participant entered the visualization room, he or she was greeted by the session coordinator. He briefly summarized the goal of the test and also described the capabilities of the room and also handed over the special goggle for 3D stereoscopic view of the screen. The participant was informed that he or she can abort any time and also that the coordinator will not answer any questions during the test itself, but after the test he or she should fill in a questioner and will have the possibility to discuss his or her feeling about the experiment.

The participant did not get any more information about navigation, usage of the system. All these information were provided inside the VirCa program itself. When the participant was sitting in front of the screens an overlay textbox were showing the basic navigation functions and instructions.

VirCa setup

In order to test the navigation capabilities of the users a room has been created inside VirCa environment. On two walls of the room there were posters with different texts and images (totally 6) and one of these posters contained the instructions for the task. There was also one poster containing the initial overlay textbox text, so the user could come back anytime to look on the basic navigation settings. The room also included an industrial robot, a table and dominos, which will be used during the tasks. An overview of the simulated room is shown in Fig. 4.

Fig. 4. Navigation test environment inside VirCa

Navigation

As already mentioned the navigation settings are introduced to the user in the beginning and also always available as a poster on the walls. The following text is shown to the use:

“There is a hand pointer in the middle of your view, what shows your direction.

You can modify the direction with using the mouse. If you want to go forward, use the keyboard and push – j – for “jumping forward”.

The length of the jump (and the pointer) is defined by the length of your hand pointer. The length could be adjusted by the mouse scrolling wheel.

By holding in the Ctrl key, the hand pointer lifts up an object you are pointing to. The object has to be in the range of the hand pointer.

Menus are shown with the right click of the mouse. You have to first point on something to query the menu. You can move in-between menu options by moving the mouse sideways. Left click enters the menu option. Escape key closes the menu.

If you feel ready left click with your mouse, find the poster on the walls with the instructions.

P.S.: You can also find this text on one of the posters, anytime you need it.”

Tasks

The user had to move inside the room and find the appropriate poster with the tasks listed on it. During this movement to the poster, they could practice the usage of the hand pointer. When the user found the poster, the following text was on it:

“1. There are dominos lying on the floor in the room. Find all the six pieces and put them on the table next to the industrial robot.

2. Make the industrial robot pick up one of the balls. You can order the robot to move with accessing the menu of the robot.

3. Signal to the session coordinator, that you are done."

After reading the tasks, the user had to go through the room and look for the dominos lying on the floor. There were four dominos in the four corners of the room, one in the center and one lying under the table. The one that was under the table, was little bit more complicated to move than the rest of the dominos. The key to the successful movement of the dominos, whether the user figured out that the translational movement of the domino could be achieved by scrolling the mouse.

Usually (4 out of 6 participants) the user had to go back to the poster with instructions on it to look on the task once more, before continuing the task with the industrial robot. The manoeuvring in the menu was always successful, never had to push Escape to start over.

In the visualization room there was one extra person (in addition to the session coordinator) sitting close to the screen to take notes and measure time for executing the tasks. As stated forehand there was no possibility to ask any questions from the supporting persons. All the participants have completed the test.

Questionnaires

Having completed all the tasks, the participants were given two questionnaires to answer on spot. One questionnaire is System Usability Scale (SUS), fine-tuned by Lewis et al. (Lewis, 2009), and the other is based on the work of Komlodi et al. (Komlodi, 2011).

The participants started first with the SUS (that was more rapid to answer) and after that they got approximately 5 minutes to answer the second questionnaire.

This questionnaire contained the following questions:

1. Please describe your general impression of the system.

2. On a scale of 1-10 how would you rate your presence in the room (1 none – 10 in the room)?

3. What was the easiest for you?

4. On a scale of 1-10 how would you rate of dizziness during the testing (1 nothing – 10 fainted)?

5. What was difficult for you?

6. What improvements do you recommend?

7. Would you like to use it again?

8. Please describe you experience with one word.

9. Would you recommend it to your friends?

It was constructed in such a manner to avoid responses that require no thinking. Question 1, 3, 5, 6, 8 are dealing with the general impressions for the system such as difficulty or easiness. Question 2 tries to target the participant's cognitive skills, while question 4 focuses on the disadvantage of such immersive environment. Question 7 and 9 measures the satisfactory level of the participant.

4. RESULTS

After all the participants have finished and filled in the two questionnaires, the session coordinator and the extra person situated in the visualization room have sit down and made a summary of the tests.

The mean SUS score was 65, while the lower bound was 55 and the upper bound was 80. This means that from the usability point of view VirCa is in the acceptability range, however further improvements are recommended.

The results from the other questionnaire and the individual assessment of the tests could be summarized in the following form:

All the participants described their impression of the system to be handsome to use. They stated also that the help they received in the beginning and the fact that they could anytime go back to read it again was crucial in accomplishing the tasks.

As the system was new to all the participants, there was a small learning period (less than a minute), when they get used to the control and navigation of the system. This time was relatively short, as all the participants had previous experience with some kind of simulators. Other key observation was that with using well known input devices (keyboard and mouse) the users were more confident as those who are introduced to newer input devices, such as in [7].

None of the participant reported dizziness or any other symptoms. In a short testing period (approximately 20 minutes per participant) this was not expected to happen.

In the current state we were focusing on simple, easily accomplishable tasks as our focus was on navigation in the immersive 3D environment. This study shows that with using standard input devices no one had any challenges with this. With introducing more advanced interaction techniques, this could turnaround.

The other founding is that even the usage of keyboard and mouse could give the feeling of presence to the users. A mean score of 6 is a quite good result for this.

The navigation inside menus was the most challenging for the users. The movement relation between the mouse movement and the menu item changing was the most difficult to follow. Participants also reported that visibility of text on the menu items should also be increased.

5. RECOMMENDATIONS

Based on the above mentioned results, the following recommendations can be formed:

It would be a nice feature if the user could put a mirror in the room. When you are manoeuvring an object to a specified spot, there is sometimes need to have another point of view also.

Beside the objects visibility, there could be small light signals (small bulbs over it) showing status, if the object is ready, busy, has error, etc. This is mostly related to machines.

Movement in the menus should be made simpler by letting the movement by keyboard also. More specific: to use the keyboard left, right and enter characters to navigate in the menu.

To give a better presence there could also be a 2D map in the corner of the screen. This increases the spatial understanding of the current situation.

6. CONCLUSIONS

In this study we have focused on the usability of an immersive 3D software tool, called VirCa. The above described methodology and results could server a stable basis for further studies. It is our goal to extend the study to include more people, with more versatile skills (novice and expert users), and to introduce more advanced input devices. The tasks executed during the experiments are simple enough to be repeatable with other input devices also, without losing the focus from the new technologies.

ACKNOWLEDGEMENT

This work has been supported by BANOROB project, the project is funded by the Royal Norwegian Ministry of Foreign Affairs

REFERENCES

Bowman D.A., Kruijff E., LaViola J.J., and Poupyrev I. “3D User Interface: Theory and Practice”. pp. 87-134. Addison-Wesley/Pearson Education, 2005.

Bowman D.A., Coquillart S., Froehlich S., Hirose M., Kitamura Y., Kiyokawa K. “3D User Interfaces: New Directions and Perspectives”. IEEE Annals of the History of Computing , pp. 20-36, 2008.

Dang N., Tavanti M., Rankin I., and Cooper M. “A Comparison of Different Input Devices for a 3D Environment”. in Proceedings of the ECCE 2007 Conference, pp. 153-160, 2008.

Galambos P., and Baranyi P. “VirCA as Virtual Intelligent Space for RT-Middleware”, in Proceedings of 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2011), pp. 140-145, 2011.

Gallo L. and Pietro G.D.. “Input Devices and Interaction Techniques for VR-Enhanced Medicine”, Multimedia Techniques for Device and Ambient Intelligence, Part 2, pp. 115-134, 2009.

Komlodi A., Jozsa E., Hercegfi K., Kucsora S. and Borics D. "Empirical usability evaluation of the Wii controller as an input device for the VirCA immersive virtual space ", in Proceedings of International Conference on Cognitive Infocommunications (CogInfoCom), pp. 1-6, 2011.

Korondi P., Baranyi P., Hashimoto H., Solvang B. "3D-Based Internet Communication and Control." Computational Intelligence in Engineering, Springer-Verlag, pp. 47-60., 2010.

Lewis J. R. and Sauro J.. “The Factor Structure of the System Usability Scale”, Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009, pp. 94-103, 2009.

Vincze D., Kovács Sz., Gacsi M., Korondi P., Miklosi A., Baranyi P. "A Novel Application of the 3D VirCA Environment: Modelling a Standard Ethological Test of Dog-Human Interactions."ACTA POLYTECHNICA HUNGARICA 9:(1) pp. 107-120, 2012.

6. Additional papers

In the time period of the thesis many additional articles were published, which were serving a strong basis of the PhD work. These articles are listed in the following, in chronological order.

Papers published but not included in the thesis:

Paper 13: Gabor Sziebig, Andor Gaudia, Peter Korondi, Ando Noriaki, Video image processing system for RT-middleware. In: Proc. 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI'06). Budapest, Hungary, 24/11/2006-25/11/2006. pp. 461-472.

Paper 14: Bjørn Solvang, Peter Korondi, Gabor Sziebig, Ando Noriaki, SAPIR: Supervised and Adaptive Programming of Industrial Robots. In: Proc. 11th IEEE International Conference on Intelligent Engineering Systems (INES'07). Budapest, Hungary, 29/06/2007-02/07/2007. (IEEE)pp. 281-286.(ISBN: 1-4244-1147-5)

Paper 15: Gabor Sziebig, Chen Hao, Lorant Nagy, Peter Korondi, Complete multimedia educational program of a DC servo system for distant learning. In: Proceedings of Computational Intelligence and Informatics (CINTI), 8th International Symposium of Hungarian Researchers, Budapest, Hungary, 15/10/2007-17/10/2007. pp. 283-293. Paper 25. (ISBN: 978 963 7154 65 2)

Paper 16: Gabor Sziebig, Andor Gaudia, Peter Korondi, Ando Noriaki, Bjørn Solvang, Robot Vision for RT-Middleware Framework. In: Proceedings of the IEEE Instrumentation and Measurement Technology Conference, IMTC '2007. Warsaw, Poland, 01/05/2007-03/05/2007. Piscataway: IEEE, pp. 1-6.(ISBN: 1-4244-0588-2)

Paper 17: Gabor Sziebig, Bela Takarics, Viliam Fedak, Peter Korondi, Virtual Master Device. In: Proc. 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI'07). Poprad, Slovakia, 25/01/2007-26/01/2007. pp. 29-40. Paper 4.

Paper 18: Gabor Sziebig, Istvan Nagy, R K Jardan, Peter Korondi, Integrated multimedia educational program of a DC servo system for distant learning. In: Proceedings of 13th Power Electronics and Motion Control Conference (EPE-PEMC 2008). Poznan, Poland, 01/09/2008-03/09/2008. pp. 2360-2367.(ISBN: 978-1-4244-1741-4)

Paper 19: Peter Korondi, Bjørn Solvang, Gabor Sziebig, Peter Baranyi, An interactive human - robot programming methodology. In: Manufacturing 2008. Biannual 19th international conference. Budapest, Hungary, 06/11/2008-07/11/2008. Budapest: pp. 125-133. (ISBN: 978-963-9058-24-8)

Paper 20: Gabor Sziebig, Peter Korondi, Zoltan Suto, Peter Stumpf, Kalman R. Jardan, Istvan Nagy, Integrated e-learning projects in the European Union. In: IECON 2008. Orlando, United States of America, 10/11/2008-13/11/2008. IEEE, pp. 3524-3529.(ISBN: 978-1-4244-1767-4)

Paper 21: Peter Korondi, Hideki Hashimoto, Gabor Sziebig, Sliding Sector Design for

Nonlinear Systems. In: Proc. International Federation of Automatic Control World Congress (IFAC WC 2008). Seoul, South Korea, 06/07/2008-11/07/2008. (IFAC)pp. 3556-3561. Paper 3477. (ISBN: 978-3-902661-00-5)

Paper 22: Peter Szekeres, Bela Takarics, Gabor Sziebig, Peter Korondi, Control of Embedded System via Internet. In: 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics. Budapest, Hungary, 06/11/2008-08/11/2008. Budapest: pp. 171-182. Paper 16.

Paper 23: Gabor Sziebig, Peter Zanaty, Vision-Based Robot Programming., In: Proceedings of Sixth Conference on Mechanical Engineering. Budapest, Hungary, 29/05/2008-30/05/2008. Budapest: Budapesti Műszaki és Gazdaságtudományi Egyetem, pp. 1-12. Paper H-02. (ISBN: 978-963-420-947-8)

Paper 24: Peter Korondi, Peter Baranyi, Bela Takarics, Gabor Sziebig, TP Model Based Sliding Control Design. In: 15th International Conference on Electrical Drives and Power Electronics (EDPE 2009). Dubrovnik, Croatia, 12/10/2009-14/10/2009. Dubrovnik: pp. 1-8. Paper TUT-004. (ISBN: 953-6037-56-8)

Paper 25: Peter Korondi, Bela Takarics, Gabor Sziebig, Friction Compensation Based on LMI Control Design with Tensor Product Transformation - Tutorial. In: 15th International Conference on Electrical Drives and Power Electronics (EDPE 2009). Dubrovnik, Croatia, 12/10/2009-14/10/2009. Dubrovnik: pp. 1-7. Paper TUT-002. (ISBN: 953-6037-56-8)

Paper 26: Bela Takarics, Gabor Sziebig, Peter Korondi, Small Batch Size Robot Programming with Human in the Loop. JOURNAL OF AUTOMATION MOBILE ROBOTICS & INTELLIGENT SYSTEMS 3:(4) pp. 228-231. (2009)

Paper 27: Peter Zanaty, Peter Korondi, Gabor Sziebig, Laszlo A Jeni, Image based automatic object localisation in iSpace environment. In: 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI 2009). Budapest, Hungary, 12/11/2009-14/11/2009. Budapest: pp. 501-512. Paper 43. (ISBN: 978-963-7154-96-6)

Paper 28: Bela Takarics, Gabor Sziebig, Bjørn Solvang, Peter Korondi, Multimedia Educational Material and Remote Laboratory for Sliding Mode Control Measurements. JOURNAL OF POWER ELECTRONICS 10:(6) pp. 635-642. (2010) IF: 0.779

Paper 29: Bela Takarics, Gabor Sziebig, Peter Korondi, Telemanipulation without Physical Contact in the Intelligent Space. ADVANCED MATERIALS RESEARCH 222: pp. 362-366. (2011)

Paper 30: Tamas Pallos, Gabor Sziebig, Peter Korondi, Bjørn Solvang, Multiple-Camera Optical Glyph Tracking. ADVANCED MATERIALS RESEARCH 222: pp. 367-371. (2011)

Paper 31: Azin Aryania, Balazs Daniel, Trygve Thomessen, Gabor Sziebig, New trends in industrial robot controller user interfaces. In: 3rd IEEE International Conference on Cognitive Infocommunications, CogInfoCom

2012. Kosice, Slovakia, 02/12/2012-05/12/2012. pp. 1-5

7. Conclusion

In the following I will summarize the scientific results of the thesis. The results could be grouped into the following 3 categorizes:

I. *Introduction of a new scientific concept of task dependent, software component based controller for flexible manufacturing cells.*

In case of small batch production, the change between products is crucial. The concept provides possibility for widespread communication between cell members. The traditional way of communication between cell members was until now limited to signaling states of machines (ready, wait, busy) or giving commands for starting/stopping specific tasks on the machine. In the new proposed concept the manufacturing cells' members, sensors are connected to a centralized network, where they can exchange information easily. In addition, this type of communication is not limited to machine-machine communication; the concept also covers human-machine or human-cell communication. All the cell members are built up from multi-layered, standardized software components, which can be freely combined to provide a new type of equipment. The cell members are instructed by a core component, which interprets the task of the manufacturing cell and forms the system architecture correspondingly. E.g.: an industrial robot, which has the task to load and unload a work-piece from a CNC milling machine, is usually pre-programmed for each single type of work-piece that is manufactured at the CNC machine. In our solution the industrial robot will not follow a pre-programmed path directly, but rather use the data (work-piece information) received directly from the CNC milling machine to grab the freshly produced work-piece.

Related papers: Paper 1, 10, and 11

II. *Development of a new paradigm of robot teaching and supervising, which opens a new dimension of robotization in the area of small and medium sized enterprises.*

The robot is considered as an unskilled worker which is strong and capable for precise manufacturing. The robot possesses a special kind of intelligence but it is handicapped in some senses (without any external sensor, the industrial robot is a "blind", programmable manipulator arm), hence it needs specific treatment. We have to command it clearly in a special way and we have to supervise its work. If we can learn how we communicate with this "new worker" we can get a new capable "colleague".

The long term goal is that the supervisor would be able to give the daily task to a robot in a similar way as he/she assigns jobs to the human workforce, for example, using CAD documentation and some verbal explanation. The robot understands the task given by the CAD drawing or verbal/visual instructions and executes it in a safe and optimal way. This way the overall programming time of the industrial robot is vastly decreased, which allows for a much higher capacity utilization.

Related papers: Paper 2, 3, 4, 5 and 11

III. Introduction of a new concept for Coginitve Telemanipualtion.

The idea behind the concept is that for feedback from the process of telemanipulation I am utilizing new type of communication channels.

In my concept, the process of telemanipulation can be observed and executed through a virtual environment, where the temperature is fed back with color, forces are represented by vibrations and the whole process can be seen through using visual feedback. The visual information can be shown on a standard computer display or using a 3D stereoscopic display or using a 3D capable head mounted display. The operator's movement in this virtual environment is tracked by a motion capture data suit. By interpreting the operator's movement information we can execute commands on the slave devices in the virtual environment, which can correspond to a mobile or industrial robot, manipulator or machining center. The virtual copy of the physical slave device interprets and executes the command, while its status can be monitored in the virtual environment. One of the benefits of this setup, that the physical slave device could be located far away from its virtual copy.

Based on this concept the virtual environment was implemented and is now used for demonstrational purposes.

To replace the complex and expensive force feedback systems I have contributed in the development of a glove utilizing tactile vibration, which is now being used in various laboratory experiments.

Related papers: Paper 6, 7, 8, 9 and 12

8. Future work

In addition to previously presented results in the Paper Section, a fully functioning flexible manufacturing cell is under implementation at Narvik University College in the time of writing the thesis. The idea is to implement the previously introduced task dependent, software component based controller in real-life situation.

One part of this work will be to compose the controller itself and the other will be in establishing the new type of human-system interaction in robotic applications.

The FMC is composed from 3 industrial robots interconnected with bi-directional conveyor belts. At the end of the production line there are two NC machines waiting for work-pieces to produce small batch sizes of one of a kind products. This cell could be seen in Fig. 4.

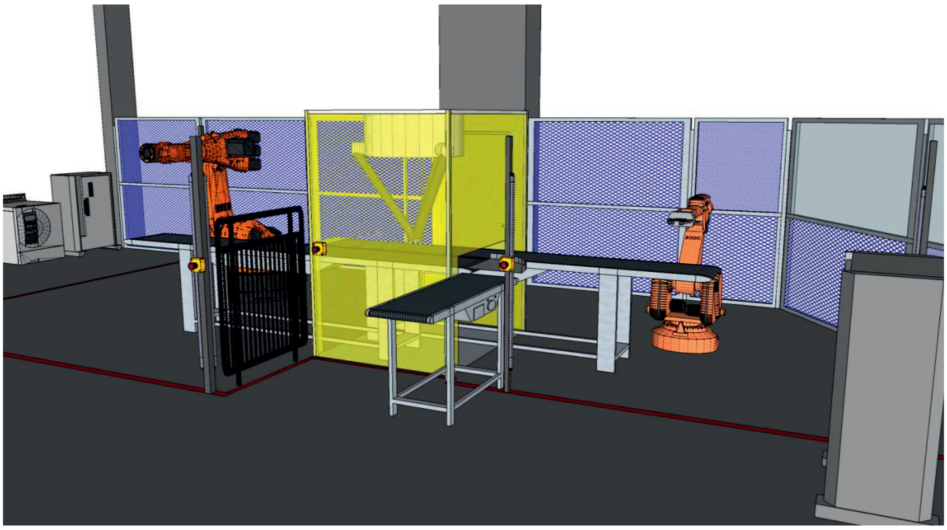


Fig. 4. 3D model of FMC at Narvik University College

The expected outcome of this pilot project is:

- Further confirmation of capabilities of software component based controller is FMC environment
- Introduction of high level and intuitive programming of FMC
- Introduction of STEP based data-exchange between cell components
- Extensive testing of previously introduced human-system interaction in industrial environment