TOWARDS MODEL MANIPULATION FOR EFFICIENT AND EFFECTIVE SIMULATION AND INSTRUCTIONAL METHODS

Ir. Pieter J. Mosterman, Ph.D.*[1]

* Institute of Robotics and System Dynamics, DLR Oberpfaffenhofen,
P.O. Box 1116, D-82230 Wessling, Germany
Tel: +49 8153 282434, Fax: +49 8153 281441,
Email: Pieter.J.Mosterman@dlr.de, URL: http://www.op.dlr.de/~pjm/

ABSTRACT

Reactive learning environments, *microworlds*, rely on *anchored instruction* and *scaffolding* to achieve *constructive learning*. Microworlds allow a learner to experiment freely and build on existing knowledge. Anchored instruction presents problems in a realistic setting and to avoid overwhelming detail, scaffolding presents assignments at a level of detail that meets the learner's competence. This is highly motivating as real life problems are immediately solved and details are successively revealed as their complexity becomes manageable. In this context, model manipulation techniques could be employed to systematically manipulate detailed continuous behavior into a simpler counterpart. In many cases, the detailed continuous behavior is abstracted into discontinuities resulting in mixed continuous/discrete, *hybrid*, models. Model manipulation can also be utilized in a reverse mode by teaching the learner where to look for unmodeled higher-order continuous behaviors to gain insight in physical phenomena that may be difficult to observe directly. If simulation is required in microworlds model reduction is critical to handle otherwise prohibitively detailed behaviors. Efficient hybrid behavior generation algorithms also enable exhaustive model analysis to identify worst-case and even possible undesired model behaviors and feed this back to the learner. Furthermore, hybrid behavior generation allows systems that are dynamically reconfigured (e.g., parts are deleted and created), facilitating interactive modeling and simulation where 'what if' scenarios can be quickly designed and studied, e.g., by 'rewind' and 'fast forward' of behaviors. Overall, to reap these benefits and achieve the overall training and knowledge capitalization goals, effective methodologies need to be developed for automatic model manipulation.

Keywords: model reduction, evolving models, microworlds, knowledge capitalization, hybrid systems, hybrid simulation, interactive modeling and simulation

1 INTRODUCTION

Recently, virtual worlds are increasingly applied for instruction. They rely on behavior generation of very realistic, and, therefore, detailed underlying models. This section gives an overview of this type of instruction, how the model complexity can be managed and how the same methods used can also be applied to otherwise improve instructional methods.

1.1 Principles of Instruction

Effective learning environments rely on the notions of *scaffolding* and *anchored instruction* to achieve *constructive learning* [Cognition and Technology Group at Vanderbilt, 1992, Dally, 1995; Von Glaserfeld, 1987]. In order to better transfer knowledge, motivate and inspire the learner, and to develop a feel for real life application of abstract theories, anchored instruction presents problems in a realistic setting. For example, instead of teaching geometry as an abstract matter, Phytagoras' theorem ($a^2 + b^2 = c^2$) can be embedded in a real life setting by computing the height of a steeple as shown in Fig. 1. This enables the learner to transfer theories and abstract concepts to real life problems and makes it easier to eventually exploit their use.

Mircoworlds facilitate embedding abstract knowledge in a real life setting. To achieve this in a very interactive and safe environment the use of high fidelity model based simulation is of paramount importance. The underlying models for simulation are necessarily complex, embodying sufficient detail to present a realistic learning environment. In order not to overwhelm the learner, scaffolding relies on less complex problems that still have a bearing to reality. Microworlds present high level assignments that can be completed successfully without the need of much initial detailed understanding. This is highly motivating as the learner immediately solves real life problems, where a Socratic educational effort successively reveals details as their complexity becomes manageable and the problem can be solved by the learner.

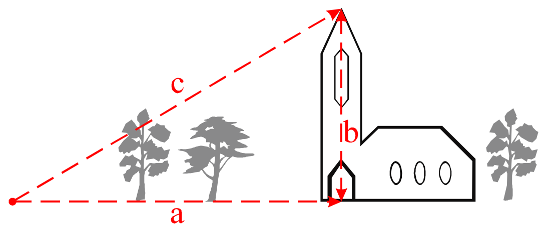

Fig. 1: Anchored instruction embeds abstract theories in a real life setting.

1.2 A Hierarchy of Behavioral Detail

Because of their complexity, realistic models are best constructed in a hierarchical fashion where very fine grained behavioral details are present at the lower levels (see, e.g., [Moormann, Mosterman and Looye, 1999]). This poses a huge numerical complexity if behaviors are to be generated all the way down at the most detailed level, which can be mitigated by a hierarchical approach to simulation that efficiently abstracts away higher order phenomena with little or no effect on gross behavior or condenses the detailed behavior into an instantaneous

[1] Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

change. These abstractions require model manipulation techniques in an intermediate stage before simulation and often generate models with mixed continuous/discrete, *hybrid*, behavior with its particular simulation idiosynchracies [Mosterman, 1999; Mosterman and Biswas, 1998]. In this scheme, hybrid behavior simulation is a key enabling technology to exploit the ambitious automated model reduction and ultimately achieve the teaching goals.

As an example, consider the primary attitude control surfaces of the airplane in Fig. 2. At a detailed level, continuous closed loop PID control moves the rudder, elevators, and ailerons to set positions. Desired setpoint values are generated directly by the pilot or by a supervising control algorithm implemented on a digital processor.

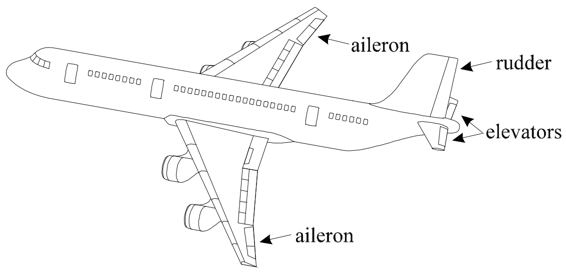

Fig. 2: Primary attitude control surface of an aircraft.

Aircraft are safety critical systems and their control systems incorporate several forms of redundancy. For example, the elevator control system may consist of two mechanical elevators (Fig. 3) that are positioned by electro-hydraulic actuators. Only one actuator per elevator is active and controls its position. The other actuator is passive and acts as a simple load. When a failure occurs, redundancy management may switch between actuators to ensure maximum control, which leads to discrete changes in system configuration. Switching actuators is realized by hydraulic valves. Model simplifications created by discretizing the fast nonlinear transients in the dynamics of these valves may produce discontinuous variable changes. Because the detailed hydraulic behavior involved may not significantly affect gross flight behavior it can be modeled as an instantaneous change in valve positions.

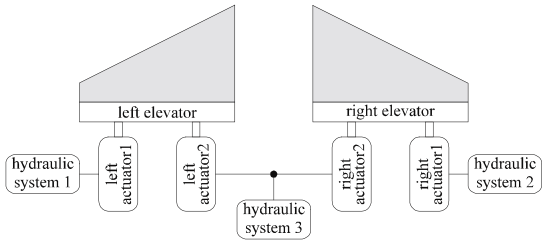

Fig. 3: Redundancy in the elevator control system.

Automated model manipulation is a necessity in order to exploit the full potential of hybrid behavior generation methods. Ideally, this allows a detailed model of tremendous continuous complexity to be systematically manipulated into a hybrid model, replacing detailed continuous behavior with instantaneous abstractions. In case of the aircraft elevator control, analysis would show that the details of the hydraulic dynamics can be captured by discontinuities, and the detailed models of the hydraulic behavior are automatically reduced to simple switches.

1.3 Model Augmentation

Algorithms that implement this model reduction can be applied for instructional purposes to familiarize the learner with abstraction techniques critical to efficient model design. Also, these techniques can be utilized in a reverse mode by teaching the learner where to look for unmodeled higher-order continuous behaviors because of the presence of abstract discontinuous changes. This is of great help, e.g., in obtaining insight in physical behavior that cannot be directly observed, and making this knowledge explicit aids in systematically extracting knowledge from complex models. In case of the aircraft elevator control system, if the change in actuators is modeled as an instantaneous switch, it can be hypothesized that higher order physical phenomena are abstracted away. These parameters can be systematically introduced and indicate the presence of physical parameters such as oil elasticity and dissipation that may explain, e.g., observed elevator velocity changes when switching between actuators.

In other work, higher order physical phenomena are systematically introduced in models to provide the basis for a physically consistent treatment of *chattering* behavior in hybrid models [Mosterman, Zhao and Biswas, 1998b]. Without these higher order phenomena, no solution trajectory would exist.

1.4 Hybrid Behavior Generation

The availability of hybrid behavior generation methods unlocks further potential with respect to instructional strategies. Efficient hybrid behavior generation enables methodologies for exhaustive model analysis to identify worst-case and even possible undesired model behaviors. This is particularly helpful, e.g., in teaching how to solve a planning task where one simulation trajectory of the proposed model that fails to meet the required constraints can be exploited by the tutoring mechanism to confront the learner with a scenario where the submitted plan fails.

In general, systems are not static but highly dynamic in a sense that parts of it evolve to behave in different modes, it may be reconfigured (e.g., in case of failure), and parts of it may be deleted and new ones created (e.g., cars entering and exiting a highway). This is facilitated by interactive modeling and simulation where 'what if' scenarios can be quickly designed and studied, e.g., by 'rewind' and 'fast forward' of behaviors. Also, during experimentation, advanced simulation allows model structure changes that are interactively made by the learner and model parts to be isolated from the rest of the system for individual scrutiny. Such interactive operation provides the optimal environment for iterative modeling and simulation and inherently embodies discrete behavior phenomena combined with continuous time behavior. In conjunction with the use of hybrid behavior generation paradigms, this flexibility requires the use of *interpreted* simulation that has reduced efficiency as main drawback. Therefore, methodologies need to be developed for model manipulation to reduce numerical complexity and, along with increasing computing power and improved numerical solvers, allow for achieving the overall training and knowledge capitalization goals.

1.5 Overview

This paper shows how automated model reduction is crucial in developing sophisticated instructional tools and methods. On the one hand, it allows behavior generation of complex systems over a wide range of detail while automatically deriving consistent parsimonious models at each of these levels of detail. This enables behavior generation in microworlds for systems of tremendous complexity (such as, e.g., the aircraft dynamics). Moreover, the problem at hand can be automatically adapted to the learner's competence, effectively hiding overwhelming detail (such as, e.g., dynamic behavior of hydraulic valves that implement the elevator actuator switch). Furthermore, model reduction *and* augmentation techniques form the basis for teaching structured modeling of physical systems and aid in gaining insight in the physics of a system.

Section 2 first discusses the hydraulics of an elevator actuator in detail and illustrates anchored instruction for systems theory by providing a practical physical representation. Section 3 discusses the need for model reduction in behavior generation for microworlds. Section 4 shows how model reduction can be used for knowledge capitalization. Section 5 shows how model reduction relates to and has a dual in model augmentation and how the two combine to build model manipulation techniques that can be applied for knowledge capitalization. Section 6 presents the conclusions and an outlook over future research.

2 ANCHORED INSTRUCTION FOR MODELING

Modern avionics systems employ electronic *fly-by-wire* control, where electronic signals generated by a digital processor are transformed into the power domain by electro-hydraulic actuators. Fly-by-wire technology allows to implement a rather sophisticated discrete event control logic that interacts with continuous physical system behavior. Such systems exhibit inherently hybrid behavior. This section focuses on the electro-hydraulic actuators used for elevator control and shows how these can be used for anchored instruction.

2.1 The Elevator System

Fig. 4 shows the operation of one elevator actuator. The continuous PID control mechanism for elevator positioning is implemented by a servo valve. A digital controller (not shown) positions the servo valve piston according to the feedback signal for PID control that may be computed based on the fluid pressure, mechanical linkage, electrical signals, and a combination of the three. The positioning cylinder consists of two chambers that are filled with oil. By controlling the oil flow through the servo valve into these chambers, the piston in the cylinder, and, consequently, the elevator can be positioned.

For a given position of the servo valve piston, there is a flow of oil from the servo valve *supply* to either the left or right cylinder chamber and a flow to the servo valve *return* from the other chamber. The position of the piston inside the servo valve can be adjusted to change the size of the orifices, thereby modulating the amount of oil flow to and from the cylinder chambers. This oil flow controls the direction and speed of travel of the piston in the cylinder that combined with the connected elevator flap constitutes the control load.

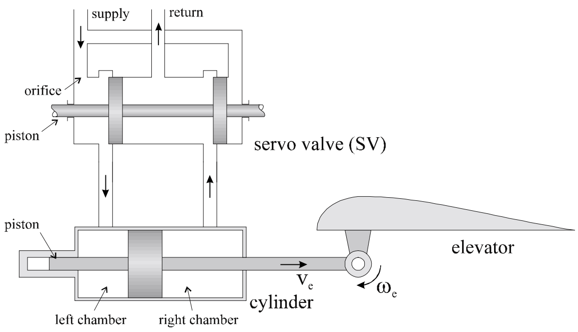

Fig. 4: Elevator control system.

When the servo valve piston is in its far right position (the situation depicted in Fig. 4), the cylinder left chamber is connected to the hydraulic *supply* pressure and the maximum amount of oil flows into the left chamber of the cylinder and the elevator moves with maximum angular velocity, ω_e, in the counter clockwise direction. When the servo valve piston is in its far left position, the right chamber is connected to the hydraulic *supply* pressure and the oil flow into the right chamber is maximal. In this configuration the elevator moves at maximum angular velocity in the clockwise direction.

2.2 Anchored Instruction in Model Design

In engineering education, models are often represented as abstract mathematical systems. For example, a MATLAB-SIMULINK representation of a 2nd order model with nonlinearity is shown in Fig. 5. Often it is neglected to explain the abstraction and embed the ensuing results of studying system behavior in a real life setting.

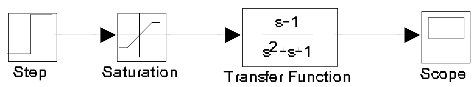

Fig. 5: Mathematical model of a second order system with nonlinearity.

In case of the elevator control system, the cylinder can be used to anchor abstract knowledge in reality. This facilitates a direct interpretation of system behavior in physical terms. First, the piston inertia, *I*, and oil elasticity, *C*, are identified as phenomena with energy storing, i.e., integrating, behavior, illustrated in Fig. 6. The combination of these two first order effects combined with oil dissipation, *R*, results in a second order system with damped, possibly oscillatory, behavior. A change in wind force acting on the elevator may correspond to a step input.

Simulation using the abstract model in Fig. 5 generates behavior that can now be interpreted in physical terms. The wind force pushes against the cylinder piston that starts building up momentum. This momentum causes a piston velocity that compresses the oil in the chamber modeled with oil elasticity and dissipation. The elasticity builds up a counter force exerted by the oil that reduces piston momentum until 0. This corresponds to the situation where the oil is maximally compressed. The force may then cause piston momentum and velocity to reverse and the piston starts moving in the opposite direction. The wind force opposes this movement and the piston oscillates back and forth until an equilibrium state is reached where the oil compression force and wind acting on the elevator are in balance. Internal dissipation effects of the oil determine the damping coefficient, i.e., how quickly this equilibrium is reached.

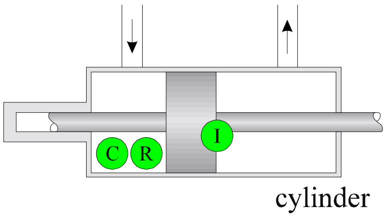

Fig. 6: Real system with second order nonlinear model representation.

This demonstrates how system modeling and theory can be embedded in real situations. The example was limited to one level of behavioral detail. In the proceedings, it will become clear that anchored instruction can act on a hierarchy of behavioral detail and this hierarchy can be systematically traversed by model manipulation.

3 BEHAVIOR GENERATION FOR MICROWORLDS

Microworlds are learning environments that allow the learner much freedom in experimenting. These microworlds can be fairly abstract representations of real life concepts such as Newton's Second Law (e.g., Thinkertools [White, 1993]), more realistic representations (e.g., STEAMER [Hollan, Hutchins and Weitzman, 1984]), and they can be high fidelity realistic environments (e.g., the Electronics Laboratory Simulator [Mosterman *et al*, 1994; 1996]). The practical viability of such microworlds is testified by the commercial success of some of this software [Mosterman *et al*, 1995].

3.1 Microworlds and Simulation

Typically, the learning environment is layered on top of a behavioral model that is the object of instruction. To convey the knowledge present in this model to the learner behavior trajectories are generated, requesting the learner to analyze and assimilate the underlying model.

In this setup, simulation is a critical ingredient for developing successful microworlds to (i) generate example trajectories, (ii) help the learner develop an intuition for the underlying model behavior, possibly by animation and visualization, and (iii) support learning from experience with model behavior. Furthermore, simulation can be performed in a networked environment and support cooperative learning in a possibly distributed setup.

The advantages of using high fidelity simulation are the capability of training how to respond in situations that are potentially dangerous and even catastrophic in real life. Furthermore, hard to generate real life behavior (e.g., certain weather conditions) as well as idealized behavior (e.g., without disturbances) can be easily invoked for training. Also, simulation allows very interactive learning environments with extreme responsiveness as well as rigidly enforced operational procedures (e.g., federal guidelines).

3.2 Constructive Learning

Constructive learning is based on the premise that the best motivation comes from within instead of from external stimuli [Von Glaserfeld, 1987]. This requires the course material to closely match the learner's interest and level of competence. Instead of trying to construct a detailed model of the learner's knowledge and generate new assignments based on this *student model* which is practiced in the area of *intelligent tutoring systems* [Wenger, 1987], microworlds leave the initiative to the learner to request instruction and training on topics and at a comfortable level of complexity that can be handled.

In this paradigm, the learner is free to experiment, but a tutoring mechanism is present to consistently produce challenging problems and respond to user input. Simulation is a necessity to visualize the results of the selected experiments that help uncover and assimilate the knowledge in the underlying model. For example, the compression and expansion of oil while moving the piston can be visualized as in Fig. 7. It presents insight in the physical phenomena that are active and what dynamic behavior they correspond to. This in turn allows the learner to construct a mental model of otherwise obscured phenomena.

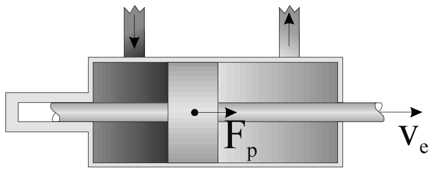

Fig. 7: Simulation and visualization of cylinder behavior.

3.3 Hierarchy in Behavior Simulation

Systems operate on a hierarchy of temporal and spatial scales. Realistic models include many phenomena that are highly nonlinear (e.g., turbulent oil flow and cylinder end stops) and often operate on a level of detail not significant for gross behavior (e.g., small leakages and elasticity effects). Though not significant for gross behavior, their presence may hamper efficient simulation tremendously as nonlinearities often require numerical iteration at each integration time step and very detailed behaviors typically operate on a time scale that requires an integration time step that is much smaller than required for simulating the dynamics of gross behavior.

To illustrate, consider the sequence of servo valve control actions in Fig. 8. Initially, the piston in the servo valve is positioned to invoke oil flow into the left cylinder chamber and the elevator starts moving towards its commanded position. When it is close to this, the servo valve piston closes the orifices to decelerate the elevator and prevent it from moving beyond the commanded position. A situation may occur where the servo valve piston completely closes the orifices, thereby blocking the oil flow into and out of the cylinder (shown on the right in Fig. 8). In this configuration the built up momentum of the cylinder piston, because of its velocity v_e, interacts with the oil elasticity. As described earlier, the piston compresses the oil somewhat but due to dominant dissipation it quickly reaches 0 velocity and the elevator comes to a halt. Because of the large dissipation involved, this behavior acts on a time scale much smaller than the time scale of interest, illustrated in Fig. 9. In this particular case, since the commanded position was not reached yet, the control quickly reopens the servo valve to continue the maneuver.

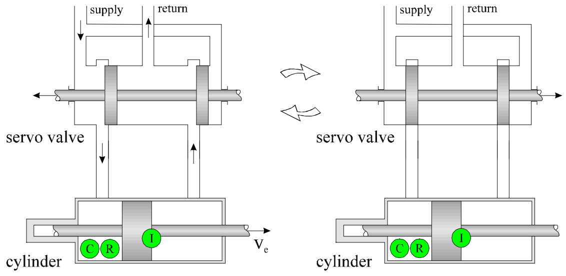

Fig. 8: Elevator control may induce fast continuous transients.

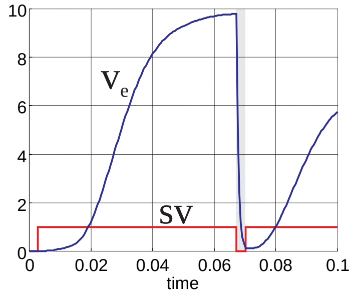

Fig. 9: Simulation of fast elevator velocity change when the servo valve is temporarily closed.

3.4 Model Reduction

The fast continuous transient shown in Fig. 9 operates on a time scale much smaller than gross behavior. Therefore, it requires a much smaller integration time step and this hampers efficient behavior generation. This is especially intolerable if real-time simulation is aimed at. Here a fixed integration time step is required to guarantee that new simulation values that have to be available at fixed time points are computed within certain temporal bounds. For the simulation in Fig. 9 this implies that the entire trajectory is simulated with an integration time step sufficiently small to handle the steep gradient when the servo valve is closed.

This problem can be mitigated by applying a *parameter abstraction* [Mosterman, 1997] to the cylinder model, removing the oil elasticity and dissipation parameters while keeping the piston inertia, as shown in Fig. 10. In this model, the steep gradient of the elevator velocity change has become an abrupt, discontinuous change, illustrated in Fig. 11. Therefore, it can be efficiently handled and numerical simulation apply an integration time step corresponding to the gross dynamics. In this particular case, the fixed time step can be chosen at least a factor 20 larger.

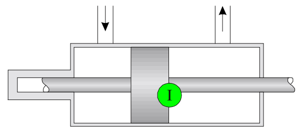

Fig. 10: Model manipulation reduces the cylinder model to a first order representation.

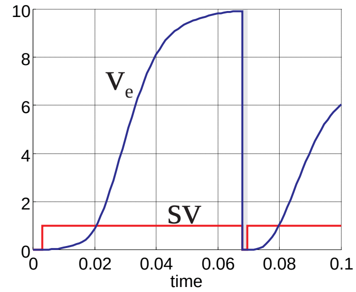

Fig. 11: Removing small oil parameters causes a discontinuous change in elevator velocity.

Note that the overall effect of the small parameters on gross behavior is preserved, i.e., the elevator velocity still changes abruptly to 0. If the detailed continuous transient was removed altogether, velocity v_e would remain constant when the servo valve was closed, and behaviors would be generated that are inconsistent with the more detailed model. This is crucial, e.g., for control design as the correct approximation reveals that the control law causes the servo valve piston to close the orifices prematurely.

In general, model reduction that leads to hybrid systems is based on (i) parameter abstraction and (ii) time scale abstraction. Parameter abstractions occur when small and large, often parasitic, dissipation and storage parameters are abstracted away from the system model causing discontinuous changes in system behavior. Time scale abstractions compress behaviors that occur on a small time scale to explicit discontinuous changes at a point in time.

If model reduction can be automated, it can be applied to dynamically adjust the level of model detail used for simulation. This allows the learner to zoom in and out on system behavior.

3.5 Hybrid Simulation for Knowledge Capitalization

As shown, the elevator actuator model with reduced complexity can be simulated more efficiently than its detailed counterpart. This is important if, e.g., real time performance in a *virtual reality* [Krueger, 1983; Rheingold, 1991] like microworld is required. As such, model reduction allows microworlds to be constructed for much more complex systems than otherwise possible.

Furthermore, efficient simulation allows exhaustive search to identify, e.g., worst case scenarios in control law design. This information can then be fed back to the learner along with the trajectory where given constraints are violated.

Also, model reduction results in simpler behaviors that include less higher order phenomena (e.g., fast oscillations). Based on the scaffolding principle, these simpler behaviors provide an optimal entry point for novices to start learning about the complexity of a particular system because they are easier to comprehend.

4 MODEL MANIPULATION FOR KNOWLEDGE CAPITALIZATION

Model manipulation can greatly aid in reducing complexity of behaviors, which results in more efficient simulation. However, model reduction itself embodies many virtues and is executed by mechanisms that can be exploited for knowledge capitalization as well.

4.1 Overwhelming Detail and Scaffolding

Whether a model is good or bad depends on the problem it needs to address. This implies that more detailed models are not intrinsically better than more abstract ones. In fact, in many situations less detail suffices to solve a problem, which makes the corresponding model better than a more detailed one.

In microworlds, instead of overwhelming the learner with complicated behaviors, basic phenomena are best presented first. In case of the cylinder, such a model includes piston inertia only, see Fig. 12(a). This presents a real life representation of a first order model that can be used, e.g., to study characteristics of first order systems such as rise time and to develop a simple control law for elevator positioning.

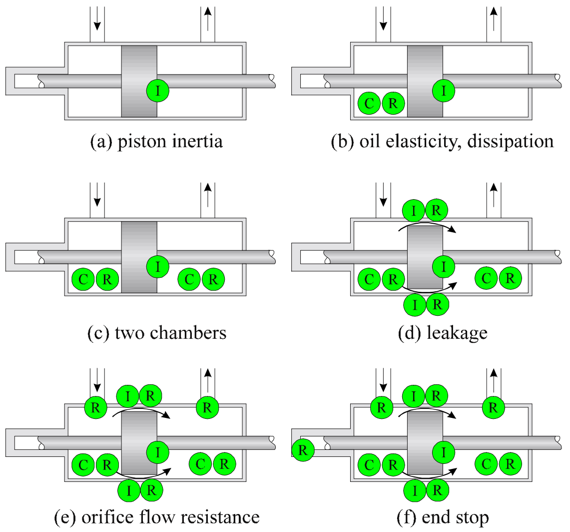

Fig. 12: The cylinder can be modeled at a number of levels of detail.

When competence at this level is achieved, the model can be augmented by including oil elasticity and dissipation effects, as shown in Fig. 12(b). This corresponds to a second order system, and again, e.g., system concepts such as damping and overshoot and control design can be studied.

The process of successively increasing model complexity may proceed by adding the dynamics of a second chamber shown in Fig. 12(c). Next, small leakage flows can be added, as shown in Fig. 12(d). These leakage effects may be the result of manufacturing imperfections, but also be designed effects to prevent extremely high pressure differences between the left and right chamber. Resistance to flow of the cylinder orifices are included in Fig. 12(e), and, finally, the nonlinearity of the end stop in Fig. 12(f).

Though the cylinder model in Fig. 12(f) mimics real behavior closest, for many purposes this model is much too detailed to be effective and the functioning of the cylinder is best explained using a simpler model. However, a model that fosters many detailed behaviors can be applied as a baseline model from which less abstract representations can be systematically derived. This enables the tutoring mechanism to gradually increase model complexity and consistently pose a challenging task and keep pace with the learner’s level of competence.

Depending on the task to be solved, certain phenomena may be included instead of others and the procedure for increasingly adding detail is not sequential. For example, the end stop nonlinearity may be of importance in combination with first order dynamics embodied by just the piston inertia, e.g., for control law design. For this task, realization artifacts such as leakage may be of little importance whereas end stop behavior may critically affect control law stability. A fundamental issue that needs to be addressed by model manipulation theory is which system characteristics are preserved given certain model simplifications. These are important issues in model manipulation that need to be solved for successful use.

5 MODEL MANIPULATION TECHNIQUES

The need to systematically reduce model complexity has long been recognized, especially in conjunction with methods such as compositional modeling and finite element modeling. However, in many cases model reduction is executed on a very abstract mathematical level, often numerically. The main drawback to this is the loss of correspondence with physical structure and topology. Another hiatus in existing model reduction theory is that though often the reduced models are shown to be good approximations within a certain operational domain in time or state space, it is not addressed how to derive the switching between domains with different approximations.

5.1 Compositional and Finite Element Modeling

In order to cope with the ever increasing complexity of engineering systems, compositional modeling techniques allow system parts to be modeled in extreme detail by domain experts. These parts can then be composed together to provide a model of the complete system. This is an appealing paradigm as it becomes possible to handle large systems, all different domains (e.g., electrical, hydraulic) involved are modeled with great expertise, the model is easier to maintain (only the modified pieces need to be changed), and model components can be reused. The disadvantage is that modeling of the individual components in tremendous detail results in stupendous complexity of the composed model. Also, behaviors that are typically collapsed into aggregate behavior are individually present and tightly coupled.

Advances in control of, e.g., flexible aircraft, have triggered a need for detailed models of structures. This is facilitated by finite element models. These rely on a very fine grid of cells that each capture behavior in a lumped parameter model. Combining all lumped parameter models of this grid results in models of very large order (often more than 1000 states).

The consequence of these modeling approaches is that a model may become prohibitively complex and unsuitable for many tasks. This complexity hampers the use of model based technologies in education and industry. In fact, in industry the same system is often modeled separately and differently by different departments within one company. For example, for each of the control design, control verification, and model based diagnosis tasks different models may be applied.

This drawback can be overcome by the use of reliable and efficient model reduction techniques that operate on one *modelbase* that includes the most detail required in any of the anticipated application tasks. Ideally, this modelbase can be composed from vendor supplied models that accompany the hardware components used in the actual system.

5.2 Methods for Model Reduction

Methods for model reduction deal mainly with abstract mathematical structures. For example, in *singular perturbation* methods [Dauphin-Tanguy, Borne and Lebrun, 1985; Kokotović, Khalil and O'Reilly, 1986] small and large parameters in the system are identified to achieve a system of first order differential equations of the form

$$\dot{x} = f(x,z,t)$$
$$\varepsilon\dot{z} = g(x,z,t)$$

where ε is small, the dot operator signifies the derivative with respect to time, and *x* and *z* are the system state variables. If ε is taken to be 0, the dynamic behavior of the *g* function becomes a set of algebraic equations in *x* and *z*. After solving for *z* and substituting into *f* a reduced order system is arrived at with state variables *x* only.

Another approach, *modal analysis* [Varga, 1994], relies on analysis of the frequency components of a system and removes phenomena with behavior significantly faster than the slow part of the sytem. The basics of this approach can be described by diagonalizing the system matrix, *A*, in

$$\dot{x} = Ax$$

to a form

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

where $\lambda_1 > \lambda_2 > ... > \lambda_n$. Now, the smallest eigenvalues λ_i can be removed to obtain a reduced system. Several criteria can be applied for this selection process [Varga, 1994]. Note that this method is best applied numerically because of the required matrix diagonalization. Also note that the diagonalization process transforms the original state vector, *x*, and direct correspondence with the physical states is lost.

A similar approach without model transformation relies on decoupling loosely connected subsystems [Iwasaki and Bhandari, 1988]. Here, model reduction is achieved by removing small entries in the *A* matrix to derive a number of tightly coupled subsystems.

To apply these linear systems methods to nonlinear systems, linearization around an operating point is required. Typically, this is performed by using the instantaneous vector components of the field at the operating point [Guckenheimer and Holmes, 1986]. The linearization is valid in a restricted area of state space around this point.

Other model reduction techniques operate on a topological model representation, i.e., a bond graph, of the system. From a bond graph model, steady state behavior can be directly derived by replacing all energy storing, integrating, elements by 0 valued sources [Breedveld, 1984]. This indicates that in steady state no change of stored energy takes place, and the resulting algebraic equations describe static behavior. For example, if the cylinder model in Fig. 8 is in steady state, the forces acting on the piston are in balance, i.e., $F_{total} = 0$, and the oil is not further compressed, i.e., $\Delta x = 0$. Thus, the inertia, *I*, is replaced by a 0 valued source of force and the elasticity, *C*, is replaced by a 0 valued source of velocity.

Another bond graph model reduction technique identifies elements that hardly affect overall power in the system, and, therefore, perform a small role in dynamic behavior [Minten *et al*, 1997]. As these elements only partake marginally in determining the dynamic behavior, they can be removed without losing much model accuracy.

Recently, model reduction techniques for specific operational domains are combined with deriving the domain transition behavior [Mosterman and Biswas, 1998; 1999; Mosterman, Zhao and Biswas, 1998a]. Because of the piecewise continuous domains that are combined with discrete domain changes, this is an inherently hybrid approach. Note that this is a crucial link in the chain to achieve comprehensive model reduction methods as all the previously sketched approaches restrict the operational domain (either in frequency or state space) on which the reduction is valid.

5.3 Applying Model Reduction Techniques

In order to develop a comprehensive model reduction approach, available model reduction techniques have to be extended, generalized and integrated. An example of the functioning of the resulting approach is given by modeling a bouncing ball at several levels of detail.

Fig. 13 shows a ball bouncing on a floor. In a continuous model, this can be modeled by a nonlinear spring constant, *C*, that represents the stiffness of the floor. When there is no contact between the ball and floor, $\Delta x < 0$, the force exerted by the floor on the ball is 0. Upon contact, $\Delta x = 0$, the stiffness quickly builds up a force that eventually reverses the ball velocity and it starts moving upward. As soon as, $\Delta x < 0$, the ball disconnects again and the force between the floor and ball becomes 0 again.

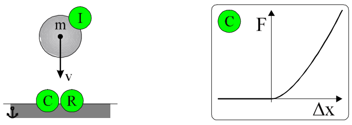

Fig. 13: A continuous nonlinear model of a bouncing ball.

In a first model reduction step, the nonlinear stiffness characteristic can be linearized as shown in Fig. 14. However, the model with linearized stiffness is only a valid approximation when the ball is in contact with the floor, i.e., on a restricted domain in state space. Thus a *mode change* is required to prevent the floor from pulling at the ball when $\Delta x < 0$ which is when it would disconnect. Therefore, the system model comprises two modes: (i) *free*: the ball is modeled as a mass with inertia, *I*, and gravity acting and (ii) *contact*: the ball is modeled as a mass in contact with a stiff floor and gravity acting. Additional complexity arises in terms of switching between modes. Note that though this is a hybrid model changes the velocity changes continuously, i.e., there is no discontinuity, and, therefore, this is referred to as a C^0 hybrid model (i.e., the 0^{th} time derivative of the variables is continuous).

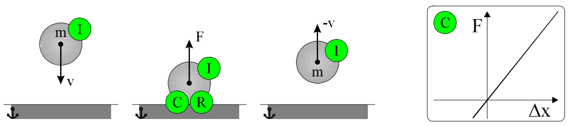

Fig. 14: A C^0 hybrid model of a bouncing ball.

In a final model reduction step, the linear stiffness can be abstracted from the model altogether. Now, an instantaneous change of velocity occurs when the ball reaches the floor. This discontinuous change in velocity requires further complexity of the mode switching function [Mosterman and Breedveld, 1999]. The advantage of this model is that no stiff gradients because of the floor stiffness are present anymore, which allows very efficient simulation.

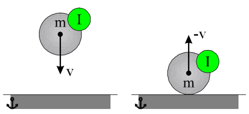

Fig. 15: A hybrid model of a bouncing ball.

In many cases, this model provides a good starting point in developing an understanding of collision behavior. Even more, it is a closer representation of our observation that the ball velocity changes instantaneously than the model that includes stiffness effects. How this velocity change occurs, however, is not clear if the detailed model including small energy storage effects is not available.

5.4 Reverse Analysis

As indicated by the bouncing ball example, model reduction may result in computationally less complex models, but often requires a conceptual investment to derive the discrete switching functions [Breedveld, 1996]. Also, certain physical phenomena are hard to understand without knowledge of the details of physical processes.

This makes a case for performing model *augmentation*. In fact, the presence of discontinuous changes can be exploited to systematically perform a model augmentation analysis. Discontinuous state changes in a physical model indicate that detailed continuous *contact behavior* is abstracted that could be systematically added to the model.

In the hybrid model of the bouncing ball, this pertains to the instantaneous change in velocity because small elasticity parameters have been removed from the model. Reintroducing these provides an excellent teaching opportunity to explain the fast detailed process of storing kinetic energy in potential energy and returning it as kinetic energy again.

Next, the nonlinear stiffness characteristic can be derived from the switched mode linearized model to show how physical system behavior can be modeled as continuous behavior governed by highly nonlinear parameters instead of a hybrid model formulation.

5.5 Summary

A variety of model reduction techniques can be applied to systematically arrive at lower order linear models. Some of these methods apply well in a numerical system formulation where each entry is the aggregate effect of several physical parameters (e.g., an *RC* time constant), but are less applicable when symbolic manipulation is required. As a result, the correspondence with physical parameters after the model reduction step may be lost. Furthermore, model reduction often relies on domains in frequency and state space on which the approximation is valid. Therefore, successful use of model reduction techniques requires methods to automatically identify behavior that occurs when transients between these domains, i.e., mode switches, occur.

Finally, though many model reduction methods exist, model augmentation is hardly addressed at all in literature. As demonstrated, this could be very helpful for instructional purposes in much the same way as model reduction and makes for a promising topic of future research.

6 CONCLUSIONS

This paper discusses and illustrates the role of mixed continuous/discrete, *hybrid*, behavior generation in reactive learning environments, *microworlds*. In order to be effective means for instruction and training, these microworlds rely on behavior generation methods to generate example trajectories of specific system behavior. The learner uses these behaviors to understand and assimilate the underlying model. Furthermore, experience with system behavior can be gained. If simulation is performed in a networked environment, it becomes an excellent means to supports cooperative learning in a possibly distributed setup.

The apparent pitfall in this scheme is the complexity of high fidelity models. These contain behaviors on a hierarchy of temporal and spatial scales that often prohibit efficent, real-time, simulation. To mitigate this problem, it can be exploited that many of these behaviors are too detailed to be of importance on a gross behavior level.

Therefore, they can be abstracted away and efficient simulation achieved.

Abstracting details is the subject of model reduction methods that come in a variety of flavors. Most of these methods deal with approximating model behavior on a restricted operational domain in frequency and state space. Consequently, it is critical to develop methodologies that handle the transition between these domains. As such, the resulting models are of a hybrid nature, and, therefore, hybrid simulation is a key enabling technology.

To effectively deploy such methods in instruction and training environments calls for robust and automatic implementations. This can be realized by generalizing and combining existing methods to establish a powerful comprehensive model reduction approach.

In addition to enabling efficient simulation, model reduction techniques embody much system theoretical knowledge that can be subject of instruction as well. Furthermore, it was shown that a dual to model reduction in the form of model augmentation exists. This can be helpful to discover unmodeled detailed physical phenomena (e.g., small elasticity effects in the bouncing ball model) that in turn explain behavior on a macroscopic level.

Finally, model reduction provides critical support for scaffolding practices in instructional efforts. In order not to overwhelm a learner with detail, the model for a given task can be simplified to match the learner’s competence. Details can be successively revealed to consistently challenge and motivate the learner.

Through all this, it is important for future research to establish theoretical results on the consistency of simplified models and to provide theorems on which characteristics (e.g., stability, steady state behavior) are preserved under certain abstraction.

REFERENCES

P. C. Breedveld; *Physical Systems Theory in Terms of Bond Graphs* (Ph.D. dissertation, University of Twente, 1984)

Peter C. Breedveld; *The context-dependent trade-off between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects* (CESA '96 IMACS Multiconference, Ecole Centrale de Lille, Lille, France, July, 1996)

J. Olin Campbell and Andrew S. Gibbons; *Progressively Challenging Evaluative Simulations That Provide Assessment, Learning, and Performance Support* (Journal of Instruction Delivery Systems, Spring issue, pp. 25-28, 1994)

Cognition and Technology Group at Vanderbilt; *An Anchored Instruction Approach to Cognitive Skills Acquisition and Intelligent Tutoring* (Cognitive Approaches to Automated Instruction, Louis A. Penner, George M. Batsche, Howard M. Knoff and Douglas Nelson, Chapter 6, pp. 135-170, L. Erlbaum Associates, Hillsdale, New Jersey, 1992)

James Dally; *Anchored Instruction in Engineering Education* (The Influence of Technology on Engineering Education, J. Bourne and A Brodersen, Chapter 5, pp. 81-102, CRC Press, Boca Raton, Florida, 1995)

G. Dauphin-Tanguy, P. Borne, and M. Lebrun; *Order Reduction of Multi-time Scale Systems Using Bond Graphs, the Reciprocal System and the Singular Perturbation Method* (Journal of the Franklin Institute, 319(1/2), pp.157-171, 1985)

Ernst von Glaserfeld; *Learning as a Constructive Activity* (Problems of Representation in the Teaching and Learning of Mathematics, C. Janiver, Chapter 1, pp. 3-17, L. Erlbaum Associates, Hillsdale, New Jersey, 1987)

John Guckenheimer and Philip Holmes; *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Applied Mathematical Sciences, vol. 42, Springer-Verlag, Berlin, 1986)

James D. Hollan, Edwin L. Hutchins and Louis M. Weitzman; *STEAMER: An Interactive, Inspectable, Simulation-Based Training System* (The Artificial Intelligence Magazine, vol. 5, nr. 2, pp. 15-27, 1984)

Yumi Iwasaki and Inderpal Bhandari; *Formal Basis for Commonsense Abstraction of Dynamic Systems* (Proceedings of AAAI-88, pp. 307-312, 1988)

Petar V. Kokotović, Hassan K. Khalil and John O'Reilly; *Singular Perturbation Methods in Control: Analysis and Design* (Academic Press, London, 1986)

Myron W. Krueger; *Artificial Reality* (Addison-Wesley, Reading, Massachusetts, 1983)

Willem Minten, Sven Vranckx, Bart de Moor and Joos Vandewalle; Bondlab, a Matlab Based GUI for Bond Graph Modeling (Journal A, vol. 38, nr. 3, pp. 11-15, 1997)

Dieter Moorman, Pieter J. Mosterman and Gert-Jan Looye; *Object-Oriented Computational Model Building of Aircraft Flight Dynamics and Systems* (Aerospace Science and Technology, 1999)

Pieter J. Mosterman; *An Overview of Hybrid Simulation Phenomena and Their Support by Simulation Packages* (Hybrid Systems: Computation and Control, Frits W. Vaandrager and Jan H. van Schuppen, Lecture Notes in Computer Science; Springer-Verlag, vol. 1569, pp. 164-177, 1999)

Pieter J. Mosterman; *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis* (Ph.D. dissertation, Vanderbilt University, 1997)

Pieter J. Mosterman and Gautam Biswas; *A Theory of Discontinuities in Dynamic Physical Systems* (Journal of the Franklin Institute, vol. 335B, nr. 3, pp. 401-439, 1998)

Pieter J. Mosterman and Gautam Biswas; *Deriving Discontinuous State Changes for Reduced Order Systems and the Effect on Compositionality* (Proceedings of the Thirteenth International Workshop on Qualitative Reasoning, pp. 160-168, Lock Awe Hotel, Scotland, June 6-9, 1999)

Pieter J. Mosterman, John R. Bourne, Arthur J. Brodersen and J. Olin Campbell; *Electronics Laboratory Simulator Instructor Manual, Laboratory Manual, and Reference Manual* (Falcon Software, Wentworth, NH, 1995)

Pieter J. Mosterman and Peter Breedveld; *Some Guidelines for Stiff Model Implementation with the Use of Discontinuities* (International Conference on Bond Graph Modeling, pp. 175-180, San Francisco, January, 1999)

Pieter J. Mosterman, J. Olin Campbell, Arthur J. Brodersen and John R. Bourne; *Design and Implementation of an Electronics Laboratory Simulator* (IEEE Transactions on Education, vol. 39, nr. 3, pp. 309-313, 1996).

Pieter J. Mosterman, Marcel A.M. Dorlandt, J. Olin Campbell, Craig Burow, René Bouw, Arthur J. Brodersen and John R. Bourne; *Virtual Engineering Laboratories: Design and Experiments* (Journal of Engineering Education, pp. 279-285, July, 1994).

Pieter J. Mosterman, Feng Zhao and Gautam Biswas; *An Ontology for Transitions in Physical Dynamic Systems* (Proceedings of AAAI-98, July, 1998a)

Pieter J. Mosterman, Feng Zhao and Gautam Biswas; *Sliding Mode Model Semantics and Simulation for Hybrid Systems* (Hybrid Systems V, Lecture Notes in Computer Science, Springer-Verlag, 1998b)

H. Rheingold; *Virtual Reality* (Summit Books, New York, New York, 1991)

Andreas Varga; *On Modal Techniques for Model Reduction* (1st IMACS Conference on Mathematical Modelling MATHMOD, Wien, 2-4 February, 1994)

Etienne Wenger; *Artificial intelligence and tutoring systems. Computational and cognitive approaches to the communication of knowledge* (Morgan Kaufmann, Los Altos, California, 1987)

Barbara Y. White; *ThinkerTools: Causal Models, Conceptual Change, and Science Education* (Cognition and Instruction, vol. 10, nr. 1, pp. 1-100, 1993)

AUTHOR'S BIOGRAPHY

Pieter J. Mosterman was born March 16, 1967 in Nes on the island Ameland off the coast of the Netherlands. In 1985 he started the B.Sc. program at the University of Twente which he received in 1987, followed by his M.Sc. in Electrical Engineering from the Control Laboratory in 1991. In 1992 he enrolled in the Ph.D. program of Vanderbilt University, Nashville, TN, and graduated at the Center for Intelligent Systems in 1997 with a Ph.D. in Electrical and Computer Engineering. Since September 1, 1997 he is a research associate at the Control Design Engineering group of the Institute of Robotics and System Dynamics at the German Aerospace Center.

In the context of his thesis, Dr. Mosterman investigated principles that govern discontinuous behavior in physical system models. Because of the thermodynamic properties of bond graphs, this modeling formalism was used and extended to *hybrid bond graphs* to specify the developed concepts. Furthermore, he established a rigorous mathematical representation based on descriptions common in the hybrid dynamic systems community. Given the application area of model based diagnosis, he developed algorithms for monitoring, prediction, and isolation of abrupt faults in dynamic physical systems. His primary interests are in compositional object-oriented modeling and analysis of complex physical systems. In the context of this work, he created the hybrid bond graph simulator HYBRSIM, that implements novel simulation algorithms based on physical principles. The concepts and notions derived from this work are included in the unifying physical system modeling language Modelica™. Application areas include a monitoring and fault isolation system for fast breeder reactors in Japan, a comprehensive test bed for diagnosing a Chevrolet V-8 internal combustion engine, modeling an electromotor for robot control, and hybrid models of the primary attitude control of aircraft that combine aircraft dynamics as switched continuous behaviors with redundancy management as extensive discrete event models. Dr. Mosterman is chair of the *IEEE CSS Virtual Action Group on Hybrid Dynamic Systems for CACSD* (Computer Aided Control System Design) and a member of the *Modelica Design Group* and of the review board of the *International Journal of Applied Intelligence*. He has served on the program committee of *Eurosim '98*, the *Tenth International Workshop on Principles of Diagnosis*, and the *IEEE SouthEast Conference 2000*.