Technical University of Denmark

DTU

Development of a robot Holon using an open modular controller

Schnell, Jakob; Andersen, Søren; Sørensen, Christian; Langer, Gilad

DTU Library
Technical Information Center of Denmark

Proceedings of the 1999 IEEE
International Conference on Control Applications
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

WP4-4   5:20

Development of a robot holon using an open modular controller.

M.Sc. Jakob Schnell (jschnell@vip.cybercity.dk)
M.Sc. Søren Andersen (c922801@student.dtu.dk)
M.Sc., Ph.D. Gilad Langer (gilad@ipt.dtu.dk)
M.Sc., Ph.D. cand. Christian Sørensen (cs@ipt.dtu.dk)

**Department of Manufacturing Engineering
Building 424, DK-2800 Lyngby, Denmark
Tel.: +45 4525 4816, Fax.: +45 4525 4803**

1   Abstract

Holonic Manufacturing Systems (HMS) has during the last period presented itself as an advantageous theoretical foundation for the problems that arise in controlling agile manufacturing systems. Previous research, at the Department, has demonstrated how modern shop floor control systems can be developed based on standard architectures for cell-control supported by engineering concepts and enabling technologies. In continuation to this research new concepts and theories for shop floor control are investigated. Ongoing research on HMS has resulted in development of the Holonic Multi-cell Control System (HoMuCS) architecture and methodology for implementing a HMS. This paper specifically reviews the development of a Robot Holon based on an open controller in the context of the HoMuCS architecture. The paper will describe the results and research work that was involved in developing a robot holon for a physical robot. The robot holon was implemented on a existing robot at the department which was upgraded by removing its native control system and replacing it with a new PC-based open controller. The development of the robot holon builds on the notion that a robot holon will be able to perform both processing tasks and material handling tasks. Based on that an attempt to draw up a robot-architecture in the HoMuCS that can easily be reconfigured for these types of tasks. The research results gave a further specification of the HoMuCS architecture by extending it with the special robot holon type.

2   Introduction

The effects of the new trends in the business environment on manufacturing system control, point clearly at the need for new concepts regarding manufacturing system control. Technology alone will not be the solution for the control problems encountered by this new type of manufacturing systems. A completely new approach is needed, which involves new control structures that are distributed and are characterised by co-ordination of different autonomous units. The theory regarding Holonic Manufacturing Systems (HMS) supports this by presenting an advantageous theoretical foundation for the control system of the manufacturing system of the future. Previous research, at the Department, has demonstrated how company tailored shop floor control can be developed by applying simulation and cell control enabling technologies (Lynggard, 1997). In order to continue this research effort new concepts and theories for shop floor control are investigated. The effort is based on the ongoing research on HMS, and the current results regarding the development of the Holonic Multi-cell Control System (HoMuCS) architecture and methodology. A holon is the basic building block of

a HMS. It is an autonomous and co-operative element of a manufacturing system that is responsible for transforming, transporting, storing and/or validating information and physical objects. A holonic manufacturing system is dependent on the existence of specific machine holons such as CNC and robot holons. This paper focuses on the research regarding development and specification of such a holon namely a Robot Holon using an open robot controller. The work is based on a theoretical study of new manufacturing system theories, research of open modular controller technology, and development of a prototype robot holon based on a case study.

3   The robot holon in a HoMuCS

The HoMuCS architecture consists of a set of customisable components described by object-oriented models (Figure 1).
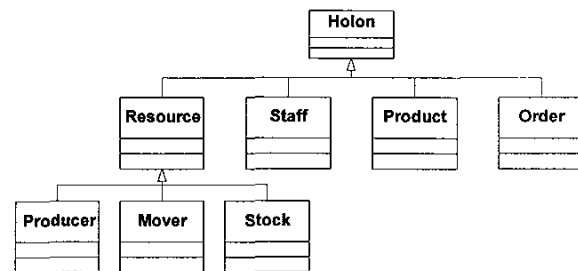
Figure 1: Part of the class hierarchy and specialisation structure of the holons. (Langer, 1999).

The root of the class hierarchy is the Holon class that provides all generic services for a holon. The Holon class is specialised into the four basic holons defined by the reference architecture (Wyns, J 1999)(Bongaerts, et al.,1996). Furthermore the basic holons are specialised into more specific classifications. The Resource class is specialised into a Mover and a Producer type thus distinguishing between the behaviour needed to perform these two basic types of tasks that manufacturing resources execute.

Implementation of a robot in a holonic system presents a problem because it can neither be classified as a mover type holon or as an producer type holon. In fact it can perform both functions since modern robots can be used for both processing and handling tasks. Therefore in a holonic system the robot should inherit from both types in order not to restrain its inherent capabilities. However in current manufacturing systems robots are typically only configured for specifically one task and thus will be seldom able to perform a processing task like welding and a moving task, such as handling a work-piece sequentially.
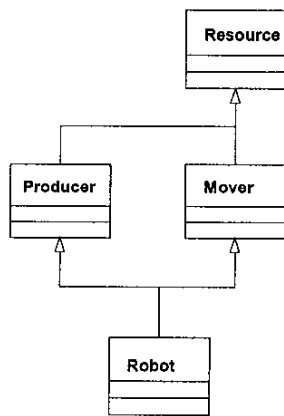

Figure 2: Specialising the robot holon.

Considering the requirements of the future manufacturing enterprise agility will become a vital parameter. Thus allowing a robot to be used for both types of operations using the same installation is essential. A holonic manufacturing systems based on the HoMuCS architecture allows just that by prescribing an agile robot holon that is able to perform both processing and handling tasks. In fact if the robot were mobile it would expand its work area and thus decrease the numbers of robots needed since no reconfiguration is needed. Robot mobility is advantageous for transportation tasks simply because one robot could cover a bigger area, yet processing tasks where the robot could move to the location where the job has to be performed also reduces costs and increases agility. This is an advantage if the work-piece is physically large where it is easier to move the processing entity rather than the work-piece. Finally since holonic systems are based on co-operation, robots can co-operate on jobs by assisting each other, e.g. one robot can handle a work-piece while the other welds it.

Current robot controllers do not have the ability to react to changes or disturbances during the execution of a program. In such a case the robot needs to be reprogrammed. This deficiency can be bettered by use of sensors, however this requires that all the possible states of a sensor are known when developing the robot programs. A traditional controller is not suitable for implementing a robot holon and therefore an open controller is applied in this research in order to obtain the required agility. In order to develop a robot holon that includes both types of tasks in a holonic manner, the research has been divided in two steps. The first being the analysis of how a robot performs as a mover holon type or a producer holon type separately

and then try to combine these two robot holons into one generic robot holon.

3.1 Robot as a mover type holon

Material handling is not part of the process plan for a specific product. Handling of material is performed to move the orders between the various resources executing the process plan. In a HMS the allocation of resources is dynamic and thus the production plan and the route an order will take through the production is not explicitly predetermined. Therefore the material handling resources need to be able to obtain the necessary information about the task and transform this data into the desired movement. In holonic manufacturing systems this is done through communication and co-operation between the holons.

Basically such a handling task consists of a sequence of linear moves and rotations, in its simplest form the mover robot will move in straight line from one point to the next. This of course will not work if the robot needs to remove a work-piece out of a machine in order to avoid collision. The same is valid when a part needs to be fixed in a CNC machine where the robot might have to slide the part slowly into the clamp on the CNC machine and hold it still while the CNC closes the clamps. This requires information exchange between the mover, the order and the machine (CNC) as the HoMuCS architecture prescribes. Furthermore when the order is free of the CNC and the robot needs to move it to a new location it has to be aware not to collide with other objects in the cell.

Considering the description above it is clear that a robot holon for material-handling tasks in a HoMuCS is a specialisation of a mover type holon robot. The extension of the HoMuCS architecture with this type of robot holon contributes to its development by adding a mover type that has extended agility because of the numbers of different jobs that it can handle.

3.2 Robot as a producer type holon

Robots are commonly used for processing tasks in industry. They are widely used for repetitive tasks often in a hostile environment, e.g. welding and painting tasks and in mass production system, where they perform the same task over a long period. The disadvantages of robots are that they are costly to reconfigure for new tasks. The right positioning of the robot is important, and the program that should be executed often has to be reprogrammed a few times before the robot performs optimally, which can be costly.

As a processing resource a robot is comparable with a CNC machine. It receives information about the processes and executes the assigned jobs. Depending on the versatility of the robot it could have tool storage, similar to FMSs (Flexible Manufacturing System), that contains tools like welding equipment, painting equipment etc. Yet, a mobile robot can move to the work-piece that needs processing, where most other processing machines needs the material delivered at a certain point.

A robot operating as a Producer in a holonic manufacturing system does not receive an IRL (Industrial Robot Language) file or any other kind of predetermined description of the job. Instead the robot should be able to generate this information from the information contained in a geometric description of the job. For example a robot holon would receive a CAD file that specifies a welding and would then plan the execution of the welding job by selecting the tool, the

specific welding parameters, and generate a path description similar to an IRL file. A robot holon used as a processing resource can be a specialisation of the Producer holon class in the HoMuCS system-architecture.

3.3 Generic robot holon

A robot in a holonic system has to be able to be both a material handler (mover) and a processing resource (producer). Resolving this dual functionality involves multiple inheritance from the producer type holon and the mover type holon (Figure 2) in the HoMuCS system-architecture. This allows for a generic robot holon that has the basic functionality of both the producer and mover holon classes.

The aim of the research described in this paper was to develop a generic robot holon class that can be used for implementing a robot for holonic shop floor control systems. Such a robot can operate as a mover as well as a producer. Physical constraints have to be considered for a robot that is capable of handling both types of tasks. Processing tasks require agility and precision, while handling/moving tasks demand strength and lifting capacity. This is one of the reasons why current robots typically are specialised for one type of functions. Other reasons are that the current production facilities do not need robots that can perform both these tasks, which leads back to the discussion regarding the future manufacturing systems need for increased agility.

Presuming that these types of robots are realisable the first step would be to develop a robot that is easily reconfigured from a mover type into producer type holon and vice-versa. Agile production systems require that cells be continuously reconfigured as new production orders are introduced. Thus the ability of the robot to be reconfigured as either a mover or producer reduces expenses and increase agility.

In it simplest form a robot holon provides ease of reconfiguration by allowing manual reconfiguration of the control application by a specialisation of the generic robot holon. The architecture of the robot holon specifically and the HoMuCS architecture generally facilitates reconfiguration since all that is needed is the definition of implementation specific control logic. If this is performed appropriately the robot can simply be configured by "plugging" it in to the system. A more advanced implementation would be a self-configuring robot that would be able to learn from experience and reconfigure itself as needed.

Obviously such a holonic robot requires a complex control system. The robot has to be able to decide what kind of functionality should be applied based on the job that the order specifies. A job is no longer just a processing job, it can vary between welding, painting and so forth. The holonic characteristics of autonomy and co-operativeness allow the robot to receive both transformation and transportation orders and execute them. A standard interface to the other resources in the system is required to allow for co-operation with the other holons in the cell. This is achieved through the HoMuCS system-architecture, open robot controllers and application of state-of-the-art computing technologies such as artificial intelligence and multi-agent technology (Langer, 1996).

4 Open controllers

Utilisation of automation equipment is steadily increasing in modern manufacturing due to decreasing costs of implementation, computer

hardware and software. Consequently manufacturing engineers are experiencing problems that arise with the implementation of this new equipment in current production systems. One of the problems is the lack of consistency in the interface, from one vendor to another. Likewise problems often appear when adjusting this new equipment to fulfil special requirements as well as integrating it with existing peripherals. This has brought about the demand for a neutral and open interface to new automation equipment, specifically their controllers. Such an open interface is the focus of several researches and standardisation initiatives, which are aimed at developing a standard architecture for automation equipment controllers that have a neutral and open interface. In the United States the General Motors Powertrain Group (GMPTG) has been working on implementing an Open Modular Architecture Controller (OMAC) since 1986. This research has had increased interest lately (GMPTG, 1996). The Open System Architecture for Controls within Automation systems (OSACA) was developed by a European commission sponsored ESPRIT III project that identified the necessary specifications for an open control system, and has suggested such architecture (Figure 3). |
The IEEE define an open system as: "A system that provides capabilities that enable properly implemented applications to run on a wide variety of platforms from multiple vendors, inter-operate with other system applications, and present a consistent style of interaction with the user" (Sperling & Lutz, 1996). Such an open interface to a controller allows the integration of equipment and other resources with less configuration problems and increased flexibility compared to conventional controllers.
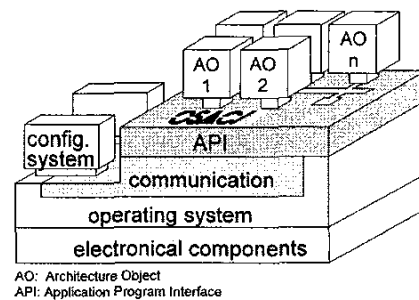

AO: Architecture Object
API: Application Program Interface

Figure 3: System architecture for open control systems (Sperling & Lutz, 1996).

Open controllers are commonly built on a four-layer architecture (see Figure 3). The bottom layer is the hardware layer, consisting of processor-boards, I/O-cards, Motion-control-boards and other peripheral equipment. The next layer is the operating system. Because of the large variety of hardware configurations the operating system should not be dependent on specific hardware components. The communication layer handles the inter-process communication. The top layer is the Application Program Interface (API) layer. This is where the user can add specific modules Architecture Objects (AO) that are developed in order to add specific functionality to the controller. The modularity of this architecture allows "plug and play" of components, reuse of components, and generally facilitates maintenance and reconfiguration.

The benefits of open controllers are that they ease reuse of applications components, upgrading of existing installations that are limited by an outdated control-system, and generally increase

1644

flexibility and performance of both new and existing hardware. They provide developers of agile manufacturing systems with a neutral interface to a large number of the numerical controlled equipment and robots and give way to an even more complex and agile shop floor control system that is ideally independent of its hardware.

4.1 The OMC

There are a variety of open controllers with different implementation solutions. The Open Modular Controller (OMC) developed at the Odense Steel Shipyard (OSS) is an example of one such implementation (Jensen, 1998). In contrast to the OSACA architecture (Sperling & Lutz, 1996), the OMC developers at OSS have chosen the PC hardware platform. This decision was based on the superior price-performance ratio that this platform offer compared to the more traditional industrial computing systems, such as 68k/VME based solutions. The PC platform additionally offers a vast variety of hardware modules for the ISA/PCI busses, which makes it easy to find the ideal board for each specific implementation case. Windows NT was chosen as the operating system also because of the advantageous price-performance ratio compared to other 'real'-time operating systems. Additionally device drivers supplied by vendor of plug-in boards offer relatively simplified communication to peripheral devices. Thus the OMC is build on standard or "de facto" standard components. The OSACA standard (Sperling & Lutz, 1996) suggests that the operating system for open controllers comply to the POSIX 1003.1b standard, which Windows NT does not. This was dealt by placing a Virtual Kernel module (VKernel) between the operating system and the OMC, which acts as a neutral operating system to the OMC.
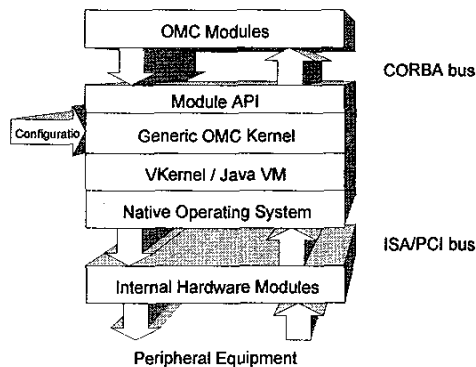

Figure 4 The OMC architecture (Jensen, 1998).

The VKernel offers an object oriented implementation of the most common multi-tasking concepts, e.g. threads, mutexes, semaphores and shared memory. The communication between the OMC modules and the OMC module API is implemented through a Common Object Request Broker Architecture (CORBA) bus. This solution offers both a dynamic modular interface to the controller as well as allowing development of modules in a vast number of programming languages such as C, C++, Java, Pascal, etc. Currently it is limited to C++ and Java. The CORBA bus also features the possibility of distributing the modules across the network.

5 Cases Studies

The case study work was performed using an ASEA IRB 6/2 robot (this robot model is from 1984) at the Department of Manufacturing Engineering, the Technical University of Denmark. It involved removing its native control system and replacing it with the new PC-based OMC controller. The aim of the case study was to develop and investigate the HoMuCS architecture and methodology further by investigating the applicability and practicality of the theory in practise. This was performed through development of a physical robot Holon installation on the ASEA robot. In the following sections the implementation of the OMC controller for the ASEA robot is described. The developed robot holon, which was implemented as part of this research is described in (Schnell & Andersen, 1999) and (Schnell et al. 1999) and exemplified through two scenarios that respectively describes the robot operating as a Mover Holon and as a Producer Holon. The description in this paper focuses on implementation of the OMC and its evaluation as an enabler for Holonic automation equipment.

5.1 Implementation of the OMC on the ASEA IRB6/2

The research work was dependent on the implementation of the Open Modular Controller (OMC), which required a large work effort. Moreover it presented many practical problems because of the outdated robot hardware and the fact that the OMC software is still under development. The advantages of such a controller are significant because of two factors; firstly it makes it possible to reuse older robots that are inhibited by their outdated control system. Secondly it allows the development and implementation of control systems without any restriction regarding their achievable functionality. The OMC made it possible to develop a control system that can react on sensor input during process execution, which is an essential feature needed in order to obtain autonomous characteristics. Another advantage of the OMC is that it is simple to acquire status data from the robot. A monitoring function can query the controller at any time regarding its status, e.g. how many hours a specific motor has been running, since its last maintenance.

The OMC was installed on the robot by replacing most of the original controller, leaving only the servo amplifiers and power supply. The OMC hardware used in this case consisted of a PC running Windows NT, a PMAC motor control board from Delta Tau technologies and an I/O card. This is the most common configuration of the OMC although it allows other devices to handle the connection to the servos, which have been implemented at OSS.

The IRB 6/2 is equipped with both velocity and position feedback. The position feedback is obtained from resolvers. The P-MAC motor control board uses encoder signals for positional feedback, which required additional resolver-to-encoder converter that can be used in conjunction with the P-MAC board and resolvers. The original controller used a phase-shifted signal from the resolver's rotor to determine their position, meaning that the resolvers used a common sinus and cosine reference on their static coils. To separate the resolver signals new wiring was needed since the resolver-to-digital converter feeds the resolver's rotor with a sinusoidal reference. Thus the resolver-to-digital converter determines the position based on the amplitude ratio between the reference and the signals from the static coils.

The controller is configured mainly with a mathematical representation of the robot, describing the lengths and types of the joints and links (rotational or translational). The other configurations are a description of how the robot is initialised (homed), various configurations regarding the used plug in boards, etc.

The Asea IRB 6/2 robot is designed with all actuators mounted on the first link (tower) that drives the individual links through various linkage mechanisms. This design has the advantage of reducing the mass that is moved and thus decreasing the load on the servomotors. The design results in a non-linear relation between joint and actuator positions that could not be handled by the OMC. The design also presents constraints between the motion of the joints. E.g. the movement of one actuator causes change to the positions of the other joints. The inverse kinematics model maps positions in Cartesian space to joint values. The OMC uses a pseudo Jacobian approach to solve the inverse kinematics model and generate the speed reference. **Figure 5** shows how a module that maps joint positions to actuator position was added to the controller in order to handle the constraints between the position of the joints. The Joint-to-actuator mapping module connects the motion controller and trajectory generator modules.
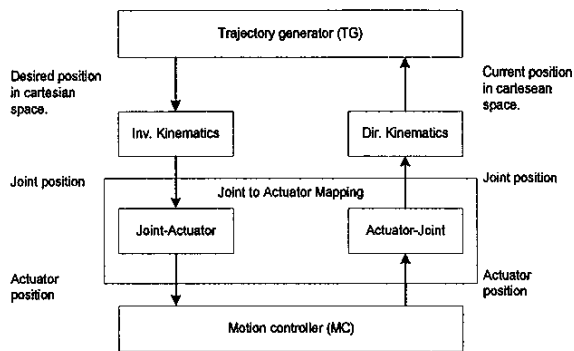

Figure 5 Joint to Actuator mapping

Besides solving the non-linearity problem the solution shows the strength of the modularity of the controller. The research results showed that installation of an open controller makes it possible to reuse outdated production equipment by significantly enhancing their functionality and features in general compared to their original control system. Additionally the implementation of the OMC supported the development of the robot holon by simplifying the realisation of autonomous and co-operative functionality in the context of a HoMuCS.

5.2 Developing a generic robot holon

In order to enhance reconfigurability between the OMC and the Robot Holon a generic controller interface was developed as an attribute of the robot holon class. This attribute provides the physical connection to the robot. A generic controller interface defines basic motion functions, functions to set speed and acceleration, acquire the current position and other generic functionality that is needed to control the robot. Thus for each specific controller this generic controller interface is customised and aggregated in the robot holon. This modularization allows the use of different controllers, for

example allows the implementation of a simulated controller that does not interface to physical equipment.

The development of the robot holon was performed by using scenarios that explore the different properties of the holon. In this case two scenarios were used, one that focused on the robot as a mover type holon and one that focused on the properties of the robot holon as a producer type holon (see section 3.The robot holon in a HoMuCS).

Mover scenario

As a mover type holon the robot should be abile to acquire the necessary information to move another holon from one position to another. The required information is the location of the holon that needs to be moved and the destination to which it has to be moved. Therefore a method of communicating physical locations between holons was developed. A scenario where the robot handled the materials within a cell containing the robot a machine and a stock, showed both the developed holons ability to co-operate and behave autonomous. The robot moved material around in the cell based upon the information that the robot acquired through interaction with the other holons. However one major aspect that was not dealt with in the scenario was the problem of obstacle avoidance, which is a complex area that was delimited.

Producer scenario

The robot as a producer type holon will not display significant different properties than any other numerical controlled equipment. Basically the robot moves some tool along a path defined by the product. For an example a welding job is given in a robot program or derived from a product description or CAD model at runtime. Traditional off line programming shows the disadvantage of binding a resource to a job. For example a NC file would be useless if the tool that the program was created for is not available. Robot programs made offline have difficulties handling jobs where the parts, due to previous processes, vary from the ideal geometry. After retrieving the description of the job, the robot should plan its execution, taking into account the actual state of the parts e.g. temperature, geometry, corrosion etc. To translate the description of job to motions an interpreter object is used. This concept enables more or less explicit descriptions of the process, the most simple would be download of a generated robot program file while a more advanced description would be extraction of, for example, the seems to be welded from a CAD model.
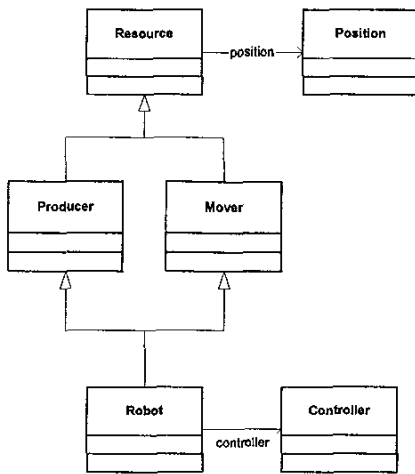
Figure 6 Robot specialisation in the HoMuCS architecture.

Figure 6 shows the resulting robot holon in a subset of the HoMuCS architecture. The attributes and methods of each class have been left out for readability reasons.

6 Conclusion

The research work presented in this paper, shows that it is practically possible to develop a holonic robot based on the HMS theory by implementing a robot controller based on an open controller architecture. The work has contributed to the development of the HoMuCS system-architecture by expanding the architecture to include a Robot Holon. The research provided some interesting results regarding the ease of obtaining agility when applying the HMS theory. Reuse and reconfiguration of the robot in different systems is possible in a minimal amount of time because of the multi-functionality provided to the robot holon. The robot holon adapts itself according the requests of a specific order. This adaptation can be either manual where an operator reconfigures the control of the robot or automatic reconfiguration by the intelligent internal logic of the robot. An interesting aspect of the research was the ability open controllers and the OMC specifically to reuse existing hardware such as robots and CNC machines that are outdated only because of their control systems. Furthermore the results of the research present some requirements for physical agility of holonic robots that have to be able to perform tasks that demand strength, precision and mobility.

7 References

Bongaerts, L., Wyns, J., Detand, J., Van Brussel, H. and Valckenaers, P (1996). Identification of Manufacturing Holons, *European workshop on agent oriented systems*. Berlin 26.-27. Sep. 1996.

GM Powertrain Group, Open (1996), Modular Architecture controls at GM PowerTrain – Technology and Implementation, Ver. 1.0, GM Powertrain Group Manufacturing Engineering Controls Council (GMPTG), USA May 14, 1996.

Jensen, S.M. (1998), Open Modular Controller, Proceedings of the 29th International Symposium on Robotics, Leuven, Belgium, 1998.

Langer, G. (1996), New Concepts and Theories For Intelligent Control of Cellular Manufacturing Systems, Presented at the Nordic-Baltic Summer School on Engineering Applications of Artificial Intelligence, Talin, Estonia, June 10-15, 1996.

Langer, G. (1999), HoMuCS - A methodology and architecture for Holonic Multi-cell Control Systems, Ph.D. Thesis, Department of Manufacturing Engineering, The Technical University of Denmark, KPD-2-99, Feb. 1999.

Lynggaard, H. J. (1997), Models And Methods For Design And Implementation Of Computer Based Control And Monitoring Systems For Production Cells, Ph.D. dissertation, Odense Steel Shipyard & Technical University of Denmark, DK, September 1997.

Schnell, J., Andersen, S., Langer, G., Sorensen, C.,(1999) "Development of a generic robot Holon for Holonic shop floor control systems." 7th IEEE Mediterrean conference on control and automation (MED99).

Schnell, J. and Andersen, S. (1999), "Integration af åben robot kontroller", M. Sc. Thesis, Technical University of Denmark – Department of Manufacturing Engineering, February 1999.

Sorensen, C., Langer, G., Alting, L. (1998). *"Developing a System Architecture for Holonic Shop Floor Control"*. Proceedings of the 5th IFAC workshop on Intelligent Manufacturing Systems (IMS'98), Gramado, Brasil, 9-11 Nov. 1998.

Sperling, W., and Luts, P. (1996), Enabling Open Control Systems – An Introduction to the OSACA System Platform, Robotics and Manufacturing, Vol 6, Edited by M. Jamshidi, et al., ASME Press New-York, May 1996.

Wyns, J (1999), "Reference architecture for Holonic Manufacturing Systems – the key to support evolution and reconfiguration", ISBN 90-5682-164-4, Ph.D. dissertation, Katholieke Univarsiteit Leuven – Faculteit Toegepaste Wetenschappen, Belgium, March 1999.