Proceedings of the ASME 2009 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA

DETC2009-86954

SYMBOLIC MODEL REDUCTION FOR INTERVAL-VALUED SCENARIOS

Lars Mikelsons
Institute for Mechatronics and System Dynamics
University of Duisburg-Essen
47057 Duisburg
Germany
Email: mikelsons@imech.de

Thorsten Brandt
Institute for Mechatronics and System Dynamics
University of Duisburg-Essen
47057 Duisburg
Germany
Email: brandt@imech.de

ABSTRACT

In many cases, the quantitative relevance of physical effects for a given technical problem is not known a priori. This holds especially for the analysis of the dynamics. Adopted from non-analog circuit design, in the last years symbolic model reduction techniques found their way towards mechatronic system modeling. Given a scenario (system inputs, initial values, parameters) and an error bound, symbolic model reduction reduces the detailed model to a less complex model, which is guaranteed to stay within predefined error bounds. However, presently symbolic reduction techniques deliver reduced models, which are only verified for a single scenario. For example a reduced vehicle model emerging from the reduction of a complex multibody vehicle model for a cornering maneuver with a small constant steering angle, is not verified to stay inside the error bounds for any other maneuver. In this contribution this drawback is addressed by the use of interval-valued scenarios.

1 INTRODUCTION

Having a detailed model of a technical system at hand, often it is not clear which physical effects have a major influence on the solution. Therefore reducing the complexity of a technical system manually is a hard task mainly relying on the intuition and experience of the design engineer. This holds even more for models build up by simulation-tools like Dymola [1] or Adams [2], because the emerging mathematical model is mostly hard to interpret. Hence, algorithms for the automated model reduction are required. Symbolic reduction techniques were first developed to generate behavioral models of non-linear analog electrical circuits [3,4]. Behavioral models are particular helpful for example in circuit design. Mostly used for behavioral model generation is the so-called DC-analysis, which considers the circuit in the steady state. Hence, ranking procedures were developed only for the DC-analysis. Transient ranking procedures were presented in [5]. Having transient ranking procedures for arbitrary DAE-systems at hand, the symbolic reduction techniques can be effectively used for the reduction of any mechatronic system [6]. As opposed to the DC-analysis, for the transient analysis a simulation scenario (initial values, system inputs and parameters) must be chosen. Thus, here the symbolic reduction techniques deliver a reduced model, which is verified only for the chosen scenario. This is a huge drawback especially for systems which allow for user engagement. For such systems the identification of significant scenarios is an additional hard task. As an example, a cornering maneuver or a straight ahead travel are typical maneuvers for a vehicle, but the parameters like velocity or steering angle can vary in wide range. Here this drawback is tried to overcome by means of interval methods for the transient analysis, which is mathematically speaking an initial value problem. Using interval methods interval-valued initial values and/or parameters can be treated [7].

This contribution is structured as follows. First symbolic reduction techniques are presented. Then interval methods for initial value problems are shown and integrated into the reduction algorithm. Afterwards the results are presented. It will be seen, that

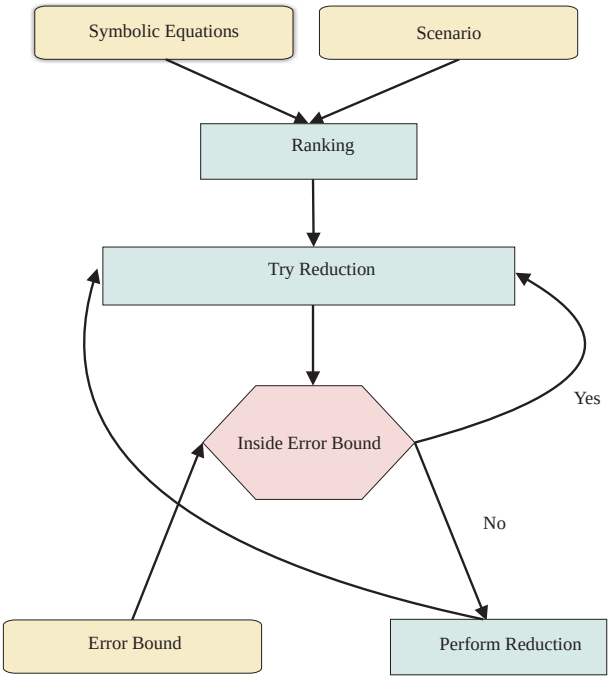

Figure 1. SCHEME OF THE REDUCTION ALGORITHM

the presented algorithm works fine for small systems, but fails for complex ones. Hence, in the outlook further approaches, which are part of current research, are discussed. Furthermore a conclusion is drawn.

2 SYMBOLIC MODEL REDUCTION

In order to extend existing symbolic reduction techniques for the use of interval-valued scenarios, first the state of the art is reviewed. The basic idea of symbolic model reduction techniques is to identify those terms of a DAE (or ODE) system, whose influence on the solution of the system is minor, and to perform a reduction on them (e.g. to neglect them). The algorithm consists of two steps, see for example [6] and [8]. First, for a specific reduction the influence of each term on the solution of the DAE-System is estimated in the so called "ranking". Then the terms are sorted ascendingly with respect to their influence on the solution in order to perform the reductions as long as the solution of the reduced DAE-System remains within a user-defined error bound ε [8]. Possible reduction techniques are negligence of terms, setting terms to constants, linearization of terms or symmetry considerations. While the first three reductions are operations on terms of the DAE-System, the last one operates on variables as follows: First, all variables are checked against each other for being symmetric in the sense, that they have similar values similar along the whole simulation. For two similar variables every occurrence of the first variable is substituted by the second variable. One equation (the one leading to the smallest error) is deleted. The symbolic reduction algorithm is visualized in Fig.1 for a chosen reduction technique. Given a scenario (system inputs,initial states and parameters) and an error bound, the algorithm starts with the ranking. Here, the influence of performing a reduction on a specific term is estimated by a simplified simulation. Afterwards it is checked whether the reductions lead to an error inside the error bounds, beginning with the smallest. Finally, a less detailed model, performing within the prescribed error bounds results.
Let now

$$\mathbf{F} : \Omega \times I \mapsto \mathbb{R}^m \tag{1}$$

be differentiable, where $\Omega \subset \mathbb{R}^n \times \mathbb{R}^n$ is an open set. Then

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) = 0 \tag{2}$$

is called DAE-system if $\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}$ is singular. Furthermore, let $\mathbf{F}$ be given in expanded form

$$\mathbf{F}_i(\mathbf{x}, \dot{\mathbf{x}}, t) = \sum_{k=1}^{l_i^1} t_{k_i}^1(\mathbf{x}, \dot{\mathbf{x}}, t), \quad 1 \leq i \leq m, \tag{3}$$

where l_i^1 is the number of terms in $\mathbf{F}_i$ and $t_{k_i}^1$ denotes the k-th term in $\mathbf{F}_i$. Each term in the first level $t_{k_i}^1$ may consist of a function $f_{k_i}^1$, whose argument is a sum of l_i^2 second level subterms $t_{k_i}^2$ $(1 \leq i \leq l_i^2)$

$$t_{k_i}^1(\mathbf{x}, \dot{\mathbf{x}}, t) = f_{k_i}^1(\sum_{k=1}^{l_i^2} t_{k_i}^2(\mathbf{x}, \dot{\mathbf{x}}, t)), \tag{4}$$

and so on. Here level indicates the hierarchy of arguments nested into each other in each single summand. Then the set $\mathcal{T}^i$ is the set of all terms in the i-th level. The manipulation of a term is called reduction in the following. Consequently, for the set of all reductions $\mathcal{K}^i$ for one reduction technique in a level i, it holds

$$\left|\mathcal{T}^i\right| = \left|\mathcal{K}^i\right|. \tag{5}$$

For $\kappa \in \mathcal{K}$

$$\mathbf{F}^\kappa = 0 \tag{6}$$

is the DAE-system emerging from the reduction κ. Then for DAE-systems of the form of Eqn. 2

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{u}) = 0 \tag{7}$$

with system inputs $\mathbf{u}$, a scenario is the set of a vector field defined on the interval I for the system inputs, the initial values and the parameters. Furthermore, $\mathcal{N}(F(\mathbf{x}, \dot{\mathbf{x}}, t), \mathbf{u})$ is the solution of eqn. 2 computed by a numerical integrator $\mathcal{N}$ at nodes $t_1, \ldots, t_N$. The solution

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_{out} \\ \bar{\mathbf{y}} \end{bmatrix} = \mathcal{N}(F(\mathbf{x}, \dot{\mathbf{x}}, t), \mathbf{u}) \tag{8}$$

consists of two components. In $\mathbf{y}_{out}$ the n_{out} output variables are contained, while $\bar{\mathbf{y}}$ consists of the remaining internal variables.

2.1 Reduction Techniques

As mentioned above possible reduction techniques are the negligence of terms ($\mathcal{U}_{neg}$), setting terms to constants ($\mathcal{U}_{const}$), symmetry considerations or the simplification of piecewise functions. Throughout this paper only the first two reduction techniques are considered. While the negligence of terms does not require any information about the term under consideration, setting to a constant does. Using $\mathcal{U}_{const}$, usually the considered term is replaced by its mean value throughout the simulation and hence depends on the scenario. Certainly, that mean value has to be computed performing a reference simulation first. However, this is no drawback, since a reference simulation is essential for the ranking anyway.

2.2 Ranking

In [5] different ranking algorithms are proposed. Here the so called One-Step- and the Residual-Ranking will be discussed. In general a ranking procedure estimates the influence of a reduction on the solution of a DAE (or ODE) system. Usually the error emerging from the reduction is considered. To estimate that error a reference solution $\mathbf{y}^\star$ is required. The crux of the matter is that the quality of the estimate raises with the duration of the ranking procedure. Clearly, an fast and accurate ranking procedure is sought. Mathematically speaking a ranking procedure $\mathcal{R}$ maps two DAE-systems on a real value, estimating the error between their solutions. Apparently, performing a simulation for each reduction would lead to the perfect estimate, but last very long.

2.2.1 Residual Ranking

Let $\mathbf{y}^\star$ be the reference solution calculated by a numerical integrator $\mathcal{N}$

$$\mathbf{y}^\star = \mathcal{N}(\mathbf{F}). \tag{9}$$

Then

$$\left\| \mathbf{F}(\mathbf{y}^\star(t_i), \dot{\mathbf{y}}^\star(t_i), \mathbf{u}(t_i)) \right\| < \eta, \quad (1 \le i \le N) \tag{10}$$

holds for a prescribed accuracy η. Hence,

$$\mathcal{R}_{res}(\mathbf{F}, \kappa) = \left\| \begin{bmatrix} \left\| \mathbf{F}^\kappa(\mathbf{y}^\star(t_1), \dot{\mathbf{y}}^\star(t_1), \mathbf{u}(t_1)) \right\| \\ \vdots \\ \left\| \mathbf{F}^\kappa(\mathbf{y}^\star(t_N), \dot{\mathbf{y}}^\star(t_N), \mathbf{u}(t_N) \right\|) \end{bmatrix} \right\| \tag{11}$$

seems to be a good estimate for the error resulting from the reduction $\kappa \in \mathcal{K}$. The residual ranking is a very fast ranking procedure due to its simplicity.

2.2.2 One-Step Ranking

Typically, computing the solution of a DAE-system, at each time step a non-linear system of equations is iteratively solved. Usually the solution of the preceding time step is used as the initial value for the solution of the system of non-linear equations at the next time-step. For the computation of the solution of Eqn.6, the reference solution $\mathbf{y}^\star$ at the corresponding time steps can be used for the initial values. Now, additionally limiting the iterations to one, a estimate of the solution of Eqn.6 $\widehat{\mathbf{y}}$ is obtained. Consequently

$$\mathcal{R}_{step}(\mathbf{F}, \kappa) = \| \mathbf{y}^\star_{out} - \widehat{\mathbf{y}_{out}} \| \tag{12}$$

is computed. The one-step ranking is more accurate than the residual ranking and delivers a good compromise between accuracy and runtime.

2.3 Term Cancellation

During the term cancellation procedure reductions are performed as long as the emerging error remains within the prescribed error bound ε. The emerging error is measured only at the n_{out} output variables. Thus, ε has dimension n_{out}. To perform as many reductions as possible, it is beneficial to start with those reductions, which lead to a small error. Thus, first the set of reductions $\mathcal{K}$ is sorted in ascending order depending on the ranking, resulting in $\mathcal{K}_{sort}$. Now, one possibility is to check one reduction of $\mathcal{K}_{sort}$ after the other. This is done by checking the computed solution of the reduced DAE-system for staying within the error bound ε. However, this method can be accelerated by the use of clusters [9]. Using clusters, the set of reductions $\mathcal{K}_{sort}$ is divided into s disjunct subsets

$$\mathcal{K}_{sort} = \bigcup_{i=1}^{s} \mathcal{S}_i, \tag{13}$$

where

$$\mathcal{S} = [S_1, \ldots, S_s]. \tag{14}$$

Each cluster $\mathcal{S}_i$ contains reductions leading to a similar estimated error (for example up to a factor of 10). Now the clusters are checked one after another, beginning with S_1 containing the reductions leading to the smallest estimated error. Thus, multiple reductions can be verified by one simulation. If a cluster $\mathcal{S}_i$ can not be verified (the reductions of $\mathcal{S}_i$ lead to errors greater than the error bound ε), $\mathcal{S}_i$ is divided disjunct into two clusters $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$. The term cancellation procedure then continues with $\mathcal{S}_i^k$ $(1 \leq k \leq 2)$. The whole reduction algorithm is shown in algorithm 1 for a reduction technique $\mathcal{U}$, a ranking procedure $\mathcal{R}$, a numerical integrator $\mathcal{N}$ and a certain level k. Here for a reduction $\kappa \in \mathcal{K}$, κ^{-1} undoes the reduction.

Algorithm 1 REDUCTION ALGORITHM

Require: DAE-system **F**, Error Bound ε, Scenario **u**
Ensure: Reduced DAE-System **G**

```
y_out^ref ⇐ N(F, u)
G ⇐ F
r ⇐ R(F, K^k)
Compute S from r
while S ≠ ∅ do
    S_t = S_1
    S = S \ S_1
    for all κ ∈ S_t do
        G ⇐ G^κ
    end for
    y_out ⇐ N(G, u)
    ε_{t_i} ⇐ ||y_{out_i}^ref − y_{out_i}||, 1 ≤ i ≤ n_out
    if ε_t ≥ ε then
        for all κ_i do
            G ⇐ G^{κ^{-1}}
        end for
        Divide S_t into S_t^1 and S_t^2
        S ⇐ [S_t^1, S_t^2, S]
    end if
end while
```

3 VALIDATED SOLUTION OF INITIAL VALUE PROBLEMS

In many applications the initial values and/or the parameters of the system under consideration are uncertain. In order to take this uncertainty into account intervals can be employed [10]. Several solvers for ODEs containing uncertainty exist [7], but none for DAEs with uncertain initial values and/or parameters. Thus, in the next sections only systems which can be modeled as an system of ODEs are considered.

3.1 Solving Initial Value Problems Containing Uncertainty

Classical interval methods for ODEs are based on Taylor series expansion. These methods are implemented for example in the AWA [11] and the VNODE(-LP) [12, 13] package. The COSY VI [14] as well as the VSPODE [15, 16] solver are based on Taylor models [17]. As VSPODE is the only solver for ODEs with interval valued initial values and interval valued parameters, it is chosen here and shortly reviewed. Given an initial value problem

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \varphi), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in X_0, \quad \varphi \in \Theta, \tag{15}$$

where φ is a p-dimensional parameter vector, X_0 and Θ are intervals and

$$\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^p \mapsto \mathbb{R}^m \tag{16}$$

is differentiable as needed. The solution of Eqn. 15 is a function that is interval valued. Now interval enclosures of that function $\mathbf{y}(t)$ at all nodes t_i $(1 \leq i \leq N)$ are desired. The algorithm consists of two steps. First a coarse a priori enclosure is computed, which is tightened in a second step. In the first phase with the help of Banach's fixed point theorem and a traditional high-order method [18] a new step size $h_i = t_{i+1} - t_i$ is computed such that uniqueness of the solution is guaranteed inside a coarse enclosure of the solution. The high order method is a generalization of the Picard-iteration [11]. In the second phase tighter bounds are computed using Taylor models and the mean value theorem. Furthermore, the result is improved applying advanced methods basing on a QR-factorization, in order to reduce the wrapping effect (error from enclosing a non-rectangular region by a rectangle) [10]. Noteworthy, the quality of the bounds depend on the order used for the Taylor models and the Taylor series, i.e. higher orders lead to tighter bounds, but longer runtime.

3.2 Application to Symbolic Model Reduction

Now, the VSPODE solver is integrated into to model reduction algorithm in order to reduce systems simulated for interval valued scenarios. Due to the absence of an interval DAE solver [7], only systems of the form Eqn.15 are be considered from now on. Setting $\mathcal{N}$ to the VSPODE solver and using the same notation as for DAEs, the structure of algorithm 1 remains

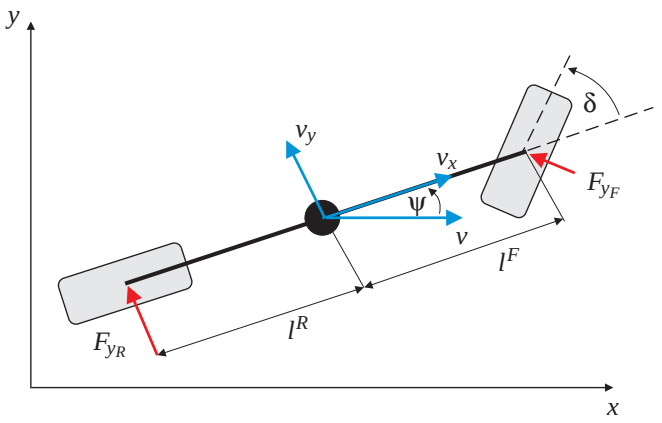

Figure 2. THE LINEAR ONE-TRACK MODEL

the same, but some modifications are necessary. The ranking procedure and the term cancellation have to be adjusted. The residual ranking can be used without modifications. For a more accurate ranking the one-step ranking could be adopted in the way, that the iterative high-order method uses initial values from the reference simulation and performs only one iteration. Though, this approach seems not very promising due to the unchanged second phase and the sensitivity of the solver on the expansion of the intervals. The second phenomenon is further explained in the next section. Here another ranking procedure is chosen. Simply the order of the Taylor models and the Taylor series is lowered. This leads to less computational effort, and worse bounds. Thus, this approach is suitable for the ranking of an interval initial value problem. In algorithm 1 a reduction κ (or a cluster $\mathcal{S}_0$ of reductions) is verified, if

$$\left\|\mathcal{N}(\mathbf{F},\mathbf{u})_i - \mathcal{N}(\mathbf{F}^\kappa,\mathbf{u})_i\right\| < \varepsilon_i,\ 1 \le i \le n_{out}. \tag{17}$$

For two intervals a and b the subtraction is defined as

$$a-b = [Sup(a)-Inf(b), Inf(a)-Sup(b)], \tag{18}$$

and therefore Eqn. 17 does not represent the change of the solution properly in case of intervals. A more adequate criterion is that the inequality

$$\max_{1\le j\le N}\max_{Op\in\{Sup,Inf\}}\left|Op(\mathcal{N}(\mathbf{F},\mathbf{u})_{i,j}) - Op(\mathcal{N}(\mathbf{F}^\kappa,\mathbf{u})_{i,j})\right| < \varepsilon_i, \tag{19}$$

holds for $1 \le i \le n_{out}$.The left hand side of Eqn.19 computes the maximum of the deviation of the supremum and the infimum from the reference solution. Hence, the maximum deviation from the reference solution can be bounded by ε.

Table 1. PARAMETERS OF THE LINEAR ONE TRACK MODEL

J_z	Inertia	$1920kg\cdot m^2$
m	Mass	$1450kg$
C_α^R	Spring constant rear	$100000\frac{N}{m}$
C_α^F	Spring constant front	$80000\frac{N}{m}$
l^R	length to rear tire	$1.45m$
l^F	length to front tire	$1.3m$

4 RESULTS

4.1 Reduction of a Linear One Track Model

The reduction algorithm is implemented in Matlab using the Maple Toolbox for Matlab. Since the VSPODE solver is implemented in C++ it is called via a MEX-function. The results are computed on a 1.8 GHz PC and visualized using INTLAB [19]. As a first example, the linear one track model is presented. The linear one track model describes the dynamics of a vehicle for a constant velocity [20]. Although the complex vehicle dynamics are already simplified during the derivation of the linear one track model some reductions are expected. Looking at Fig.2 and using the approximations

$$F_{y_F} \approx -C_\alpha^F\cdot\left(\frac{v_y+\dot\psi\cdot l^F}{v}\right) \text{ and } F_{y_R} \approx -C_\alpha^R\cdot\left(\frac{v_y-\dot\psi\cdot l^R}{v}\right), \tag{20}$$

for small steering angles δ the ODE of the linear single track model reads

$$\dot v_y = -\frac{C_\alpha^R+C_\alpha^F}{m\cdot v}v_y + \left(\frac{C_\alpha^R\cdot l^R - C_\alpha^F\cdot l^F}{m\cdot v} - v\right)\dot\psi + \frac{C_\alpha^F}{m}\delta \tag{21}$$

$$\ddot\psi = \frac{C_\alpha^R\cdot l^R + C_\alpha^F\cdot l^F}{J_z\cdot v}v_y - \left(\frac{C_\alpha^R\cdot l^{R^2} + C_\alpha^F\cdot l^{F^2}}{J_z\cdot v}\right)\dot\psi + \frac{C_\alpha^F\cdot l^F}{J_z}\delta \tag{22}$$

$$\dot r_x = v\cdot\cos\psi - v_y\cdot\sin\psi \tag{23}$$

$$\dot r_y = v\cdot\sin\psi + v_y\cdot\cos\psi \tag{24}$$

in state space, where v_y is the lateral velocity, ψ is the yaw angle and r_x respectively r_y denote the position of the vehicle. In Tab.1 the parameters of the linear one track model are shown. The reduction is performed for a straight-ahead travel and a cornering maneuver, each lasting ten seconds.

4.1.1 Straight-Ahead Travel The straight travel is performed twice. In the first scenario the velocity v is set to $v = 16.\overline{6}\frac{m}{s}$, while in the second scenario v is interval valued and set to $v = [16.\overline{6}, 19.\overline{4}]\frac{m}{s}$. Furthermore, the steering angle at the wheel δ is set to $\delta = [-\frac{\pi}{1800}, \frac{\pi}{1800}]^\circ$. Hence, the linear one track

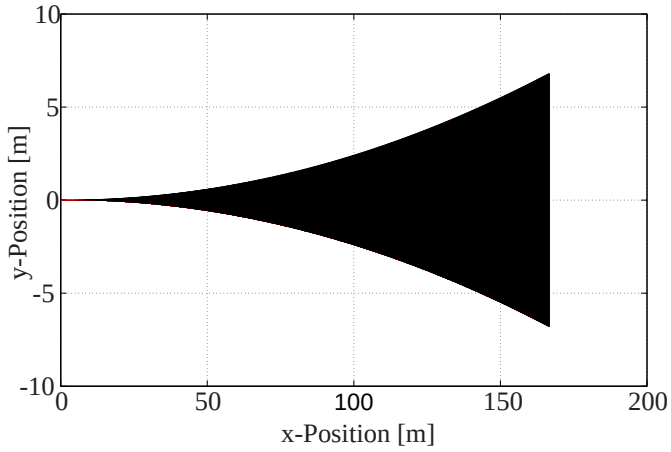

Figure 3. INTERVAL-VALUED POSITION OF THE DETAILED MODEL FOR THE STRAIGHT-AHEAD TRAVEL WITH $v = 16.\overline{6}\frac{m}{s}$

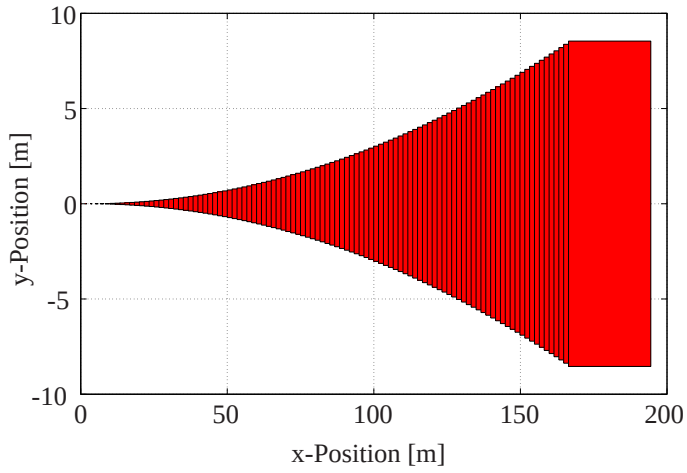

Figure 4. INTERVAL-VALUED POSITION OF THE DETAILED MODEL FOR THE STRAIGHT-AHEAD TRAVEL WITH $[16.\overline{6}, 19.\overline{4}]\frac{m}{s}$

model is tried to be reduced not only for a straight-ahead travel, but also for small steering angles and a variable velocity. In both cases the position of the vehicle is chosen as output. Setting an error bound of $\varepsilon_{r_i} = 0.1m$ $(i \in \{x,y\})$ on r_x and r_y for the first scenario, the model reduces to

$$\dot{v}_y = [-0.13772, 0.13772] + [-0.13745, 0.13745] \quad (25)$$
$$\ddot{\psi} = [-0.00825, 0.00825] \quad (26)$$
$$\dot{r}_x = [16.64796, 16.667] \quad (27)$$
$$\dot{r}_y = \sin\psi \cdot v. \quad (28)$$

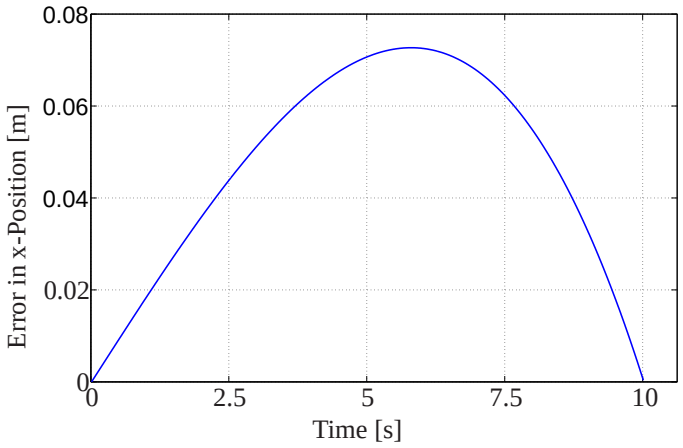

Figure 5. ERRROR IN r_x BETWEEN THE ORIGINAL AND THE REDUCED MODEL

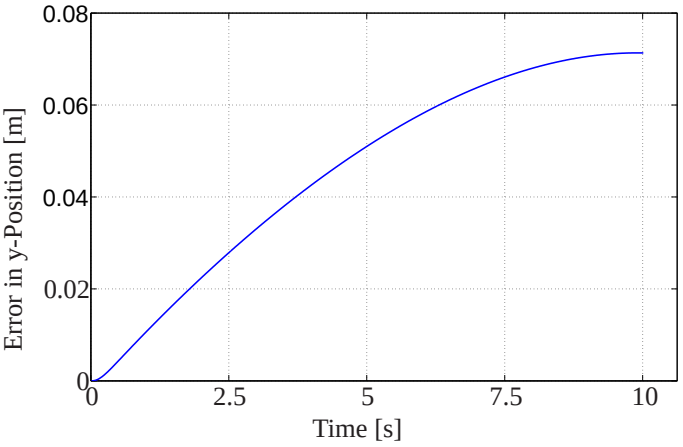

Figure 6. ERRROR IN r_y BETWEEN THE ORIGINAL AND THE REDUCED MODEL

Obviously, the lateral acceleration, the yaw rate and the velocity in x-direction are set to intervals with constant bounds, i.e. their mean values. Furthermore, the term of the right hand side of the the lateral velocity, which depends on the lateral velocity itself is neglected, since the lateral velocity is quite low. In Fig. 5 and 6 the error in the position is plotted. The error is computed as for the term cancellation using the left hand side of Eqn. 19. Since the velocity in x-direction is set to the mean value, the error in r_x is maximal at $t = 0.5s$. Contrary, the error in r_y grows with time. In Fig. 3 the output variables r_x and r_y of the original model are plotted. Note that the plot consists of many thin intervals due to the real valued velocity.
The plot of the output variables r_x and r_y for the second scenario

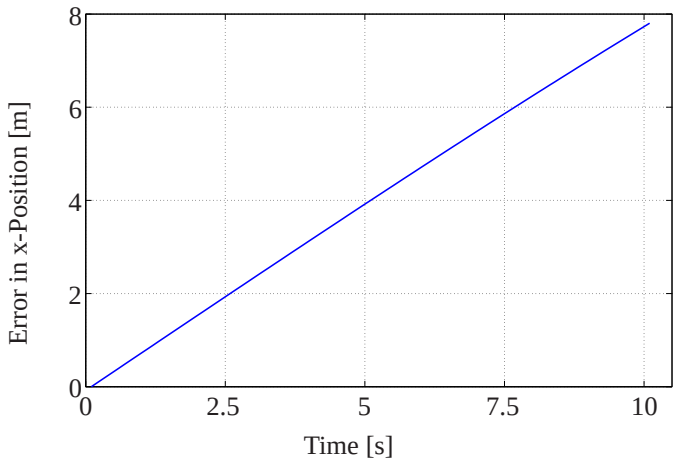

Figure 7. ERRROR IN r_x BETWEEN THE ORIGINAL AND THE REDUCED MODEL

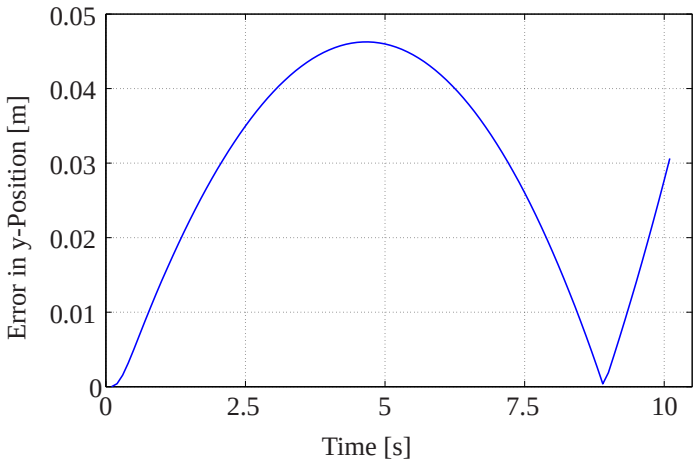

Figure 8. ERRROR IN r_y BETWEEN THE ORIGINAL AND THE REDUCED MODEL

is given in Fig. 4. Raising the error bound to $r_x = 10m$ and $r_y = 0.6m$, the reduced model reads

$$\dot{v}_y = [-0.18137, 0.18137] + [-0.14601, 0.14601] \quad (29)$$
$$\ddot{\psi} = [-0.10917, 0.10917] + [-0.09457, 0.09457] \quad (30)$$
$$\dot{r}_x = \cos[-0.04335, 0.04335] \cdot v + [16.6667, 16.6667] \quad (31)$$
$$\dot{r}_y = \sin\psi \cdot v. \quad (32)$$

As can be seen in Fig. 7 and 8 here the actual error in r_y seems to stay far from the maximum. However, during the reduction algorithm the error reaches $0.3071m$, but drops to a maximum of $0.0463m$ again. That effect occurs, if two reductions produce errors, which prone to erase each other. This effect can also be observed in Fig. 8. Without the last reduction the error would not start to decrease at the value of $0.0463m$, but raise further to $0.3071m$. The last reduction causes an error in the other direction and the leads to the shown plot. Obviously, this is an additional argument for the use of clusters in the term cancellation procedure.

4.1.2 Cornering As well as the straight-ahead travel the cornering maneuver is performed with a real valued velocity of $v = 16.\overline{6}\frac{m}{s}$ and an interval valued velocity of $v = [13.\overline{8}, 16.\overline{6}]$. This time the lateral acceleration $\dot{v}_y$ and the yaw rate $\ddot{\psi}$ are chosen as output variables, since they are often more significant for vehicle dynamics than position. Hence, the output $[\dot{v}_y\ \ddot{\psi}]^T$ is calculated via the output matrix **C** and the feedforward matrix **D**

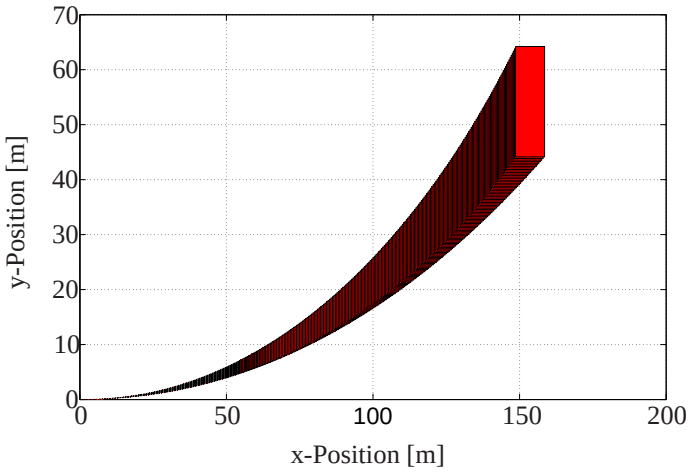

Figure 9. INTERVAL-VALUED POSITION OF THE DETAILED MODEL FOR A CORNERING MANEUVER TRAVEL WITH $v = 16.\overline{6}\frac{m}{s}$ and $\delta = [\frac{2}{3}, 1]^\circ$

by

$$[\dot{v}_y\ \ddot{\psi}]^T = \underbrace{\begin{bmatrix} -\frac{C_\alpha^R + C_\alpha^F}{m \cdot v} & (\frac{C_\alpha^R \cdot l^R - C_\alpha^F \cdot l^F}{m \cdot v} - v) \\ \frac{C_\alpha^R \cdot l^R + C_\alpha^F \cdot l^F}{J_z \cdot v} & (\frac{C_\alpha^R \cdot l^{R2} + C_\alpha^F \cdot l^{F2}}{J_z \cdot v}) \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} v_y \\ \ddot{\psi} \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{C_\alpha^F}{m} \\ \frac{C_\alpha^F \cdot l^F}{J_z} \end{bmatrix}}_{\mathbf{D}} \delta. \quad (33)$$

First the case of a real valued velocity is considered. The position of the vehicle is plotted in Fig.9. For a constant real valued

steering angle the system reaches steady state after a while. For an interval valued steering angle of $\delta = [\frac{2}{3}, 1]^\circ$ at the front-wheel the same holds as can be seen in Fig.10 and 11. Consequently, one would expect the right hand sides of Eqn. 21 -24 to vanish. Starting in the interval steady state the model reduces as expected to

$$\dot{v}_y = 0 \tag{34}$$
$$\ddot{\psi} = 0 \tag{35}$$
$$\dot{r}_x = 0 \tag{36}$$
$$\dot{r}_y = 0. \tag{37}$$

Here the error bounds $\varepsilon_{\dot{v}_y} = 0.01\frac{m}{s}$ and $\varepsilon_{\ddot{\psi}} = 0.001$ are used. As expected the dynamics of the model vanish and the lateral acceleration as well as the yaw rate remain constant. On the one hand this is the expected result, on the other hand the result is not only obtained for cornering with a constant steering angle, but also for cornering with a steering angle between $\frac{2}{3}^\circ$ and 1°.
As a last example the velocity is interval valued again and set to $v = [13.\overline{8}, 16.\overline{6}]$. Furthermore, the system does not start from steady state this time. Using the same steering angle as above and the error bounds $\varepsilon_{\dot{v}_y} = 0.3\frac{m}{s^2}$ and $\varepsilon_{\ddot{\psi}} = 0.1$ the reduced system reads

$$\dot{v}_y = 0 \tag{38}$$
$$\ddot{\psi} = -\left(\frac{C_\alpha^R \cdot l^{R^2} + C_\alpha^F \cdot l^{F^2}}{J_z \cdot v}\right)\dot{\psi} + \frac{C_\alpha^F \cdot l^F}{J_z}\delta \tag{39}$$
$$\dot{r}_x = 0 \tag{40}$$
$$\dot{r}_y = 0. \tag{41}$$

Here the simulation time was set to $2s$, since the system reaches steady state within this time and the error remains constant as can be seen in Fig. 12 and 13.

4.2 Reduction of a Planar Two Track Model

The underlying planar two track model consists of 18 states. One for the rotation of each wheel, two for the planar forces acting on each tire and six for the position and orientation of the vehicle. The steady state tire forces are calculated using the linear part of the Dugoff tire model [21]. Simulating that model with an interval-valued steering angle at the wheels of $\delta = [-10^{-6}, 10^{-6}]$ the integration aborts after $0.05s$, since the diameter of the resulting interval valued states gets to large. Hence, even for a very small steering angle the simulation can not be performed. Consequently, the reduction could not be performed for interval scenarios. The results for single scenarios can be found in [22]. The break-down time of the solver shrinks with the complexity of the system. Hence, in order to perform a symbolic reduction

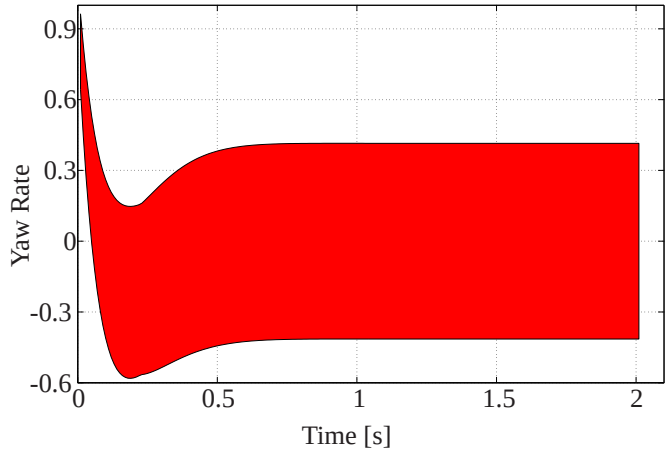

Figure 10. YAW RATE FOR THE FIRST TWO SECONDS OF A CORNERING MANEUVER WITH $v = 16,\overline{6}\frac{m}{s}$ and $\delta = [\frac{2}{3}, 1]^\circ$

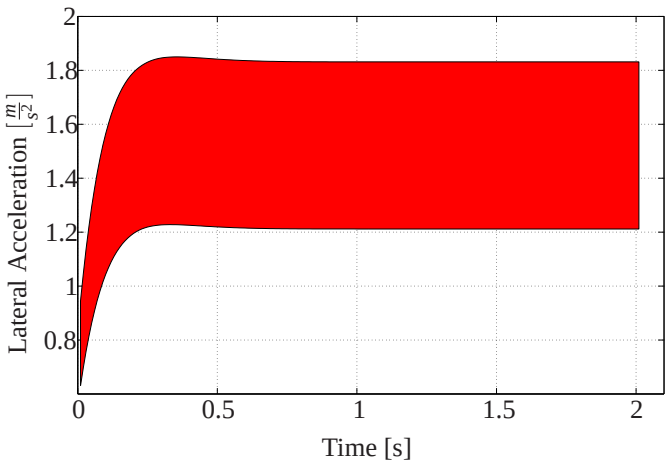

Figure 11. LATERAL ACCELERATION FOR THE FIRST TWO SECONDS OF A CORNERING MANEUVER WITH $v = 16,\overline{6}\frac{m}{s}$ and $\delta = [\frac{2}{3}, 1]^\circ$

one has to check first if the system under consideration can be simulated for the chosen interval valued scenario.

5 CONCLUSION AND OUTLOOK

In this contribution an approach for the extension of symbolic reduction techniques is presented. The use of interval scenarios leads to reduced models, which are valid not only for one scenario. The algorithm delivers fine results in the case of small systems as for example the linear one track model. For complex systems the algorithm fails, due to the absence of a feasible

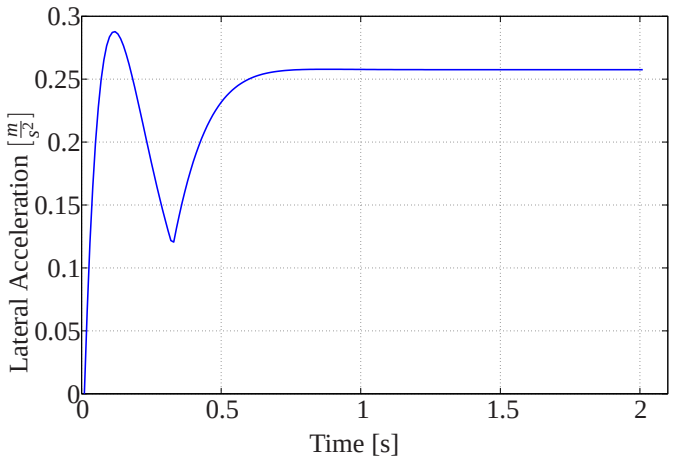

Figure 12. ERROR IN THE $\dot{v}_y$ FOR A CORNERING MANEUVER WITH $v = [13.\overline{8}, 16.\overline{6}]\frac{m}{s}$ and $\delta = [\frac{2}{3}, 1]^\circ$

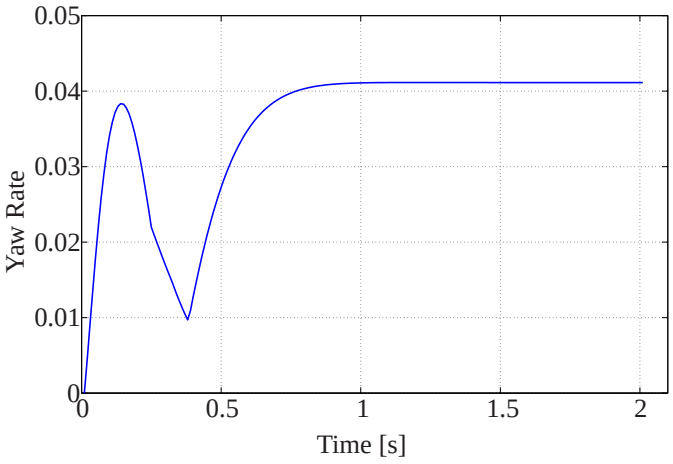

Figure 13. YAW RATE FOR THE FIRST TWO SECONDS OF A CORNERING MANEUVER WITH $v = 16, \overline{6}\frac{m}{s}$ and $\delta = [\frac{2}{3}, 1]^\circ$

solver. Here other approaches are part of current research. Since, discretizing scenario regions would lead to very high computational effort the procedures shown in [23] or [24] seem promising. Another way could be the backward computation of reachability sets as presented in [25]. Though, the computation of reachability sets is computationally even more expensive than solving initial value problems with uncertain initial values. A compromise might by the forward computation of reachability sets using piecewise linear approximations [26]. The choice of an efficient method for complex systems and its examination is part of future work.

REFERENCES

[1] Elmqvist, H., Brück, D., and Otter, M., 1995. "Dymola-User's Manual". *Dynasim AB, Research Park Ideon, Lund, Sweden*.

[2] Pfaltz, J., 1993. *The ADAMS Language: A Tutorial and Reference Manual*. University of Virginia Charlottesville, VA, USA.

[3] Borchers, C., 1998. "Symbolic behavioral model generation of nonlinear analog circuits". *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on]*, **45**(10), pp. 1362–1371.

[4] Popp, R., Oehmen, J., Hedrich, L., and Barke, E., 2002. "Parameter controlled automatic symbolic analysis of nonlinearanalog circuits". *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*, pp. 274–278.

[5] Wichmann, T., 2003. "Transient Ranking Methods for the Simplification of Nonlinear DAE Systems in Analog Circuit Design". *PAMM*, **2**(1), pp. 448–449.

[6] Sommer, R., Halfmann, T., and Broz, J., 2008. "Automated behavioral modeling and analytical model-order reduction by application of symbolic circuit analysis for multi-physical systems". *Simulation Modelling Practice and Theory*.

[7] Nedialkov, N., 2006. Interval tools for ODEs and DAEs. Tech. rep., Technical Report CAS 06-09-NN, Dept. of Computing and Software, McMaster University, 2006.

[8] Wichmann, T., 2004. "Symbolische Reduktionsverfahren für nichtlineare DAE-Systeme". *Berichte aus der Mathematik. Shaker Verlag, Aachen, Germany*.

[9] Wichmann, T., 2000. "Computer aided generation of approximate DAE systems for symbolic analog circuit design". *Proc. Annual Meeting GAMM*.

[10] Moore, R., 1966. *Interval analysis*. Prentice-Hall Englewood Cliffs, NJ.

[11] Lohner, R., 1988. "Einschließung der Lösung gewöhnlicher Anfangs-und Randwertaufgaben und Anordnungen". PhD thesis, PhD thesis, University of Karlsruhe.

[12] Nedialkov, N., and Jackson, K., 2002. "The design and implementation of an object-oriented validated ODE solver". *Draft, available via http://www. cas. mcmaster. ca/nedialk*.

[13] Nedialkov, N., 2006. VNODE-LP. Tech. rep., Department of Computing and Software , McMaster University.

[14] Makino, K., and Berz, M., 2006. "COSY INFINITY Version 9". *Nuclear Inst. and Methods in Physics Research, A*, **558**(1), pp. 346–350.

[15] Lin, Y., and Stadtherr, M., 2006. "Validated Solution of Initial Value Problems for ODEs with Interval Parameters". *2nd NSF Workshop on Reliable Engineering Computing, Savannah,*.

[16] Lin, Y., and Stadtherr, M., 2007. "Validated solutions of

initial value problems for parametric ODEs". *Applied Numerical Mathematics*, **57**(10), pp. 1145–1162.

[17] Neher, M., Jackson, K., and Nedialkov, N., 2007. "On Taylor Model Based Integration of ODEs". *SIAM Journal on Numerical Analysis*, **45**(1), p. 236.

[18] Nedialkov, N., Jackson, K., and Pryce, J., 2001. "An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE". *Reliable computing*, **7**(6), pp. 449–465.

[19] Rump, S., 1999. "INTLAB-INTerval LABoratory". *Developments in Reliable Computing*, pp. 77–104.

[20] Riekert, P., and Schunck, T., 1940. "Zur Fahrmechanik des gummibereiften Kraftfahrzeugs". *Archive of Applied Mechanics (Ingenieur Archiv)*, **11**(3), pp. 210–224.

[21] Dugoff, H., Fancher, P., Segel, L., of Michigan, U., of Standards, N. B., States, U., Institute, H. S. R., and Bureau, N. H. S., 1969. *Tire Performance Characteristics Affecting Vehicle Response to Steering and Braking Control Inputs*. Michigan Highway Safety Research Institute.

[22] Mikelsons, L., Unterreiner, M., and Brandt, T., 2009. "Generation of Continuously Adjustable Vehicle Models using Symbolic Reduction Methods". *To appear in ECCOMAS Multibody Dynamics 2009*.

[23] Girard, A., and Pappas, G., 2006. "Verification Using Simulation". *Lecture Notes in Computer Science*, **3927**, p. 272.

[24] Kapinski, J., Krogh, B., Maler, O., and Stursberg, O., 2003. "On Systematic Simulation of Open Continuous Systems". *Lecture Notes in Computer Science*, pp. 283–297.

[25] Mitchell, I., Bayen, A., and Tomlin, C., 2005. "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games". *Automatic Control, IEEE Transactions on*, **50**(7), pp. 947–957.

[26] Asarin, E., Dang, T., and Girard, A., 2003. "Reachability Analysis of Nonlinear Systems Using Conservative Approximation". *Lecture Notes in Computer Science*, pp. 20–35.