PROGRESS IN INTEGRATED SYSTEM ANALYSIS AND DESIGN SOFTWARE FOR CONTROLLED VEHICLES

W. Kortüm* and W. Schiehlen** and M. Hoffmann***

* DLR-Institute for Robotics and System Dynamics, Wessling
** Institute B for Mechanics University of Stuttgart
*** TEDAS Mechanical Systems Software GmbH, Marburg

SUMMARY

This paper represents one of the results of the second Herbertov Workshop in 1992 on "Research Issues in Automotive Integrated Chassis Control Systems". It was decided to bring together engineers who are actively involved in current software developments to report on the progress in integrated system analysis and design software for controlled vehicles. Therefore, after mentioning some of the principal requirements for such software, the state-of-the-art is summarized. Promising recent research and software developments are described and the application of some of these concepts to an actively controlled vehicle with preview is discussed. Throughout the article the ideas of *concurrent analysis and design strategies* are emphasized.

1. INTRODUCTION

For the computer assisted analysis of the dynamical behaviour of mechanical systems such as road and rail vehicles the *multibody approach* has become well established within the last two decades. Indicators are not only the number of research papers given, for instance at the preceding IAVSD symposia, but in particular the widespread use of *multibody software* by vehicle analysists in industry and in research organizations. As a service to the vehicle dynamics community, IAVSD organized a first workshop in Herbertov devoted to describe and summarize the existing vehicle–oriented multibody software and especially define some realistic vehicle benchmarks on which the programs could demonstrate their capabilities. The final document is now published as a special issue of the Vehicle System Dynamics Journal, [1], and will be available at the 13th IAVSD Symposium. Also, in the Multibody System Handbook, [2], many of the available multibody programs are described.

Multibody system (MBS) software has proved to be very effective for vehicle system dynamics especially since the development cycle for new vehicles has to be shortened, leading also to smaller turnaround requirements for simulations, and when the analysts are asked for rapid "trouble–shooting" advices. However, it remains a matter of fact that most of these programs had been originally developed for investigating well specified vehicles. In other words, the programs were mainly designated for system *analysis* rather than system *design*; the latter could only be performed to some extent through extensive parameter variation. Even parameter studies were indeed rather tedious for some MBS programs. Those MBS codes, which generate symbolical expressions for the equations of motion as for instance NEWEUL have an advantage for such studies as they maintain the mechanical parameters even in the final equations.

The design issue has been stressed even more through the fact that passive vehicle suspensions using conventional springs and dampers have been driven to their limits with respect to ride comfort and safety. On the other hand, suspensions with *active components* may offer significant advantages over purely passive systems, [3, 4]. This aspect of electronically controlled systems has become popular also for steering, braking (ABS), accelerating (ASC) etc. that one can speak today of *chassis control systems*. A recent conference in Japan was devoted entirely to this topic of Advanced VEhicle Control (AVEC), [5]. It should be noted that the automotive industry, predominately in Japan, seems to be the leader in this domain, however, as pointed out already in [6], the traditionally more conservative railroad industry is now seriously developing active feedback controlled components (car body steering, active suspensions, steered bogies etc.), [7].

In contrast to the complexity of the feedback control schemes, however, the mechanical models used for design remained pretty poor, i. e. mostly quarter-car models were used, in some cases the designers were advancing to half-car models. One major reason was the missing link between the multibody and the control design software, partially due to the gap between the mechanical and the control engineers. Advanced vehicle systems today rely both on mechanics *and* electronics expressed by the increasing popularity of the term *"mechatronics"*. Mechatronic systems call for an *integrated* process of modelling, simulation and control system design based on realistic models because of the unavoidable interaction of the various components. Such methods must be part of a *Concurrent Engineering* process defined as an approach for designing, manufacturing and its quality control, all at the same time, Fig. 1. Concurrent Engineering is superior to the traditional sequential engineering both with respect to time and costs involved but also with respect to the capabilities of the final product. In our context concurrent means, that a control loop is not added onto an existing passive device but designed *together* with the mechanical system from the outset.

2. REQUIREMENTS FOR MECHATRONIC SYSTEM DYNAMICS SOFTWARE

As pointed out in the previous section the existing software packages have some major drawbacks as the multibody packages were mainly designed for the analysis of complex passive mechanical systems whereas the corresponding control-oriented packages mainly of the MATLAB-type such as PRO-MATLAB and MATRIX_X – for a survey, see [8] – lack modelling and simulation capabilities. Moreover, their control system design strategies were, more or less, based on synthesis methods for linear systems such as pole-assignment, RICCATI design, linear observers and KALMAN filters. The also quite popular block-oriented simulation languages like ACSL, [9], were concentrating on time-simulation of nonlinear models but again have no modelling aid for mechanical systems; i. e. the user has to establish and

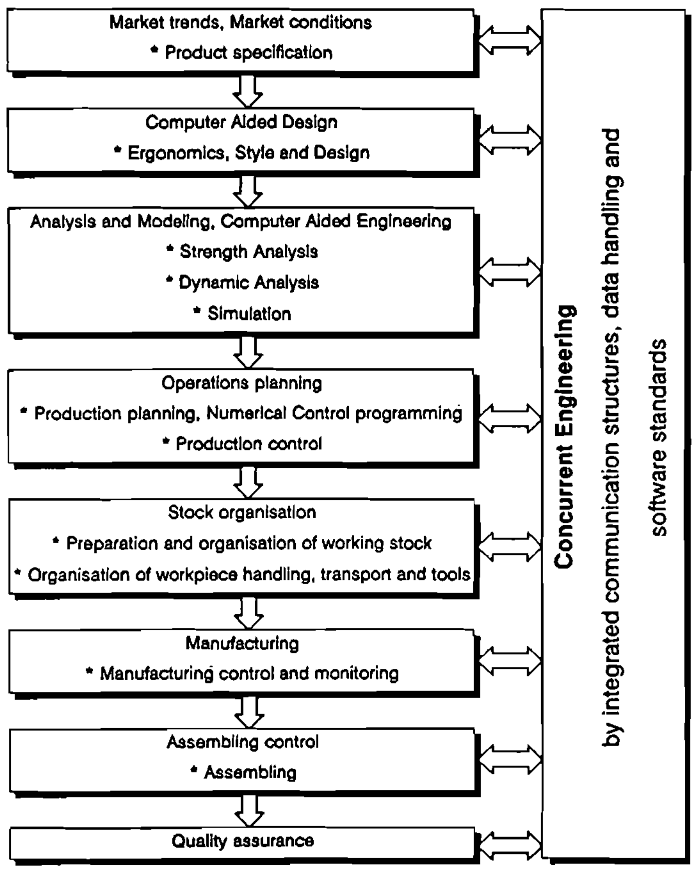

Fig. 1 The Main Ideas of Concurrent Engineering.

provide the code for the equations of motion, [10].

For setting a better perspective it may be worthwhile to formulate independent of the capabilities of the existing software the major modelling, dynamical analysis and design features needed for mechatronic vehicle systems, see Fig. 2,

- modelling as a multibody system with structural flexibility and actuators for control;
- establishing the nonlinear equations of motion symbolically *and* numerically;
- analyzing the equations of motion, e. g. simulation via numerical integration and linearization of the system equations;
- design studies, e. g. selecting control strategies, parameter studies and selecting optimal design parameters;

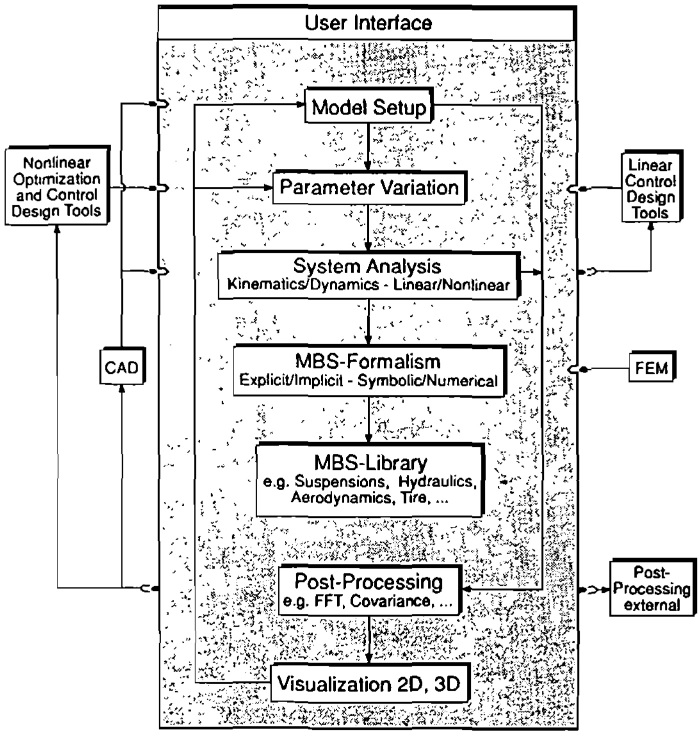

Fig. 2 Major Modelling, Dynamic Analysis and Design Features for Mechatronic Vehicle Systems, Example: SIMPACK

- evaluation of the dynamical performance and visualization of the results, e. g. animation;
- open system for an interdisciplinary use together with CAD–, FEM– and control design–tools.

A few comments may illustrate the needs especially if the reader compares the desirable features with the presently available software:

1. Even if it is agreed that the "core model" of a vehicle can be described by a set of multibody system elements such as rigid bodies, bearings, joints supports, mechanical springs and dampers, there are many more modelling items to be incorporated for an advanced *mechatronic* vehicle, as for instance hydraulic, pneumatic, electric or magnetic actuators, electronics, control logic, propulsion systems, tyres etc. Not forgetting that modern vehicle designs strive towards lightweight constructions; thus, flexible body modes have to be taken into account. Fig. 3 indicates some of the modelling items necessary for realistic modelling of automotive vehicles.

   Some of the required data can be obtained from CAD systems, e. g. the multibody geometrical data, the weight, center of gravity and moments of inertia. For the data of flexible bodies, usually in form of modal data (mass–, damping– and stiffness–matrices, modal shape functions), Finite Element Modelling (FEM) programs can be used.

However, the modelling issues are far from being satisfactorily solved, e. g. many of the existing modellers are *closed systems*, i. e. their models cannot be exported into an integrated modelling concept. Even more general concepts such as the *Bond Graph* idea, [11], are far from being professional and user friendly.

2. If the major modelling elements are of the multibody type there exist a number of well established formalisms, see [1, 2]. However, many of them lack important modelling facets, for instance ADAMS, perhaps the most widely used MBS–software worldwide in industry, and NEWEUL lack real elastic modelling, and most of the more professionally applied tools like ADAMS (automotive industry) and MEDYNA, the leading MBS–program in the railway

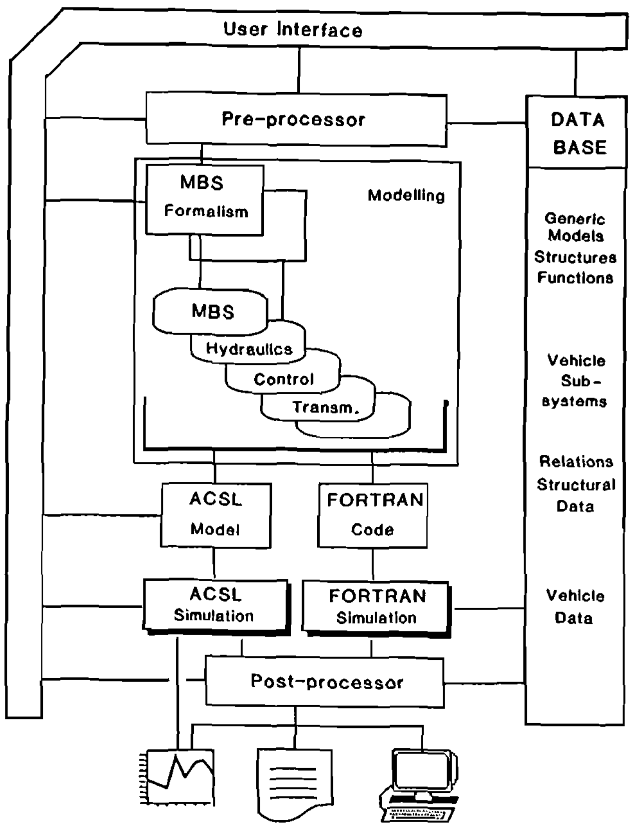

Fig. 3 Modelling and Simulation Requests for Automobile Industry (Source: Audi AG).

field are closed systems, i. e. their models cannot be exported into other modelling environments. It would be desirable to establish a ***standardization*** for defining MBS data in order to use the most appropriate multibody formalism for a given vehicle dynamics problem, see below.

3. Simulation is often defined as performing *experiments* with the mathematical model. Therefore, analyzing nonlinear system equations was, for a long time, thought of as solving these equations for specified parameters, given inputs and specified initial conditions via numerical integration. However, desirable multibody system analysis, Fig. 4, features static analysis (equilibrium states and loads), kinematic analysis (system assembly, computation of consistent initial conditions, motion of the system neglecting dynamic properties) and linearization producing the state–space system matrices for linear system analysis.

   Further analysis items are stochastic analysis in the time– or frequency domain for linearized models and Monte Carlo simulation for nonlinear models. Other desirable *"experiments"* may be sensitivity analysis, parameter variation, stability analysis and perhaps the determination of periodic motions (limit cycles).

4. Because of the feedback control loops in mechatronic vehicles, the issue of their design has become more and more important recently. Since many of the control system synthesis procedures are traditionally based on a linear system representation of the type

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \,, \tag{1}$$

$$\underline{y} = C\underline{x} + D\underline{u} \,, \tag{2}$$

   a linearized version of the nonlinear equations of motion is the starting point for

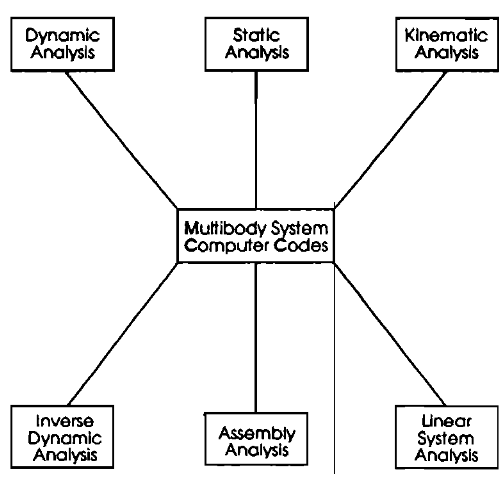

Fig. 4 Major Analysis Features Required for Multibody Systems.

these type of control design. Further features such as testing the observability, controllability, designing observers, KALMAN filters, feedback gains for pole-assignment or quadratic synthesis are needed. Most of these features are today readily available in software packages built upon MATLAB, e. g. $MATRIX_X$, PRO-MATLAB etc. One major drawback of the linear design strategies, however, is that the inherent nonlinearities frequently encountered in vehicle dynamics are not taken into account and that the plant parameters in this design approach, i. e. the matrices A, B, C, D have to be specified from the outset. This is a serial process, resulting in more cost and time than concurrent design. These requests automatically lead to what we may call *simulation based design of vehicle system*, see [12]. Major efforts have been undertaken in this area, e. g. [13, 14, 15]. A very promising effort in this field is near completion for general release at DLR under the tempting name ANDECS (= Analysis & Design of Controlled Systems), [16, 17], see also Sec. 4.

5. An important item which has not yet been addressed specifically is the *software engineering* aspect. It became clear that closed software packages have restricted use and a degree of flexibility is gained through *open software structures*, which can be combined to make use of their different functionalities much more easily. Along with developing open systems naturally came along the requests for *standardization* of *software interfaces* as well as for the standardization of *data structures*. As a new term in this context, *object-oriented data* was introduced for the fact that for instance the necessary model data for a multibody vehicle can be formulated regarding the multibody model as an object and *not* relating to the specific data of the multibody *formalism* which may be applied later, see [18].

3. THE STATE-OF-THE-ART IN ANALYSIS & DESIGN SOFTWARE

In the following section we intend to describe briefly the presently *available* software in the light of the preceding requirements. This very short survey is by no means exhaustive, it is needed to give the reader an impression on the present situation characterized through some leading software with respect to efforts on integrating the analysis and design process.

First of all, we have to acknowledge that the *market* we are dealing with is – as usual – dominated by some well-established products on which the efforts of the alternatives especially the *newcomers* have to be measured; in some cases the established products set a *quasi-standard* in the corresponding field. A new product may fail if these market-driven laws are not taken seriously into account (an example may be the long life-time of FORTRAN versus newcomers like ALGOL etc.). In our present discussion, we have to deal with the following leading products:

- ADAMS, [1, 2] for multibody system simulation;
- MATLAB/$MATRIX_X$, [8], for control system analysis and design;
- ACSL, [9], for block-oriented dynamic system simulation.

These well-established packages either were developed with respect to existing standards (CSSL-standard for ACSL) or have, more or less, created some *quasi-standard* as the MATLAB-type of linear system description and the multibody system definition used for ADAMS.

Within the automotive industry the software package ADAMS, which is the acronym for Automatic Dynamic Analysis of Mechanical Systems, has become widespread for simulating the kinematics and dynamics of an automobile. The systems simulated with ADAMS include full vehicle models with suspension linkages, anti-roll bars, struts, steering mechanisms, brakes, drive-trains, engines, tyres, cam and valve mechanisms. Electronic control systems can also be included into the ADAMS model. However, the user has to implement the control algorithms into a user-written subroutine which is linked to ADAMS. Tools concerning control design, like the determination of optimal feedback gains, are not offered within this software package.

These features are available in control system analysis and design tool-boxes built upon MATLAB. As the most complete representative, the $MATRIX_X$ family offers full support for the complete dynamic system design process starting from system analysis via control design up to prototype testing. One of the newer additions to $MATRIX_X$ is a graphical modelling system, called System Build, for building block diagrams, which are similar to data flow charts. The graphical representation of system and control algorithms, serves as the basis for documentation, system analysis, control design, simulation and automatic real-time code generation, which then can be tested with the controller AC100.

Relying solely on $MATRIX_X$ for the development of vehicle control systems means, however, in contrast to using a multibody software as ADAMS, that the analyst would have to generate the complex equations of motion for the mechanical system by himself. A model export from ADAMS into $MATRIX_X$ is not possible, because of ADAMS being a closed system. There have been efforts to overcome these handicaps and the present state is that ADAMS and $MATRIX_X$ can interchange linear

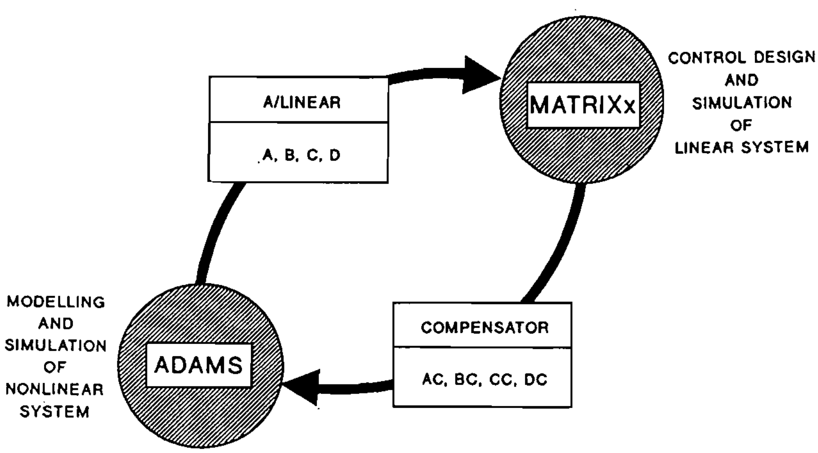

Fig. 5 Interchange of Linear Plant and Controller Models Between ADAMS and $MATRIX_X$.

plant and controller models respectively, Fig. 5, [19]. Here, the nonlinear mechanical system including sensor and actuator dynamics are modelled inside ADAMS. Using the module ADAMS/Linear this system is linearized at a certain operating point and transferred to MATRIX_X in state space matrix form. Control design and simulation of the linear closed loop system is then performed within MATRIX_X. Afterwards, the control laws – whether an ideal linear state–feedback control or more sophisticated estimators and compensators – will be imported back into ADAMS. There the nonlinear closed loop system may be simulated and the control performance on the realistic model is evaluated.

With ADAMS version 6.0 all the necessary data exchange is done automatically for linear continuous control laws. For linking discrete, nonlinear controllers, e. g. being developed by MATRIX_X, to ADAMS the following detour is presently advised: Auto Code of MATRIX_X may be used to translate the graphical description of the control algorithms into a FORTRAN subroutine; this subroutine will be embedded into ADAMS. With this nonlinear interface the closed loop performance of mechanical systems with a complex nonlinear and/or discontinuous controller may be evaluated through ADAMS simulations.

In summary, it must be recognized that the connections between some of the major codes remained unsatisfactory. The main stumbling block is that nonlinear models could not be exported into other analysis and design environments. Such an model export can be most easily done with a *symbolic* multibody formalism generating analytical expressions for the mechanical system equations of motion. Examples are NEWEUL, MESA VERDE, SD–FAST and more recently AUTOSIM, see [1, 2]. Some of these formalisms did originally not even supply simulation capabilities, i. e. solvers to integrate the equations; they were intended to be linked to special simulation software and many of them were applied in combination with ACSL. However, there have been encountered two major drawbacks of these combinations:

- Some of the symbolic codes have problems with real multibody systems with many degrees–of–freedom if all expressions are completely evaluated.

- The numerical integration codes of external packages like ACSL were not prepared for the specific requirements of multibody system equations; among other flaws, implicit differential equations or differential–algebraic equations are not allowed.

SIMPACK, see [1, 2], uses another approach: it is based on an efficient essentially *numeric* $0(N)$–formalism and has proven its effectiveness even for large multibody problems. Designed in structured functional modules it is *open* for intrinsic interfaces with external software tools, see Fig. 2. SIMPACK can be linked easily as a fully nonlinear block into simulation codes like ACSL or simulation and control design environments like MATRIX_X, MATLAB and ANDECS. Practical experience has been achieved so far with MATRIX_X and ANDECS, [14].

In addition to generating numerically the equations of motion, SIMPACK can also establish *symbolic code*. A FORTRAN file can be created representing the condensed equations of motion for a specific multibody system. The advantages of such a symbolic code are twofold: first a very significant saving in simulation time and second the possibility to extract the FORTRAN code and implement it on any platform independent from the source code, e. g. on an onboard computer of a vehicle for the purposes of real-time applications such as hardware–in–the–loop simulations. Condensed equations of motion are also available by NEWEUL.

Very recently also an ADAMS subroutine interface has been developed. This new interface, available with version 7.0, enables the incorporation of nonlinear ADAMS models into other simulation environments; e. g. one block in $MATRIX_X$/System Build can represent a nonlinear multibody model of ADAMS.

Let us finally comment on the third standard package in the field of modelling, simulation and control software, namely ACSL; this is the acronym for Advanced Continuous Simulation Language and probably the most widely used continuous system simulation language (CSSL–standard). ACSL has a number of attractive features: modelling aids as attractive features: modelling aids as nonlinearities, transfer functions etc.; it can handle state–dependent discontinuities as well as mixed discrete–continuous systems which are for instance required to simulate digital control of continuous plants. However, a number of shortcomings became apparent over the years which have not been solved satisfactorily yet:

- There is for complex mechanical systems no model generation directly available, i. e. the equations of motion have to be established by the user in state–space form. For this reason it became popular to put a multibody formalism in front of ACSL as it was for instance proposed by some automobile companies, see Fig. 3. However, this proposal has some other significant draw–backs.

- The numerical integration schemes implemented in ACSL have not been kept up–to–date and the introduction of user–supplied integration routines is problematic. Furthermore, implicit differential equations and differential–algebraic equations cannot be handled at present, i. e. the integration routines are not well suited for multibody system equations, especially in the situation of closed loops and stiff systems, as they appear frequently in vehicle system dynamics.

- ACSL is to narrowly focussed on the numerical integration of initial value problems for ordinary differential equations; e. g. linearization features been added but these work only for the submodels which are generated by ACSL.

As a summary, ACSL is another closed system which has a number of useful features but in the context of the requirements posed a major handicap is that its models cannot be exported into other environments where they may be more effectively solved, used for control design or parameter optimization etc.

4. CURRENT RESEARCH AND SOFTWARE DEVELOPMENTS

Needless to say that ongoing research and software development can only be described if it occurs within the limelight of the authors. For this reason we essentially concentrate on reporting two open German activities which are not only directly related to the main topic of this contribution, i. e. modelling, simulation and control of complex mechanical systems but also have a number of cross–coupling relations as shall soon become apparent.

The two activities are:

1. The Multibody System Research Project, of the German Research Foundation (DFG) which has been intermediately reported at IAVSD Lyon, [18]; it was recently completed with an international symposium and its proceedings are readily available, [20].

2. The concurrent control engineering project ANDECS, [16], i. e. concurrent engineering here means multi-disciplinary dynamics design integration with emphasis on controlled systems.

Both projects have created a vast-number of results in software engineering technology of mechatronic systems; they have influenced each other significantly and yielded some concepts, which at least point into the direction of the requirements put together in Sec. 3. Even, if not all the facets will survive the roughness of the real world, we are convinced that the main conceptual ideas point the way ahead.

As was mentioned already, [14], all three software categories described in Sec. 3 have been designed with specific forms of *shortsightedness*: the multibody system packages mainly had in mind dynamic analysis, i. e. simulation, of (purely) mechanical systems; the control system packages of the MATLAB-type had the dominant view of analysis and design of linear state-space control systems, whereas the continuous simulation languages were focussed on the time-integration of explicit first-order nonlinear differential equations defined by block-oriented methods.

Most of the developers realizing their shortcomings tried to overcome it by extending their model, method and handling restrictions. As an example all MATLAB-type packages, being originally restricted to linear algebra, have now modelling and simulation environments for nonlinear systems, e. g. SystemBuild for $MATRIX_X$ and SIMULINK for MATLAB.

Nevertheless, none of the packages will ever be complete as far as present and future engineering requirements are concerned. Hence the idea of linking together or even integrating various packages came into being; recall the discussion of Sec. 3. Some of the proposals are quite stable while others collapsed. A major problem for that is, that in most case integration was attempted by a high-level user interface and supervisor program on top of various stand-alone proprietary software packages, while those packages themselves did not adhere to any common standards – since there were none.

Both at DLR within the ANDECS project and within the DFG project an alternative view was exemplary developed: to define a low-level neutral data definition for multibody dynamics model generation and to use a neutral dynamics *model-bus* for all types of computer-assisted dynamics engineering experiments. These are the basic ingredients for an open computational-data/model exchange, i. e. for data/model import *and* export, Fig. 6.

Within the DFG-project and in cooperation with the DLR ANDECS-project a prototype for an object-oriented multibody data definition has been developed and both projects have chosen the database RSYST for its implementation, [20]. Since the multibody system datamodel has been described at the Lyon symposium, see [18], it is not repeated here.

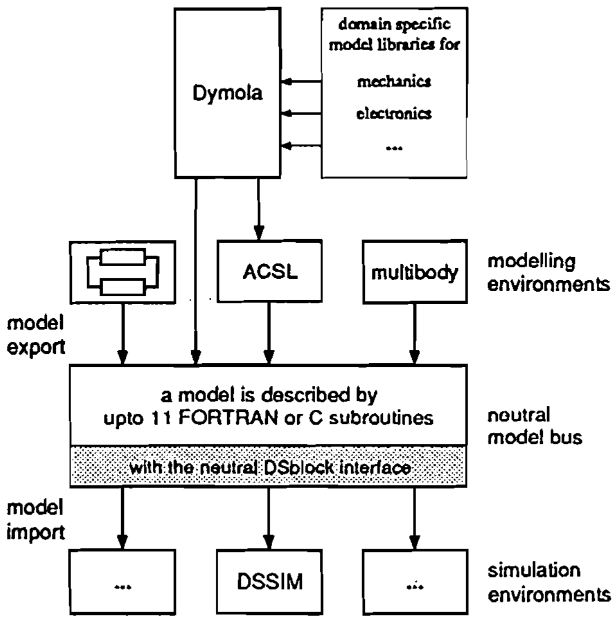

Fig. 6 Separation of Modelling and Simulation in ANDECS

The goal of this concept was to combine different multibody formalisms with different simulation programs via standard interfaces; these potentials were demonstrated, [20], for NEWEUL and SIMPACK, see Fig. 7. This figure additionally shows the intimate connections between the two projects.

As pointed out before, vehicle dynamics engineering requires dynamics analysis *and* design experiments. Whereas the multibody and vehicle system dynamics analysis features have been frequently discussed, e. g. [1], design variations have been addressed to a much lesser extent. However, advanced concepts and active mechatronic subsystems in vehicles certainly require advanced computer-aided design strategies.

Within ANDECS a conceptual framework for analysis and design was developed, [17], based on multi-criteria parameter optimization. Its performance criteria can be based on simulation results of the nonlinear system. It is possible to find the best parameters of a feedback control or the best synthesis parameters of a mechanical system or both at the same time, i. e. *concurrently*, see Fig. 8a, b.

5. OPTIMIZATION OF AN ACTIVELY CONTROLLED VEHICLE SYSTEM

As an application, which demonstrates the effectiveness of the concepts being discussed so far, we have selected one of the major IAVSD benchmark problems, the Bombardier ILTIS, see [1]. As in [21] an active suspension with preview control is envisaged. The design and optimization of active vehicle suspensions is still today

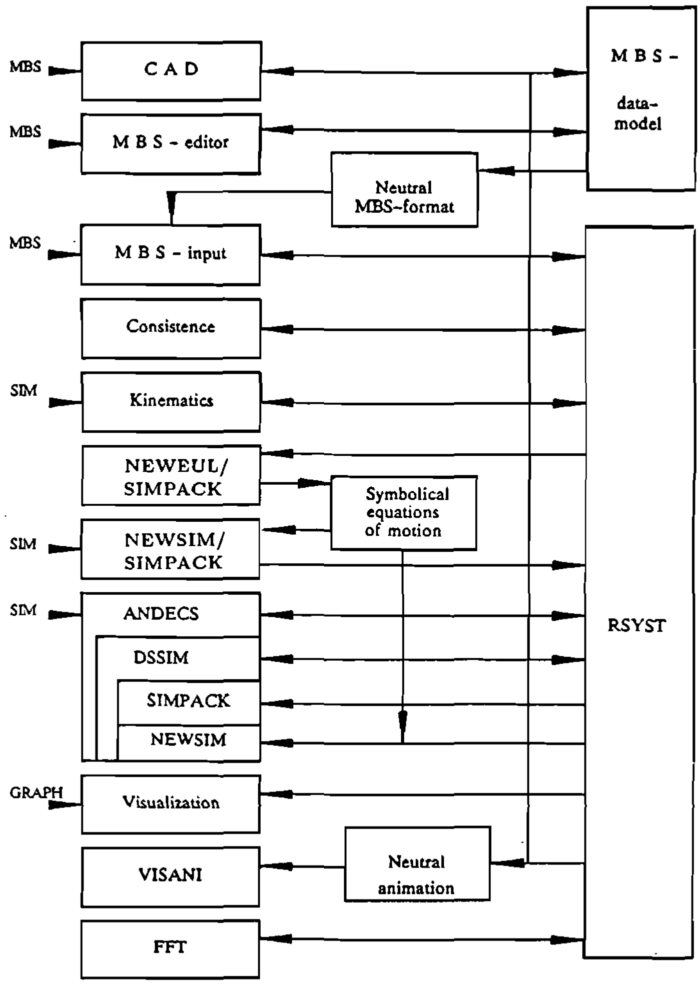

Fig. 7 NEWEUL and SIMPACK within the DFG-Project

mostly exercised on very simple mathematical models even in connection with the availability of multibody software for simulation. A typical procedure is to develop the control scheme and its parameters based on simplified design models of the type "quarter-car" (2 DOF) and then to simulate the full spatial vehicle being equipped with the so-designed feedback, [22, 23]. However, since optimized designs based on drastically reduced models applied to a full vehicle model, may reveal significant less than optimal behaviour, great effort is presently undertaken to allow for a design being based on more realistic simulation models, [12, 14, 13]. In order to perform such design procedures efficiently, the full nonlinear multibody simulation models must be made available with *free synthesis parameters* within the optimization and design software. The open structure of NEWEUL and the symbolic code generation of SIMPACK support such efforts significantly and software environments have been

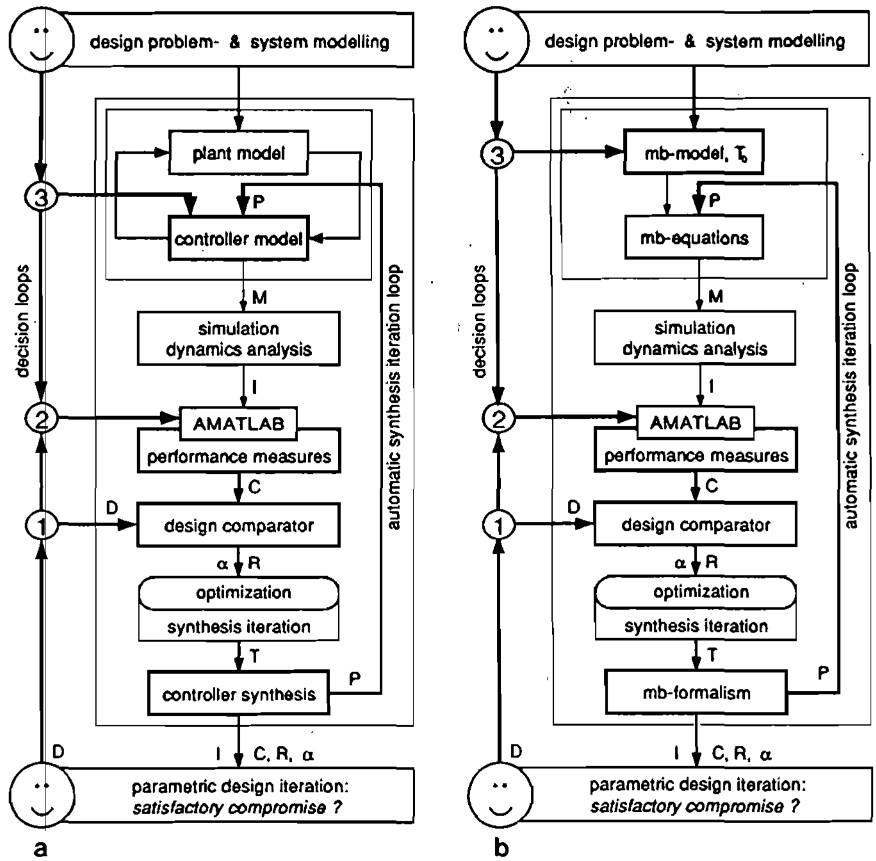

Fig. 8 Concurrent Design of Controller Synthesis and Mechanical Systems with ANDECS

implemented at DLR and at the University of Stuttgart, [13, 17], which allow the parameter optimization of nonlinear multi-degree-of-freedom multibody systems.

The following results are obtained using SIMPACK *and* the multi-objective programming system MOPS of ANDECS, [17], see Fig. 9, as described in [24]. Here several – often conflicting – performance criteria can be minimized through multicriteria optimization theory to find *pareto-optimal* solutions.

The Conventional Vehicle (IAVSD Benchmark)

Fig. 10 shows the side view of the vehicle; details of the suspension arrangements – these are identical in the front and rear – are given in Fig. 11. In addition to the nonlinear force laws of the leaf spring and the shock absorbers, the nonlinear kinematics of the suspension (similar to a McPherson strut) involves two kinematically closed-loops. The also nonlinear tire-model is based on the Calspan model. The full passive vehicle involves ten degrees-of-freedom (6 DOF for the car body plus

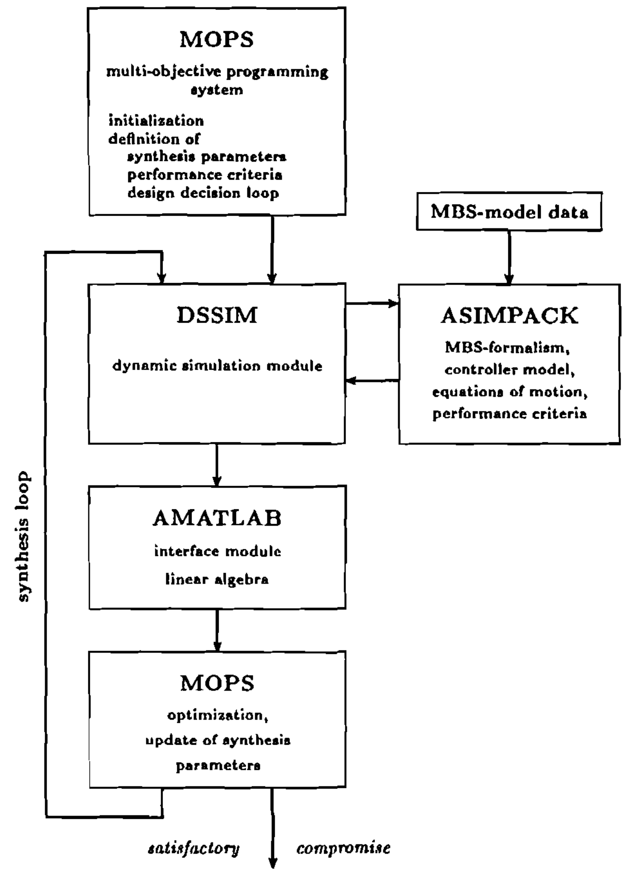

Fig. 9 Applied Software Modules Optimizing SIMPACK Models in the ANDECS Environment

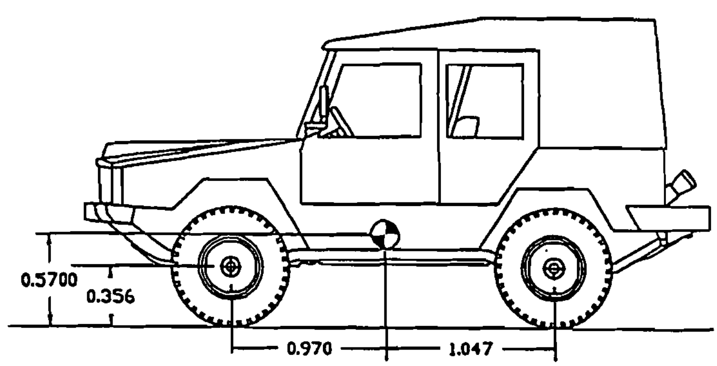

Fig. 10 The Bombardier ILTIS–Side View, see [1, 21].

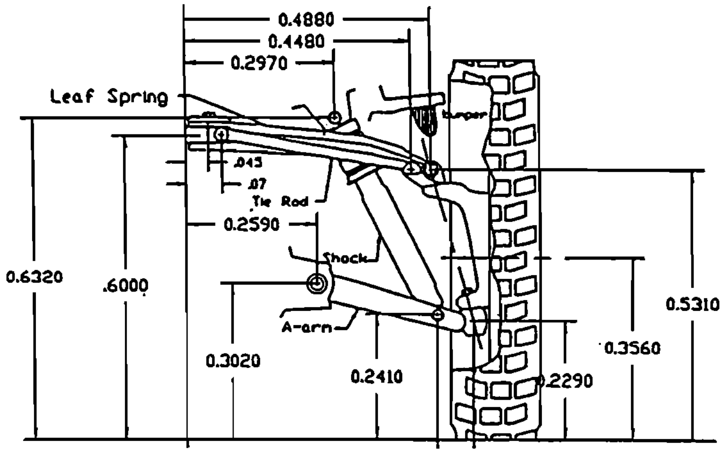

Fig. 11 The ILTIS Suspension Arrangement.

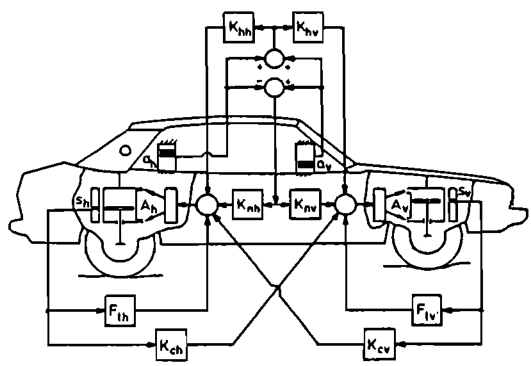

Fig. 12 Control Structure of the Shock Absorbers Without Preview (Source Foag [25]).

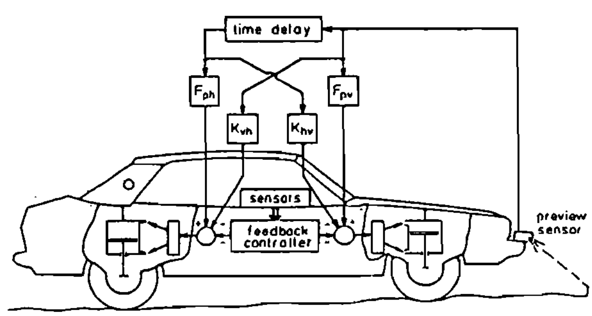

Fig. 13 Control Structure of the Shock Absorbers with Preview (Source Foag [25]).

one DOF for each wheel). For further details of the model as well as simulation results for a variety of maneuvers the reader is referred to [1].

Actuator and Controller for the Active Vehicle

For activating the passive vehicle, in addition to the passive shock–absorber an hydraulic actuator is considered, which has controlled oil–volumina. The force law of this actuator is taken from [25]; it is approximately being described by a first–order linear differential equation.

For controlling the oil–volumina $q_k, k = 1 - 4$, we consider the following sensors as available: the relative motion of the actuator piston, δ_k, the vertical absolute acceleration of the car body, a_{zk}, its pitch and roll angular accelerations, $\ddot{\varphi}_{1k}, \ddot{\varphi}_{2k}$. Based on these measurements, the control structure is shown in Fig. 12, as in [26]. The control law based on this structure has, because of the symmetry of the vehicle, 2 x 8 = 16 free synthesis parameters for the four actuators.

As a speciality, in addition two *preview sensors* are considered at the front of the vehicle measuring the vertical displacement s_p of the road in front of the car at the right and left track, see Fig. 13. The preview measurements are adapted with the aid of time–delays to the wheel positions and with dynamic filters of third-order added to the control of the oil–volumina, Fig. 13. For the simulation of the controlled vehicle each of the four time–delays are approximated by a second–order Padé–approximation. With the differential equations for the actuator, the feedback filters and these Padé–filters the total system order of the preview controlled vehicle becomes 44 and a total of 38 free parameters to be chosen.

Case Study and Resulting Model Reduction

As a first case study the benchmark test situation, where the ILTIS travels over a symmetrical "cosine–bump" of 0.2 m height and 5 m length with a speed of 10 m/sec, was selected. For a 5 sec maneuver the simulation time on a HP-9000/720 workstation of the complete ILTIS model takes about 50 sec. This is because of the 8 kinematic loops, the very stiff differential–algebraic equations in connection with the suspension characteristics and the discontinuities of wheel lift and landing. For the parameter optimization (not for the design comparisons) a half car model was chosen in order to save computer time. However, in order to keep the potential deviations of this model as small as possible, the exact 3D–kinematic model of the suspensions was projected into the 2D design model and all force laws were properly adapted to this situation. As a result the "active ILTIS" can be modelled with a MBS in tree–structure with a total number of 22 states and for active suspension without preview 10 synthesis parameters and with preview 28 synthesis parameters were to be selected.

Results of the Design Optimization

Figures 16 – 19 show some major results of a multi–objective parameter optimization study with MOPS, all being based on the simulation with the complete 3D–model of the ILTIS; they present

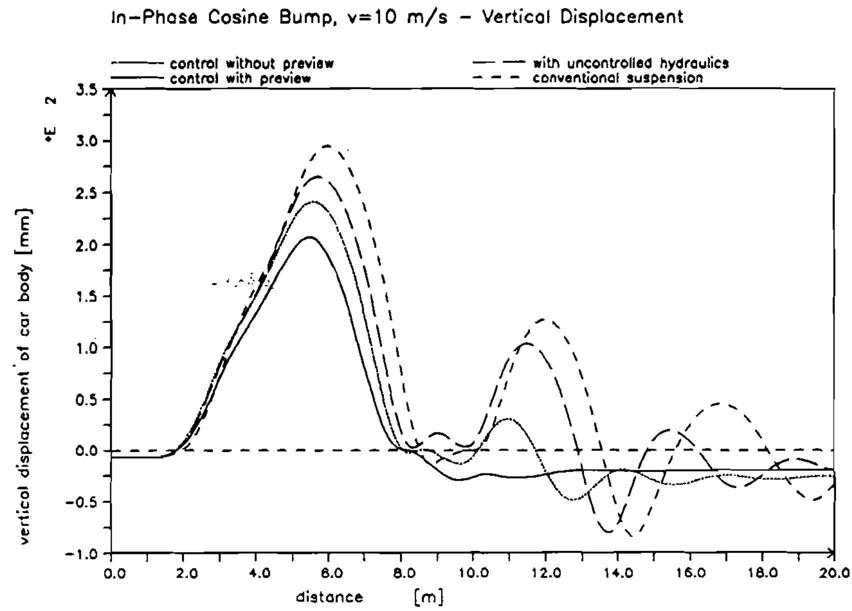

Fig. 14 Comparison Between Different Suspension Concepts: Vertical Displacement of Car Body.

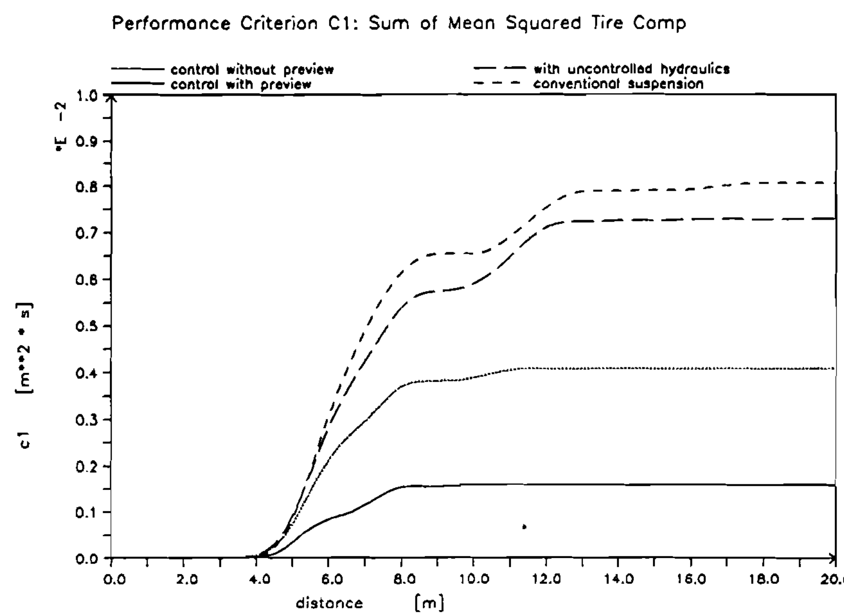

Fig. 15 Comparison Between Different Suspension Concepts; Performance Criterion for the Wheel Deflections: $c_1 = \int_0^t \sum_{k=1}^{2} \Delta r_k^2 \, dt$.

- the vertical deflection of the car body, (Fig. 14);
- the performance: wheel–load variation, (Fig. 15);
- the performance: vertical acceleration, (Fig. 16);
- the performance: pitch angular acceleration, (Fig. 17),

each for four different design cases. The different designs which are compared are

- the conventional passive system;
- the suspension with uncontrolled hydraulic actuator;
- the active suspension design without preview;
- the active suspension with preview control.

As expected the conventional passive design is the "worst" case, whereas the active preview control provides the most favorable performances, which significantly exceed the results of the active suspension without preview sensors; details can be found in [24].

Fig. 18 shows two action shots taken from an animation of the preview controlled vehicle (solid model) and the conventional model (wire frame model) at two selected positions after running over the bump clearly, demonstrating the better performance of the preview controlled case.

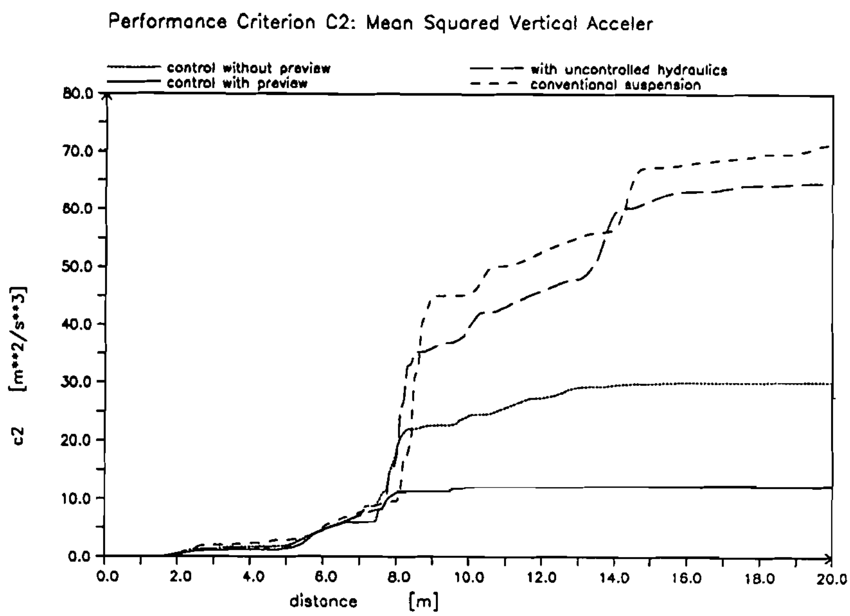

Fig. 16 Comparison Between Different Suspension Concepts; Performance Criterion for the Vertical Acceleration: $c_2 = \int_0^t a_z^2\,dt$.

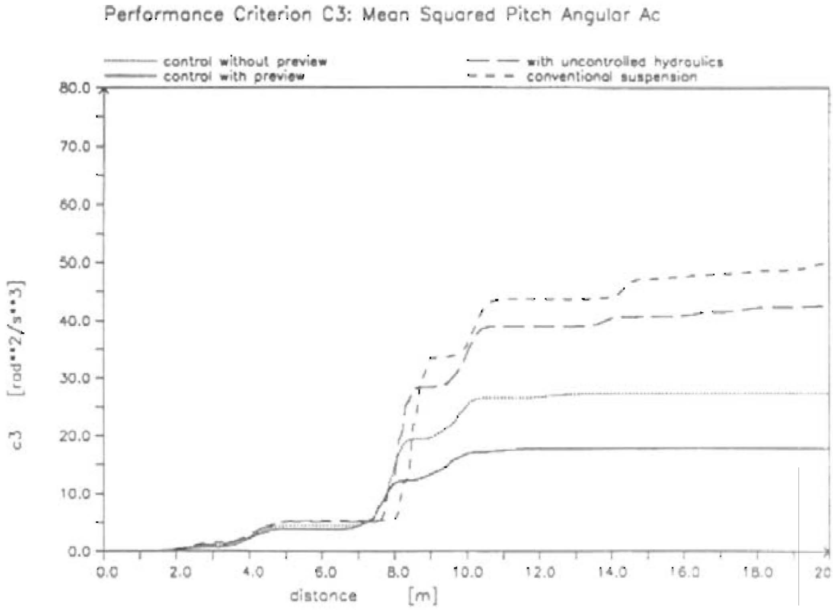

Fig. 17 Comparison Between Different Suspension Concepts; Performance Criterion for the Pitch Angular Acceleration: $c_3 = \int_0^t \ddot{\theta}^2 \, dt$.

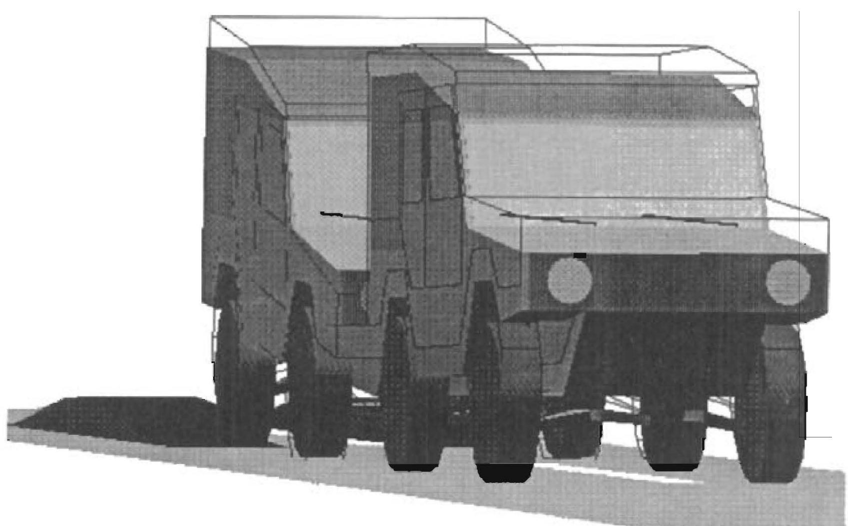

Fig. 18 Animation of the Preview Controlled Vehicle (solid model) as Compared with the Conventional Design (wire frame model).

Some care should be applied with respect to the practical implications of this study: So far only one major maneuver has been considered for the design and analysis and also no hardware restrictions (sensors, costs etc.) have been taken into account.

6. CONCLUSION AND PROSPECTS

Advanced vehicle systems will be equipped with electronics and feedback control systems, denoted as mechatronic vehicles. The analysis and design of the mechanical parts as well as the electronic components has to be performed *concurrently*. The main analysis and design features for mechatronic vehicles have been summarized and most available popular software has been critically evaluated versus the future requirements. Recent progress made has been described and in particular two current research and software developments in Germany have been described in some detail as they seem to be pointing the way to an integrated system of analysis and design software for controlled vehicles. The market will decide how successful some of the concepts will prove.

References

[1] W. Kortüm and R.S. Sharp. Multibody Computer Codes in Vehicle System Dynamics. *Supplement to Vehicle System Dynamics*, 22, 1993.

[2] W.O. Schiehlen. *Multibody Systems Handbook*. Springer, 1990.

[3] D.C. Karnopp. Are Active Suspensions Really Necessary? *ASME, 78-WA/DE/2*, 1978.

[4] R.M. Goodall and W. Kortüm. Active Controls in Ground Transportation - a Review of the State-of-the-Art and Future Potential. In *Proc. 8th IAVSD-IUTAM Symposium*, pages 225–257, Cambridge, Mass., 1983.

[5] M. Nagai M. Iguchi, M. Abe, editor. *Proceedings of the International Symposium on Advanced Vehicle Control, AVEC'92*, Yokohama, Japan, September 14-17 1992.

[6] R.M. Goodall and W. Kortüm. Active Suspension for Railway Vehicles - an Avoidable Luxury or an Inevitable Consequence? In Ü. Jaaksoo V. Utkin, editor, *Proc. 11th IFAC World Congress, "Automatic Control in the Service of Mankind", Tallinn, Estonia, USSR, Aug. 13-17, Vol. 10*, pages 219–225, 1990.

[7] I MECH E. *Proceedings of Railway Dynamics for Today and Tomorrow*, London, England, 1992. Seminar S043.

[8] F.E. Cellier. *Systems Modeling and Computer Simulation*, chapter Computer-Aided Control Systems Techniques and Tools, pp.631-679. Marcel Dekker, Inc., New York, Basel, 1988.

[9] F.E. Cellier. *Continuous System Modeling.* Springer, 1991.

[10] W. Kortüm. Software zur Modellbildung und Simulation der Dynamik mechatronischer Systeme. In *Proc. Modellbildung für Regelung und Simulation,* Langen, März 1992. VDI/VDE-GMA-Aussprachetag.

[11] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: A Unified Approach.* John Wiley & Sons Inc., 1990.

[12] W.O. Schiehlen. Simulation Based Design of Automotive Systems. In E. J. Haug, editor, *Concurrent Engineering: Tools and Technologies for Mechanical System Design (NATO ASI Series F, Vo. 108),* pages 303–337. Springer, 1992.

[13] D. Bestle, P. Eberhard, and W. Schiehlen. Optimization of an Actively Controlled Vehicle System. In *Proc. IUTAM Symposium "Optimal Control of Mechanical Systems", Moscow.* University of Stuttgart, Institute B of Mechanics, 1992.

[14] G. Grübel and W. Kortüm. Towards a Coherent Technology for Computational Vehicle System Dynamics. In *Proc. of AVEC'92, International Symposium on Advanced Vehicle Control, Yokohama, Japan, September 14-17,* 1992.

[15] W. Schiehlen. Multibody Dynamics Software for Controlled Vehicle Simulation. In *Proc. of AVEC'92, International Symposium on Advanced Vehicle Control, Yokohama, Japan, September 14-17,* pages 37–42, 1992.

[16] G. Grübel. AnDeCS: a Concurrent Control Engineering Project. In *Proc. IEEE CACSD '92 Symposium,* pages 37–46, Napa, 1992.

[17] G. Grübel, H.-D. Joos, M. Otter, and R. Finsterwalder. The ANDECS Design Environment for Control Engineering. In *IFAC World Congress,* Sydney, 18-23 July 1993.

[18] W. Schiehlen. Prospects of the German Multibody System Research Project on Vehicle Dynamics Simulation. In *Proc. 12th IAVSD-Symposium, Lyon, France, August 26-30, 1991 (Supplement to Vehicle System Dynamics, Vol. 20),* pages 537–550, 1991.

[19] M. Hoffmann and B. Richter. The Balancing Briefcase: Interfacing ADAMS and MATRIX_X. In *Int. ADAMS Users' Conference,* Ypsilanti, Michigan, USA, 1992.

[20] W. Schiehlen, editor. *Advanced Multibody System Dynamics - Simulation and Software Tools,* Dordrecht, Boston, London, 1993. Kluwer Academic Publishers.

[21] R.G. Langlois, D.M. Hanna, and R.J. Anderson. Implementing Preview Control on an off-Road Vehicle with Active Suspension. In G. Sauvage, editor, *Proceedings of the 12th IAVSD-Symposium, Supplement to Vehicle System Dynamics 20,* pages 340–353, 1991.

[22] W. Schwartz. Optimierungsgestützte Auslegung einer aktiven Fahrzeugfederung innerhalb der Entwurfsumgebung von ANDECS und der Simulation mit SIMPACK. DLR-IB 515-93/18, DLR, 1993.

[23] M. Hoffmann. Development of Control Strategies for Suspension Systems. In *24th Int. Symposium on Automotive Technology and Automation*, Florence, Italy, May 1991.

[24] W. Kortüm and W. Schwartz. Design of Optimal Active Vehicle Suspensions Based on Realistic Models. In *Proc. Second Conference on Mechatronics and Robotics*, Duisburg, September 27–29 1993.

[25] Werner Foag. *Regelungstechnische Konzeption einer aktiven PKW-Federung mit "preview"*. PhD thesis, Ruhruniversität Bochum, VDI-Verlag, 1990.

[26] W. Foag and G. Grübel. Multi-criteria control design for preview vehicle-suspension systems. In *Proc. 10th IFAC World Congr. on Automatic Control, Munich*, volume 3, pages 190–196, 1987.