

# Mechatronic Integration Modeling

Dr. Kevin Craig, *Member, IEEE*, Dr. Mary DeVito, Mike Mattice, Chris LaVigna, Dr. Carole Teolis

**Abstract** - The major challenge facing the Mechatronic community in the development of the next generation of smart systems lies in the integration complexity of an enormous number of requirements, physical structures and the software components inherent in these systems. Meeting this challenge has dictated the need for a fundamental change in the mechatronic design process in order to ensure affordability, reliability, maintainability, adaptability and a built-in growth potential for these large scale Mechatronic systems.

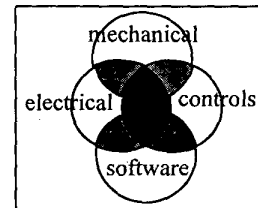
The efforts discussed in this paper were aimed at developing a design process for large-scale systems which would make multidiscipline integration an integral part of the design process.

## I. INTRODUCTION

"Mechatronics is a technical interdisciplinary, built upon the basis of classical mechanical, electrical, controls, electronic engineering, computer science and system engineering. "The field of mechatronics emerged because many systems could hardly be improved with established methods and reasonable effort." "The main task of a mechatronic expert is not to solve specific parts of a task, but to plan the whole product design appropriately."

As the size and complexity of Mechatronic systems increases the system design problem goes beyond the physical structures and software components. The major challenge facing the Mechatronic community in the development of the next generation of smart systems lies in the integration complexity of an enormous number of requirements, physical structures and the software components inherent in these systems. Such systems, to be effective, must integrate seamlessly into the infrastructure. Interface complexity emerges as a new kind of problem. The inherent integration complexity of these large scale mechatronic systems impacts not only system development cost and risk but also total system life cycle maintenance and support cost. Meeting this challenge has dictated the need for a fundamental change in the mechatronic design process in order to ensure affordability, reliability, maintainability, adaptability and a built-in growth potential for large scale Mechatronic systems. The mechatronic design process now needs to focus on complex integration issues.

The efforts discussed in this paper were aimed at developing a design process for large-scale systems which would make multidiscipline integration an integral part of the design process.



*Mechatronic Integration*

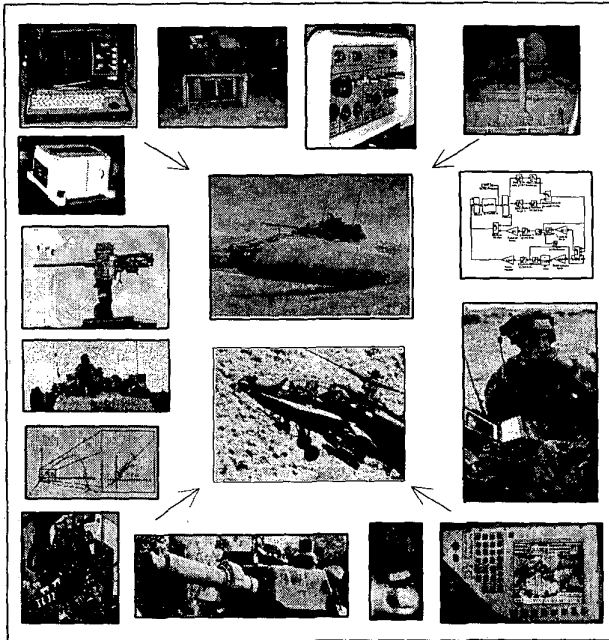
## II. INTEGRATION ISSUES

The design and integration of large scale systems needs to start at a higher level than we are used to. The IEEE/ASME journal recently had 3 special issues on Mechatronics. The 3rd volume dealt with Mechatronic integration. Below are some comments from the journal articles concerning integration:

- Sensor Technology Integration in an Intelligent Machine for Herring Roe Grading, E. A. Croft, C. W. de Silva, and S. Kurnianto - "intelligent" decision-making system, with multiple levels of abstraction"
- Mechatronics-An Industrial Perspective, N. Kyura and H. Oho - the importance of "intimate" integration in mechatronics product designs is raised. Several issues pertaining to attaining this ideal integration mechanical, electrical controls, and system engineering in mechatronics product designs are cited. To further advance the current mechatronics products, areas of improvement are needed.
- Modeling and Design Methodology for Mechatronic Systems, R. Isermann - The information processing can be organized in multi-levels, ranging from low-level control through supervision to general process management.
- Modeling, Design, and Control Integration: A Necessary Step in Mechatronics, K. Youcef-Toumi - Future products require more sophistication and flexibility from both the hardware and software points of view. In particular, these products involve several energy domains, which is one reason for the new requirements for system design and integration. These are important aspects not only for industrial research and development personnel but also for academicians.
- Sensor Technologies and Microsensor Issues for Mechatronics Systems, R.C. Luo - Intelligence and flexibility are essential in a mechatronic product. To achieve the primary function of an integrated system, it is essential

that the functional interaction and spatial integration between mechanical, electronic, control, and information technologies be accomplished in a synergistic way.

The government for example, deals with large-scale, multidiscipline, intelligent, complex systems. Due to its multidiscipline nature, large-scale Mechatronics systems involve multiple in-house departments and vendors each with a different specialty all at different physical locations. The military considers large-scale integration issues to be the most difficult/expensive issue associated with their systems.



*Example Military Components and Systems*

This research effort was designed around the concept of developing an environment which would provide for both the incremental development of design integration models and analysis of these integration models. Mechatronic design integration models represent, in their most basic form, the system's building blocks, its' interconnectivity, and its' information transfer. This Mechatronic Integration Modeling environment provides a framework and methodology for designing the integration of Mechatronic systems. The integration description language portion of the environment will provide a means to formally describe systems which are required to: integrate both hardware and software components. The integration metric formulas will provide a means to evaluate the integration model under development.

Currently, there exists several engineering tool environments for system development, some of which are: MatrixX, Matlab, AutoCad, etc. There also exists several computer science tool environments for system development some of which are: CADRE, ObjecTime, C++, Rational Rose, etc. All of the tools mentioned above

are considered excellent commercial tools and dramatically increase productivity in their particular domain. As a result Mechatronic system integration design cannot be designed using any one of the commercial systems listed above. As the complexity of Mechatronic systems increases the complexity of the integration design to build these systems also increases. Therefore new Mechatronic tool environments must be developed in order to capture the various domain (mechanical engineering, software engineering, controls engineering, electrical engineering, etc.) integration design information that will be necessary in order to build these multidiscipline systems. The function of this Mechatronic tool environment is *not* to replace such tools as MatrixX, Matlab, CARDE, etc. This tool environment would provide for a means to develop and analyze design integration models for mechatronic systems and once this has been completed then such tools as Matlab and ObjecTime would be utilized as the next step in the design process. This research effort was aimed at supporting integration of mechatronic systems. It provides a mechanism to make multidiscipline integration a part of the initial design process. This research effort concentrates on the integration of the parts.

### III. TOOL AND EXPERIMENT OVERVIEW

The authors have developed a Mechatronic Design Integration Tool to support the design integration process and analysis formulas to support quantitative design integration analysis.

The Mechatronic Design Integration Tool is based on the development of a Description Language (DL). DLs are formal, yet concise, understandable and flexible languages for describing objects. Description Languages have been developed for such items as VHDL (VHSIC Hardware Description Language) which is utilized to capture complex digital electronic circuits, it is considered a standard in the electronic design community. Control tools such as Matlab, Simulink, SystemBuild, and MatrixX are also partially built upon DLs. The tool developed by the author through formal notations can be utilized to design mechatronic system integration models which become an integral part of the initial design process. The tool provides the means to formally describe design integration models which will make integration a part of the design process. Mechatronic design integration models represent, in their most basic form, the system's building blocks, their interconnectivity, and information transfer. Design integration issues include such items as organization and global structures, protocols for communication, synchronization, data access, physical connections, real-time requirements, messages, connectivity, requirements, constraints, addresses, interrupts, assignment of functionality to design elements, composition of design elements, services, signals, bandwidth, update rates, resolution, accuracy, formats, power requirements, connectors, pin-outs, etc.

The integration metric formulas provide a means to evaluate the integration model under development. It

provides a quantitative measurement by which design integration analysis can be computed to assist in the evaluation of the complexity of system design integration models.

Two research experiments were performed to evaluate the Mechatronic Design Integration Tool and the design integration analysis formulas. These research efforts were funded by the US Army. The objective of the first experiment was to develop a testbed to determine the feasibility of utilizing robots to depalletize, fuse, and load shells for howitzer systems. This mechatronic testbed would be responsible for such objectives as: operating in dynamic environment, locating objects i.e. shells, pallets, fuses, gun tubes, etc. handling fuses, fusing shells, removing pallet lids, handling shells, loading breaches, closing breaches, pulling firing mechanisms, opening breaches, etc. The mechatronic testbed would need to configure and integrate: software (planning, databases, tracking, monitoring, filters, etc.), sensors (camera, force/torque, range, tactical, infra-red, etc.) and actuators. The components were to be designed by both contractors and in-house efforts. The authors developed a Mechatronic Design Integration Model, using the Mechatronic Design Integration Tool developed under this research effort to support the intelligent integration of this testbed. Under this experiment the Mechatronic Design Integration Tool was extended, where necessary, in order to better formalize the required description sets.

The scope of the second experiment was to develop an improved 30mm weapon system testbed. The Army is interested in this problem because the current 30mm weapon system is: (1) inaccurate, (2) requires expensive O&S (operating and support) costs, and (3) designed around hydraulic fluids. Some of the problems associated with hydraulic drive systems are: safety and environmental hazard, the gearbox produces backlash, valves and seals cause friction problems, filter replacement, characteristics of the system change over time, dynamic compressibility of the fluid, leakage, required maintenance in a sterile environment "clean room", hydraulics work best within certain environment conditions i.e. temperature, etc. The components were to be designed by multiple contractors. In order to improve the weapon system mechatronic testbed we needed to integrate sensors (piezo, resolver, accelerometers, joystick, etc.), actuators (electric drive), and software. The authors again developed a Mechatronic Design Integration Model, using the Mechatronic Design Integration Tool to support the intelligent configuration and integration of the improved 30mm weapon system. The new system was to provide: (1) improved accuracy, (2) reduced O&S costs, and (3) replace the hydraulic drive with an electric drive. Some of the advantages of a direct electric drive is: it's gearless, encoder is embedded in the drive, requires almost no maintenance, improve accuracy, no environmental or safety hazard, no fluid to lead, no valves or seals which need to be replaced, etc. The accuracy was also to be improved by integrating piezo stacks near the muzzle (the tip of the gun). Piezo Stacks are smart materials

which can act as both sensors and actuators i.e. sense the vibration/bend of the barrel and use the actuator portion to bend the gun barrel to keep it pointing straight.

#### IV. A CLOSER LOOK

The Mechatronic Design Integration Tool is a frame-oriented language. It provides a number of definition types, each of which is like a form with specific description sets, specific fields, and specific types of values that are allowed in each field. A Mechatronic Design Integration Model description is a collection of these frames, with some number of fields filled in with values. The authors will often refer to these as "objects". The definition types in the Mechatronic Design Integration Tool are organized into sublanguages. Each sublanguage is viewed as a complete description of the system of interest, but only at one level of description or abstraction. The Mechatronic Design Integration Tool currently has sublanguages for domain modeling, for systems of abstract components, and for designs consisting of implemented components. The sublanguages and the definitions that belong to each are listed below.

<u>Structural Model</u>	<u>Detailed Struct. Model</u>	<u>Domain Model</u>
Element	Component	Domain-entity
Structural Port	Implementation Port	Domain-relation
Message	Operation	Domain-op
Connection	Communication	Domain-con
Message Protocol	Communication Type	
Information	Data	Domain-attrib
Constraint	Constraint	Constraint

The Mechatronic Design Integration Tool which the authors developed consists of three hierarchical levels: (a) domain modeling, (b) structural modeling, and (c) detailed structural modeling. More specifically the schema consists of object oriented constructs. The description sets associated with the objects above are not discussed in detail in this paper, due to the page limitation, for a detailed description refer to pointers in the reference section of this paper. We will however give a very brief description of some of the higher level objects, in order for the reader to get an idea of how the objects interconnect.

The domain modeling constructs consists of such objects as: entity, attribute, and relation. The entity represents a thing, or a fundamental concept, in the domain of a system. The attribute refers to some aspect of a domain entity, it provides additional information for an entity, whether quantitative or qualitative. The relation object provides information concerning relationships between entities. The structural modeling construct consists of such objects as: element, port, connection, connection type, message, information, capability, and constraints. The element is the basic entity within a system it describes structural decomposition. Ports describe interfaces within a hierarchical system, hardware interface configuration

and/or software configuration. Connections define the topology of the system by defining which elements communicate with which others and the specific information exchanged. Connection type defines a class of connections to be used in the system. Messages are the system's unit of communication. Each message contains specific kinds of information, and is transmitted between specific elements. Information definition represent the data and data type content of a message. The detailed structural modeling portion of the system layout consists of its own unique set of objects some of which are: implementation port, component, operation, communication, communication type, and data definition. The implementation-port specifies an interface to services. The operation object is a formal description of a function, as realized in some component. The communication object defines the network of the design by defining which operations communicate with others. The communication type defines a class of communications to be used in the design. Finally, the data definition describes the way information is represented in the design implementation.

Distinguishing features of the "Mechatronic Design Integration Tool":

- A description language that allows for the integration of interdisciplinary components.
- Tool models have minimal complex syntax, and are as easy to read as possible.
- Models can be expressed in ASCII files of simple and open format thus facilitating publishing, filtering, and reading them into tools.
- Tool is designed as an open language. It is easy to add new objects and/or fields of objects to cover domain-specific information.
- Tool encompasses several views or layers of system description. These layers are interlinked in very specific ways. It should be clear what is derived from what, and what objects are pertinent in checking that derivation.
- Places more emphasis on the description of communication and on non-functional attributes.

The Mechatronic Design Integration Tool can be utilized to assist in the process of working out a new system or enhance an existing system. As a general system representation language, the authors have tried to adopt ideas from the CASE (Computer-Aided Software Engineering) community and from the design communities. Rumbaugh's OMT (Object-oriented Modeling Tool) is a source of representative of this world. The Mechatronic Design Integration Tool is used to develop integration models for the purpose of facilitating integration, maintenance, upgrades, and improve probability of component reuse.

Integration Analysis: When trying to define System Integration Complexity ( $C_i$ ) we first break the system down into subsystems. Each of the subsystems can be analyzed separately to determine the complexity of the subsystem

and then each of the subsystem complexities can be added to define the complexity of the entire hardware/software interface system. In the following analysis we break the interface into hardware and software subsystems.

To define the complexity of each subsystem we think in terms of a tree structure. The 'trunk' of the tree represents the entire subsystem. Branches, which represent the first level breakdown of the subsystem, come from the trunk. Each of these branches may also break into branches, and so on until the subsystem has been broken down to the most basic element of the system, the leaves. The complexity of the most basic element can easily be defined. Then the subsystem complexity can be computed by summing the complexity of each of the leaves of the tree structure. Mathematically this definition of complexity in terms of a weighted 'leaf count' can be represented by nested summations. System Integration Complexity ( $C_i$ ).

$$C = \left( \sum_{i=1}^n \left( \sum_{j=1}^n \sum_{k=1}^{n_j} \sum_{l=1}^{n_{jk}} \sum_{m=1}^{n_{jkl}} \alpha_{ijklm} \right) \right) + \sum_{i=1}^m \left( \sum_{j=1}^m \beta_j \right)$$

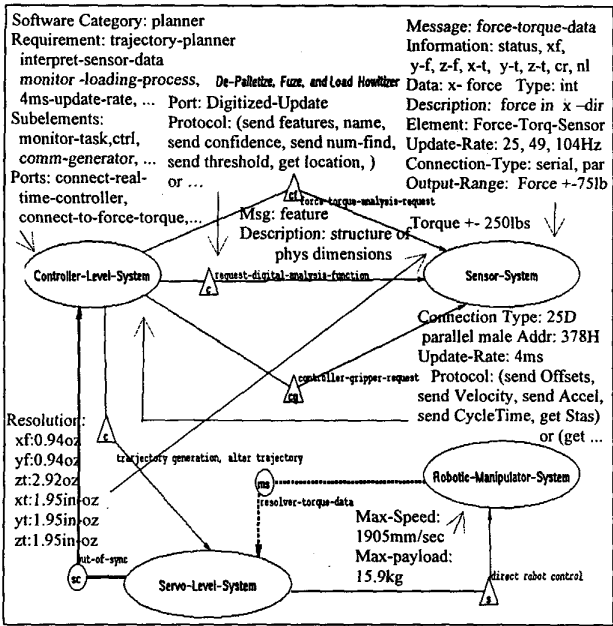
$$\text{Interface complexity for a single software component} = \sum_{m=1}^{n_{ijkl}} \alpha_{ijklm}$$

$$\text{Interface complexity for a single hardware component} = \sum_{j=1}^{m_i} \beta_j$$

## V. EXPERIMENTS

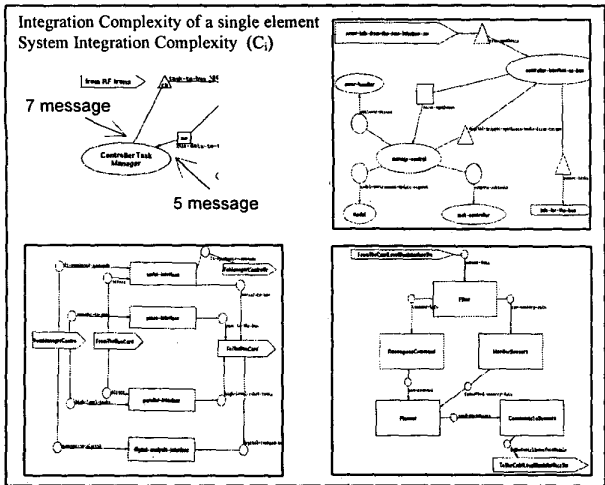
As we stated earlier, the first experiment performed as a part of this research effort was to design, integrate, and build a robotic testbed for material ammunition handling utilizing a 7 DOF (Degree Of Freedom) robot. The transition from the initial configuration to the final configuration was supported by; (1) the mechatronic integration model that was developed using the "Mechatronic Design Integration Tool" and (2) redesign indicators were supported by the integration analysis formulas, shown above.

The following examples are a very small portion of the integration model that we developed using the graphical user interface provided with the Mechatronic Design Integration Tool. The diagrams below are only a small indication of the amount of information that is available in the integration model.



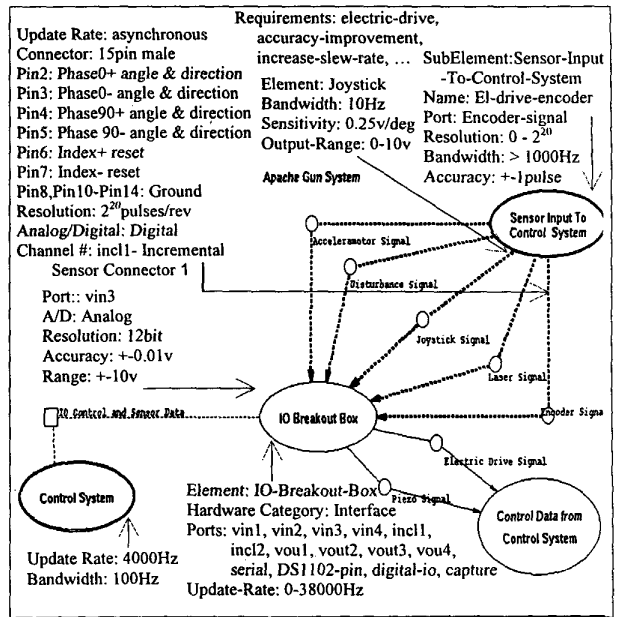
Small Example of the Robotic Integration Model

Below is an example of a component that we redesigned as a result of integration modeling analysis computations.

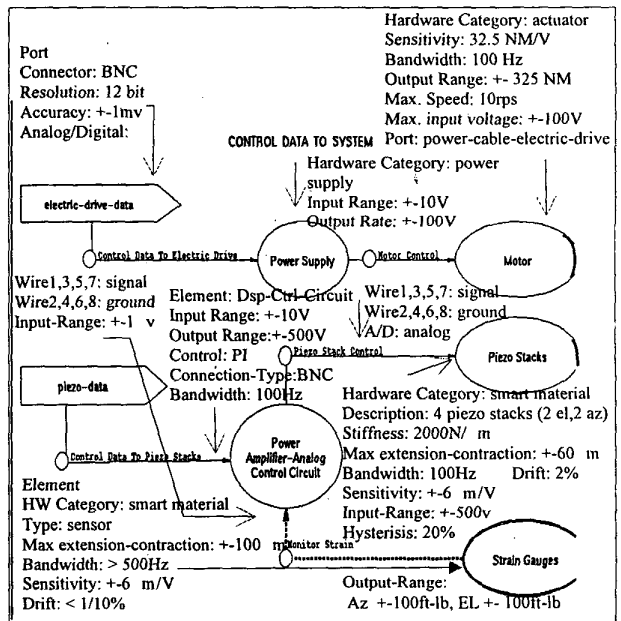


Example of Integration Redesign

The second experiment, as we discussed earlier, was the 30mm apache weapon. The purpose of this experiment was to develop a Mechatronic Design Integration Model, for integrating an improved 30mm weapon system program, using the Mechatronic Design Integration Tool developed under this research effort and enhanced in the first experiment. Below are two examples of a small portion of the integration model that we designed. The first example is of a higher level component, the example that follows is subcomponents of one of the components (Control Data from Control System) in the first example.



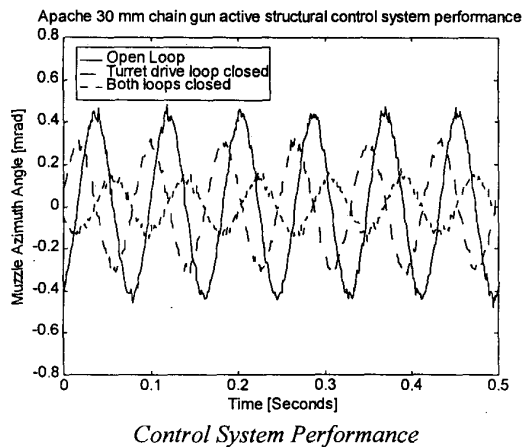
Small Example of the Weapon System Integration Model



Integration Model - Control Data from Control System

Below is a plot which indicates the ability of the new 30mm weapon system to reduce muzzle motion during firing. Open loop indicates the muzzle motion during firing when we are not using the electric drive or the piezo stacks to try to control it. The turret drive loop closed indicates the reduction in muzzle motion when we use just the electric drives to try to reduce the muzzle motion. Both loops closed indicates the reduction in muzzle motion when we use both the electric drives and the piezo stacks to try and reduce the muzzle motion. The result is that

muzzle motion is reduced/decreased by approximately 70%. This translates to having the ability to fire twice the distance with 50% fewer rounds with the same hit probability ( $P_h$ ). The idea is to allow military pilots to stay further away from their enemy i.e. out of the enemy's lethality range and also conserve projectiles/bullets. In effect increase the probability of survivability of pilots during battle.



## VI. SUMMARY

In conclusion, this effort was motivated by the need to make integration and redesign an integral part of the mechatronic design process. The research efforts resulted in two main contributions: (1) development of a Description Language entitled "Mechatronic Design Integration Tool" which are utilized to develop integration design models and (2) development of metrics for model redesign integration analysis. These efforts can be utilized to standardize and analyze the development of integration models for mechatronic systems.

## VII. REFERENCES

- Aushlander, D., "What is Mechatronics?", IEEE/ASME Transactions on Mechatronics, Vol. 1, No.1, March, pp.5-9.
- Booch, Grady, Object - Oriented Design with Applications, Benjamin/Cummings, 1991.
- Coad, Peter, Yourdon, Edward, Object - Oriented Analysis, Prentice-Hall, 1991.
- Gajski, Daniel; Specification and Design of Embedded Systems, Prentice Hall Inc., 1994.
- Gomaa, H., Software Design Methods for Concurrent And Real-Time Systems, Reading, MA, Addison Wesley, 1993.

## Mechatronic Web Sites:

- <http://dfs.iis.utokyo.ac.jp/IEEE.ASME/Mechatronics>  
<http://www.cage.curtin.edu.au/mechanical/staff/rogers/grp-gprj.htm>  
<http://www.sloan.mit.edu/HomePages/ckemerer/kemerer.html>  
[http://cims5.marc.gatech.edu/mechatronic\\_definition.html](http://cims5.marc.gatech.edu/mechatronic_definition.html)  
<http://hart.me.berkeley.edu/~me290p/mechatronics.html>  
<http://rcs.ee.washington.edu/CR/MECH/>  
<http://mecha.re.pusan.ac.kr/>  
<http://www.cyber.reading.ac.uk/research/>  
<http://biosys2.me.berkeley.edu/>

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorenzen, W., Object-Oriented Modeling and Design, Prentice-Hall, Inc. Englewood Cliffs, NJ 07632, 1991.

IEEE/ASME Transactions on Mechatronics Volume 1,2, 3 (March 1996)

"Sensor Technology Integration in an Intelligent Machine for Herring Roe Grading", E. A. Croft, C. W. de Silva, and S. Kurnianto

"Ten Years of Mechatronic Research and Industrial Applications in Finland", V. Salminen

"Mechatronics-An Industrial Perspective", N. Kyura and H. Oho, Vol1, No. 1, pp 10-15

"Modeling and Design Methodology for Mechatronic Systems" R. Isermann

"Modeling, Design, and Control Integration: A Necessary Step in Mechatronics" K. Youcef-Toumi, Vol1, No.1, March, pp 29-38.

"Sensor Technologies and Microsensor Issues for Mechatronics Systems", R.C. Luo

"Intelligent Control for Human-Machine Systems", M. Buss and H. Hashimoto

Warner Electric, "Instructions for Superior Electric SLO-SYN Model MX-2000 Programmable Multi-Axis Motion Controller", 1994.