A Systems Approach for Modelling Mechatronics Systems

Bassam A. Hussein
Production and Quality Engineering Department, The Norwegian University of Science and Technology, NTNU, Trondheim, Norway
Received on December 3, 1997

ABSTRACT

This paper presents a unified approach based on utilizing multidimensional arrays in order to model the physical and logical properties of mechatronics systems. A mechatronics system model consists of two interacting submodels. A submodel that describes aspects related to energy flow in the physical system, and another submodel that describes aspects related to information flow in the control system. The multidimensional array based approach of modelling provides us with the possibility to use one terminology and the same formalism for modelling both subsystems. The consequence of using the same formalism is that simulation of the mechatronics system can be performed using only *one* simulation environment.

Keywords: Mechatronics, System, Modelling

1. INTRODUCTION

Mechatronics system is defined as the synergetic integration of mechanical engineering with electronics, and intelligent computer control in the design and manufacturing of industrial products and processes [5]. The components of mechatronics systems must be designed concurrently, that is, the constraints imposed on the system by each discipline must be considered at the very early stages. Therefore, proper system design will depend heavily on the use of modelling and simulation throughout the design and prototyping stages.

The integration within a mechatronics system is performed through the combination of the hardware components resulting in a *physical system* and through the integration of the information processing system resulting in an *intelligent control system* [7]. The mechatronics system then, is the result of applying computer based control systems to physical systems. The control system is designed to execute commands in real time in order to select, enhance, and supervise the behavior of the physical system. The only possible way to guarantee that these control functions will keep the behavior of the whole system within certain boundaries before we actually build it, is to create a model of the real system that takes into account all the imposed constraints by both the hardware and software components. This implies that a model of the real system must be powerful enough to capture all the properties of mechatronics system. That includes; the dynamic, static, discrete event, logic, as well as cost related properties of the real system, a task we believe, defies any fragmented approach of modelling.

In this paper we present a unified approach for modelling mechatronics systems. This unified approach utilizes geometric objects or multidimensional arrays to formulate models of mechatronics systems. The multidimensional array based approach of modelling provides us with the possibility to use the same formalism for a large variety of systems [2,3,4,9]. The consequence of using the same formalism is that simulation of mechatronics systems can be performed using only *one* simulation environment.

2. MODEL STRUCTURE

Intuitively speaking, a model that describes the dynamic behavior of a given system can not be used to investigate the static behavior of the very same system.

Therefore, in order to capture all aspects, we need a variety of models, each one of them encapsulates some aspects of the real system.

We will consider the mechatronics system model as *a set of connected submodels*, each submodel corresponds to some realizable aspects. In this regard, the term connected was used to emphasize the dependency between the variables in these submodels.

Throughout the process of modelling, we shall distinguish between the following concepts, see Figure 1.

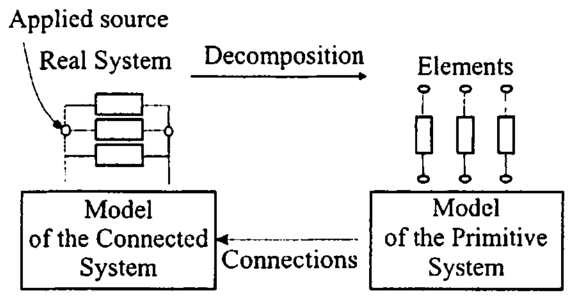

Figure 1. Process of Modelling

Decomposition: in order to handle the complexity of mechatronics systems, they should be decomposed into subsystems. This decomposition is carried out on a multilevel fashion until we reach the basic elements that constitute the total system.

The primitive system model: is a description of the system in the disconnected state. It expresses the relation between the variables in the individual elements when the bonds between these elements are removed. By this model we isolate a specific behavior; static, dynamic, etc., in each element. A pair of local variables defines the behavior of a given element locally.

The connected system model: is a description of the same system after taking the internal constraints into account. The internal constraints within the system are given by the way the local variables are connected or related directly as well as indirectly by the variables of the connected system. The connected system model resembles the actual structure of the real system.

The applied sources are generated due to interaction between the system and its environment. They could be seen as the external constraints imposed on the system or even inherent constraints in the form of stored energy in system elements.

3. APPLICATION EXAMPLE

Consider, the manufacturing system shown in Figure 2. The system consists of a boring spindle powered by a direct current motor. The feed forward motion of the boring spindle is carried out by means of a hydraulic linear actuator. The hydraulic actuator is powered by a constant pressure hydraulic pump. The volumetric flow in the hydraulic circuit is controlled by a servo valve [8].

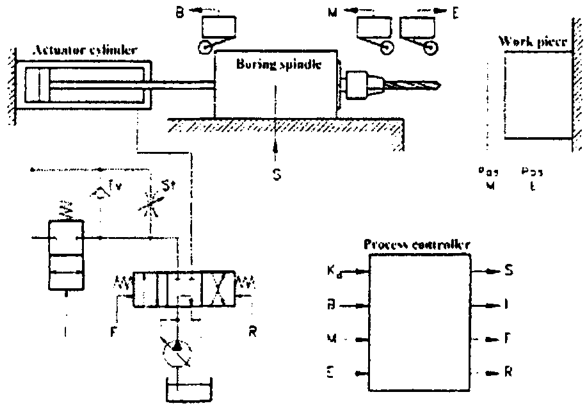

Figure 2. A Manufacturing System

The above manufacturing system has the following specifications:

The positions of boring spindle are sensed by three micro breakers. Breaker (*B*) which indicates that the boring spindle is at the rear position. At the rear position the rapid phase valve (*I*) will be switched on in order to allow a rapid forward motion (*F*) and the signal (*S*) will switch on the spindle motor. Breaker (*M*) indicates that the boring spindle has reached the feeding position. At this position the rapid phase valve will be switched off in order to start a controlled feed forward motion. This motion is regulated by the servo valve (*St*). Breaker (*E*) which indicates that the boring spindle has reached its final position, at this position and the backward motion (*R*) will begin, simultaneously the rapid phase valve (*I*) will be switched on in order to allow a rapid backward motion. It is also specified that the rotating speed of the spindle motor should be kept at 3000 rpm. during boring the work piece and the feed forward speed must be kept at 2cm/sec under all loading conditions.

Our objective is to set up a complete model of the given system using multidimensional arrays and to carry out necessary experiments on the model to verify that specifications are satisfied.

3.1 Physical System Modelling

When modelling physical systems, we are concerned with modelling the evolution of the physical variables that lives within this system. The decomposition of the physical system is shown in Figure 3.

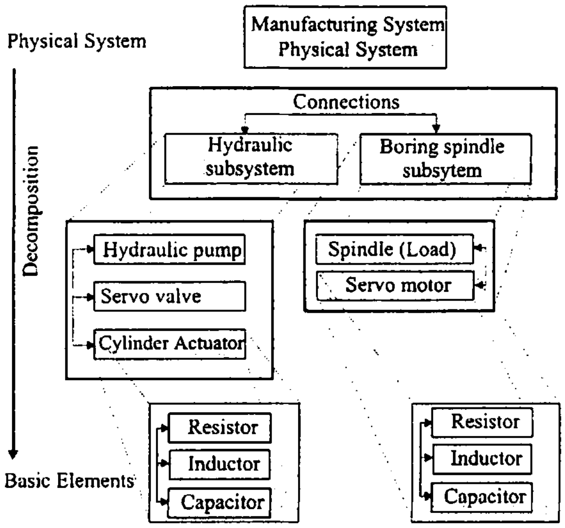

Figure 3. Decomposition of the Physical System

The groups of basic physical elements are classified into three categories:

Generalized resistor: examples of this category are; electric resistor, mechanical damper, and hydraulic resistor.

Generalized capacitor: examples of this category are; electric capacitor, mechanical spring, and hydraulic reservoir.

Generalized inductor: examples of this category are; electric inductor, mechanical mass, and hydraulic inductor.

Breaking down the physical system into subsystems and further into basic elements will provide us with a sharp insight about the evolution of the physical quantities within each subsystem, yielding to better understanding of the modes and the states that each subsystem would attain. The advantages of having such insight will become visible during the design phase of a local control system.

Modelling can be considered as the opposite procedure of decomposition. The difference is that, in decomposition, we divide the system into independent physical entities, while in modelling we reconnect the models of these physical entities. Therefore, modelling can be seen as the procedure of connection.

In modelling, we start at the bottom level of this hierarchy and move upwards. At each level, we propagate from a primitive system model to a connected system model. In the succeeding level, the primitive system model would then be established by aggregating the connected system models from the former level as shown in Figure 4.

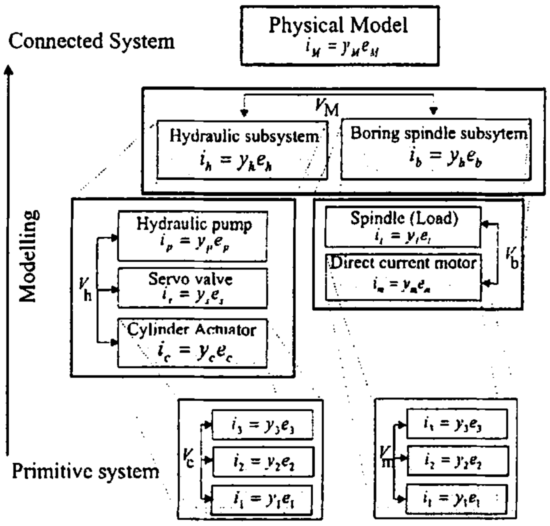

Figure 4. Modelling of the Physical System

At the bottom level of each subsystem, the primitive system model will be established by utilizing the governing equation or the fundamental law of each individual element. That fundamental law, such as Newton's law or Ohm's law, describes the local behavior of that element. Direct and indirect connections that resemble the internal constraints within the boundaries of each subsystem define the transformation from the primitive system model to the connected system model. For systems with linear connections such as direct current servomotor, the internal constraints are given by one connection object, the velocity object (*V*). The velocity object is a 2-dimensional array, the rows in that array correspond to the variables in the primitive system (local variables) and the columns correspond to the variables in the connected system (global variables). Thus, the velocity object is a transformation from the global variables in the connected system model to local variables in the primitive system model.

The model of the physical system is set up by aggregating diagonally the connected system models of the hydraulic subsystem and the boring spindle. Modelling the physical system resulted in a set of differential/algebraic equations [7]. In a state space form, the behavior of the physical system is given by: $y = f(A, x, u, t)$

Where (x) is the set of initial state variables, (u) is the set of input sources, (A) is the state transition matrix for the physical system (connected system), the index (t) is the independent time variable and finally (y) is the set of output state variables.

3.2 Control System Modelling

Before a control algorithm can be designed and implemented we need a description of its required properties or behavior. A precise and comprehensive mathematical model of the properties of the control system could be expressed by employing logic notation. This mathematical model provides us with means to reveal the inconsistency and conflicts in the control system and to verify that the control system meets design specifications.

In order to carry out all control functions outlined in problem description, the control system should be decomposed into three subsystems. A process controller subsystem, which will be responsible to issue start and stop commands for the different physical entities and two continuos controllers. One controller for the servo valve in the hydraulic subsystem in order to regulate the feed forward motion of the hydraulic actuator. The second controller is for the servomotor in order to regulate the angular speed of the spindle motor. The decomposition is shown in Figure 5.

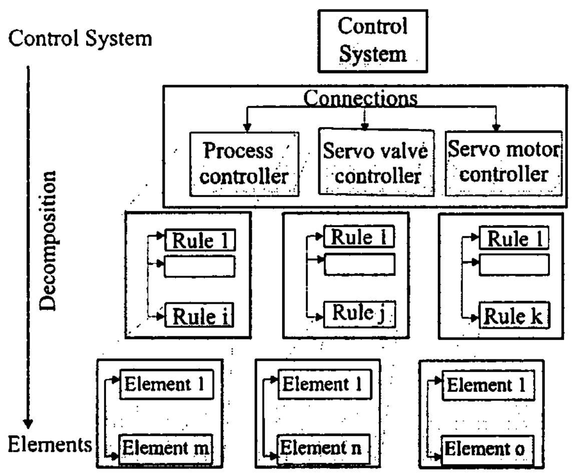

Figure 5. Decomposition of the Control System

The functions of each subsystem are described by a set of logical arguments or rules. Each of these logical arguments could be considered as a subsystem that can be decomposed further into a number of *factious logical elements*. These elements could be literally anything that could carry a logical variable that assumes either the state of truth or falsehood (1 0). These elements represent the primitive system model of a specific control function.

The procedure of modelling the control system will also move upward along the hierarchy until a total model is obtained as shown in Figure 6.

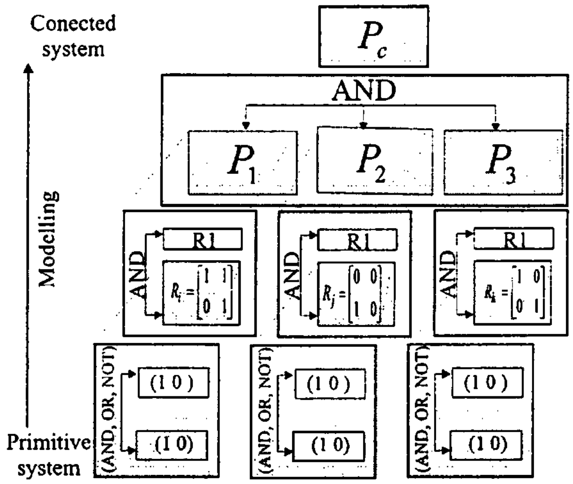

Figure 6. Modelling of the Control System

In the primitive system model, the connections between the logical variables are defined by three connection objects. In classical logic, they are referred to as *basic logical connectives*. The group of basic logical connectives includes; conjunction (***AND***), disjunction (***OR***), and negation (***NOT***). We propagate to the connected system model by aggregating the logical variables in the primitive system using the above logical connectives.

A connected subsystem is nothing else but the truth table of a logical argument expressed in a multi-dimensional array form. The number of axes in that array should be equal to the number of variables, therefore all repeated axes must be fused together by the method of colligation. The connected system expresses all the possible states of the system after imposing the internal constraints on the structure by connecting its individual elements. The behavior of the control system could be represented in the following form $s = f(P_c, i, n)$.

Where, (i) is a set of input variables that is external constraints due to interaction with the environment. (P_c) is the state transition matrix of the control system expressed in multidimensional array format. (s) is a set of output variables. The index (n) is analogues to a time index in that it specifies the order of a given state.

3.3 Model of The Total System

Since both systems utilize different types of signals internally, then intuitively speaking, the only possible interface between the physical and the control system model will take place externally, through the environment by means of the impressed sources. In the above manufacturing system, we can distinguish between two ways of interface between the physical and the control system.

Discrete interface: takes place in the process controller when the purpose of the control system is to coordinate asynchronous tasks to satisfy system requirements. For example, when an event command "start the spindle motor" is issued by the process controller, the spindle motor starts rotating. The process of rotation itself is controlled by the lower level controller (continuos controller).

Continuous interface: takes place locally on lower level control schemes when the purpose of the control system is to keep the behavior of the physical system within given boundaries such as implementing speed control. The resultant system model in this case is said to be a *hybrid system model.* The identifying characteristics of hybrid systems are that they incorporate both continuos dynamic behavior, i.e., the evolution of physical quantities governed by differential and algebraic equations ($y = f(A,x,u,t)$), and discrete event dynamic behavior governed by logic equations: ($s = f(P_c,i,n)$).

A total model can be obtained by generating a simple interface between the physical system model and the control system model. The interface will be consisting of two simple memoryless mapping functions (α) and (β) [1]. The first map (α) converts the controller output (s) into a constant incremental input to the physical system as follows: $u(t) = \alpha(s_n)$

The second map (β) converts the physical system output into a set of input logic variables to the control system as follows: $i_n = \beta(y(t))$, as shown in Figure 7.

What we have gained so far is establishing a consistent and complete mathematical description of mechatronics system model by using arrays to identify the properties of the whole system. The interface between the submodels is kept as simple as possible by employing simple mapping functions.

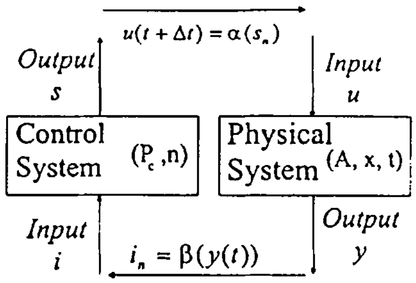

Figure 7. Continuous Interface

4. SIMULATION

Considering that the whole system is at rest and the boring spindle is at the rear position and the user has just pressed start button. The combination of input signal from the breakers and from the interface with the physical system will cause the control system to attain a new state and consequently a new set of output logical variables will be generated. This combination of output signals will cause the boring spindle to start moving forward in a rapid phase motion (uncontrolled motion). At the same time the spindle motor will be switched on and start rotating. However, since the spindle motor has not yet reached the feeding position, this rotation speed will remain unaffected by the servo motor control algorithm. Simulation for the angular velocity of the spindle motor is shown in Figure 8.

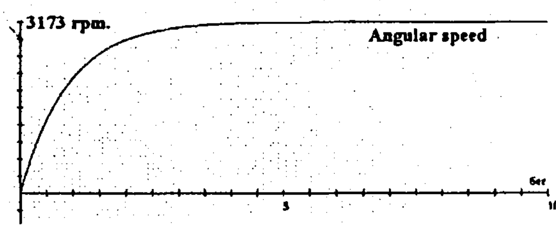

Figure 8. Angular Speed of Spindle Motor

It is shown from Figure 8 that the spindle motor will attain a constant rotation speed of 3173 rpm. after a transient period of about 5 seconds. The spindle motor was simulated assuming zero load torque on the spindle that is because the boring spindle has not yet reached feeding position. The objective of the control system will be to keep spindle motor within 3000 r.p.m. under all loading conditions. Simulation of the linear speed and the differential pressure of the hydraulic actuator is shown in Figure 9. It shows that the rapid phase velocity of the actuator is about 6cm/sec.

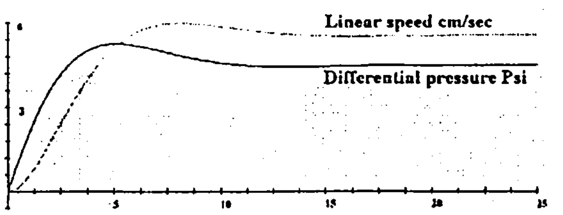

Figure 9. Linear Speed and Differential Pressure in the Actuator Cylinder

The system will continue to operate within the boundaries shown in Figure 8 and Figure 9 until it receives a new set of input sources. That set will be initiated when the boring spindle reaches position *M*.

Due to the signals generated from the interface with physical system, which is no longer at rest, combined with a new set of signals from the micro breakers. The control system will attain a new state and generate another set of output signals to be interpreted by the mapping function and converted into new input physical signals. In this case, the boring spindle will go from rapid phase motion (6cm/sec) to a controlled feed forward motion in such way that the feed forward motion will be kept at 2cm/sec, and the rotating speed of the spindle motor should be reduced from 3173 r.p.m. to be within 3000 r.p.m. under all loading conditions. The actuator linear velocity will be controlled by the servo valve controller algorithm. And the boring spindle motor will be controlled by the servo motor controller algorithm. Assuming that the servomotor is subjected to cosine load torque given by ($T_L = 2 \times \cos t$) and the hydraulic cylinder is subjected to load force given by ($F_L = 0.05 \times \cos t$). Simulation results are shown in Figure 10 and Figure 11.

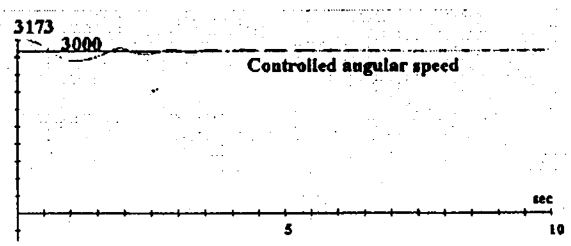

Figure 10. Controlled Angular Speed

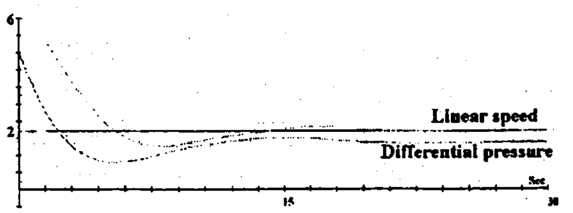

Figure 11. Controlled Feed Forward Speed

The simulation shows that the output speed of both the spindle motor and the actuator cylinder are kept within the boundaries specified by control algorithm.

5. CONCLUSIONS

A systems approach that utilizes multidimensional arrays for modelling mechatronics systems has been proposed and presented in this paper. The array approach provided us with a powerful mathematical representation of the real system. By utilizing multidimensional arrays we set up two submodels embodying the physical and the logical properties of mechatronics system. The interface between these two submodels is kept as simple as possible by employing a simple mapping functions. The practical advantages of using multidimensional arrays to describe the dynamic as well as the logic behavior is that one simulation environment suffice.

6. REFERENCES

[1] Antsaklis, et. al. 1993, Hybrid system modelling and autonomous control systems. Hybrid systems workshop, Technical university of Denmark.

[2] Bjørke, Ø., 1989, Manufacturing systems theory - A survey paper-keynote paper at the CIRP general assembly, Trondheim-Norway

[3] Bjørke, Ø., 1995, Manufacturing systems theory - A geometric approach to connection. Tapir-Trondheim

[4] Franksen, Ole I., 1992, The geometry of logic, from truth tables to nested arrays. The 4th international symposium on systems analysis and simulations. Berlin

[5] Harashima, et. al. 1996, Mechatronics-What is it, why, and how?. IEEE/ASME Transactions on Mechatronics, Vol.1, No.

[6] Hussein, B.A., 1997, On modelling mechatronics systems. NTNU report, Trondheim, Norway

[7] Isermann, R., 1996, Modelling and design methodology for mechatronics system. IEEE/ASME Transactions on Mechatronics, Vol.1, No.

[8] Lien, T.K., 1995, Digital styring for mekatronikk, Tapir Forlag.

[9] Møller, G. L, 1995, On the technology of array based logic. Ph. D. dissertation Electric power engineering department, Technical University of Denmark