COMPLETE AIRCRAFT SYSTEM SIMULATION FOR AIRCRAFT DESIGN -PARADIGMS FOR MODELLING OF COMPLEX SYSTEM

Petter Krus* **Jonas Nyman****

****Department of Mechanical Engineering**
Linköping University, S-58183 Linköping, Sweden

***Department of Mechanical Engineering**
Linköping University/Volvo Aero Corporation

Abstract

Modelling and simulation is of crucial importance for system design and optimisation. In aircraft, simulation has been strong in the area of flight dynamics and control. Modelling and simulation of the hydraulic systems has also a long tradition. The rapid increase in computational power has now come to a point where complete modelling and simulation of the sub systems in an aircraft is possible.

In this paper it is demonstrated how the actuation system control surfaces can be simulated using a flight dynamics model of the aircraft coupled to a model of the actuation system. In this way the system can then be optimised for certain flight condition by "test flying" the system. The distributed modelling approach used, makes it possible to simulate this system faster than real time on a 650 MHz PC. This means that even system optimisation can be performed in reasonable time.

Introduction

There are several methods for modelling and simulation of dynamic systems. Figure 2 shows the taxonomy of dynamic system modelling and simulation. There are basically two ways of representing systems. One is the signal flow approach using block diagrams and the other is the power port modelling representation. Using power port modelling there are two different ways of representing the equations. One is the conventional lumped parameter approach where derivatives of states are calculated in components and subsystems. These are then integrated in a centralised solver.

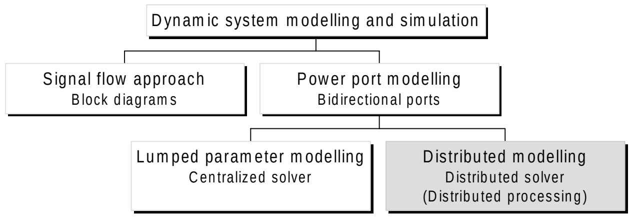
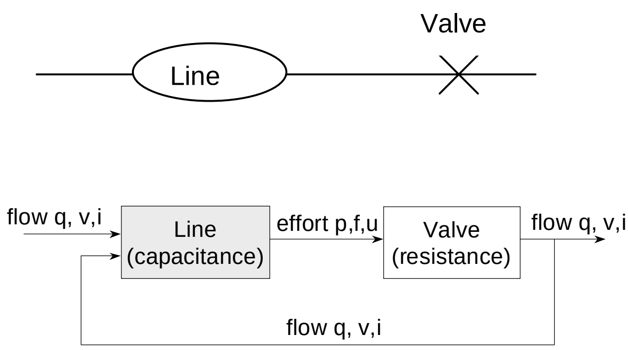


Figure 1. Taxionomy of modelling and simulation methods.

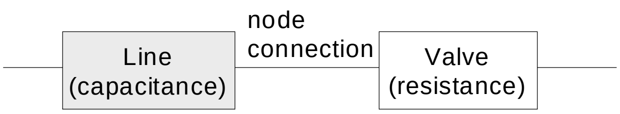
The other approach is to use a distributed solver where all the differential equations are solved locally in each component or subsystem. This is discussed in more detail later on. To illustrate the difference between different Signal flow and power port modelling, a simple Valve-volume (transmission line) system is used.

Using signal flow modelling the system is represented by the block diagram below

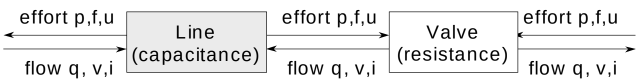


The block diagram clearly shows all variable couplings in the system. This is very useful for system analysis and is therefore very popular for representing control systems and systems connected to control systems.

Using power port modelling the diagram below is obtained



Using power port modelling there are bi-directional nodes containing the transfer of several variables (in this case, one in each direction). Therefore the power port representation is more compact. The figure below shows the signals transferred at each node. In one direction there are the effort variable, which could be pressure, force, voltage etc. In the other direction there is the flow variable, which could be, flow, speed, current, etc.



In addition the power port diagram closely match the real system in that each node connection represents a real physical connection, and physical connection are by nature bi-directional. This makes power port modelling much more suitable, than block diagrams, for modelling of physical systems. It should, however, be pointed out that block diagram representations could be used for parts of a system.

Distributed Modelling for Simulation of Fluid Power Components and Systems

Simulation of fluid power systems are characterised by difficulties such as very strong nonlinearities, stiff differential equations and a high degree of complexity. Using conventional integration techniques it is necessary to use a very small time step in order to be able to deal with numerically stiff problems.

Distributed modelling using transmission line elements represents a very suitable method for modelling and simulation of large complex dynamic systems. The origin of this concept goes back to Auslander 1968. This method evolves naturally for calculation of pressures when pipelines are modelled with distributed parameters. This approach was adopted for simulation of fluid power systems with long lines in the HYTRAN program.

A related method is the *transmission line modelling* method (TLM) presented by Johns and O'Brien for simulation of electrical networks. This method has also been used by Kitsios and Boucher, for modelling of a hydraulic control system, where both the fluid part as well as the mechanical part was modelled using TLM. In the TLM method the basic dynamic element used for modelling is the lossless, dispersionless transmission line. All capacitances and inductances are replaced by transmission line elements where the length of the element is adjusted to correspond to the distance a wave propagates in one time step. These elements are referred to here as *unit transmission line elements* (UTL). A method related to TLM was also proposed by Fettweis 1971 for modelling electric filters where the electrical system is modelled with UTL-elements. This method has given rise to a whole branch of digital filters called wave filters.

Jones and O'Brien pointed out that an important aspect of modelling using UTL-elements is that most of the numerical errors introduced by an ordinary solver are avoided. The errors made due to the introduction of UTL-elements, are better described as modelling errors.

An attractive feature with this is that laws of conservation of mass and energy still hold for the solution, since there always exists a plausible physical system for the model although the line lengths may vary compared to the original system. This also implies that the user may tolerate a larger numerical error since, generally, quite large modelling errors are present anyway (errors of the order of 10\% are generally considered acceptable from an engineering point of view).

1 shows different ways of modelling and simulating a system. All real physical systems are distributed although they often can be accurately described by a lumped model. On the other hand if we have a lumped model we can transform it into a distributed model by substituting some capacitances or inductances with UTL-elements and make substantial gains in the numerical properties of the solution.

Real physical world obeys the laws of information processing as well and using UTL-elements, the propagation of information that takes place in real systems is simulated. Calculation of state variables are all handled by the different component subroutines associated with them, and the transmission of information between components is restricted by the transmission speed (e g the speed of sound or ultimately by the speed of light). Consequently, there is no immediate communication between components that are separated by some distance. This is also called the principle of local causality. This indicates that, provided the delay time is sufficiently large, there is no need to solve one big system of non-linear differential equations at each time step. Instead, there will be many small systems of equations, which are much easier to solve. To obtain sufficient delays the distances between components can be altered using UTL-elements.

It should, however, be pointed out that using the UTL-element to represent all dynamics in the system is not always justified. Some components or small systems can be more effectively solved using an exact solution, or a numerical solver for the differential algebraic equations describing the component. In this way, the UTL-element is used only for communication between components or subsystems. Another way to put it is, that the UTL-element can be used to split a system into elements that are small enough to be solved effectively using numerically robust methods.

use of UTL-elements for partitioning of systems is a non-exclusive approach. Conventional simulation techniques can still be used within the subsystems. This means that UTL-elements can be used to connect simulation models developed in different simulation packages.

The Unit Transmission Line Element

In transmission line modelling the basic dynamic element is the unit transmission line. In the HOPSAN package this is used to connect different components to each other. In the general case it can be used to model both capacitances and inductances. In the HOPSAN-package, however, it is used only to represent capacitances (oil volumes and mechanical springs).

The complete set of equation that describes a loss less transmission line are:

$$P_1\,(t+T) = p_2(t) + Z_c q_1(t+T) + Z_c q_2(t) \qquad (1)$$

$$P_2\,(t+T) = p_1(t) + Z_c q_1(t) + Z_c q_2(t+T) \qquad (2)$$

Where Zc is the characteristic impedance of the line, p and q are pressures and flows respectively. Note that the main property of these equations is the time delay they introduce in the communication between the ends. Note that the main property of them is the time delay they introduce in the communication between the ends. Another interesting observation is found if p2 in eq (1) is substituted with eq (2) (shifted T) and the outlet in 2 is closed (q2=0).

$$p_1(t+T) = p_1(t-T) + Z_c(q_1(t-T) + q_1(t+T)$$

If this equation is compared to the trapezoidal method for integration

$$y_{h+1} = y_1 + \frac{1}{2} h\,(f(y_t,t) + f(y_{h+1}, h+t))$$

These are the same if T=h/2

The implication of this is that if we use the trapezoidal method to integrate pressure in a volume (capacitance) between two components, this corresponds to introducing a short pipe instead of a pure capacitance see Figure 2.

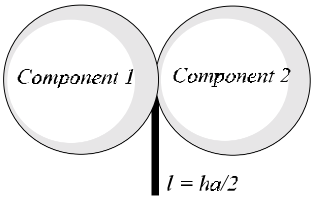


Figure 2. Modelling of capacitance using the trapezoidal method.

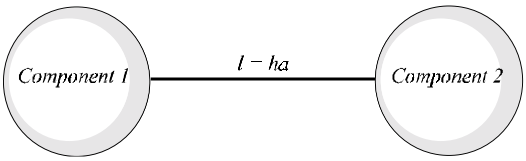


Figure 3. Modelling of capacitance using UTL-element.

If, however the volume is modelled as pipe to begin with, this can be oriented so that it isolates the two components from each other and thereby isolate them numerically from each other since there is a physically motivated time delay between the components. In order to represent a pure capacitance C with an UTL-element the length of the element will correspond to $h\ a$ where h is the time step and a is the speed of sound in the fluid. The characteristic impedance is simply set to Zc=h/C. For a more detailed discussion on the UTL-element as an integrator see Krus-Jansson-Palmberg-Weddfelt 1990 where also elements with more than two nodes are described.

Structure of model subroutine

During the simulation it is suitable to have all C-type components in one section and the Q-type components in another section. Since all communication between Q-type components will go through the C-type components there will be no communication between Q-type components in one time step. Consequently there are no restriction of the ordering of Q-type components, except for the case of a feedback controller. In a position servo the load position will be feed back to the servo valve without any time delay. This aspect makes it recommendable to place the Q-type components that represents control elements such as valves last in the section of Q-type components. This will yield the principle structure of the model subroutine shown in Fig

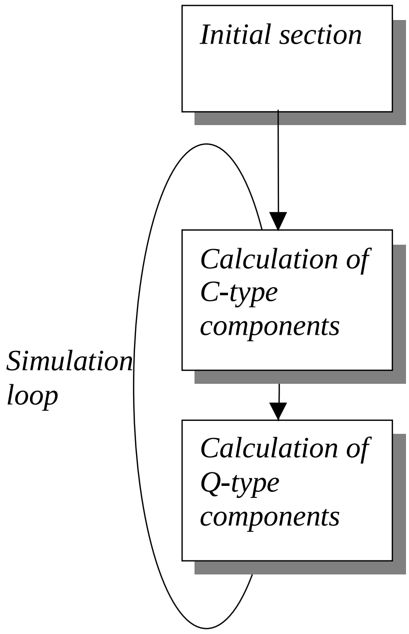


Figure 4. Simulation loop.

When simulating systems the evaluation of the system is performed in two steps:
1. The first step in the simulation loop is to calculate the characteristics from all teh C-type components presenting characteristics as output, such as in transmission line elements.
2. The flows (and pressures) from the Q-type components can then be calculated. Step one is then repeated for the next time step.

Note, however, that a component may represent a system with its internal transmission lines. This does not matter as long as all the external connecting nodes for the component are of the same type, ie calculates flows and pressure from characteristics.

Parallel processing

The formulation of the problem using distributed modelling is very suitable for parallel processing, since it mimics the way signals propagates and communicate in the real system. The formulation is such that different parts of the simulated system can be simulated on different processors. In this way, even highly complex systems can be simulated in real-time, it is only a matter of splitting the system into subsystems small enough to be run in real -time. A description of early efforts in this area can be found in Krus et al 1990. Very promising work in this area, where subsequently also carried out at the University of Bath, UK using transputer technology (Burton et al 1992). The transputer technology represented somewhat of a dead end, but parallel processing will ultimately be the main road to increasing the performance beyond the next decade. There is plenty of more potential in increasing the complexity of processors than increasing the clock speeds which is limited by the speed of light since the wavelength need to be longer than the physical dimensions of the processor. The dimensions of the processor are limited downwards because of quantum effects.

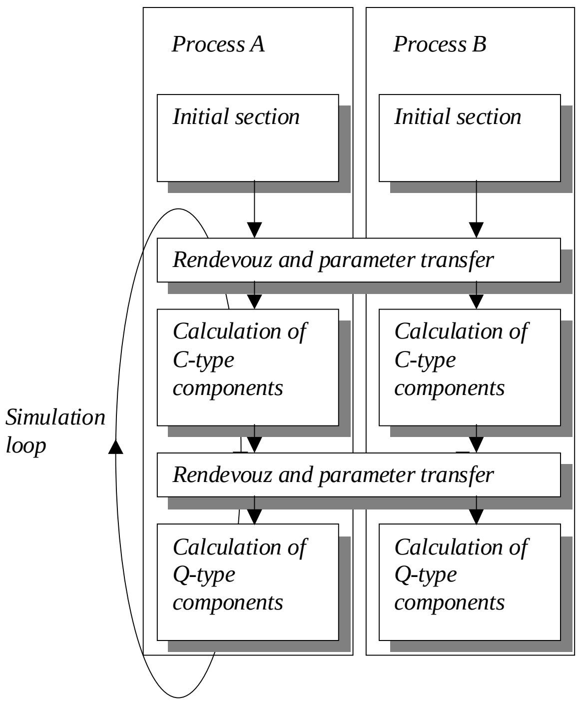


Figure 5. Parallel processing loop.

Hierarchy

In order to handle large system it is necessary to introduce hierarchy in the models. Using hierarchy it is possible to hide complexity by defining parts of the systems as subsystems that can be treated as components at the top level. Each such subsystem can be composed of both C-typ and Q-type components provided that the interface is of one type (most commonly Q-type). It is also possible to have a local time step in the subsystem. Figure 6 shows how one subsystem can be composed by other components and subsystems. In fact the structure of system model and the subsystem model is almost identical. The main difference is that the system model does not have to interface with other systems.

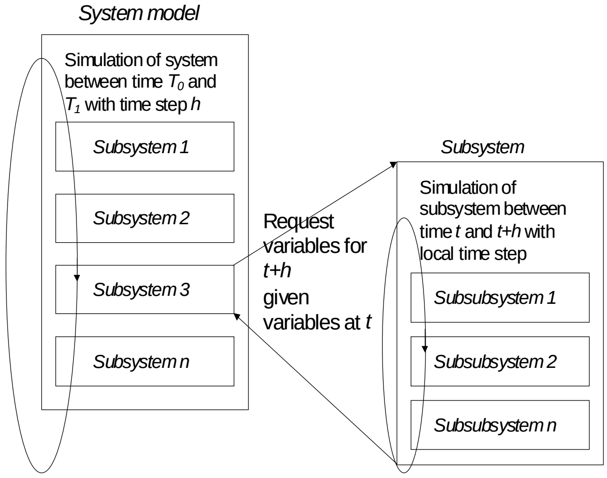


Figure 6. Hierarchy within the simulation loop

A Simulation Environment

The simulation paradigm described here can be translated into a set of tools for modelling and simulation. In the figure below an example of such an environment is shown. There is one tool for defining and assembling a system. There is another tool to create components and subsystems at a detailed equation level using support from a symbolic math package. Finally there is a shell for running the simulation. This shell can in principle be very simple since the system model is highly self contained. All computations are processed within the model so the shell should only administrate the simulation. It is, however, desirable with powerful analysis tools and simulation management tools accessible within the shell. This includes frequency analysis and general capabilities to perform calculations on simulation results. Furthermore, it is possible to define parameter variations for series of simulations. This also makes it possible use optimisation on the simulation model for system optimisation.

A simulation environment for simulation of complex fluid and systems

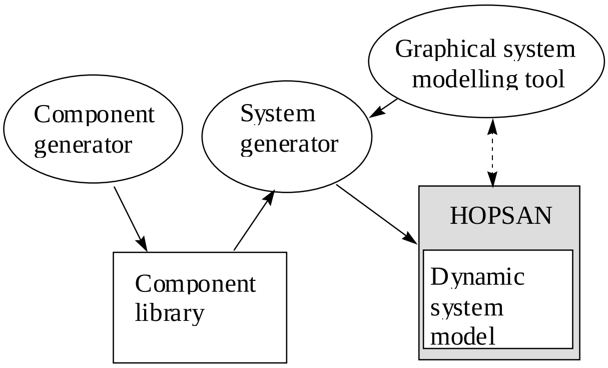


Figure 7. The HOPSAN simulation environment.

Simulation of an engine control system using two different paradigms

As one example an control system for the variable nozzle in a jet engine is modelled. This is essentially a piston controlled by a variable pump. The other components are there mainly to maintain the pressure level in the system. Figure 8 represents the HOPSAN model, which is a power port representation very similar to the system schematic.

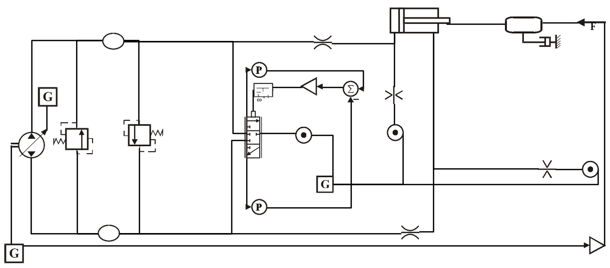


Figure 8. Power port model of nozzle control system.

Figure 9 represents the same system but in block diagram form used in Simulink. Here, all variable connections are made explicit, resulting in approximately twice as many connections. An effort has here been made to maintain the physical structure of the system, which brings it closer to the power port modelling representation. However, if a connection is made to the wrong connection it can result in totally unphysical behaviour, while a power port model yields the same faulty behaviour as the real system if a mistake is made in the connection. This clearly is an advantage in debugging models. The simulation results are shown in Figure 10 where the piston position is shown and in Figure 11 where the pressures are shown. The simulation results for both models are shown, but at this resolution they are identical so only one curve is visible in each diagram.

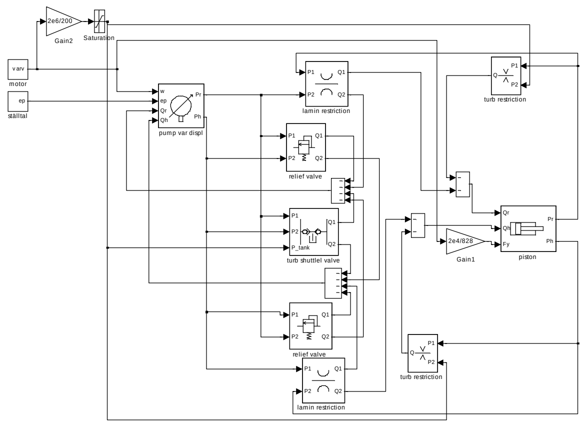


Figure 9. Block diagram model of nozzle control system.

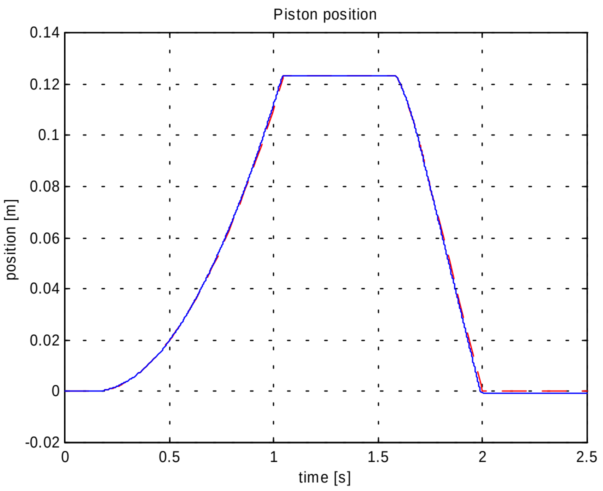


Figure 10. Simulation result showing the piston position

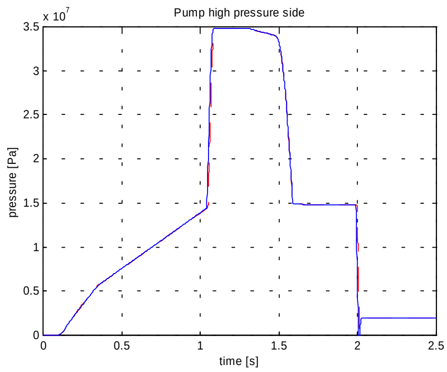


Figure 11. The pressure in the high-pressure side of the piston.

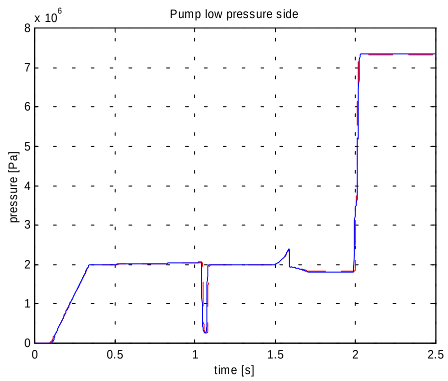


Figure 12. Pressure in the low-pressure side of the piston.

Figure 13 shows another example of a system that has been simulated using the HOPSAN package. This represents a three dimensional dynamic model of an aircraft coupled to an engine model and a hydraulic actuator system. This system can be simulated in real time using a 650 MHz Pentium III machine.

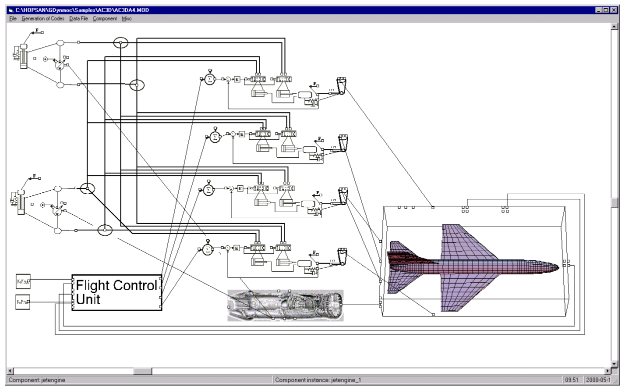


Figure 13. 3D-flight dynamics model connected to actuator system and propulsion.

The three-dimensional flight dynamic model was developed using the HOPSAN component generator (HOPSAN-COMPGEN) see Krus 1996. This employs the formula manipulating power of *Mathematica* to generate a FORTRAN subroutine for the flight dynamics. In this way it is very easy to do changes to the flight dynamics since modifications can be done at a high level of abstraction, i.e. if entirely different control surface configurations are to be used.

Using this model it is possible not only to optimise the actuator system, but to optimise the actuators together with parameters in the aircraft such as control surface area. As a consequence it is possible to obtain a better over all solution than if each domain is designed separately. It also forms a platform for communication. Engineers from different disciplines can see and understand consequences of their design decisions on other disciplines.

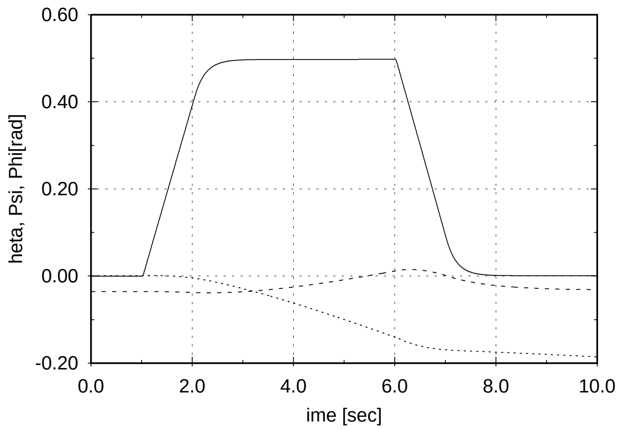


Figure 14. Aircraft angles during a turn. The bank angle goes up to .5 rad and the yaw angle levels out at -0.2 rad.

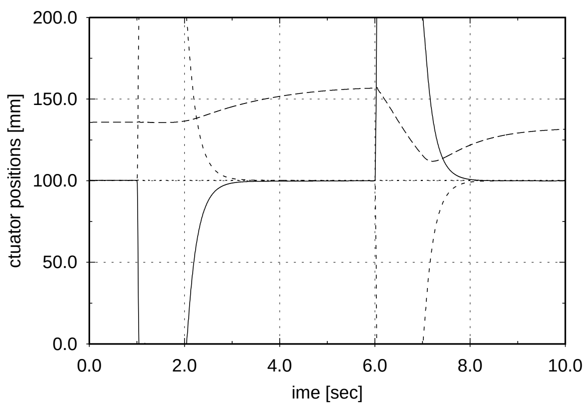


Figure 16. The actuator positions for ailerons, horizontal stabiliser, and rudder (not moving).

Complete functional aircraft modelling

The development in computational speed has been enormous over the last decade. An increase in simulation speed of at least a factor of thousand has been experienced over the last decade and this rate shows no signs to diminish.

The development in systems modelling has come to the point where *complete modelling* of systems is possible, e.g. the complete hydraulic system and interfacing systems in an aircraft.

Complete modelling does not mean that all components are dealt with down to the very smallest details of their behaviour. It does, however, mean that all functionality is modelled, at least qualitatively. Furthermore, in contrast with the usual problem oriented approach, the tests to be simulated with the model are not explicitly known when the model is established. The aim would be to develop query independent models that can be used as test beds for analysis of a wide range of test applications.

During a project the first objective for the different groups would be to quickly produce highly simplified models that defines the interface and produce a reasonable boundary condition to the rest of the system. A model of the complete system can then be assembled that can be used by all of the groups for the next phase, where the design groups gradually refine their models in the context of a complete system model. Eventually all the design groups have refined their models and a complete model with detailed subsystem can be obtained.

The reason to develop models of this kind is, that the same model can be used, by various groups in a project, as a test bed. This means that subtle couplings between different subsystems can be detected and dealt with at an early stage of system development. It also becomes easier to ensure that all groups use the latest most updated model. The main objections against this approach has been the computational resources needed to simulate such a large system, the difficulty to model large systems and to deal with the results from large models.

A way of realising (more) complete modelling is using *web-based simulation*, where different team models different subsystems that can be accessed through internet/intranet and used by other teams, to assemble complete systems models. This does, however, require extremely high reliability and robustness, from a numerical point of view, since no one can be expected to have complete overview of the complete model, and the dynamic phenomena that may results from interaction between subsystems. However, using distributed modelling, where the numerical properties of a subsystem are indifferent whether it is simulated by itself or together with others, this can be assured.

Conclusions

One of the most important shifts in paradigm occurring in engineering system design is the adoption of common system models as a foundation for system design. This allows for a much more effective product development process since a system can be tested in all stages of design.

The distributed modelling and simulation stands out as *the only viable option* for large-scale system. This is because it has both good scaling properties as well as a potential for parallelisation. Although conventional (centralised) solver can be shown to be superior for some problem with low complexity the linear scaling properties of distributed modelling will always be orders of magnitude more efficient for large-scale systems. Furthermore, distributed modelling is a non-exclusive approach that can accommodate conventional centralised solver in subsystems.

Using distributed modelling it is possible to use one model for a wide range of analysis. From mission performance to detailed behaviour on a component level, this clearly simplifies the analysis, since not only the data are shared for different analyses but also the model itself. Modelling and simulation has come to a point where it is changing the way engineering is done, and the true potential of simulation is relished when it is connected with other tools, such as optimisation, to support the design process.

Reference

[1] D M Auslander, 'Distributed System Simulation with Bilateral Delay-Line Models' Journal of Basic Engineering, Trans. ASME p195-p200, June 1968.

[2] P.B. Johns and M. O'Brien, 'Use of the transmission line modelling (t.l.m) method to solve nonlinear lumped networks'. *The Radio and Electronic Engineer*, 50:59-70, Jan/Feb 1980.

[3] Burton, J.D., Edge, K.A. and Burrows, C.R., "Modelling Requirements for the Parallel Simulation of Hydraulic Systems", ASME *Journal of Dynamic Systems and Control* March 1994

[4] P Krus, 'An Automated Approach for Creating Components and Subsystems for Simulation of Distributed Systems' Presented at the 'Ninth Bath International Fluid Power Workshop', Bath, UK 1996.

[5] P Krus, A Jansson, J-O Palmberg, 'Optimization Using Simulation for Aircraft Hydraulic System Design', Procedings of IMECH International Conference on Aircraft Hydraulics and Systems, London, UK, 1993

[6] P Krus, A Jansson, J-O Palmberg, K Weddfelt. 'Distributed Simulation of Hydromechanical Systems'. Presented at 'Third Bath International Fluid Power Workshop', Bath, UK 1990.