Interdisciplinary Modelling of Mechatronic Systems by Coupling Simulation Tools

Oliver Lenord, Thomas Huber, Daniel Franitza, Manfred Hiller

***Abstract* – Mechatronic Systems are characterised by interacting subsystems of different types. To model this class of dynamic systems an interdisciplinary approach is required. In this paper a way to model a controlled hydro-mechanical system using three different kinds of simulation tools is presented. The mechanical system, described in MOBILE, is coupled with a DSH*plus* model of the hydraulic system and a controller modelled in MATLAB/SIMULINK. This concept is applied to the oscillating boom of a concrete pump.**

***Keywords* – Interdisciplinary modelling, virtual prototyping, coupling dynamic systems, flexible multibody systems, hydraulic, damping control.**

I. INTRODUCTION

Modelling mechatronic systems always carries the challenge of handling different types of dynamic systems. In the case of large-scale manipulators you will find a hydraulic drive on the actuator side with electronic sensors and controllers. Furthermore the mechanical structure has to be taken into account, as you want to get a good representation of the real system.

For each of the classes of dynamic systems mentioned specialised simulation tools are available. To use the specialist properties of these tools for your modelling task interfaces between the tools are required. A simulation environment for dynamic systems that provides a powerful interface that is supported by many simulation software developers is MATLAB/SIMULINK. So-called S-Functions are used to add dynamic transmission elements to the control scheme. As long as these elements have only one direction of transmission, their use is quite straightforward. In the case of multiple dynamic elements with bidirectional interaction some boundary conditions have to be fulfilled. For our controlled hydro-mechanical system, the mechanical state of the hydraulic cylinder influences the hydraulic pressure. This pressure itself results in an applied force on the mechanical system.

After some remarks on the coupling of dynamic systems in general, the functionality of S-Functions will be discussed. Based on this a concept for how to use S-Functions to model a coupled hydro-mechanical system is presented. Simulation results for the controlled concrete pump boom follow. Finally the conclusions are presented.

Oliver Lenord, Daniel Franitza and Manfred Hiller work in the Mechatronics Laboratory of the Mechanical Engineering Department at the Gerhard-Mercator-Universität Duisburg, Lotharstr. 1, 47057 Duisburg, Germany.
Email: {lenord, franitza, hiller}@mechatronik.uni-duisburg.de

II. COUPLING OF DYNAMIC SYSTEMS

Fig. 1 shows the coupling of dynamic systems based on the kinetostatic transmission elements, as used in the multibody system program library MOBILE [4]. The bidirectional transmission, divided into push forward and pull back operations, is especially important for

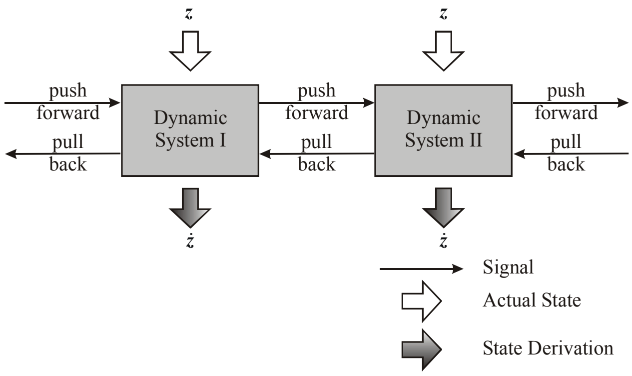

Fig. 1: Bidirectional coupling of dynamic systems

mechanical systems. It allows the transmission of kinematic variables on the position, velocity and acceleration levels from the root of the system to the leaves of the branches. This *push forward* operation leads to the solution for the global kinematics. The following *pull back* operation from the leaves to the root evaluates all resulting forces by taking into account all inertial and external forces.

In other dynamic systems (electric circuits, hydraulic systems) one unidirectional transmission is sufficient to

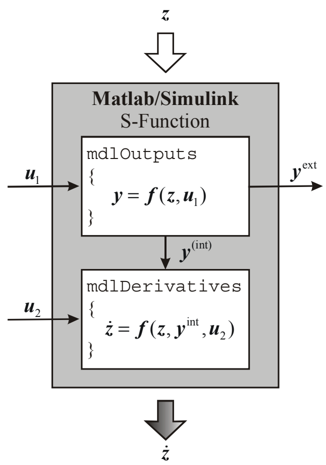

Fig. 2: Transmission element interface in SIMULINK

update all system variables. According to [2], [8], [6] this update process is the *across* operation while the evaluation of the state derivatives is called the *through* operation.

As an analogy to the *across* and *through* operations, MATLAB/SIMULINK provides the functions `mdlOutputs` and `mdlDerivatives` within the transmission element interface (S-Function). However, as shown in Fig. 2 the *across* operation is only unidirectional since it provides only a *push forward* and no *pull back*.

III. INTERDISCIPLINARY MODEL

As described above the S-Function does not exactly correspond to a bidirectional transmission element. Still it is possible under certain conditions to couple a mechanical system with other dynamic systems as will be exemplified by the modelling of a controlled hydro-mechanical system.

The application considered here is a concrete pump boom with four arms with an overall reach of 34 meters. Because of the light structure and the excitation through the pumping process, oscillations with an amplitude of up to one meter can occur. To prevent such dangerous behaviour the *virtual spring damper* control concept was proposed [2] and successfully implemented. This control concept functions in such a way that the third of the four hydraulic drives is used as a linear spring damper. The measured cylinder force and position are used to determine what velocity a linear spring damper would have under the given circumstances. The hydraulic drive is dynamically adjusted to give the corresponding behaviour by means of a velocity controller.

For further studies, and optimisation of the control parameters, the existing MOBILE elastomechanical model of the manipulator had to be extended by adding hydraulic and electronic components. This produced an interdisciplinary model.

The control concept already existed in MATLAB/SIMULINK. The simulation toolbox DSH^{plus} offers a wide variety of hydraulic components to easily model the hydraulic drive, including pressure supply and servo valves. Further more DSH^{plus} provides an export function to produce C code from the model. MOBILE itself is object oriented and is programmed in C++. To take advantage of the graphical user interface in MATLAB/SIMULINK a way was found to implement the interdisciplinary model using the S-Function interface instead of implementing all components as source code in MOBILE.

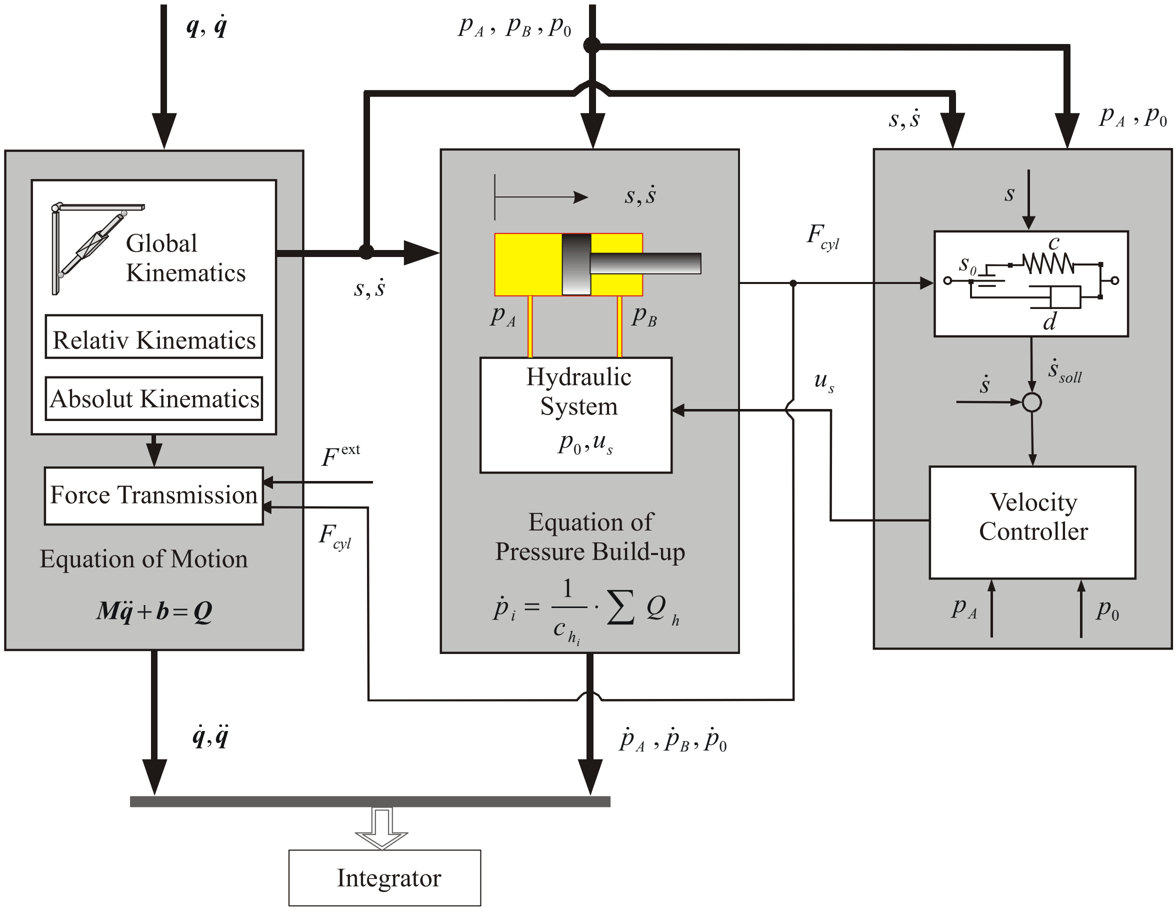

Fig. 3: Signal flow chart for the coupled controlled hydro-mechanical system

The signal flow is shown in Fig. 3, and starts with the actual values of the mechanical state variables $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and the hydraulic states p_0, p_A, p_B on top of the chart. They result from the previous integration step taken by the integration algorithm. Following the signals from left to right, first the forward kinematics of the mechanical system is solved. The updated position and velocity of the hydraulic cylinder, s and $\dot{s}$, can now be processed in the hydraulic system and the controller. The actual pressure in the hydraulic system and the given position and velocity of the cylinder, s and $\dot{s}$, enable evaluation of the resulting cylinder force. The position and velocity of the cylinder and its chamber pressures, p_A and p_B, are used as input values for the controller. The set velocity $\dot{s}_{\mathrm{set}}$ is determined depending on the stiffness c, damping d and the unloaded length s_0 of the virtual linear spring damper. The non-linear velocity controller outputs the control voltage for the servo valve u_s. Since all internal variables of the hydraulic system are now updated the pressure build-up equation can be solved and the derivatives of the hydraulic states be determined. The same is true for the mechanical system after the *pull back* operation for the force transmission has been carried out. The vector of derivatives can now be transmitted to the integration algorithm to perform the next integration step. Each of the operations mentioned is carried out by a function within one or other of the dynamic systems. The flow chart in Fig. 3 shows that there is a certain order that has to be followed. Therefore the relevant functions need to be identified and placed in right order.

All the function calls are listed in order in Table 1. The structure of MOBILE allows the *across* and *through* operations (`doMotion`, `doForce`, `giveYd`) to be called separately. This is not the case for DSH^{plus}. The exported C-code provides only one function (`dgl`) that calculates the *across* variables ($\tilde{Q}_i$, F_i) and the *through* variables ($\dot{p}_i$). Therefore the `dgl` function has to be called twice. The draw back of superfluous operations can be eliminated through an additional case sensitive exception of certain function calls.

The implementation in MATLAB/SIMULINK can be performed as follows. While the controller is represented by a number of static transmission elements from the SIMULINK library, the mechanical and hydraulic systems are each added as dynamic S-Function elements (Fig. 2) to the MATLAB/SIMULINK scheme. In the next step the

	Subsystem operation	**Calculation Performed**	**Function Call** MOBILE / DSH^{plus}
1	**Mechanical** push forward	Forward Kinematics, $s = f(\boldsymbol{q},\dot{\boldsymbol{q}})$; $\dot{s} = f(\boldsymbol{q},\dot{\boldsymbol{q}})$	MOBILE::`doMotion()`
2	**Hydraulic** across	Flow Rates, $\tilde{Q}_i = f(\Delta p_i)$ Cylinder Force, $F_{cyl} = f(\Delta p_{cyl})$	DSH^{plus}::`dgl(…)`
3	**Control** across	Control Voltage, $u_s = f(s, \dot{s}, p_0, p_A)$	
4	**Mechanical** pull back	Inverse Dynamic, $\boldsymbol{Q} = \boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{F}^{\mathrm{ext}})$	MOBILE::`doForce()`
5	**Mechanical** through	Equations of Motion state derivatives $\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{Q} - \boldsymbol{b}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{F}^{\mathrm{ext}}))$	MOBILE::`giveYd(…)`
6	**Hydraulic** through	Pressure Build-up Equation state derivatives $\dot{p}_i = {}^{1}\!/_{c_{h_i}} \cdot \sum \tilde{Q}$	DSH^{plus}::`dgl(…)`

Table 1: Order of function calls

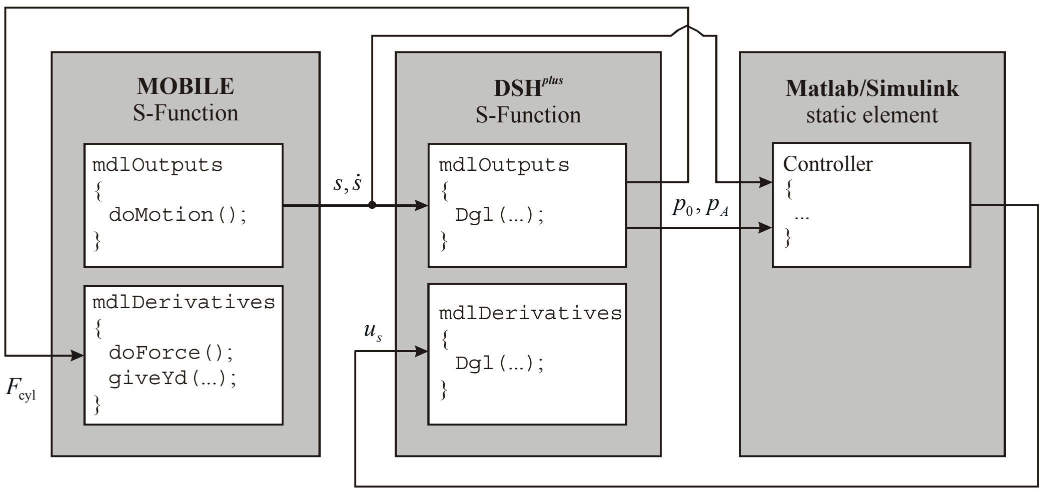

Fig. 4: Linking and function calls of the transmission elements in MATLAB/SIMULINK

inputs and outputs for each S-Function have to be set, the linkages between the elements added and the *across* and *through* functions called according to Table 1. Fig. 4 gives an overview of how the S-Functions are arranged.

After integration MATLAB/SIMULINK updates all output variables and calls the static transmission elements and the `mdlOutputs` functions in the S-Function elements. The order of execution follows the links from sources to sinks. In this case it starts with the MOBILE `mdlOutputs` function, continues with the DSH^{plus} `mdlOutputs` and ends up in the controller. After this all output variables are updated. The `mdlDerivatives` functions of the S-Function elements are called in arbitrary order to determine the state derivatives. Since the *pull back* operation `doForce` of the mechanical system has not yet been performed, it is called right before the equations of motion are built in the `giveYd` function. A `giveYd` call in the `mdlOutputs` function would lead to an unsolvable closed loop.

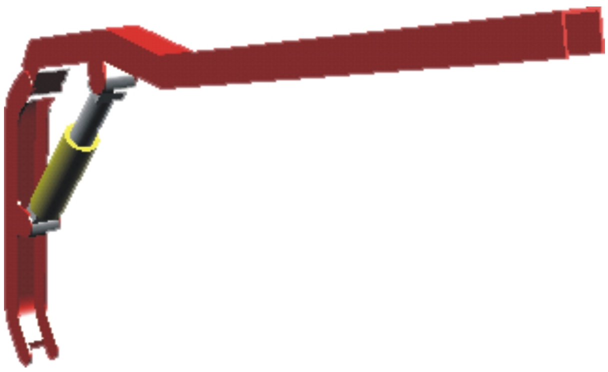

Fig. 5: Simple controlled elastic hydro-mechanical system

IV. SIMULATION RESULTS

For initial study and investigation of efficient optimisation strategies a simple manipulator was modelled, as shown in Fig. 5. The mechanism, which has one closed loop, is modelled in MOBILE, using the method of the characteristic pair of joints to retrieve an explicit solution for the constraint equations. The elastic arm is modelled using jointed rigid links with two degrees of freedom. The stiffness of the arm was chosen to be very low to clearly show the damping effect of the *virtual spring damper* controller. The linear hydraulic drive was modelled as a prismatic joint with friction. The friction law is formulated as an approximated Stribeck curve with a steady zero crossing. The dynamics of the hydraulic drive, including the pressure supply, servo valve, pressure relief valves and differential cylinder, are described in the separate DSH^{plus} model. As already mentioned the system is controlled using the *virtual spring damper* control concept. The parameters of the controller are basically the virtual stiffness and damping.

Optimisation of this interdisciplinary system is very time consuming if the optimisation criterion is formulated as an integral in the time domain, such as with the integrated time multiplied quadratic error (ITE^2) method. To handle complex interdisciplinary models a linearisation method was developed [3] and used to find parameters for optimal damping of the given manipulator. This is about a hundred times faster than by evaluating the ITE^2 criterion, and has only a minimal reduction in quality.

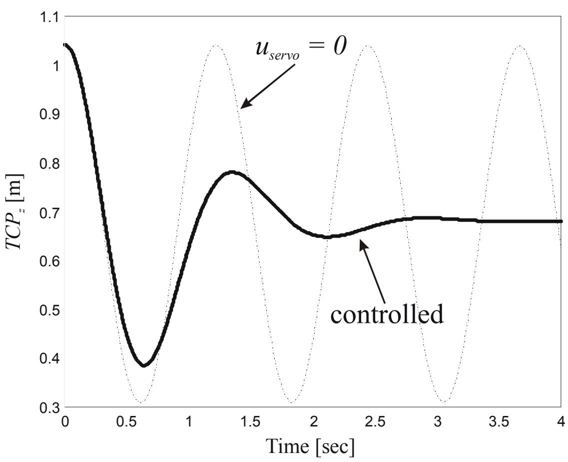

Fig. 6: Vertical end-effector position of the simple model in the controlled and uncontrolled case

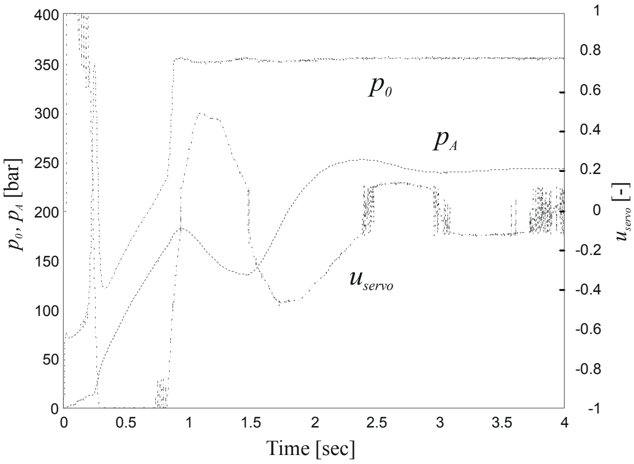

Fig. 7: Run of hydraulic pressure and control voltage for the controlled manipulator

The simulation results in Fig. 6 show a comparison between the controlled and uncontrolled system behaviour. In both cases the initial condition of the elastic arm is unbend without any load and the whole system in rest. In the uncontrolled case the servo valve is locked. The curves of the control voltage and the hydraulic pressure for the controlled case are shown in Fig. 7.

In the uncontrolled case the system oscillates with very low damping and large amplitude around static equilibrium. In contrast to this the amplitude of oscillation decreases rapidly in the controlled case.

In addition to the simple manipulator, a complex interdisciplinary model of a concrete pump boom was developed [3]. The mechanical structure is shown in Fig. 8. There are two closed loops in the transmission linkages of joint B, C and D. Each elastic arm can be

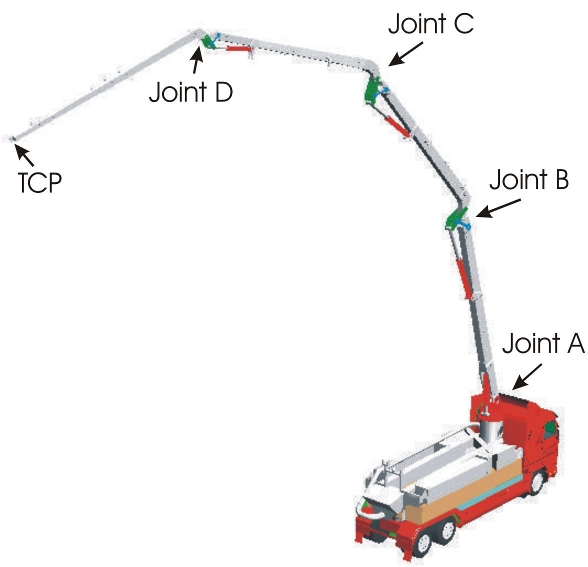

Fig. 8: Structure of the Schwing™ KVM34x mobile concrete pump

described by six planar elastic degrees of freedom, but the elastic deformation has not been taken into account in the following simulations (this will be considered in

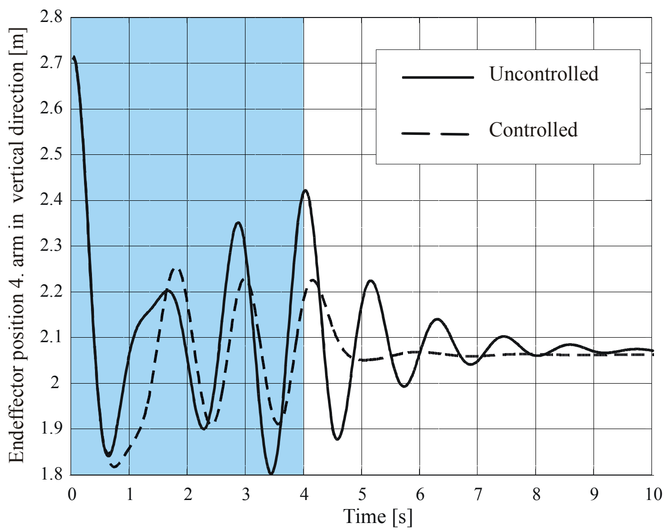

Fig. 9: Excitation of the concrete pump boom in the controlled and uncontrolled case

future work). The control concept provides only the third drive in joint C with damping capabilities. Therefore only this one is modelled as a controlled hydraulic drive in the same way as the simple manipulator. The stiffness of the other three hydraulic drives is approximated using a linear spring-damper approach, whose parameters depend on the enclosed oil volume in the cylinders. The optimisation of the control parameters has not yet been carried out for the concrete pump, but some good results have been still achieved in the simulation as documented in Fig. 9. In the cases displayed the system is excited for four seconds by an oscillating external load. In the uncontrolled case an increasing end-effector oscillation is built up and slowly decreases after the external load is removed. When the controller is activated the oscillation is limited to a reasonable level and is completely absorbed shortly after the excitation stops.

V. Conclusion

In this paper the coupling of dynamic interdisciplinary systems was discussed. As an example, a controlled hydro-mechanical system was implemented in the well-known simulation environment MATLAB/SIMULINK. To do this a bidirectional transmission element was implemented without violating the required order of execution. This way each of the three coupled dynamic systems could be modelled using specialized tools and these could later be linked together using the S-Function interface.
The simulation results demonstrate the functionality of the interdisciplinary approach. The damping control concept for oscillating manipulators was successfully adapted to the model of a simple three degree of freedom manipulator as well as to the complex model of a concrete pump boom.

Acknowledgements

The authors would like to express their thanks to the Deutsche Forschungsgemeinschaft (DFG) who funded SFB291 Project A1 on the Modelling of Large Scale Manipulators, and to the company Schwing GmbH who provided much practical assistance.

References

[1] Anantharaman, M., Fink, B., Hiller, M. und Vogel, S. An *Integrated Development Environment for Mechatronic Systems*. In J. Lückel (Ed.): Proceedings of the 3rd Conference on Mechatronics and Robotics (pp. 54-69). Paderborn, Germany, 4th. - 6th. October 1995.

[2] Cellier, F. E. *Continous System Modeling*. Springer-Verlag: New York, Berlin, Heidelberg, London, Paris, Tokyo, 1991.

[3] Huber, T. *Interdisziplinäre Modellbildung eines geregelten hydromechanischen System*, Diplomarbeit am Fachgebiet Mechatronik, Gerhard-Mercator-Universität Duisburg, 2001.

[4] Kecskeméthy, A. MOBILE - *An object-oriented tool-set for the efficient modeling of mechatronic systems*. In Hiller, M. Fink.B., Proceedings of the 2nd Conference on Mechatronics and Robotics, September 27-29, Moers, Germany, pages 447-462. Heinig & Müller, Mülheim a.d.R 1993.

[5] Lenord, O., Fang, S., Huber, T., Franitza, D., Hiller, M. *Multi-criteria Optimisation of the System Dynamics Based on Interdisciplinary Modelling*, presented at the Euromech Colloquium No. 427 on Computational Techniques and Applications in Nonlinear Dynamics of Structures and Multibody Systems, Cachan, France, 24-27 September 2001.

[6] J. Müller, M. Hiller *A Mechatronic Simulation Model for the Large-Scale Hydraulically Driven ALDURO*. Held at the International Workshop on "Computer Software for Design, Analysis and Control of Fluid Power Systems"; Trondheim, Norway, 1999.

[7] Nissing, D., Bernzen, W. and Schwarz, H. *On Vibration Control of a Concrete Pump*, F1058-2. Karlsruhe, 1999.

[8] Schneider, M. *Modellbildung, Simulation und nichtlineare Regelung elastischer, hydraulisch angetriebener Großmanipulatoren*, Fortschritt-Berichte, Reihe 8, Nr. 756, VDI-Verlag, Düsseldorf, 1999.