INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19–21, 2003

MODULARISATION OF MECHATRONIC MECHANISMS WITH DEPENDENT DEGREES OF FREEDOM

Björn Möller, Sören Andersson and Jan Wikander

Abstract

Modularisation and mechatronic solutions are two prominent trends in modern product development. For some mechatronic systems, modularisation is difficult. Such systems often involve functionally complex products such as robots and vehicles, where the mechanical function of a subsystem is strongly dependent on all the interacting subsystems. This paper addresses one of the problems posed by such systems.

In order to illustrate the problem and a possible solution to it, a feasibility study of a modularised mechanism test bench was undertaken using a behaviour control approach. The modelling and simulation tools SimMechanics and Simulink were used to evaluate the behaviour of this mechanism. The results indicate that it is possible to modularise the control of a complex mechanism, even if the degrees of freedom are strongly dependent.

Keywords: Distributed system, module control, behaviour-based control, perception sensor

1 Introduction

Modular products are composed of building blocks (modules) with standardised physical interfaces chosen for specific purposes [1, 2]. Modularisation has great potential both technically and commercially and modular products can be found in many different areas. Examples include washing machines, measuring instruments, personal computers and vehicles.

Modularisation is often closely associated with the generation of product platforms and product families. In recent years it has gained ground because of what is known as mass customisation and the need for shorter development times. Intense international competition, demanding customers in a crowded market, exploding product variety and rapidly changing technologies require companies to be fast, responsive and highly productive when developing products with distinction and integrity.

Product development is normally expected to run concurrently with other operations. A modular approach facilitates this. Another important advantage of a modular system is that individual components can be developed without disturbing the composition of the range of products produced. It also allows distributed development and specialisation. Modularisation opens up many possibilities; however, it also places higher demands on the process and the product.

Most methods and principles used in the development of modular products were developed primarily for purely mechanical systems [1, 3]. However, the trend towards increased use of mechatronic solutions means that there is a need to integrate mechanics, electronics, computers and software in order to obtain new and better mechanical functions. It is thus important to develop relevant basic principles and product development theories and methods for modular mechatronic systems.

2 The background to this study

Traditional development methods assume that modules should be functionally self-contained and independent [1, 4]. While many products can successfully be modularised in accordance with this principle, this is not the case for some important classes of products, such as robots and vehicles. Attempts to use this principle when modularising such products have been limited and are unlikely to be successful. The reason for the difficulty is that the main parts of the product are not functionally independent. This is true of most types of robots and also of the new mechatronic concepts or solutions that are used in vehicles and that will become increasingly common in future vehicles.

Previous studies demonstrate the difficulty of using traditional principles and modularisation methods when developing modular mechatronic systems [2]. One concept studied was a prototype for a wheel-based mobile robot for difficult terrain designed using a modular approach (see figure 1). This robot was constructed using a few main modules that together constituted an articulated, steered all-wheel drive vehicle.

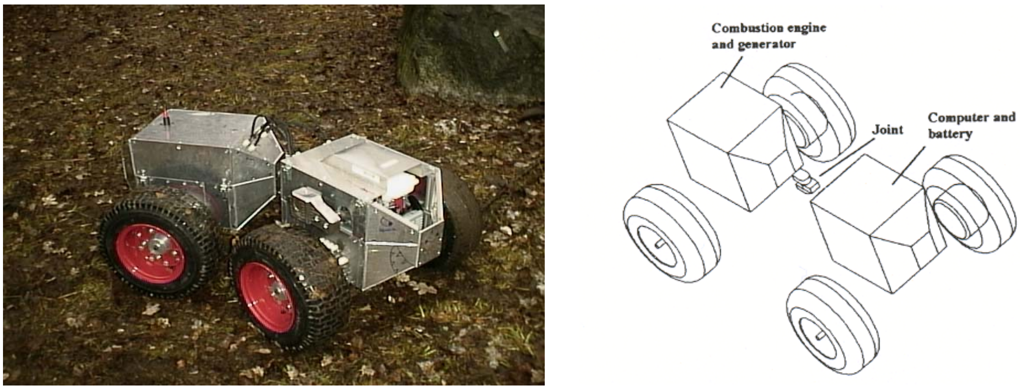

Figure 1. Wheel-based mobile robot

The vehicle is easy to steer in the ideal case with smooth ground and good friction. However, the steering becomes very difficult when the terrain is rough, particularly if the friction between the wheels and the ground is unpredictable and varying. The steering then involves controlling the motion or applied torque of each wheel-motor in order for the robot to produce the desired overall behaviour. Thus while the all-wheel drive robot prototype is mechanically very simple (few modules) it is difficult to control. The reduction in mechanical complexity obtained by modularisation necessitates increased complexity in the control of the vehicle [5].

A general trend in vehicle engineering is the development of x-by-wire systems. Such systems can also be regarded as modular mechatronic systems. The product structuring of a vehicle can be made much more flexible than has been possible before. High performance subsystems or modules are used, and these modules may often themselves be highly integrated mechatronic systems.

The control of x-by-wire systems involves both a decrease and an increase in complexity in comparison with traditional mechanical solutions. However, the problems posed are not completely new, since the x-by-wire concept has long been used in aeroplanes. Signal transmission through mechanical linkages or hydraulic transmissions has been replaced by data transmission through electrical communication buses. The timing properties and reliability of such communication have been subject to intensive research in the area often referred to as distributed real-time systems. In the embedded systems domain, the term *distributed system* corresponds to a *modular system* in the mechanical domain. Standardised interfaces have made it fairly easy to build systems from components and reconfigure systems, even on-line. However, there is a major difference between distributed systems and their mechanical counterparts. The joint mechanical/physical behaviour of an interconnected set of mechanical modules is intrinsic to the system. By contrast, in a distributed computer control system, the behaviour is largely defined by control algorithms that typically are not modular and that are usually implemented after the system *hardware* configuration has been defined.

The use of distributed systems is the rule rather than the exception in machines and mechanical systems. Automotive applications, aircraft, robots, construction equipment and production systems all use the concept.

A distributed mechatronic system comprises a set of communicating mechatronic units with more or less cooperative actions. A low degree of cooperation reduces the complexity of the system and thus simplifies design. However, in many cases the cooperative actions of the different parts are closely related to the function of the system. Our work has focused on this area, and particularly on the following research question:

Given that mechanical complexity can be significantly reduced by a modular approach, can the control complexity in a distributed mechatronic system be reduced by a completely modularised control solution, thus minimising the communication requirements?

3 Modularisation methods for mechanical systems

Erixon [1] developed a support method, modular function deployment (MFD), for the generation and evaluation of modules. He defines modularisation as decomposition of a product into building blocks (modules) with specified interfaces for company-specific reasons. *Module drivers* are important elements in the MFD method, which consists of the following steps: 1. Clarify customer requirements (QFD). 2. Select technical solutions (functional decomposition, Pugh selection matrix). 3. Generate concepts, (MIM, questionnaire). 4. Evaluate concepts, (interface matrix, evaluation chart, MEC). 5. Improve each module (DFX). The MFD method has mainly been used for restructuring and clustering already existing technical systems. The natural interfaces between the components are there from the beginning, and the question to be resolved is where the interfaces for the modules should be placed. The method therefore uses an interface matrix to study different options. Stake [6] has done further work on the conceptual development of modular products and has made important contributions with respect to module drivers.

Blackenfelt [7] studied the problem of managing complexity by product modularisation. He found that “the actual modularization is done in the embodiment phase where the technical solutions are grouped to modules by considering both the functional and the strategic aspects . “The detailing of the modular structure is done by optimising the degree of variety and by freezing the interfaces after the variety has been considered.” He used both a strategic and a functional design structure matrix (DSM) in his studies of different technical systems.

A method of design for variety described by Martin and Ishii [3] has many similarities with the MFD method. They examined external factors that may cause the requirements for designs to changes over time. They developed a general variety index (GVI) as an evaluation aid and a coupling index (CI), which indicates the strength of the coupling between the components in a product. The result of a CI analysis using the method proposed by Martin and Ishii is similar to that obtained used the extended DSM method presented by Blackenfelt [7].

4 Mechatronics and distributed control

Mechatronic products include mechanical, electronic and software elements. A basic mechatronic product comprises sensors that gather information, computers that process the information, and actuators that act on the information and control mechanical parts.

Designing mechatronic products involves at least the disciplines of mechanics, electronics, software and control engineering. Thus mechatronic products are not only complex in terms of the number of components or functions but also in terms of the integration of different engineering disciplines.

The idea behind mechatronics is to enhance the functionality of a system by adding sensors, actuators and computers, with control algorithms implemented in the software. The success of mechatronics is due to its ability to improve functionality and develop new functions that were previously either impossible or very difficult to achieve.

An impressive example of the successful use of mechatronics is a telemanipulator in which a remote slave is controlled by an operator through a master device that is mechanically decoupled from the slave. The ultimate design goal can here be formulated as "the telemanipulation controller should behave as a massless and infinitely stiff mechanical link carrying motion references from master to slave and providing force reflection from slave to master" [8]. In other words, a complex mechanical linkage has been completely replaced by a software solution using sensors, actuators and processing elements.

As already mentioned, distributed systems are the mechatronic equivalent of modular systems in the mechanical and hardware domains. A distributed mechatronic system is a set of communicating mechatronic units. A low degree of coordination and cooperation between the modules simplifies control of the system. An industrial Cartesian-type gantry robot with no, or at least low, dynamic coupling between the degrees of freedom could thus easily be controlled by a modular distributed system with one mechatronic unit per degree of freedom. On the other hand, a revolute-type of robot structure with intense dynamic coupling and non-linear effects is substantially more complex, and communication between the distributed units could become overly complex [9, 10].

Mechatronic distributed systems have evolved for very much the same reasons as modular mechanical systems. However, a distributed mechatronic system offers the additional advantages of redundancy, improved signal quality and reduced cabling. Unfortunately, the distributed concept also brings some difficulties in terms of partitioning of the overall control problem [11], communication bandwidth bottlenecks [12] and the testing and verification of functionality and correctness [13]. All these problems are rooted in the so far unavoidable couplings and interdependencies between the distributed modules [11].

5 The modular mechatronic systems project

The overall goal of the modular mechatronics systems project, for which this work is an initial study, is to develop general principles for clustering complex mechatronic systems with strongly dependent functional components into robust modular mechatronic systems with minimal intermodule communication. The focus will be on systems that cannot easily be clustered into functionally self-contained, independent blocks with current modularisation principles or methods.

The principles we will develop require that the complex system level control problem be solved by several cooperating module level controllers. The control of the modules should be such that the exchange of information between the modules is kept to a minimum.

It is hoped that these principles will counter the problems posed by the growth in complexity when larger, dynamically coupled systems are developed. It is envisaged that the total system will become substantially simpler at the expense of slightly more complex modules.

5.1 Study case

Even in this case, an attempt is made to observe the general principle that modules should be functionally self-contained and independent, mean that the modules are given a high degree of autonomy. The goal is to make them comparatively simple, even if they include closely integrated mechanical, electronic and software elements.

The system studied in this paper is a revolute mechanism structure comprising a five-link system. The system is built up of five identical link modules and one perception module. Each link module consists of an electrical motor (a stepper motor), a revolute joint with an angle sensor, and a link. A microcomputer and the electronics for the electrical motor are integrated within the link. The five links all move in the horizontal plane. The total system is complex with strong functional coupling between the modules. Figure 2a shows an exploded 3D drawing of one link module and figure 2b shows the physical prototype without the perception module. The angle of each joint is recorded and the front link module is equipped with a perception module, which may be an ultrasonic sensor, that retrieves information about an obstacle or the distance to a goal and its angle relative to the front link module. The perception module also contains a microcomputer to transform data into a convenient form for further transformations in the link module control computers.

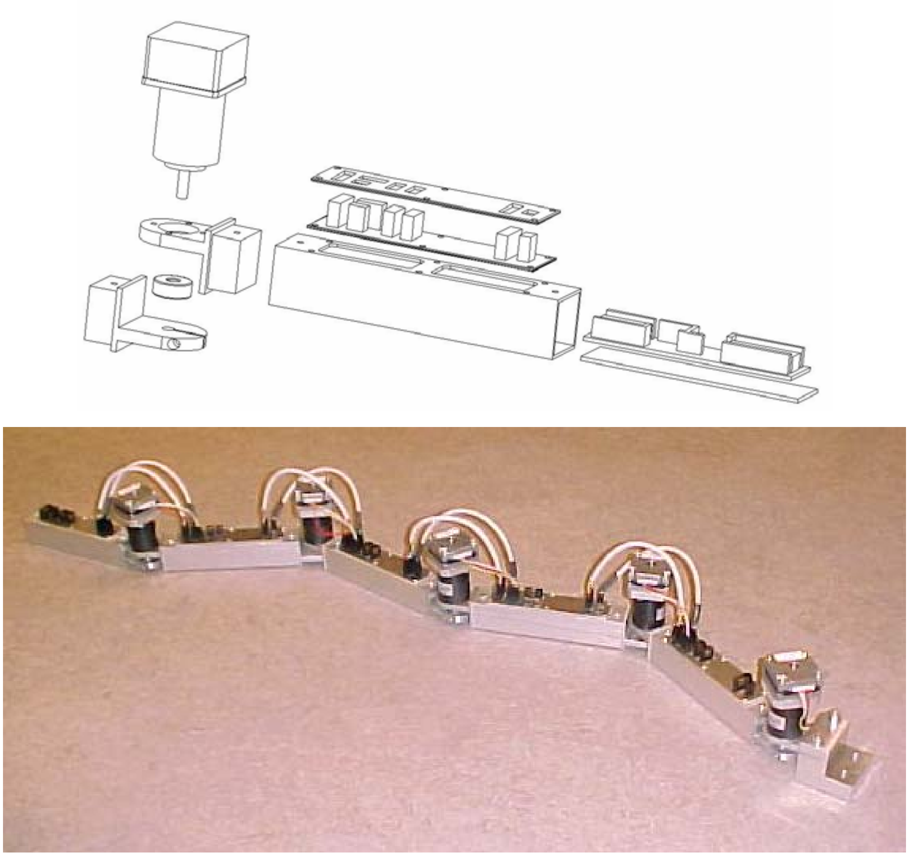

Figure 2. a) Link module (top) b) Five-link prototype system with revolute joints (bottom)

The control concept applied in this case can be referred to as behaviour-based control. The control actions are regarded as fairly simple reactions to external stimuli. The control algorithms themselves are seen as modules, each responding to specific sensor input with appropriate actuator output [14]. Given that we have adopted a behavior-based approach to control, the control is designed such that each module reacts to its neighboring module.

The perception module measures the distance and angle from the module to a target object and transforms these data into distances perpendicular to and along the link. If a target object appears in the working area, the task is to move the end point of the front link module towards it. To achieve this, the front link module first tries to solve the task by itself. If unable to do that, it requests help from other modules in the chain. If the link module connected to the front link module can only partly meet the request, it will in turn request help from the next module in the chain. This process continues until the final link module in the chain, that is, the link module that is connected to ground, is activated. Throughout this process, the front module is unaware of the number of modules attached and activated. All it can do is use the perception sensor to record the overall effect of the actions of the modules in the chain. This control concept is presented schematically in figure 3.

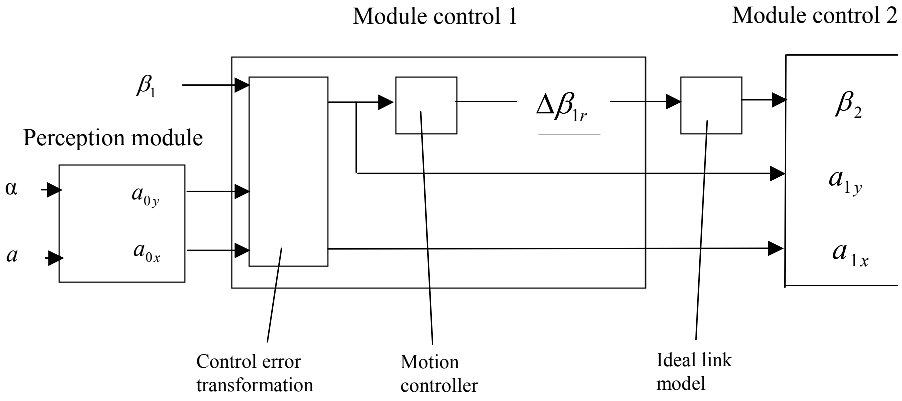

Figure 3. Control system

The front link module of the five-link revolute system is not regarded as a master module but as a module with perception capability. The perception derives from a sensor that can record the direction α and distance a to a goal. These sensor readings can be seen as the control error for the end point of the mechanism. The direction and distance control errors are transformed to the orthogonal distance errors $a_{0x} = k_0 \cdot a \cdot \cos\alpha$ and $a_{0y} = k_0 \cdot a \cdot \sin\alpha$ along the front link and perpendicular to the front link respectively. k_0 is a controller gain.

In the simulation, $k_0 = 0.2$ gave a smooth motion of the system. The orthogonal distance errors a_{0x} and a_{0y} are read by module controller number 1, which is the front module. Module controller number 1 will affect a_{0y}. The motion controller for link module 1 was designed as follows

$$\Delta\beta_{1r} = \arctan\left(a_{0y} / l_1\right) \tag{1}$$

where l_1 is the link length. The altered reference angle $\Delta\beta_{ir}$ is added to $\beta_{(i+1)}$. Hence, with this controller the front link module plans its own motion but also asks for assistance from other modules in the chain. We have found it beneficial to transform both distance errors a_{0x} and a_{0y} into new distance control errors for the second link. The transformed distance control errors for the second link module (i = 2), one perpendicular and one along the link, are

$$a_{2x} = k_1 \cdot \left(a_{1x} \cdot \cos\beta_2 - a_{1y} \cdot \sin\beta_2\right) \tag{2}$$

$$a_{2y} = k_1 \cdot \left(a_{1x} \cdot \sin\beta_2 + a_{1y} \cdot \cos\beta_2\right) \tag{3}$$

where β_2 is the angle between link 2 and 3. k_1 is a controller gain set = 0.5 in these simulations. Each link module has the same controller gain and the transformation can be formulated according to

$$a_{ix} = k_1 \cdot \left(a_{(i-1)1x} \cdot \cos\beta_i - a_{(i-1)y} \cdot \sin\beta_i\right) \tag{4}$$

$$a_{iy} = k_1 \cdot \left(a_{(i-1)x} \cdot \sin\beta_i + a_{(i-1)y} \cdot \cos\beta_i\right) \tag{5}$$

where i is the number of the link module. i = 1, 2, 3 ….

The control algorithms for all link modules are identical. For the particular experimental setup described here, the design parameter k_1 (controller gain) is tuned manually.

The concept was simulated using SimMechanics, a toolbox for Matlab, and can easily be integrated with Simulink. The SimMechanics model is shown in figure 4. A sequence from one simulation is shown in figure 5. The results of the simulations show that it is possible to design a modular mechatronic system even if the degrees of freedom are strongly dependent, as in this case with a five-link mechanism with revolute joints. In principle, the addition of any number of link modules to the system will have only a marginal effect on the complexity of the system.

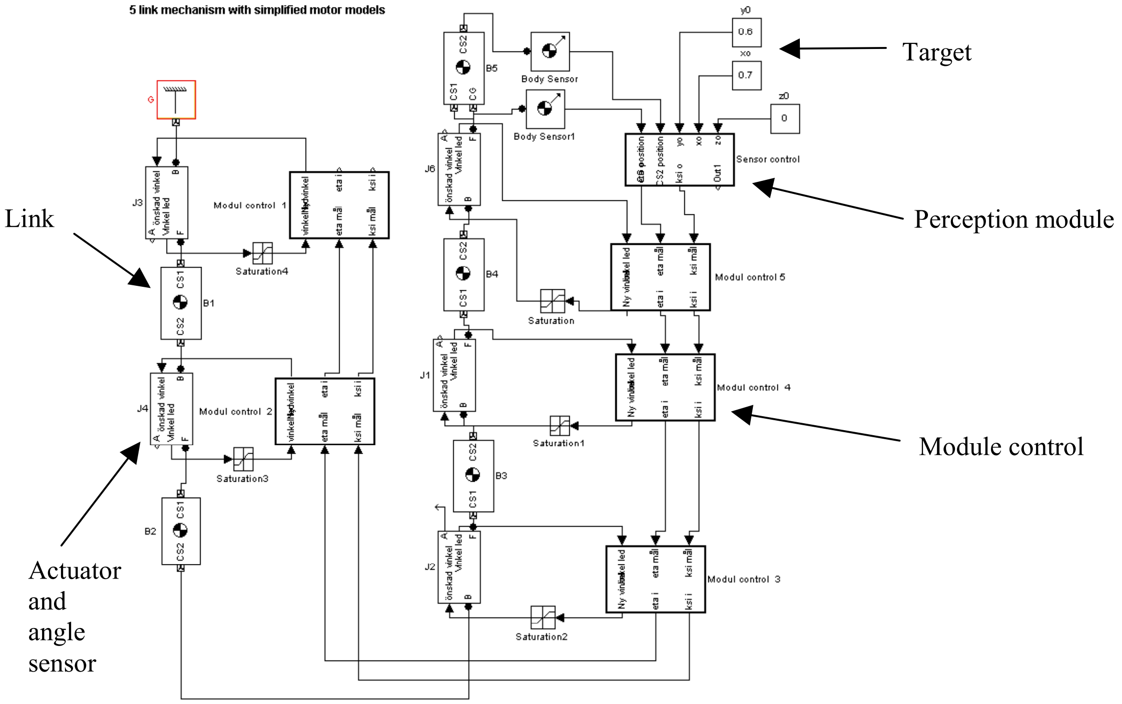

Figure 4. SimMechanics model of the five-link system

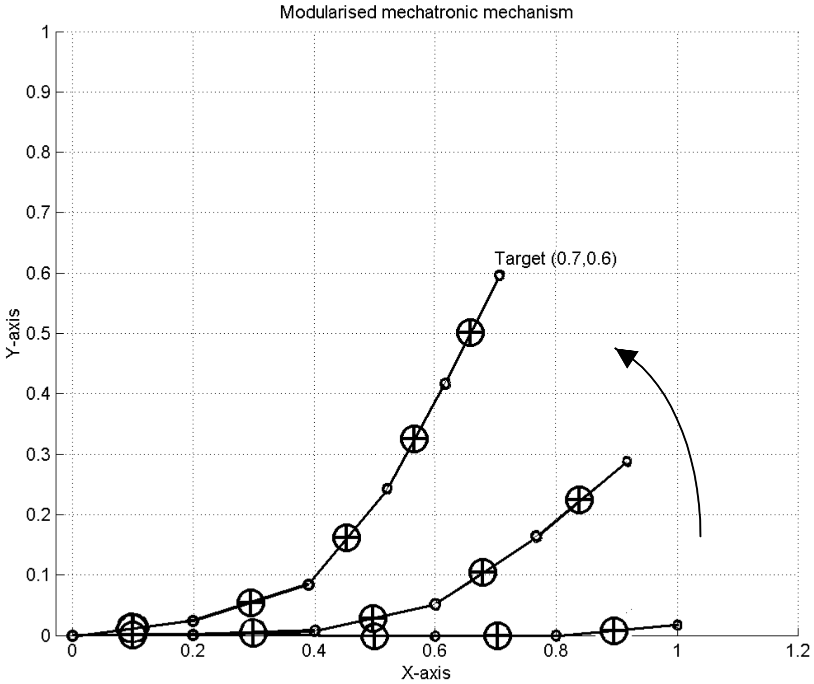

Figure 5. A motion sequence from the simulation with the SimMechanics model of the five-link system

6 Conclusions

Modularisation of products has great potential both technically and commercially. Most methods and principles used in the development of modular products were, however, developed for purely mechanical products. Consequently some problems may occur during product development, since mechatronic design solutions are used for many different types of products. Mechatronic solutions mean that new and better mechanical functions can be obtained by integration of mechanics, electronics, computers and software. The x-by-wire concept, which has long been used in aeroplanes, is an example of a modular mechatronic systems. In developing such systems, the need for integration between different fields of engineering poses a serious problem.

Previous studies of robots and vehicles clearly illustrate the difficulty of using traditional development principles and modularisation methods when developing modular mechatronic systems. The reduction in mechanical complexity achieved by modularisation results in increased complexity of the control system.

Timing properties and the reliability of communication between modules have also been studied in an area often referred to as distributed real-time systems. The term “distributed system” is often used and corresponds to a modular system in the mechanical domain. A distributed mechatronic system may thus be viewed as a set of communicating mechatronic units with more or less cooperative actions.

In the modular mechatronic systems project, for which this work is an initial study, the cooperative actions of the different parts are regarded as closely connected to the function of the system. The work reported here focuses on whether it is possible to significantly reduce the control complexity in a distributed mechatronic system when using a modularised control solution.

In order to study different control approaches, a modular mechatronic mechanism test bench was developed and studied. The test bench is a revolute type of mechanism structure, with five links. Each link module consists of an electrical motor (a stepper motor), a revolute joint with an angle sensor, and a link. A microcomputer and the electronics for the electrical motor are integrated within the link The five links all move in the horizontal plane. The system has strong functional coupling between the modules. The front link module is equipped with a perception module. The feasibility of a modular control concept for the five-link mechanism was studied.

The control concept studied can be referred to as behaviour-based control. The behaviour of the system with the control concept was modelled and simulated with the aid of the Matlab toolboxes SimMechanics and Simulink. The results of the simulations indicate that it is possible to modularise the control of a complex mechanism such as the five-link mechanism studied so that only simple communications between the different modules are necessary.

References

[1] Erixon, G., 1998, “Modular function deployment: A method for product modularization”, Doctoral thesis, Dept. of Manufacturing Systems, Royal Institute of Technology (KTH), Stockholm.

[2] Blackenfelt, M. 1999, “On the development of modular mechatronic products”. Dept. of Machine Design, Royal Institute of Technology (KTH), Stockholm, TRITA-MMK 1999:2.

[3] Martin, M. V. and Ishii, K., 2002. “Design for variety: Developing standardized and modularized product platform architectures”. Research in Engineering Design 13, 2002, 213–235.

[4] Erixon, G., Erlandsson, A., von Yxkull, A. and Östergren, B. M., 1994, “Modularize the product” (in Swedish), Sveriges Verkstadsindustrier, ISBN 91-7548-321-1.

[5] Blackenfelt M. and Andersson S., 1998 “Wheel-based mobile robot prototype designed with a modular approach: A case study”. *NordDesign '98*, 26–28 Aug., 1998 in Stockholm.

[6] Stake, R. 2000. “On conceptual development of modular products”. Doctoral thesis. Dept. of Production Engineering. Royal Institute of Technology, KTH, Stockholm.

[7] Blackenfelt, M. 2001. “Managing complexity by product modularization”. Doctoral thesis. Dept. of Machine Design. Royal Institute of Technology, KTH, Stockholm.

[8] Flemmer, H., Eriksson, B., and Wikander, J., “Control design and stability analysis of a surgical teleoperator”, Journal of Mechatronics: The Science of Intelligent Machines, Vol. 9, pp. 843–866, 1999.

[9] Kuroki, M., Wikander, J., Yagawa, G. and Lind, H. “A fast adaptive control scheme on a network of feed forward control”, in Proc. of 11th Int. Conf. on Structural Mechanics in Reactor Technology, August 19–23, 1991, Tokyo, Japan.

[10] Lind, H., Wikander, J. and Törngren, M. “On decentralization of control functions for distributed real-time motion control”. In Proc of the Int. Symposium on Robotics and Manufacturing, August 1994, ASME Press, ISBN 0-7918-0044X.

[11] Törngren, M. and Wikander, J., 1996. “A decentralization methodology for real-time control applications”. Journal of Control Engineering Practice, Vol. 4, No. 2, February 1996.

[12] Törngren, M. “Fundamentals of implementing real-time control applications in distributed computer systems”. Journal of Real-Time Systems, Vol. 14, No. 3, 1996.

[13] Thane, H. “Monitoring, testing and debugging of distributed real-time systems”. Doctoral thesis, Dept of Machine Design, Royal Institute of Technology (KTH), Stockholm, 2000.

[14] Arkin, R. C. “Behaviour Based Robotics”, MIT Press, 1998, ISBN 0-262-01165-4.

For further contact:

Machine Design, Royal Institute of Technology (KTH),
Stockholm, Sweden.
bjornm@md.kth.se, soren@md.kth.se, jan@md.kth.se