Model Semantics and Simulation of Time Scale Abstractions in Collision Models

Pieter J. Mosterman*
Inst. of Robotics and System Dynamics
German Aerospace Research Center
D-82230 Wessling
Germany
Pieter.J.Mosterman@dlr.de

Jan F. Broenink†
Department of Electrical Engineering
University of Twente
7500 AE Enschede
The Netherlands
J.F.Broenink@el.utwente.nl

Gautam Biswas‡
Department of Computer Science
Vanderbilt University
Nashville, TN 37235
USA
biswas@vuse.vanderbilt.edu

Abstract

Effective design of models requires simplification of different components of the system. For example, interactions of the system under scrutiny with its environment are modeled as idealized sources and sinks, and system behavior is derived from idealized interactions among constituent components. Some of the more detailed behavior is governed by small parameters, that produce continuous but complex nonlinear behaviors. These nonlinear continuous behaviors are often simplified to piecewise continuous forms, which then results in discontinuities at the temporal or spatial scale of interest. The resulting models are of a mixed continuous/discrete, or *hybrid*, nature. Physical systems theory based on *continuity of power* and *conservation of energy* is well understood. However, once discontinuous effects are included, these laws may be violated at the macroscopic level. The challenge is to endow hybrid models with semantics that do not violate the underlying physical nature of the system. This paper studies the effects and semantics of time scale abstraction, such as embodied by Newton's collision rule. The results show that hybrid modeling of physical systems can be systematic without the need for *ad hoc* hysteresis effects.

Keywords— **hybrid systems, bond graphs, modeling, simulation**

* Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

† Jan F. Broenink is with the Control Laboratory of the University of Twente.

‡ Gautam Biswas is supported by grants from PNC, Japan and Hewlett-Packard, Co.

1 Introduction

Models are constructed to help analyze and design behavior of a system under investigation. An advantage of analyzing models is that one can focus on the system under consideration, and the environment in which the system operates can be made implicit to the analysis. However, most real-world system behaviors are governed by interactions with their environment, which makes it important to explicitly model these interactions. The part of the environment with which the system interacts is its *context*. Details of the rest of the environment are of no concern to the modeler. Because the only interest is the interaction with the system under investigation, the context can be specified by idealized sources and sinks [11].

Another important observation about the purpose of models is that they are constructed to analyze behaviors of interest. Most physical systems operate on a hierarchy of temporal and spatial scales. Often, the modeler's interest is not to analyze behavior at very fast time scales, but more to study the gross continuous system behavior at a macroscopic level. Just like specifying the details of the system context may not be of interest, the details of the fast time phenomena may be of little interest to the modeler, other than how it affects the system at the desired level of detail. Introducing abstractions that lead to simplified interactions between temporal and spatial scales often causes discontinuities in the behaviors at the macroscopic level. The resulting models cover mixed continuous/discrete behaviors, and are called *hybrid systems* [14; 17].

Physical system behavior is governed by the laws of *continuity of power* and *conservation of energy* [19] and the resulting continuous behavior exhibited by these systems is well defined. However, the more detailed continuous behavior, exhibited by complex systems is often nonlinear, and this complicates behavior analysis because [7; 13]:

- Well known linear system analysis techniques and qualitative analysis techniques generate imprecise results.

- System order required to describe such behavior is much higher than that required to study the behavior of interest, and this can result in too much effort in designing models and in ill-constructed models. Further, knowing the correct values of system parameters is critical to analyzing the properties of the system model, yet the exact parameters of the microscopic phenomena are typically hard to obtain.
- Small parameter values producing behaviors at a very fast time scale require very small step sizes for numerical simulation algorithms. This causes the simulation algorithm to be computationally inefficient.

To overcome these difficulties, abstractions of the microscopic phenomena may be introduced into the system model. In previous work [16] we identified two types of abstractions in physical system models: (i) time scale abstraction, and (ii) parameter abstraction. Both time scale and parameter abstraction can cause discontinuities in system behavior. To illustrate, consider a bouncing ball. When the collision is considered perfectly elastic [2], the ball reverses its momentum the moment it makes contact with the floor. In reality, small elasticity effects in the ball and floor store the kinetic energy as potential energy and quickly return it during a short interval of time. The time scale of these phenomena is abstracted into a point of momentum reversal. When the collision is considered perfectly non-elastic, the ball loses all its initial momentum when it makes contact with the floor. In reality, small deformation and other dissipation effects (acoustic, thermal) are present that dissipate the ball's energy very quickly. However, because these parameters are abstracted away the ball appears to intantaneously lose its momentum. Note that this differs from the time scale abstraction where small elasticity effects were not abstracted away, but condensed into a point of time.

Good models of physical systems have a sound theoretical basis in that they are based on laws from the physical domain. The bond graph modeling language [3] provides a well defined formalism for expressing and developing physical system models where the basic physical principles, such as conservation of energy and continuity of power are directly enforced. When model abstractions require discontinuities to be introduced into physical system models, the basic laws that govern physical system behavior seem to be violated [8; 17]. Our goal in this paper is to go beyond a pure mathematical description for hybrid system models [1; 9; 10] by establishing physical semantics that govern the behavior of hybrid system models. These established semantics are then incorporated into the bond graph framework to facilitate the building and analysis of hybrid system models.

In particular, this paper investigates the nature and effect of discontinuities created by introducing time scale abstractions to simplify models for 1-dimensional collision effects. Section 2 describes the collision chain phenomena in the device Newton's Cradle, and presents a mathematical representation of these in the form of *pinnacles* in state space [16]. It shows how these collision chains can be systematically modeled by the bond graph simulator *20sim* [5; 6]. Section 3 investigates the semantics for the collision points. Section 4 presents the conclusions and a discussion of this paper.

2 Newton's Cradle

Collision phenomena typically occur at time scales much smaller than the time scale of gross system behavior. Therefore, analysis of collisions use short duration *impulsive forces*, P, rather than the typical *forces*, F, observed in nature [2]. These impulsive forces result from abstracting the effect of collision forces that exist for very short intervals of time. In the limit, it can be assumed that the collision impulses act only for a point in time. At this point, the velocities of the colliding bodies instantaneously change from one value, the *a priori* velocity, to another, the *a posteriori* value.

2.1 Problem Analysis

Consider the collision between the two bodies in Fig. 1. In reality, when collision occurs, small elasticity effects store the kinetic energy over a short period of time and return the stored energy during a contiguous short interval of time. Because the time scale of this phenomenon is very small compared to the behavior of interest, it can be modeled as an instantaneous change at a point in time governed by Newton's collision rule [4]

$$v_2^+ - v_1^+ = -\epsilon(v_2 - v_1), \tag{1}$$

where ϵ is the coefficient of restitution, which determines the loss of energy in the system. When $\epsilon = 1$ this represents an ideal or perfectly elastic collision where no energy is lost, whereas $\epsilon = 0$ represents a perfectly nonelastic collision where a significant amount of energy is lost. Note that this does not imply that all energy is lost for an inelastic collision.

Time scale abstraction, which results in the fast collision phenomena being reduced to a point in time, may create the impression that the conservation of energy principle is violated. But the physical principle of *conservation of state* still needs to hold despite the abstraction of a small behavior to a point. This principle is a generalized form of conservation of charge, momentum,

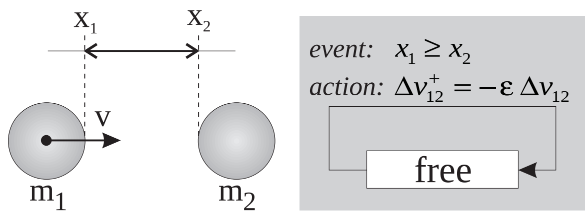

Figure 1: **A 1-dimensional collision between two bodies.**

etc. [17]. In this system, the generalized state is momentum and conservation requires ($m_1 = m_2$)

$$v_2^+ + v_1^+ = v_2 + v_1 \tag{2}$$

Combined with the collision rule in Eq. (1) this results in the instantaneous change of velocity upon collision

$$\begin{cases} v_1^+ = \frac{1-\epsilon}{2} v_1 + \frac{1+\epsilon}{2} v_2 \\ v_2^+ = \frac{1+\epsilon}{2} v_1 + \frac{1-\epsilon}{2} v_2 \end{cases} \tag{3}$$

This algebraic relation only holds at a point in time. As a result, the application of this relation may cause a discontinuous change in the velocities v_1 and v_2. Therefore, these changes have to be implemented as transition actions, where the value change is accompanied by a change in the mode or state. This corresponds to a Mealy type state machine [20].

Fig. 1 illustrates the state machine implementation, where the collision action that changes v_1 and v_2 is executed as a self-loop when $x_1 \geq x_2$. However, the corresponding action does not disable the discrete transition, and in a corresponding implementation model this transition would be executed infinitely many times. In fact, no new configuration is realized once the transition condition is invoked. Given that discontinuous changes are instantaneous, no real time elapses during the transitions. This violates the principle of *divergence of time* that states that any physical system behavior always progresses in time [15].

To prevent this loop of instantaneous transitions, the assumptions under which the coefficient of restitution yields a correct approximation need to be investigated. In reality, this coefficient is a function of the impact velocity [2] and tends to 0 when this impact velocity becomes small. Therefore, an additional condition to ensure the validity of the approximation is given by $v_1 - v_2 > v_{th}$, i.e., the difference in velocities has to exceed a threshold for the elastic collision phenomena to occur. If this difference is taken to be an infinitesimal value ($v_{th} \to 0$), the additional constraint becomes $v_1 > v_2$. Now, when first $x_1 \geq x_2$, collision occurs and the velocities are modified according to Eq. (3). The new values violate the $v_1 > v_2$ constraint and no further discrete events follow.

This shows that switching specifications have to embody assumptions made with regard to the abstractions that result in discontinuous behavior. A necessary condition for incorporating these assumptions is the divergence of time principle. In previous work, we have employed a multiple energy phase space analysis to verify physical correctness of the hybrid models [14; 15]. In other work, *ad hoc* hysteresis effects are introduced to prevent the loop of instantaneous changes [4; 12]. Though this eliminates the problem during simulation, it may cause modeling misconceptions.

In general, to enforce divergence of time, one can either:

- model the discontinuous phenomena in more continuous detail, or
- strengthen the discrete switching conditions to ensure there is only one possible mode for a given state vector.

Adding more continuous detail is often undesired because it increases the computational complexity of the simulation task. To strengthen switching specifications modeling assumptions that lead to discontinuity have to be made explicit. For the collision example, divergence of time can be enforced by adding the $v_1 > v_2$ condition to the transition condition.

2.2 Simulation in 20sim

Hybrid system models mix continuous behavior evolution with discontinuities that can be attributed to configuration or *mode* changes. These mode changes are invoked by discrete events that are generated when during continuous simulation pre-specified threshold values are crossed. At this point, continuous simulation is halted, and the discrete model is executed. One discrete event may cause a sequence of further events, and continuous simulation resumes after the configuration changes have converged to a new consistent mode. When a mode change occurs, the state vector values may change discontinuously, or even change in size. In this paper, discontinuous state changes are explicitly specified by transition actions in the discrete model. Situations where discrete transitions cause changes in the state vector are dealt with elsewhere [13; 14].

To develop a simulation system for 1-dimensional collisions, we used the physical modeling and simulation package *20sim* [6]. *20sim* provides an elegant modeling environment for continuous physical system models. It incorporates both the bond graph and block diagram modeling formalisms. However, *20sim* does not support discrete event modeling, e.g., in the form of finite state machines and Petri nets. The challenge is whether physical system modeling and simulation packages such as

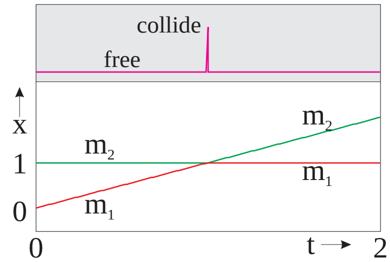

Figure 2: **Simulation of a collision between two bodies.**

20sim can be augmented to facilitate modeling and simulation of hybrid systems.

Presently, *20sim* has extensions that facilitate modeling of hybrid systems by explicitly constructing (sub)models as sorted assignment statements instead of unsorted equations. Furthermore, it offers a reset function for state variables. The collision models in this paper are formulated as assignment statement descriptions, that consist of two parts:

- the continous model differential equations of the bodies in the discrete *free* state.
- the finite state machine, separated into four parts (Algorithm 1):
  1. event detection,
  2. discrete state transition,
  3. transition actions, and
  4. restarting the continuous model part by solving the initial value problem of the newly inferred discrete state.

A *20sim* simulation result of the two colliding bodies in Fig. 1 is shown in Fig. 2. Body m_1 moves with a constant velocity towards m_2 that is at rest. Upon collision all of m_1's momentum is transferred instantaneously to m_2. Therefore, after the collision, m_1 is at rest and m_2 moves with a constant velocity.

Algorithm 1 FSM description of two bodies model in *20sim* version 2.2.

```
# FSM event detection: a collision
collide := (discrState == free) and
  collide == 0 and abs(distance) <= 0 and abs(v1 - v2) > 0
# FSM state transition: change the state of the FSM
discrState := if collide then free else discrState
# FSM action: at collide transition: momentum redistribution
newp1 := (m1 * (p1 + p2) - eps * m1 * m2 * (v1 -v2)) / (m1 + m2)
newp2 := p1 + p2 - newp1
dummy1 := resetfunc(collide, p1, newp1)
dummy2 := resetfunc(collide, p2, newp2)
# 'Restart' the IVP: recalculate the velocities,
# since the momenta changed after a state transition
v1 := p1 / m1
v2 := p2 / m2
```

From Algorithm 1 it is clear that it is straightforward to implement a graphical finite state machine editor that automatically generates discrete model code. Note that the *20sim* simulator currently only implements the state reset function without root-finding (i.e., locating the discrete event within a small temporal margin), which can lead to numerical inaccuracies, and make the result dependent on the integration method used.

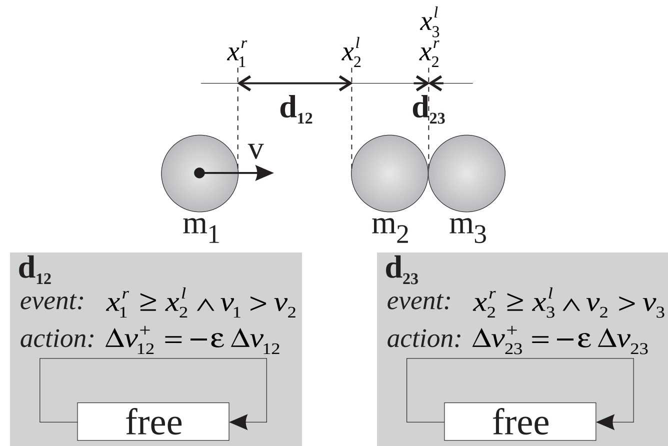

Figure 3: **A collision between three bodies.**

2.3 Collision Chains

The described model handles one discontinuous change in state well. However, a number of collisions may occur, seemingly at the same instant in time. Consider a collision chain between three bodies (Fig. 3) where each transition action embodies Newton's collision rule. Fig. 4 shows a *20sim* simulation of the transfer of momentum between the three bodies. Note that body 2 takes on the complete initial velocity of body 1 before transferring it to body 3. This point in state space represents a so-called *pinnacle* and in simulation occurs at the same point in time where body 3 takes over the initial momentum [13; 16].

Pinnacles represent points in state space that are immediately departed when the new initial values are calculated. The reason for the immediate departure is the *dynamic coupling* between the discrete switching specifications. When the transition actions modify energy state variables, e.g., velocities, this may result in further transition conditions that are enabled. For example, in case of the three colliding bodies (Fig. 3) the $v_2 > v_3$ pre-condition for transition is satisfied immediately after the collision between m_1 and m_2 has changed v_2. All these discrete changes are executed at the same simulation time point. Once no further changes occur (given divergence of time this is guaranteed [14]), simulation of the continuous behavior resumes.

In previous work [13; 16] we have shown that such a sequence of mode changes can be attributed to time scale abstractions. The continuous state vector is mod-

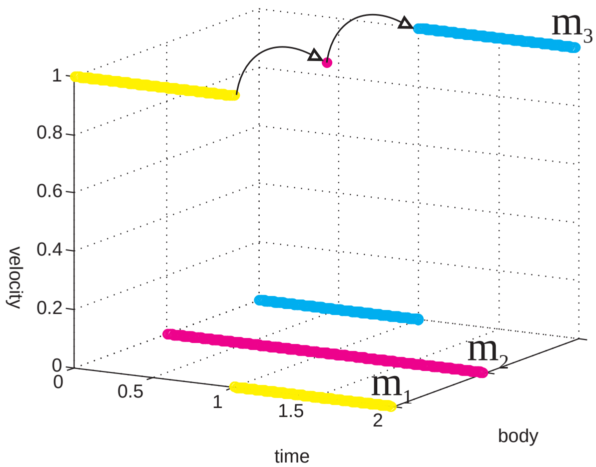

Figure 4: **Simulation of an ideal elastic collision between three bodies.**

ified as part of a transition action. Therefore, each mode in a sequence affects the system state, even though it is departed immediately after it becomes active (exemplified by the collision chains). A second class of mode changes occur because of parameter abstractions used to simplify nonlinear system behavior. The modes generated because of parameter abstraction correspond to virtual points in state space, i.e., they have no real existence on the time line, and, therefore, the mode is departed before the state vector is affected. Such mode changes are discussed elsewhere [14; 15].

3 Collision Semantics

In case of multiple collisions, several pinnacles may be traversed which in simulation all occur at the same point in time. However, semantically these points do not coincide but they follow each other immediately in time. This can be illustrated by investigating the underlying physical behavior. In reality, small elasticity coefficients are present that store the kinetic energy before passing it on, which requires a small amount of time. When this elasticity is modeled (Fig. 5), simulation shows that the point where body 2 takes on the velocity does not coincide with the point where velocity has been transferred to body 3 ($m_1 = m_2 = m_3 = m$). No matter how small the coefficient of elasticity, which governs the rate of transfer, body 2 captures the initial momentum from body 1 before it passes it on to body 3.

Fig. 6(a) shows this behavior for a perfectly elastic collision, $m_1 = m_2 = m_3 = m$. After collision, m_1 comes to rest. If the collisions were modeled to occur at the same instant in time, m_2 and m_3 would act as one body with mass $2m$, and behavior would be distinctly different. Fig. 6(b) shows a simulation of this phenomenon. In

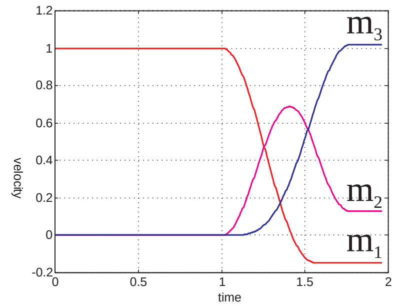

Figure 5: **An elastic collision between three bodies with small elasticity coefficients.**

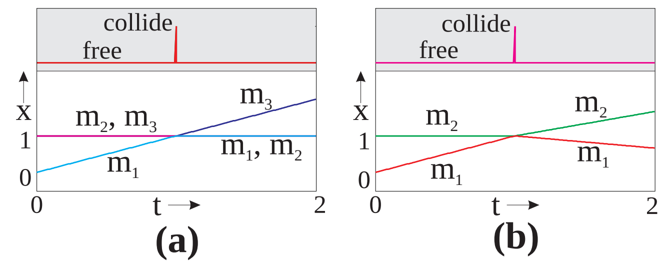

Figure 6: **Differences in behavior of one body, m_1, with mass m when (a) colliding with two bodies of equal mass, m, and (b) one of mass $2m$.**

this scenario, m_1 collides with body, m_2, with twice the mass of m_1, and m_1 obtains a negative velocity instead of coming to rest.

To investigate the consistency of the hybrid models, consider the collision in Fig. 7. If $m_1 = m_3 = m$ and $m_2 = 2m$, m_1 has an initial velocity $v_1 = v$, and velocity of m_3 is $v_3 = -v$. Depending on the initial positions, three scenarios may occur:

1. m_1 collides with m_2 first. In this case, $v_1^+ = -\frac{1}{3}v$ and $v_2^+ = \frac{2}{3}v$ after which m_2 collides with m_3 and $v_2^+ = -\frac{4}{9}v$ and $v_3^+ = \frac{11}{9}v$.

2. m_3 collides with m_2 first. In this case, $v_3^+ = \frac{1}{3}v$ and $v_2^+ = -\frac{2}{3}v$ after which m_2 collides with m_1 and $v_2^+ = \frac{4}{9}v$ and $v_3^+ = -\frac{11}{9}v$.

3. m_1 and m_3 collide with m_2 at exactly the same time. In this case the momentum transfers through m_2 between m_1 and m_3 and $v_1^+ = -v$, $v_2^+ = 0$ and $v_3^+ = v$.

These scenarios depict three very distinct behaviors, depending on the order in which the collisions occur as determined by the initial distances between the bodies. This sensitivity to a small perturbation in initial conditions (in this case the distance between the bodies) is inherent to hybrid systems in particular and nonlinear systems in general. In this system, the continuous state

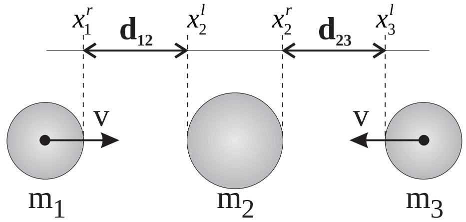

Figure 7: **Collision between three bodies with different mass.**

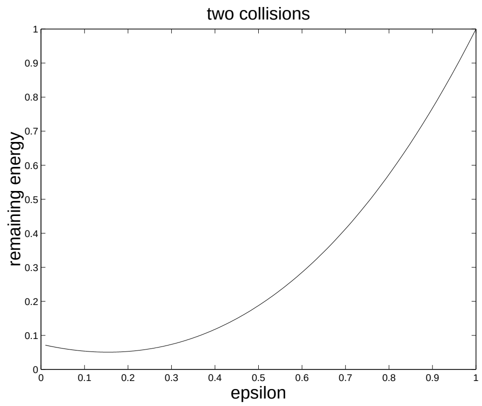

Figure 8: **Energy remaining after two collisions for different ϵ.**

space is partitioned into three segments, each with distinctly different behavior. A small change in initial condition may move the system from one segment into the other. In reality, the boundaries between these segments are not as crisp as in the idealized situation because of higher order, nonlinear, continuous effects. Therefore, when the distance of m_1 to m_2 and that of m_2 to m_3 are close but not quite equal, the actual behavior is somewhere in between both extremes. Future research includes finding correct physical semantics for hybrid systems that are less sensitive to the initial conditions. This may be possible based on energy analysis. Note that the energy in the system after two collisions is a function of the coefficient of restitution, Fig. 8. The degradation of the quality of the collision approximation is shown by the behavior of this function for small ϵ. When $\epsilon = 0$, the maximum loss of energy should be obtained. However, the function has a minimum around $\epsilon = 0.154$.

Now that we have a system sensitive to the exact time of collision, we can investigate what semantics yield a consistent formulation. Consider the system in Fig. 9 with $m_1 = m_3 = m_4 = m$, $m_2 = 2m$. If m_1 and m_4 are positioned such that the distance between m_1 and m_2 (d_{12}) equals the distance between m_2 and m_4 (d_{24}) minus the diameter of m_3 (two times its radius r_3), i.e.,

$$d_{12} = d_{24} - 2r_3, \tag{4}$$

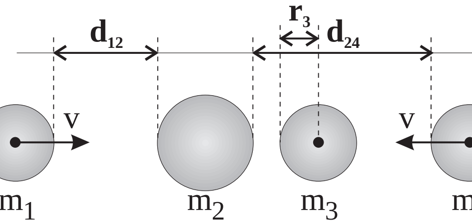

Figure 9: **Collision between four bodies with different mass.**

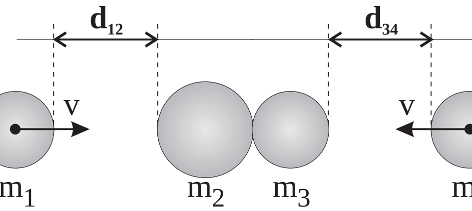

Figure 10: **Another collision between four bodies with different mass.**

and when m_1 and m_4 are given an initial velocity with equal magnitude, v, but opposite directions, m_4 first collides with m_3. If the collision is assumed perfectly elastic, m_3 takes on all of the intial momentum of m_4. Next, one of the three previously discussed scenarios occurs. If the collision between m_3 and m_4 occurs in 0 time, then m_1 and m_3 collide with m_2 at the exact same time, momentum transfers through m_2 without affecting its velocity, i.e., it remains at rest.

A special case of the collision experiment between the four bodies is presented in Fig. 10. Now, m_3 is positioned such that it is in contact with m_2, $d_{23} = 0$. When m_1 and m_4 are located as before, $d_{12} = d_{34}$, and the collision between m_4 and m_3 coincides with the collision between m_1 and m_2. However, this latter collision causes m_2 to obtain a new velocity $v_2 \neq 0$. Therefore, the behavior is distinctly different, though for consistent model semantics it should be the same. Different behavior for $d_{23} = 0$ could be justified by attributing this to m_2 and m_3 being connected. However, if m_2 and m_3 are truly connected bodies, upon collision they act as one body with mass $m_2 + m_3$ and this results in yet a different behavior. This behavior conflicts with the collision model that results in correct behavior for the collision in Fig. 3 where m_2 first takes on all initial momentum before passing it to m_3, i.e., they are separated.

To obtain consistent semantics, it is reiterated that the collisions are phenomena that are abstracted to occur in an infinitely small interval. This implies that when m_1 and m_4 are positioned at equal distances from m_2, accounting for the diameter of m_3, their collisions with m_2 do not coincide. In fact, m_1 collides first, which is exactly what is obtained when m_3 is positioned to be in contact with m_2.

This thought experiment shows that in order to achieve consistent hybrid systems, the semantics of configuration changes due to time scale abstraction should be implemented to occur in an infinitesimally small amount of time. This is in contrast with the mathematical hybrid systems approach [1; 9; 10] where all configuration changes are considered to occur in 0 time and, therefore, result in inconsistent models.

4 Conclusions

Physical systems operate on a hierarchy of temporal and spatial scales. Much like interactions with the environment, the detailed, continuous, behavior can be modeled by ideal effects. Often this results in discontinuities and the corresponding models are of a mixed continuous/discrete, hybrid, nature. In previous work we have identified two types of abstraction that lead to discontinuous behavior: (i) time scale abstraction and (ii) parameter abstraction [13; 16]. In this paper we discuss simulation and model semantics of time scale abstraction in detail.

Hybrid models exhibit continuous behaviors within modes of operation, and discrete transitions between modes when threshold value are crossed. During these discrete changes, conservation of energy and continuitiy of power appear to be violated. In previous work we have established the principle of *divergence of time* as a necessary condition for physical correctness of discrete mode changes [17]. This paper shows how this principle aids in revealing implicit modeling assumptions which yields insight in the modeling abstractions. Furthermore, it shows how these assumptions, when made explicit, result in models that are consistent with physical principles. In other work, *ad hoc* hysteresis effects are introduced to facilitate the simulation task [4; 12]. Though this can be physically justified, it cannot be linked to the modeling assumptions that form the basis for inconsistencies.

Due to dynamic coupling, one discrete change may invoke a number of consecutive changes. We show how the semantics of these changes caused by time scale abstraction, have to rely on infinitely small intervals of time ($\Delta t \rightarrow 0$) to result in consistent models. This differs from the mathematical approach where such discrete changes are said to occur in 0 time, and sequences of changes occur at the same point in time, distinguished by an additional index that represents their place in the transition sequence [1; 9; 10].

The examples in this paper illustrate that hybrid systems are sensitive to small perturbations in initial conditions. In general, this is a characteristic of nonlinear systems, where entirely different behavior may ensue as a result of a small perturbation in the initial value. However, in reality this sensitivity is much less extreme. Presently, these phenomena are not facilitated by state of the art simulation tools. Future research focuses on physical semantics for hybrid behavior closer to reality and an implementation in modeling and simulation tools such as *20sim* to obtain robust hybrid system simulation results. This research is related to exloiting the effect of small parameter values in sliding mode semantics [18], which allows behavior generation for a mathematically singular plane in state space. Eventually, we aim to develop an extensive ontology for hybrid models of dynamic physical systems. Furthermore, we will enhance *20sim* with proper graphical editor facilities to elegantly specify hybrid systems with finite state machines, and add exact event localisation to the simulation engine.

References

[1] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Lecture Notes in Computer Science*, volume 736, pages 209–229. Springer-Verlag, 1993.

[2] Raymond M. Brach. *Mechanical Impact Dynamics*. John Wiley and Sons, New York, 1991.

[3] P.C. Breedveld. Multibond graph elements in physical systems theory. *Journal of the Franklin Institute*, 319(1/2):1–36, January/February 1985.

[4] Peter C. Breedveld. The context-dependent tradeoff between conceptual and computational complexity illustrated by the modeling and simulation of colliding objects. In *CESA '96 IMACS Multiconference*, Lille, France, July 1996. Ecole Centrale de Lille.

[5] Jan F. Broenink. *Computer Aided Physical Modeling and Simulation: A Bond Graph Approach*. PhD dissertation, University of Twente, Enschede, The Netherlands, 1990.

[6] Jan F. Broenink. Modelling, simulation and analysis with 20-sim. *Journal A*, 38(3):22–25, January 1998. Special CACSD issue.

[7] Jan F. Broenink and Paul B.T. Weustink. A combined system simulator for mechatronic systems. In *Proceedings of ESM 96*, Budapest, Hungary, June 1996.

[8] Jan F. Broenink and Klaas C.J. Wijbrans. Describing discontinuities in bond graphs. In *Proceedings of the International Conference on Bond Graph Modeling*, pages 120–125, San Diego, California, 1993.

[9] Akash Deshpande and Pravin Varaiya. Viable control of hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 128–147. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[10] John Guckenheimer and Stewart Johnson. Planar hybrid systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999, pages 202–225. Springer-Verlag, 1995. Lecture Notes in Computer Science.

[11] Derek J. Hatley and Imtiaz Pirbhai. *Strategies for Real-Time Systems Specification*. Dorset House Publishing Co., New York, New York, 1988.

[12] Yumi Iwasaki, Adam Farquhar, Vijay Saraswat, Daniel Bobrow, and Vineet Gupta. Modeling time in hybrid systems: How fast is "instantaneous"? In *1995 International Conference on Qualitative Reasoning*, pages 94–103, Amsterdam, May 1995. University of Amsterdam.

[13] Pieter J. Mosterman. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis*. PhD dissertation, Vanderbilt University, 1997. http://www.op.dlr.de/FF-DR-ER/staff/pjm/papers/thesis/p.html.

[14] Pieter J. Mosterman and Gautam Biswas. A Formal Hybrid Modeling Scheme for Handling Discontinuities in Physical System Models. In *AAAI-96*, pages 985–990, Portland, Oregon, August 1996. AAAI Press, 445 Burgess Drive, Menlo Park, CA, 94025.

[15] Pieter J. Mosterman and Gautam Biswas. Verification of dynamic physical system models. In *ASME-96*, pages 707–714, Atlanta, GA, November 1996.

[16] Pieter J. Mosterman and Gautam Biswas. Principles for Modeling, Verification, and Simulation of Hybrid Dynamic Systems. In *Fifth International Conference on Hybrid Systems*, Notre Dame, Indiana, September 1997.

[17] Pieter J. Mosterman and Gautam Biswas. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*, 334B(6), 1997.

[18] Pieter J. Mosterman, Feng Zhao, and Gautam Biswas. Model semantics and simulation for hybrid systems operating in sliding regimes. In *AAAI Fall Symposium on Model Directed Autonomous Systems*, 1997.

[19] Henry M. Paynter. *Analysis and Design of Engineering Systems*. The M.I.T. Press, Cambridge, Massachusetts, 1961.

[20] K.C.J. Wijbrans. *Twente Hierarchical Embedded Systems Implementation by Simulation: a structured method for controller realization*. PhD dissertation, University of Twente, CIP-DATA Koninklijke Bibliotheek, Den Haag, The Netherlands, 1993. ISBN 90-9005933-4.