THE ROLE OF CONTROLS IN MECHATRONICS

3.1 Introduction

Mechatronic design deals with the integrated and optimal design of a mechanical system and its embedded control system. This definition implies that the mechanical system is enhanced with electronic components in order to achieve a better performance, a more flexible system, or just reduce the cost of the system. In many cases the electronics are present in the form of a computer-based embedded (control) system. This does not imply that every controlled mechanical system is a mechatronic system because in many cases the control is just an add-on to the mechanical system in a sequential design procedure.

A real mechatronics approach requires that an optimal choice be made with respect to the realization of the design specifications in the different domains. In control engineering the design of an optimal control system is well understood and for linear systems standard methods exist. The optimization problem is formulated as: given a process to be controlled, and given a performance index (cost function), find optimal controller parameters such that the cost function is minimized. With a state feedback controller and a quadratic cost function, solutions for the optimal controller gains can be found with standard controller design software, such as Matlab (Fig. 3.1).

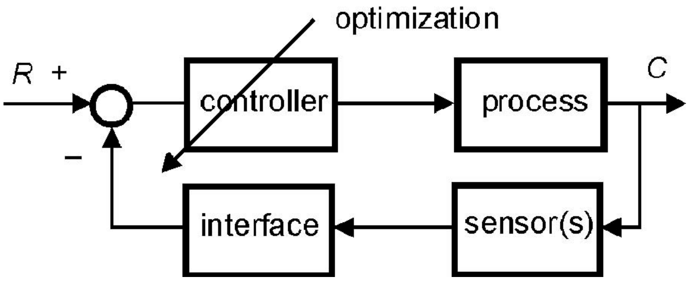


FIGURE 3.1: Optimization of the controller.

Mechatronic design on the contrary requires that not only the controller be optimized. It requires optimization of the system as a whole. In the ideal case all the components in the system: the process itself, the controller, as well as the sensors and actuators, should be optimized simultaneously (Fig. 3.2).

In general this is not feasible. The problem is ill defined and has to be split into smaller problems that can be optimized separately. Later on the partial solutions have to be combined and the performance of the complete system has to be evaluated. After eventually readjusting some parts of the system this leads to a sub-optimal solution

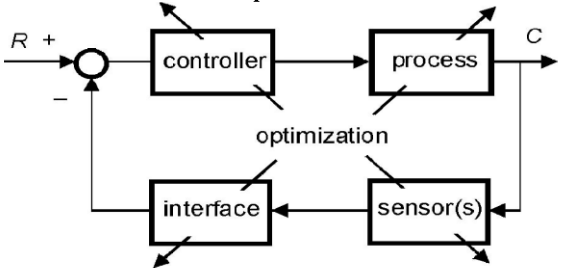


FIGURE 3.2: Optimization of the all system components simultaneously.

In the initial conceptual design phase it has to be decided which problems should be solved mechanically and which problems electronically. In this stage decisions about the dominant mechanical properties have to be made, yielding a simple model that can be used for controller design. Also a rough idea about the necessary sensors, actuators, and interfaces has to be available in this stage. When the different partial designs are worked out in some detail, information about these designs can be used for evaluation of the complete system and be exchanged for a more realistic and detailed design of the different parts.

Although the word mechatronics is new, mechatronic products have been available for some time. In fact, all electronically controlled mechanical systems are based on the idea of improving the product by adding features realized in another domain. Good mechatronic designs are based on a *real systems approach*. But mostly, control engineers are confronted with a design in which major parameters are already fixed, often based on static or economic considerations. This prohibits optimization of the system as a whole, even when optimal control is applied.

In the last days of gramophones, the more sophisticated designs used tacho feedback in combination with a light turntable to achieve a constant number of revolutions. But a really new design was the compact disc player. Instead of keeping the number of revolutions of the disc constant, it aims for a constant speed of the head along the tracks of the disc. This means that the disc rotates slower when tracks with a greater diameter are read. The bits read from the CD are buffered electronically in a buffer that sends its information to the DA converter, controlled by a quartz crystal. This enables the realization of a very constant bit rate and eliminates all audible speed fluctuations. Such a performance could never be obtained from a pure mechanical device only, even if it were equipped with a good speed control system. In fact, the control loop for the disc speed does not need to have very strict specifications. It should only prevent overflow or underflow of the buffer. The high accuracy is obtained in an open loop mode, steered by a quartz crystal (Fig. 3.3).

The flexibility introduced by the combination of precision mechanics and electronic control has allowed the development of CD-ROM players, running at speeds more than 50 times faster than the original audio CDs. A new way of thinking was necessary to come to such a new solution. On the other hand, the CD player is still a sophisticated piece of precision mechanics. No solid-state electronic memory

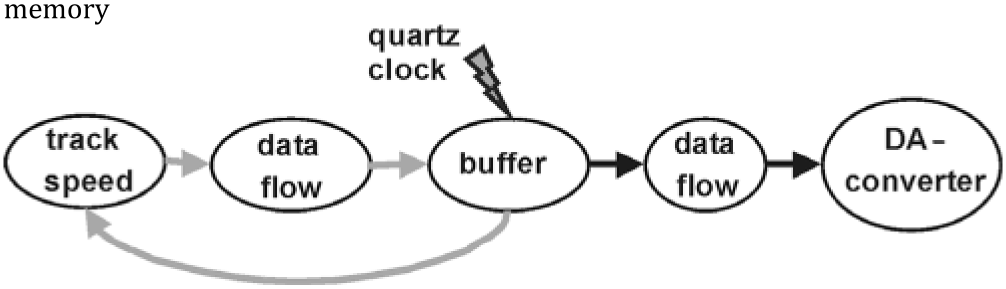


FIGURE 3.3
Combination of closed-loop and open-loop control in a CD player.

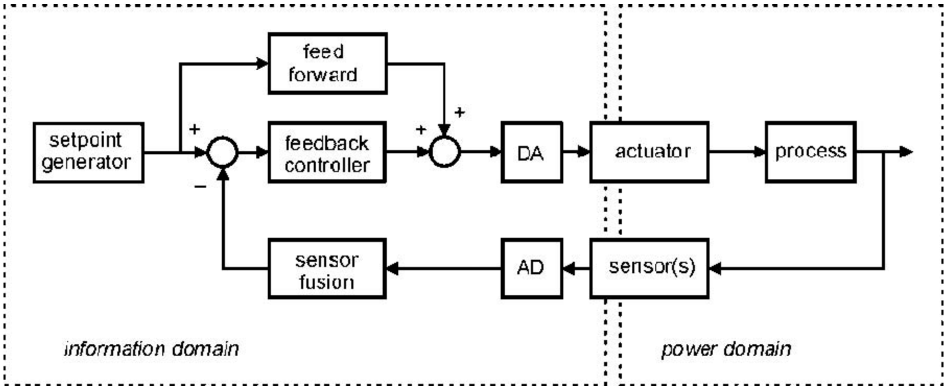


FIGURE 3.4: Mechatronic system.

3.2 Key Elements of Controlled Mechatronic Systems

A mechatronic system consists by definition of a mechanical part that has to perform certain motions and an electronic part (in many cases an embedded computer system) that adds intelligence to the system.

In the mechanical part of the system power plays a major role. This in contrast to the electronic part of the system where information processing is the main issue. Sensors convert the mechanical motions into electrical signals where only the information content is important or even into pure information in the form of numbers (if necessary, through an AD converter). Power amplifiers convert signals into modulated power. In most cases the power supply is electrical, but other sources such as hydraulic and pneumatic power supplies are possible as well. A controlled mechanical motion system thus typically consists of a mechanical construction, one or more actuators to generate the desired motions, and a controller that steers the actuators based on feed-forward and sensor-based feedback control (Fig. 3.4).

3.3 Integrated Modeling, Design and Control Implementation

Modeling

During the design of mechatronic systems it is important that changes in the construction and the controller be evaluated simultaneously. Although a proper controller enables building a cheaper construction, a badly designed mechanical system will never be able to give a good performance by adding a sophisticated controller. Therefore, it is important that during an early stage of the design a proper choice can be made with respect to the mechanical properties needed to achieve a good performance of the controlled system. On the other hand, knowledge about the abilities of the controller to compensate for mechanic imperfections may enable that a cheaper mechanical construction be built. This requires that in an early stage of the design a simple model is available that reveals the performance limiting factors of the system. Still there is a gap between modeling and simulation software used for evaluation of mechanical constructions and software used for controller design. Mechanical engineers are used to finite element packages to examine the dynamic properties of mechanical constructions. It is only after reduction to low-order models (modal analysis) that these models can be used for controller design. On the other hand, typical control-engineering software does not directly support the mechatronic design process either; in the modeling process the commonly used transfer functions and state space descriptions often have lost the relation with the physical parameters of the mechanical construction. Tools are required that allow modeling of mechanical systems in a way that the dominant physical parameters (like mass and dominant stiffness) are preserved in the model and simultaneously provide an interface to the controller design and simulation tools control engineers are used to.

Simulation is an important tool to evaluate the design of mechatronic systems. Most simulation programs like Simulink use block diagram representations and do not support physical modeling in a way that direct tuning of the physical parameters of the mechanical construction and those of the controller is possible as required in the design of mechatronic systems. Recently, programs that allow physical modeling in *various physical domains* became available. They use an object-oriented approach that allows hierarchical modeling and reuse of models. The order of computation is only fixed after combining the subsystems. Examples of these programs are 20-sim, described by Broenink as CAMAS and Dymola.

In this section the modeling and simulation program 20-sim (pronounced Twente Sim) will be used to illustrate the simultaneous design of construction and controller in a mechatronic system. 20-sim supports object-oriented modeling. Power and signal ports to and from the outside world determine each object (Weustink, De Vries, and Breedvel). Inside the object there can be other objects or, on the lowest level, equations. Various *realizations* of an object can contain different or more detailed descriptions as long as the interface (number and type of ports) is identical. Modeling can start by a simple interconnection of (empty) submodels. Later they can be filled with realistic descriptions with various degrees of complexity. De Vries refers to this as *polymorphic* modeling. Submodels can be constructed from other submodels in hierarchical structures. Proper physical modeling is achieved by coupling the submodels by means of the *flow of energy*, rather than by *signals* such as voltage, current, force, and speed. This way of modeling is well suited for mechatronics system design. It will be illustrated with an example.

We want to consider the design of a simple servo system, considering the use of a voltage source, a DC motor, and a mechanical load driven through a transmission (Fig. 3.5).

The transmission is disregarded for the time being. The belt is considered as infinitely stiff and the transformation ratio is taken care of by changing the motor constant. If a power amplifier driven by a signal generator describes the voltage source, we can draw the iconic diagram of Fig. 3.6. At this stage the different *components* in this model are still empty. But all components have electrical and/or mechanical "ports." With the proper interfaces (ports) defined, the components can be connected to each other.

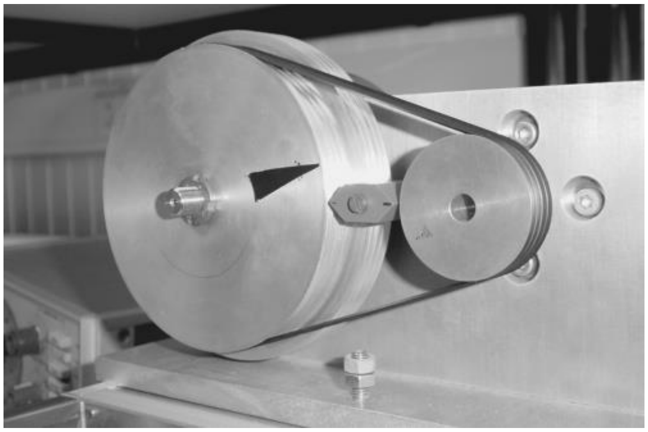


FIGURE 3.5: Simple DC-servo system.

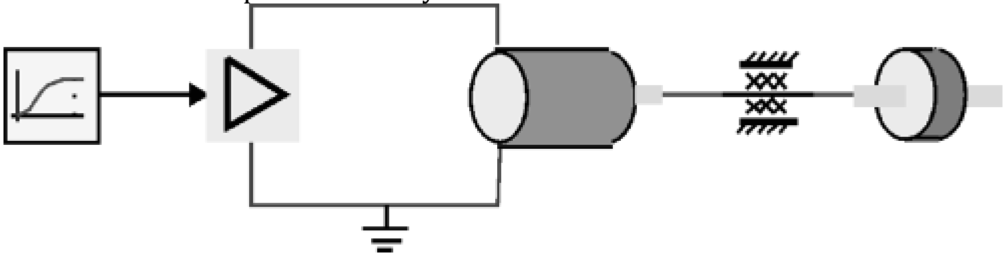


FIGURE 3.6: Iconic diagram of the simple servo system.

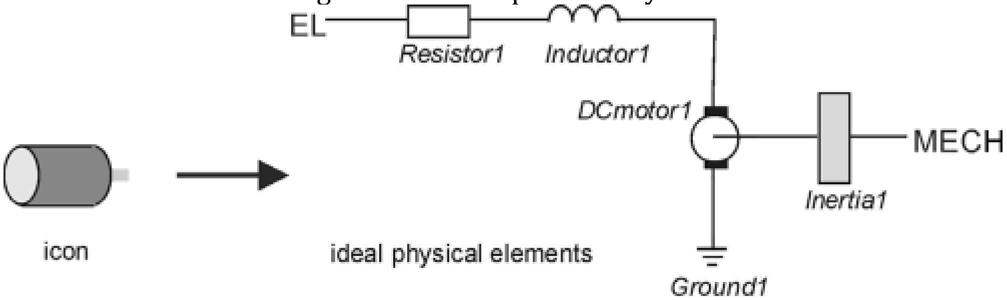


FIGURE 3.7: Icon of the motor expanded to ideal physical elements.

In the next step we can detail the description of the DC motor. One solution could be the description given in Fig. 3.7. The motor is now described by a number of *ideal physical elements*, each representing a basic physical relation. The motor has an electrical (EL) as well as a mechanical port (MECH).

Each of the *elements* in this figure can be described as an element with an electrical and/or mechanical port. The idea of ports is made more explicit in so-called bond graphs.

For the electrical elements these are the voltage difference over the element and the current through the element. For the mechanical elements these are the torque and the (angular) velocity. The products of these conjugated variables (*P=ui* or *P=T*ω) represent power.

If we go down a step further into the hierarchy, we arrive at the level of equations. For instance, an electrical resistor can be described by the equation:

p.u = R*p. I (3.1)

where the variables *p.u* and *p.i* indicate the conjugated variables *u* and *i* of the electrical port *p* . Note that this is an equation and not an assignment statement. It could have been written equally well in the form:

p.i = 1/R*p.u (3.2)

In a similar way the inductance can be described by the equations:

p.u = L*ddt(p.i) OR (p.i) = 1/L * int(p.u) (3.3)

where ddt(*p.i*) denotes di/dt and int(*p.u*) denotes $\int udt$. In case of an R-element there is no preference for one of the two forms. For the I-element the integral form is preferred in the simulations. 20-sim determines the preferred causal form and derives the equations automatically.

The energy flow or *power P* is the product of *two conjugated signals*, called effort (*e*) and flow (*f*):

$P = ef$ (3.4)

Examples of this expression in the mechanical and electrical domain are

$P = Fv$ or $P = T\omega$ (3.5)

$P = ui$ (3.6)

Where *F* is force, *v* is velocity, *T* is torque, ω is angular velocity, *u* is voltage, and *I* is current.

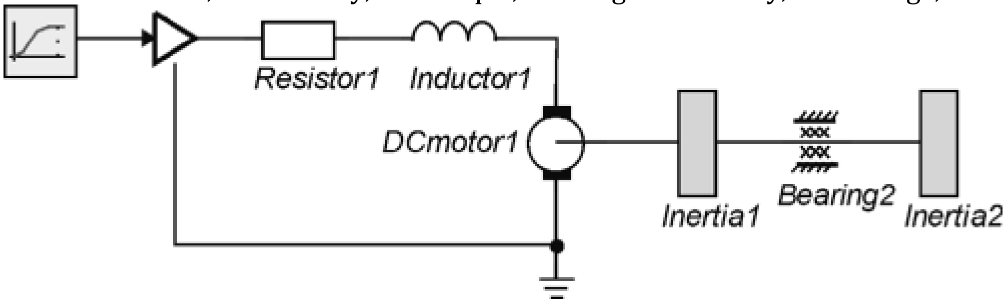


FIGURE 3.8: Complete model in the form of ideal physical elements.

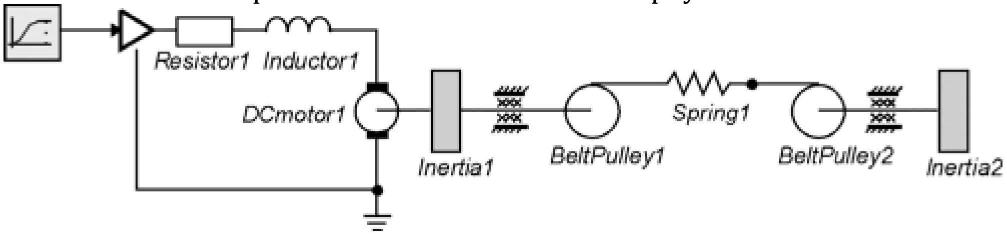


FIGURE 3.9: Model extended with transmission

When we expand the complete Fig. 3.6 we obtain Fig. 3.8. When this model is processed a message pops up that indicates that inertia 2 has a dependent state. The two inertias in this model always have the same speed, and therefore, they are dependent. They cannot have independent initial conditions.

The message indicates that this element can only be written in derivative form:

There are several ways to deal with this problem.

1. The two inertias can be combined into one inertia (the program will do this automatically). A message pops up that the dependency of the two inertias has been solved symbolically.
2. Dealing with the derivative causality by means of an implicit integration algorithm.
3. The transmission can be added, including some flexibility in the belt.

If the flexibility is negligible, solution 1 leads to the simplest model. On the other hand, the warning raises the question whether the flexibility of the belt can be disregarded indeed. If not, the model has to be extended with a spring element. It should be noted that this should not be done for numerical reasons only. If the transmission were very stiff, this would result in high-frequency dynamics and lead to unnecessary slow simulations. On the other hand, if the flexibility is important, as it is in this system, the warning draws the designer's attention to the fact that the model may be oversimplified. In Fig. 3.9 the transmission, including a spring element, has been added. Processing of this model does not produce any warnings.

This example illustrates how modern software can help to come up with a model that has the complexity that is needed for a particular problem. Physical models, in the form of an iconic diagram, based on connecting elements by means of power ports, may help in this modeling process. The user can select the preferred view, whether this is a bond graph, an iconic diagram with ideal physical element, or a view using higher lever submodels, like in Fig. 3.6. In the next section it will be shown how to use this model for the design of controllers.

Control System Design Methodologies

Many processes can be reasonably well controlled by means of PID controllers. This is due to the fact that these processes can be more or less accurately described by means of a second-order model. Tuning rules, like those of Ziegler Nichols, enable less experienced people to tune such controllers. Relatively simple models can also describe many mechatronic systems. A mechatronic system mostly consists of an actuator, some form of transmission, and a load. A fourth-order model can properly describe such a system. The performance-limiting factor in these systems is the resonance frequency. A combination of position and tacho feedback (basically a PD controller) can be applied here as well. But due to the resonant poles proper selection of the signals to be used in the feedback is essential. Efforts have been made to derive recipes for tuning such systems, in addition to selecting the proper feedback signals. Computer support tools are essential to enable less experienced designers to use these recipes describe a structural design method for mechatronic systems. The method starts with reducing the conceptual design to a fourth-order model that represents the dominant properties of the system in terms of the total mass to be moved and the dominant stiffness. This model still has physical meaningful parameters. In this model appropriate sensors are chosen, as well as a path generator. In the conceptual design phase a simple controller is developed and mechanical properties are changed, if necessary. Then a more detailed design phase follows where also parameter uncertainties are taken into account.

Servo System Design

Here we will consider some simple aspects of the design of a servo system in order to illustrate the advantage of the use of physical models and to illustrate the need for an integrated design approach. We consider the model discussed before, a load driven by an electric motor, through a flexible transmission.

The iconic diagram of this model was given in Fig. 3.9. In this example a current amplifier has replaced the voltage amplifier allowing the removal of the electrical resistor and the inductance. In the step responses of Fig. 3.10 the resonance due to the flexible transmission is clearly visible.

From the equations used for the simulation, 20-sim can automatically derive a model in a form suitable for controller design, such as a state-space description, a transfer function, or poles and zeros. An interface is provided to Matlab enabling, for instance, to use Matlab algorithms to compute the gains of advanced controllers like an LQR (optimal state feedback) or LQG controller (with a Kalman filter for state estimation and optimal state feedback). The diagram of the process together with an LQG controller is given in Fig. 3.11 and some responses in Fig. 3.12.

A properly designed P(I)D controller is able to perform almost similarly, especially when the amount of noise is small. A first attempt could be to use only measurements of the load angle and load speed.

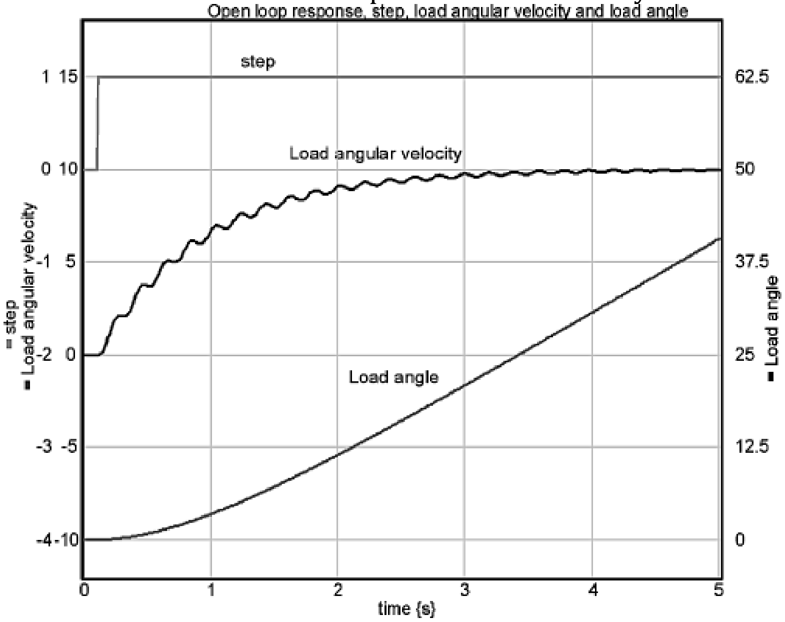
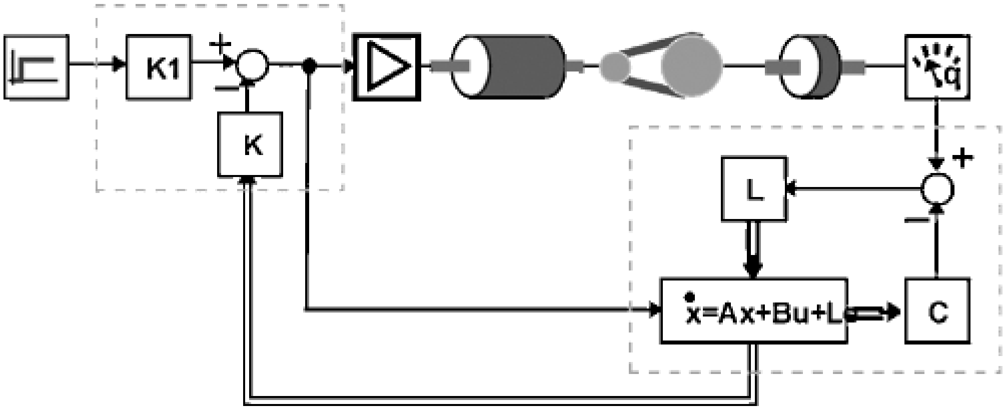


FIGURE 3.10
Open loop responses.


K1
+
−
K
L
+
−
ẋ=Ax+Bu+Le
C
q̇