MULTIBODY DYNAMICS 2005, ECCOMAS Thematic Conference
J.M. Goicolea, J. Cuadrado, J.C. García Orden (eds.)
Madrid, Spain, 21–24 June 2005

EFFICIENT SIMULATION OF BUSH AND ROLLER CHAIN DRIVES

Gerhard Hippmann[⋆], Martin Arnold[⋆⋆] and Marcus Schittenhelm[⋆]

[⋆]Intec GmbH
Argelsrieder Feld 13
D-82234 Wessling, Germany
e-mails: gerhard.hippmann@simpack.de,
marcus.schittenhelm@simpack.de
web page: http://www.simpack.de/

[⋆⋆] Institute of Numerical Mathematics
Department of Mathematics and Computer Science
Martin–Luther-University Halle–Wittenberg
D-06099 Halle (Saale), Germany
e-mail: martin.arnold@mathematik.uni-halle.de
web page: http://www.mathematik.uni-halle.de/~arnold/

Keywords: Multibody Dynamics, Chain Drive Simulation, Numeric Time Integration.

Abstract. *Bush chains and roller chains are frequently used in valve train systems of combustion engines. Their complex dynamical behaviour is dominated by the effects of high velocities and transient driving torques as well as by polygonal action and rotatory impacts.*

These phenomena may be studied efficiently by the methods of multibody dynamics. However, the high-frequency characteristics and the comparatively large number of degrees of freedom cause challenging numerical problems. There are two basic strategies to keep the numerical effort in time integration within reasonable bounds: On the one hand it may be reduced by adapted modelling, on the other hand specific time integration schemes may be used to solve the equations of motion. The paper introduces methods in both fields and presents simulation examples to show their effect on efficiency and to validate their implementation.

1 INTRODUCTION

Bush chains and roller chains are frequently used in valve train systems of combustion engines. Their usual function is to drive the cam shafts by form-locking transmission of torque from the crank shaft. Thereby the transient driving torques of the shafts as well as typical kinematic and dynamic effects of chain drives result in relevant excitations of the system. As a result the motions of the valves are significantly disturbed which leads to problems in gas exchange and durability and the drive can also produce inacceptable noise at certain working points. In addition to intense dynamics, chain drives are characterised by highly non-linear kinematics and discontinuous contacts which include friction.

These phenomena may be studied efficiently by the methods of multibody dynamics. However, several specifics turn the simulation of chain drives into a challenging problem.

First of all we have to deal with several thousands, instead of hundreds, of degrees of freedom. This is especially problematic for operations where the numerical effort grows quadratically with the system's state vector, e. g. Jacobian calculations.

Secondly the chain links are characterised by low inertias and high contact stiffnesses resulting in notedly stiff equations of motion and a frequency range up to several tenthousand Hertz. In general implicit, comparatively complex, integration schemes are required to solve this kind of problems with acceptable calculation effort.

Since hundreds of chain links can come into contact with all wheel sprockets and guides, we have to deal with a huge number of possible collisions. As a consequence, tailor-made strategies of collision detection are required to find the active contact pairings and thus permit efficient contact evaluation.

Last, but not least, the initial state determination of chain drives is not trivial. The positions and velocities of all links, wheels and the tensioner depend on each other and have to be made consistent by a suitable method.

This paper presents ideas and methods addressing the above mentioned problems of chain drive simulation. In Section 2 we deal with the modelling aspect. Section 3 introduces methods to achieve efficient numerical time integration. Two example simulations are presented in Section 4 which is followed by a summary.

2 MODELLING

Chain drives are always subsystems of superior machines. Therefore we choose the system boundaries so that only essential elements are regarded whilst the rest of the system is modeled using standard MBS elements. As an example Figure 1 shows the *links*, *wheels* and *guides* of a chain drive used in a passenger car V6 engine. Bush and roller chains consist of inner and outer links standardised by ISO 606 [1] together with the wheels' tooth geometry. The guides are used to avoid transversal oscillations of the chain. Another important element is the *tensioner* which adjusts the chain load whilst damping longitudinal oscillations. However, tensioners are not regarded in this paper since they usually can be modeled by standard MBS elements, e. g. force elements representing hydraulic lash adjusters.

Obviously numerous contacts occur in chain drives. First the bushes resp. rollers of the inner links contact the teeth of the wheels to enable form-locking transmission of motion. Secondly there are contacts between the links' plates and the guides. To enable dynamic simulation close to reality we also have to consider the clearance of the link joints resulting in one more contact per link. The modelling of these contacts is presented in more detail in Section 2.3.

The chain drive modelling presented in this paper follows the maxim "as simple as possible,

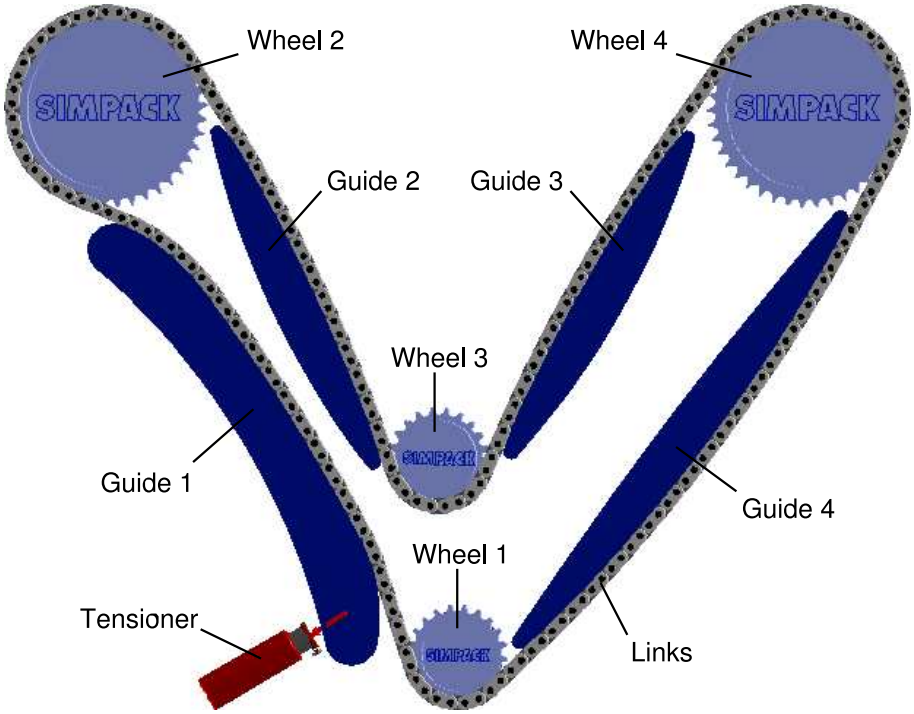

Figure 1: MBS model of a chain drive of a V6 passenger car engine.

as detailed as necessary". Therefore it is based on the following collection of relevant phenomena in chain drive dynamics [2] to be considered. All other effects are wittingly neglected to achieve optimal efficiency.

The best known phenomenon of chain drive dynamics is polygonal action: At wheel inlets and outlets the chain kinematics causes excitations in longitudinal and transversal direction. This behaviour accompanies impacts of the links hitting wheel teeth and guides. In addition, dynamic effects gain importance at higher speeds: At wheel inlets and outlets the rotatory speed of the links abruptly changes resulting in corresponding rotatory impacts. During contact centrifugal forces can substantially influence the links' strains and dynamics. Moreover, load and wear cause chain elongation which can significantly disturb the contact kinematics between the links and teeth. Last, and most importantly, transient driving torques act as excitations of the chain drive dynamics.

2.1 2D Modelling

A standard method for reducing modelling and calculation effort of multibody systems is to consider planar dynamics only. This approach is applicable for chain drives since they are normally arranged symmetrically and all relevant dynamic effects occur along their plane of symmetry.

The benefit of this simplification is substantial: By considering three degrees of freedom instead of six the large number of system states representing the links is halved. Moreover the numerical effort of matrix operations is reduced notedly, see Table 1.

When the chain drive is part of a three-dimensional MBS the assumption of a planar model can be violated. In particular wheels mounted on flexible shafts provoke small tilting motions which have to be considered. This problem can be solved by transforming the actual wheel (index A) to a virtual wheel (index W) lying in the chain drive plane (Figure 2).

	2D	3D
scalar product	3	5
vector product	3	9
vector transformation	6	15
matrix product	12	45

Table 1: Number of floating point operations of frequently used matrix operations.

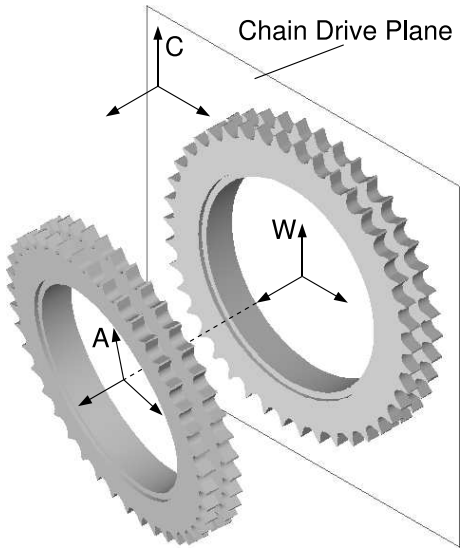

Figure 2: Transformation of the actual wheel A to the virtual wheel W.

Based on a chain drive reference system C the position and velocity vectors of the virtual wheel centre are projected onto the chain plane, e. g.

$$ {}_C\mathbf{r}_{CW} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot {}_C\mathbf{r}_{CA} \, . \tag{1} $$

Since the contact profiles must not be distorted the wheel is in addition folded into the plane,

$$ T_{CW}(\gamma) = T_{CA} \cdot T_{AW}(\alpha, \beta) \, , \tag{2} $$

where T_{CW} consists of an elementary rotation matrix about the z-axis and T_{AW} represents the virtual folding as cardan orientation matrix.

Thus the virtual wheel used in the chain drive evaluations is determined by the two-dimensional kinematics T_{CW}, $\mathbf{r}_{CW}$, ω_{CW} and $\mathbf{v}_{CW}$. The resulting contact forces and torques are transformed correspondingly to the actual wheel:

$$ \mathbf{F}_A = \mathbf{F}_W \, , \tag{3} $$

$$ \mathbf{M}_A = \mathbf{M}_W - \mathbf{r}_{WA} \times \mathbf{F}_W \, . \tag{4} $$

2.2 Chain Trajectory

Using state-of-the-art DAE methods the numerical effort of time integration is dominated by the evaluation of Jacobian matrices [3]. Thereby the number of evaluations corresponds to the degree of non-linearity of the system. Hence the non-linear kinematics of the chain links is disadvantageous from the viewpoint of efficient time integration.

This problem can be widely avoided by the trajectory approach known from wheel-rail applications [4, 5]: Instead of using cartesian coordinates and orientation matrices with respect to the chain drive reference system the kinematics of the links are determined by generalised coordinates relative to the ideal trajectory of the chain [6]. Each link has three states, namely its longitudinal trajectory position s, its lateral position deviation y and its angle deviation γ (Figure 3). As a consequence the degree of non-linearity of the system is notedly reduced during normal operation.

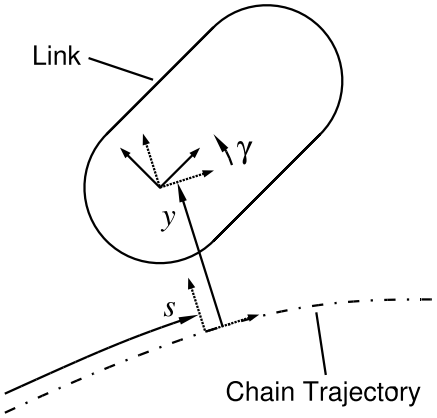

Figure 3: Generalised link states of the trajectory approach.

The ideal trajectory of common chain drives consists of circular arcs and straight lines only, the arcs along wheels and guides and the lines at unbound sections. In addition we use polynomial transitions to avoid discontinuous time derivatives of the rotatory link states. So the trajectory can be composed of three element types of simple mathematical representation.

The trajectory elements for given wheel and guide positions are generated based on the trivial determination of common tangential lines of two circles. However, multiple combinations have to be considered for guides consisting of multiple arcs. Furthermore the whole process depends on the variable tensioner state τ.

Therefore we implemented an algorithm which iteratively determines consistent trajectory elements, link states and the tensioner state. In each function evaluation the trajectory is generated and the links are positioned along it considering their joint clearance, elongation and pre-stress force. So we obtain the signed distance $\Delta L = f(\tau)$ between the first and the last link of the chain depending on the tensioner state. ΔL represents the scalar function residual of the non-linear problem. Note that in addition the rotatory joint states of the wheels have to be consistent to the longitudinal chain position. So they have to be determined by secondary iterations in each function evaluation. On velocity level all we have to do is setting the desired longitudinal speed v as link speeds $\dot{s} := v$ and as wheel speeds $\omega := v/r$.

Although the whole process is quite complex, a standard solver for non-linear systems yields reliable convergence for reasonable initial states τ_0. As a great advantage of this approach we find a consistent state of the chain drive which can directly be used as initial state for dynamic simulation. Moreover the chain trajectory can be used in a first stage of collision detection

between the links and the wheels and guides. Therefore each trajectory element holds a pointer to its corresponding wheel resp. guide. So the contact candidate for each link is determined incidently when evaluating its trajectory kinematics.

2.3 Contact Modelling

The functionality of chain drives is determined by numerous contacts. During dynamical simulation, the contact analysis is in general carried out in four steps: Step 1 involves collision detection to find out if two contact candidates possibly collide and therefore have to be considered in the following steps. In Step 2 the point of contact and the corresponding relative kinematics in normal and tangential direction are determined. Assuming suitable force laws the contact forces are calculated in Step 3. And finally the forces are applied to the contacting bodies so that they are considered in the equations of motion.

In some aspects of contact modelling we follow the approved methods of Fritz [7]. In particular we use the same multibody discretisation of chains which differs quite notedly from reality (Figure 4): Real chains consist of inner links which hold the bushes resp. rollers and outer links which act as intermediate bodies. They are joined by cylindric pins of the outer links and hollow-cylindric bores of the inner links. In contrast all links of the chain model are of identical geometry with one inner and one outer link half. So it is possible to model only one body per link. Note that inner and outer links still can be considered seperately concerning their inertia properties, plate geometry and longitudinal stiffness.

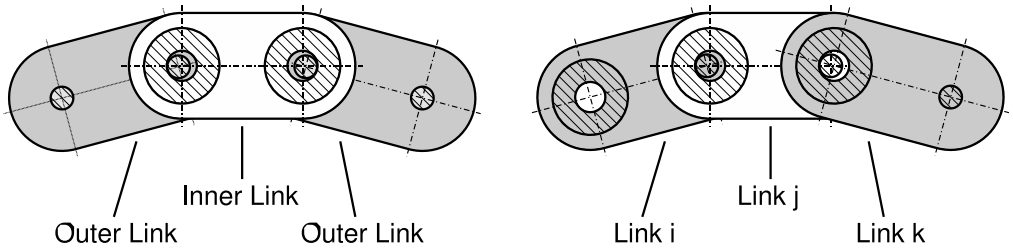

Figure 4: Real (left) and modeled (right) chain structure.

In the current implementation we model all contacts by elastic contact forces and do not follow the rigid body approach proposed by Fritz [7]. In our experience the calculation effort of dynamical simulation is dominated by the chosen friction model and parameters rather than by the high physical stiffness of the wheel-link contacts. Besides compliant contact modelling appears to be a more realistic approach than assuming ideal rigid bodies.

2.3.1 Link-Link Contact

Since the contact pairings between the links are unchangingly defined by the chain topology, no collision detection is required for the link-link contacts. As mentioned above, all links are joined by cylindric contacts which result in contact between interleaved circles in two-dimensional modelling. Therefore the determination of the contact point and kinematics is trivial. As proposed in [7] we use a combined linear contact stiffness which includes the longitudinal stiffnesses of the links. Additionally to linear damping during contact, non-linear radial damping, which decreases with growing contact distance, can be defined to model the oil film in the link joint. Furthermore, friction can be considered by a viscous or regularised Coulomb force law applied to the rotatory degree of freedom.

2.3.2 Wheel-Link Contact

As mentioned above, the wheel and guide contact candidate of each link is obtained from its current trajectory element. If this pointer refers to a wheel, a classic bounding volume approach is used to find out if the link bush resp. roller may collide with its teeth [8]. For this the distance between the wheel centered reference system W and the bush centered link reference system L is compared to the sum of their radii, see Figure 5:

$$|\mathbf{r}_{WL}| - \frac{d_1 + d_a}{2} \begin{cases} \geq 0 & \rightarrow \quad \text{no contact} \\ < 0 & \rightarrow \quad \text{contact possible} \end{cases} \tag{5}$$

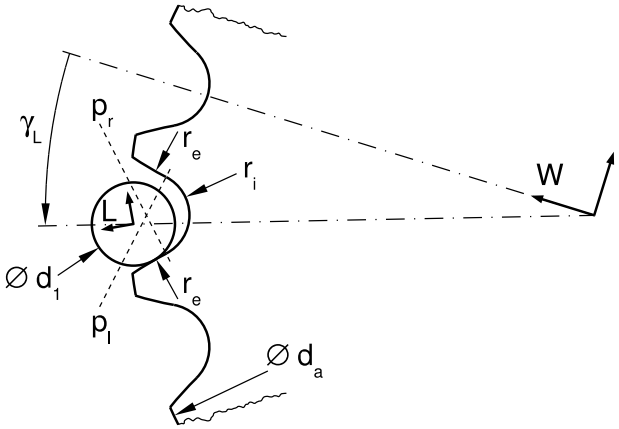

Figure 5: Contact between wheel W and link L.

The next step is to determine the tooth gap which may collide with the bush resp. roller. With the position vector ${}_W\mathbf{r}_{WL}$ expressed in coordinates ${}_Wx_{WL}, {}_Wy_{WL}$ of the wheel system the angle between the first tooth and the link can be computed as

$$\gamma_L \quad = \quad \text{atan2}\,(\,-{}_Wx_{WL},\, {}_Wy_{WL}\,)\ . \tag{6}$$

Thus the tooth gap index is given by

$$i_T \quad = \quad \begin{cases} \text{int}\left(\,\frac{\gamma_L}{2\pi}z + \frac{3}{2}\,\right) & : \quad \gamma_L \geq 0\,, \\ \text{int}\left(\,(\frac{\gamma_L}{2\pi} + 1)z + \frac{3}{2}\,\right) & : \quad \gamma_L < 0\,, \end{cases} \tag{7}$$

where z is the number of teeth.

Each tooth gap profile consists of the tooth flanks of radius r_e and the intermediate roller seating of radius r_i (Figure 5). In a preprocessing step we calculate the separation lines p_l and p_r which are determined by the flank centres and the roller seating centre. Using the normal form coefficients n_x, n_y, d_0 the signed distances between the lines and the bush centre can be calculated by

$$d_l \quad = \quad n_{lx} \cdot {}_Wx_{WL} + n_{ly} \cdot {}_Wy_{WL} + d_{l0}\,, \tag{8}$$

$$d_r \quad = \quad n_{rx} \cdot {}_Wx_{WL} + n_{ry} \cdot {}_Wy_{WL} + d_{r0}\,. \tag{9}$$

Therewith for possible contact between the bush resp. roller and the profile segments holds

$$\begin{cases} d_l < 0\,,\ d_r < 0 & \rightarrow \quad \text{roller seating,} \\ \quad d_l > d_r & \rightarrow \quad \text{left flank,} \\ \quad d_l < d_r & \rightarrow \quad \text{right flank.} \end{cases} \tag{10}$$

At this point we have reduced the problem to trivial contact between two circles again. For determination of the contact force we use force laws similar to Fritz [7]. A contact distance for damping without solid body contact can be defined to model the oil film. Additionally viscous or regularised Coulomb friction can be considered. In doing so, roller chains differ from bush chains as their rollers are assumed to be a part of the wheel and e.g. the viscous friction law reads

$$M_L \;=\; (\omega_R - \omega_L) \cdot \mu_{rot} \tag{11}$$

with the rotatory roller velocity ω_R. In contrast, the tangential force of bush chains is determined by standard body surface contact, viz

$$F_L \;=\; (v_W - v_B) \cdot \mu_{trans} \tag{12}$$

with the tangential relative velocity $v_W - v_B$ of wheel and bush at the contact point.

2.3.3 Guide-Link Contact

The contact evaluation between guides and links is based on similar concepts as used for the wheel-link contacts. The algorithm is executed for links L positioned at trajectory elements pointing to a guide G as possible contact partner. We proceed from guide profiles composed of circular arcs with centre positions C_i and radii r_i (Figure 6). The direction angles γ_{ij} of the separation lines of adjacent arcs are calculated during preprocessing. The used link discretisation suggests itself to approximate the link plate profiles by circles with the plate depth h_2 as diameter [7].

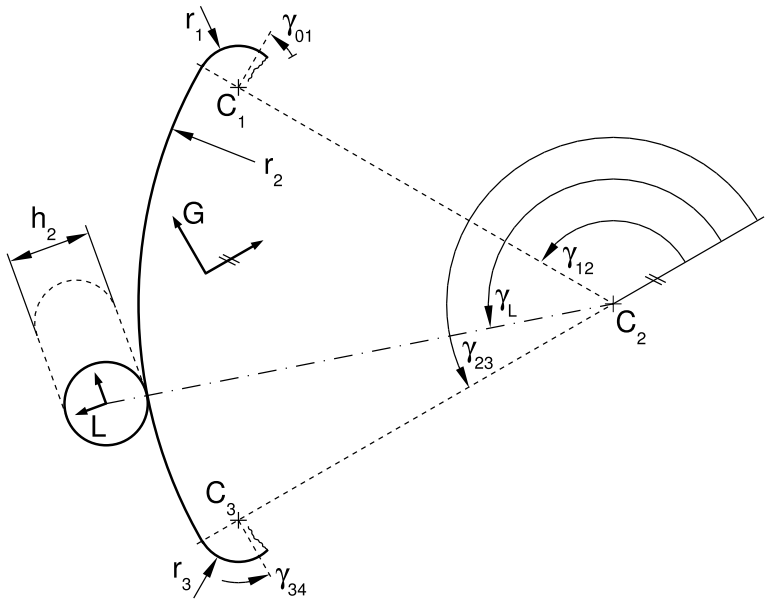

Figure 6: Contact between guide G and link L.

For finding the arc which may contact with the link the arcs are tested successively in two steps. In Step 1 the circles are checked for collision by a simple distance evaluation:

$$|\mathbf{r}_{C_iL}| - \left(\frac{h_2}{2} + r_i\right) \;\begin{cases} \geq 0 & \rightarrow \quad \text{no contact} \\ < 0 & \rightarrow \quad \text{contact possible} \end{cases} \tag{13}$$

Only if Step 1 gives a positive result the link position is checked by Step 2. Using the coordinates ${}_G x_{C_i L}, {}_G y_{C_i L}$ of the position vector ${}_G \mathbf{r}_{C_i L}$ from the arc centre to the link system the link angle results in

$$\gamma_L \;=\; \text{atan2}\,(\,{}_G y_{C_i L},\, {}_G x_{C_i L}\,)\;. \tag{14}$$

Thus the condition

$$\gamma_{i-1,i} \le \gamma_L \le \gamma_{i,i+1} \tag{15}$$

indicates contact between the current arc i and the link.

The force laws implemented for guide-link contacts are nearly identical to those of the wheel-link contacts for bush chains. Note that Coulomb's friction law does not need to be regularised here since we always have sliding friction during normal operation.

3 NUMERICS

Chain drives show a very complex dynamical behaviour that causes substantial problems in the dynamical simulation. Specially adapted numerical methods have to be used in time integration.

3.1 Basics

The degrees of freedom of links, wheels, guides and tensioner define a set of minimal coordinates $\mathbf{q}_c$ for the multibody system model of the chain drive. The equations of motion are derived following the principles of classical mechanics [9, 10]. They form a set of 2nd order ordinary differential equations

$$\mathbf{M}_c(\mathbf{q}_c)\ddot{\mathbf{q}}_c(t) = \mathbf{f}_c(\mathbf{q}_c, \dot{\mathbf{q}}_c, t) \tag{16}$$

with the symmetric, positive definite mass matrix $\mathbf{M}_c(\mathbf{q}_c)$ containing mass and inertia properties of the bodies and the vector of applied and gyroscopic forces $\mathbf{f}_c(\mathbf{q}_c, \dot{\mathbf{q}}_c, t)$.

In most applications, the chain drive is just a subsystem of more complex technical systems like combustion engines. In a multibody system model of the overall system, the minimal coordinates $\mathbf{q}_c$ of the chain drive form a subset of the position coordinates $\mathbf{q} = (\mathbf{q}_e, \mathbf{q}_c)^\top$ of the overall system with $\mathbf{q}_e$ describing, e. g., the degrees of freedom of the core engine model. The equations of motion (16) of the chain drive model are a subsystem of the equations of motion for the overall system.

In contrast to the minimal coordinates $\mathbf{q}_c$ of the chain drive model, the position coordinates $\mathbf{q}$ of the overall system may be redundant. In general, the components of $\mathbf{q}$ are not independent of each other but have to satisfy n_g constraints

$$\mathbf{0} = \mathbf{g}(\mathbf{q}, t) \tag{17}$$

see, e. g., [11, 12]. The equations of motion

$$\begin{aligned} \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}}(t) &= \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathbf{G}^\top(\mathbf{q}, t)\boldsymbol{\lambda}\,, \\ \mathbf{0} &= \mathbf{g}(\mathbf{q}, t) \end{aligned} \tag{18}$$

form a 2nd order differential-algebraic equation (DAE) with constraint forces $-\mathbf{G}^\top(\mathbf{q}, t)\boldsymbol{\lambda}$ that are determined by the constraint matrix $\mathbf{G}(\mathbf{q}, t) := (\partial \mathbf{g}/\partial \mathbf{q})(\mathbf{q}, t)$ and the Lagrangian multipliers $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_g}$, see [11].

For the numerical solution of (18) we look for DAE time integration methods that may handle both the constraints $\mathbf{g} = \mathbf{0}$ in DAE (18) and the large-dimensional ODE subsystem (16) for the chain drive model efficiently. The results of numerical tests in Section 4.1 illustrate that these two requirements contradict each other.

Classical DAE time integration methods for (18) like BDF or implicit Runge–Kutta methods [13, 14, 15] involve the solution of systems of non-linear equations in each time step and require frequent evaluations of the Jacobian of the right hand side and frequent LU-decompositions of iteration matrices in the corrector iteration. In the application to multibody system models that include chain drives these methods get very inefficient because of the large dimension of subsystem (16).

In the standalone time integration of the chain drive model equations (16) classical non-stiff ODE integrators like explicit Runge–Kutta methods or predictor-corrector methods of Adams type [16, 17] are much more efficient than implicit DAE time integration methods, see Section 4.1 below. These methods are, however, not applicable to constrained systems (18).

There are two basic strategies to combine the benefits of explicit and implicit time integration methods in the application to DAEs (18) with large-dimensional subsystems (16), see also [18]: On the one hand the numerical effort of implicit methods may be strongly reduced exploiting the sparsity structure of the Jacobian. On the other hand the benefits of implicit and explicit methods may be combined in a modular time integration approach that has been used successfully in the framework of co-simulation [19, 20, 21].

3.2 Adapted Implicit Time Integration Methods

Practical experience for large scale models shows that the corrector iteration is the most time consuming part in implicit time integration methods. In state-of-the-art implicit solvers like DASSL [13] and RADAU5 [14] much effort is spent on improving the numerical efficiency of the corrector iteration. In the application to the dynamical analysis of multibody systems further savings are achieved by adapting these solvers to the special structure of the equations of motion (16) and (18), see [3].

Chain drive models have a very specific topology since each individual link interacts only with its two neighbouring links and possibly with wheels and guides. Therefore, the Jacobian of the right hand side in (16) is sparse with a characteristic sparsity structure that may exploited to speed-up the time integration by implicit methods. To simplify the presentation we will consider in the following the time integration of ODE (16) but the proposed methods may be carried over straightforwardly to the equations of motion in general DAE form (18).

With the velocity coordinates $\mathbf{v}_c := \dot{\mathbf{q}}_c$ the second order system (16) may be rewritten as a first order system in residual form that may be handled by DAE integrators like DASSL [13]:

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0} \quad \text{with} \quad \mathbf{x} := \begin{pmatrix} \mathbf{q}_c \\ \mathbf{v}_c \end{pmatrix}, \quad \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t) := \begin{pmatrix} \dot{\mathbf{q}}_c - \mathbf{v}_c \\ \mathbf{M}_c(\mathbf{q}_c)\dot{\mathbf{v}}_c - \mathbf{f}_c(\mathbf{q}_c, \dot{\mathbf{q}}_c, t) \end{pmatrix}. \tag{19}$$

The corrector iteration in time step $t_{n-1} \to t_n$ relies on a simplified Newton method

$$\mathbf{x}_n^{(k+1)} := \mathbf{x}_n^{(k)} + \mathbf{p}_n^{(k)}, \quad \dot{\mathbf{x}}_n^{(k+1)} := \dot{\mathbf{x}}_n^{(k)} + \alpha \mathbf{p}_n^{(k)}, \quad \mathbf{J}\, \mathbf{p}_n^{(k)} = -\mathbf{F}(\mathbf{x}_n^{(k)}, \dot{\mathbf{x}}_n^{(k)}, t_n) \tag{20}$$

with an iteration matrix of the form

$$\mathbf{J} := \alpha \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}, t) + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}(\mathbf{x}, \dot{\mathbf{x}}, t) \tag{21}$$

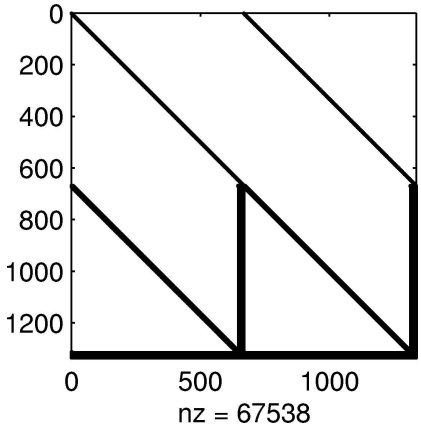

Figure 7: Sparsity structure of the Jacobian in implicit time integration: Chain drive model.

that is defined by a weighted sum of Jacobians of $\mathbf{F}$ with a scalar weighting factor α that depends on the current stepsize and order of the time integration method [13]. As a typical example, Figure 7 shows the sparsity structure of $\mathbf{J} \in \mathbb{R}^{1335\times 1335}$ for the chain drive model of Section 4.1. The topology of the chain drive results in a characteristic block band structure with less than 5% non-zero elements.

A substantial speed-up in the frequent re-evaluations of the iteration matrix $\mathbf{J}$ during time integration may be achieved by separate updates of the Jacobians $\partial\mathbf{F}/\partial\dot{\mathbf{x}}$ and $\partial\mathbf{F}/\partial\mathbf{x}$ in (21) and by an algorithm for the finite difference approximation of sparse Jacobians that generalises the classical Curtis–Powell–Reid (CPR) algorithm [22] known from non-linear optimisation, see [3]. Both methods are implemented in the industrial multibody system tool SIMPACK and will be used in the numerical tests of Section 4.

To speed-up the time-consuming solution of the systems of linear equations $\mathbf{J}\mathbf{p} = -\mathbf{F}$ in (20) we exploit the 2×2-block structure of

$$\mathbf{J} = \begin{pmatrix} \alpha\mathbf{I} & -\mathbf{I} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix} \tag{22}$$

that results from the second order structure of (16). The first block row is composed of multiples of the identity matrix. Therefore a block Gauss elimination is much more efficient than classical Gauss elimination for dense matrices $\mathbf{J}$. The number of floating point operations for the LU-decomposition of $\mathbf{J}$ is reduced from $\mathcal{O}((2n_{q_c})^2/3)$ to $\mathcal{O}(n_{q_c}^2/3)$, i. e., by a factor of 8.

The contact forces between the n_{link} links of the chain drive are modelled according to Section 2.3.1. Each link is connected to its two neighbouring links and has no direct dynamical interaction with the remaining $n_{\text{link}} - 3$ links. Therefore, the blocks of the second block row in (22) are sparse with an arrow structure, see Figure 7:

$$\mathbf{J}_{2i} = \begin{pmatrix} \mathbf{B}_i & \mathbf{J}_{2i}^{12} \\ \mathbf{J}_{2i}^{21} & \mathbf{J}_{2i}^{22} \end{pmatrix}, \quad (i = 1, 2). \tag{23}$$

The upper left blocks $\mathbf{B}_i \in \mathbb{R}^{3(n_{\text{link}}-1)\times 3(n_{\text{link}}-1)}$ are band matrices with bandwidth 11 that represent the couplings between neighbouring links. Wheels, guides, the chain closing link and the contact forces wheel-link and guide-link contribute to the small dense blocks $\mathbf{J}_{2i}^{12}$, $\mathbf{J}_{2i}^{21}$, $\mathbf{J}_{2i}^{22}$, see also Section 2. The arrow structure of $\mathbf{J}_{2i}$ may again be exploited by a block Gauss elimination that reduces the numerical effort for the solution of $\mathbf{J}\mathbf{p} = -\mathbf{F}$ substantially.

3.3 Modular Time Integration

State-of-the-art DAE time integration methods are the method of choice in industrial multibody system simulation [13, 12]. Applied to coupled problems, these classical methods are, however, often less efficient since all subsystems have to be integrated by one and the same method.

Alternatively, the modular structure of coupled problems may be exploited explicitly in time integration [19]. We discuss this approach in more detail for the equations of motion (18) that are rewritten as

$$\begin{aligned} \mathbf{M}_e(\mathbf{q}_e)\,\ddot{\mathbf{q}}_e(t) &= \mathbf{f}_e(\mathbf{q}_e,\dot{\mathbf{q}}_e,t) + \mathbf{f}_{ec}(\mathbf{q}_e,\dot{\mathbf{q}}_e,\mathbf{q}_c,\dot{\mathbf{q}}_c,t) - \mathbf{G}^\top(\mathbf{q}_e,t)\boldsymbol{\lambda}\,, \\ \mathbf{0} &= \mathbf{g}(\mathbf{q}_e,t)\,, \\ \cdots\cdots & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathbf{M}_c(\mathbf{q}_c)\,\ddot{\mathbf{q}}_c(t) &= \mathbf{f}_c(\mathbf{q}_c,\dot{\mathbf{q}}_c,t) + \mathbf{f}_{ce}(\mathbf{q}_c,\dot{\mathbf{q}}_c,\mathbf{q}_e,\dot{\mathbf{q}}_e,t) \end{aligned} \tag{24}$$

with the differential-algebraic equations of motion of the core engine model (coordinates $\mathbf{q}_e$) that are coupled to the ODE model equations of the subsystem chain drive (coordinates $\mathbf{q}_c$). In (24), the force vectors $\mathbf{f}_{ec}$ and $\mathbf{f}_{ce}$ represent the force coupling of the chain drive to the overall system.

A *modular* time integration method proceeds in macro steps $T_m \to T_{m+1}$ of stepsize $H := T_{m+1} - T_m$. The data exchange between subsystems is restricted to the discrete synchronisation points T_m. Between these synchronisation points the subsystems are integrated completely independent of each other. In macro step $T_m \to T_{m+1}$, each individual subsystem may be integrated by its own time integration method that is tailored to this subsystem. In (24), the core engine model is integrated by a classical implicit DAE method whereas an explicit ODE method is used for the subsystem chain drive.

Modular time integration is well known from co-simulation (or simulator coupling) methods that are based on the coupling of several simulation tools to simulate the dynamical interaction of coupled technical systems [19, 20, 21, 23]. In the application to chain drive simulation, the modular time integration is used *inside* the multibody system tool. Both subsystems are integrated in one and the same simulation tool but with different time integration methods. From the view point of engine simulation the chain drive may be considered as a force element with highly complex internal dynamics described by (16) that defines the force vector $\mathbf{f}_{ec}$, see (24).

The separate time integration of the subsystems in macro step $T_m \to T_{m+1}$ requires suitable approximations of coupling terms like $\mathbf{f}_{ec}$, $\mathbf{f}_{ce}$ in (24) since data are exchanged only at the synchronisation points T_m. In a series of numerical tests a combination of quadratic extrapolation of $\mathbf{q}_c$ with linear interpolation of $\mathbf{q}_e$ was found to be useful. The time integration from $t = T_m$ to $t = T_{m+1}$ is performed in two stages. In Stage I, the core engine model is integrated to get $\mathbf{q}_e(T_{m+1})$. The coupling term $\mathbf{f}_{ec}$ is approximated by

$$\mathbf{f}_{ec}(\mathbf{q}_e(t),\dot{\mathbf{q}}_e(t),\mathbf{q}_c(t),\dot{\mathbf{q}}_c(t),t) \approx \mathbf{f}_{ec}(\mathbf{q}_e(t),\dot{\mathbf{q}}_e(t),\bar{\mathbf{q}}_c(t),\bar{\dot{\mathbf{q}}}_c(t),t)$$

with a quadratic extrapolation of the link positions:

$$\bar{\mathbf{q}}_c(t) := \mathbf{q}_c(T_m) + (t-T_m)\dot{\mathbf{q}}_c(T_m) + \frac{(t-T_m)^2}{2}\ddot{\mathbf{q}}_c(T_m), \quad \bar{\dot{\mathbf{q}}}_c(t) := \dot{\mathbf{q}}_c(T_m) + (t-T_m)\ddot{\mathbf{q}}_c(T_m).$$

In Stage II, the numerical solution $\mathbf{q}_e(T_{m+1})$, $\dot{\mathbf{q}}_e(T_{m+1})$ is used for linear interpolation of the input data $\mathbf{q}_e$, $\dot{\mathbf{q}}_e$ in the integration of the subsystem chain drive:

$$\mathbf{f}_{ce}(\mathbf{q}_c(t),\dot{\mathbf{q}}_c(t),\mathbf{q}_e(t),\dot{\mathbf{q}}_e(t),t) \approx \mathbf{f}_{ce}(\mathbf{q}_c(t),\dot{\mathbf{q}}_c(t),\bar{\mathbf{q}}_e(t),\bar{\dot{\mathbf{q}}}_e(t),t)$$

with

$$\begin{aligned}\bar{\mathbf{q}}_e(t) &:= \mathbf{q}_e(T_m) + \frac{t-T_m}{T_{m+1}-T_m}(\mathbf{q}_e(T_{m+1}) - \mathbf{q}_e(T_m))\,,\\ \bar{\dot{\mathbf{q}}}_e(t) &:= \dot{\mathbf{q}}_e(T_m) + \frac{t-T_m}{T_{m+1}-T_m}(\dot{\mathbf{q}}_e(T_{m+1}) - \dot{\mathbf{q}}_e(T_m))\,.\end{aligned}$$

The macro stepsize H has essential influence on the numerical stability, on the accuracy and on the efficiency of the modular time integration method. The data exchange at the discrete time grid $\{\, T_0,\, T_1,\, \ldots \}$ restricts the maximum frequency in the coupling terms $\mathbf{f}_{ec}$, $\mathbf{f}_{ce}$ to $1/H$. Therefore, H has to be chosen sufficiently small to resolve the full technically relevant frequency range in all coupling terms. In the application to chain drive simulation, typical macro stepsizes H are in the range of $1.0\,\mu$s.

4 EXAMPLES

The presented methods for modelling and time integration of chain drives have been implemented in the industrial MBS tool SIMPACK [5] and tested by numerous simulation runs. In the following we present some results which allow to rate efficiency and fidelity of the implementation.

4.1 Efficiency

The focus of the present paper is on the efficient simulation of complex multibody system models that contain a chain drive as *sub*system. In view of the huge numerical effort it is, however, favourable to compare and to optimise the time integration methods in the application to an isolated chain drive model with transient measured excitations, viz the kinematics of the crank shaft and the driving torques of the cam shafts.

Throughout this Section we consider a multibody system model of the chain drive in Figure 1 that performs one full cycle at a speed of 6000 rounds per minute. The chain drive consists of $n_{\text{link}} = 216$ links with $3n_{\text{link}} = 648$ degrees of freedom. Wheels, flexible return wheel shaft and tensioner are described by 19 additional second order state variables $\mathbf{q}_c$ in (16) and an additional first order state variable describing the internal dynamics of the hydraulic tensioner resulting in a first order state vector $\mathbf{x}(t) \in \mathbb{R}^{1335}$ in (19).

In principle, the problem could be solved by classical implicit DAE time integration methods. In the numerical tests, the time integration by the standard implementation of DASSL [13] was stopped after $24\,\text{h} = 86400\,\text{s}$ cpu-time on standard PC hardware since less than $1/2$ cycle of the chain drive was simulated in that huge amount of computing time. Because of this very large numerical effort the present chain drive model can not be used in industrial design processes unless a substantial speed-up of time integration is achieved by specially adapted numerical solution methods.

Column 2 of Table 2 shows the statistical data for the modified version of DASSL that is available in the industrial multibody system tool SIMPACK. The computing time for one full cycle of the chain drive is reduced to about 5 h by exploiting the simple structure of $\partial\mathbf{F}/\partial\dot{\mathbf{x}}$ in (19) and the sparsity structure of $\mathbf{J}$, see Section 3.2 and [3].

Additional savings of about 48% computing time result from the use of block Gauss elimination for the 2×2-block matrix $\mathbf{J}$ in (22) that has been implemented in SIMPACK, see column 3 of Table 2. The LU-decomposition of $\mathbf{J}$ and the solution of linear equations $\mathbf{J}\mathbf{p} = -\mathbf{F}$ contribute now less than 0.9% to the overall computing time and there are no further reductions

	DASSL SIMPACK	DASSL block Gauss	DASSL band matrix	LSODE	DOPRI5	SIMPACK modular integration
cpu-time[1] (s)	18344.4	9585.5	9772.9	3539.7	2487.9	748.1
# time steps	41287	43716	44405	281494	60209	103384
# function evaluations	1705326	1325287	1393471	508891	361255	108013
# function evaluations (without Jacobian)	60606	62377	63811	508891	361255	107272
# Jacobian evaluations	1232	946	996	0	0	19
# failed error tests	144	140	145	–	1	57
# failed corrector	230	147	164	0	0	8

Table 2: Numerical effort of time integration for the subsystem chain drive.

of the overall computing time if the band structure of $\mathbf{J}_{2i}$ in (23) is exploited, see column 4 of Table 2. The computing time even increases slightly because of an increased sensitivity of the band matrix solver w. r. t. round-off errors that results in more Jacobian evaluations than before ($\#\mathtt{Je} = 996$ vs. $\#\mathtt{Je} = 946$).

In general, stiff multibody systems like the chain drive model should be integrated much more efficiently by implicit methods than by explicit ones [14, 17]. But, practical experience shows that the nearly undamped high-frequency oscillations in the chain drive and the discontinuities caused by unilateral force elements in the elastic contact models restrict the time stepsize strongly for *all* integration methods. In implicit time integration methods, the large numerical effort per time step is therefore not compensated by larger time stepsizes.

For the chain drive model, non-stiff integrators like Adams methods (LSODE) or explicit Runge–Kutta methods (DORPI5) are more efficient than DASSL, see columns 5 and 6 of Table 2. These methods are, however, not applicable to general multibody system models including constraints $\mathbf{g} = \mathbf{0}$ in (18).

The last column of Table 2 illustrates that modular time integration with a combination of DASSL for the overall system and DOPRI5 for the chain drive is an interesting alternative. The present simulation with macro stepsize $H = 1.0_{\mathrm{E}} - 6\,\mathrm{s}$ benefits not only from the combination of two powerful time integration methods but additionally from the elimination of unphysical very high frequency components in the coupling terms $\mathbf{f}_{ce}$, $\mathbf{f}_{ec}$ of the subsystem chain drive by the discrete sampling with sampling points T_m, see Section 3.3.

For this numerical test, the degrees of freedom of wheels, guides and tensioner are considered as components $\mathbf{q}_e$ and the degrees of freedom of the n_{link} chain links define $\mathbf{q}_c$. With this splitting of $\mathbf{q}$, the coupling terms $\mathbf{f}_{ce}$, $\mathbf{f}_{ec}$ in (18) are defined by the contact forces wheel-link and guide-link, see Sections 2.3.2 and 2.3.3 .

The simulation results for all six numerical tests in Table 2 coincide within the user defined error bounds. Compared with standard implementations of classical DAE time integration

[1] Xeon 2.4 GHz, SIMPACK 8.7, SuSE Linux 9.0

methods, the computing time is reduced by more than one order of magnitude. In the SIMPACK simulation environment, the computing time is reduced by an additional factor of 1.9 using block Gauss elimination in implicit methods, by a factor of 7.5 using non-stiff methods (not applicable to DAEs (18)) and by a factor of 24.5 using the novel modular time integration approach.

4.2 Validation

For validation we tested the chain drive model as subsystem of a detailed V6 engine MBS model (Figure 8). It includes the complete valve train system with two gear drives, four flexible cam shafts, hydraulic cam shaft bearings and 24 valve gears with flexible cam followers, dynamic valve springs and detailed dynamic hydraulic lash adjusters. This results in 1564 rigid and 29 flexible bodies and 1876 force elements which require a total of 3403 first order states.

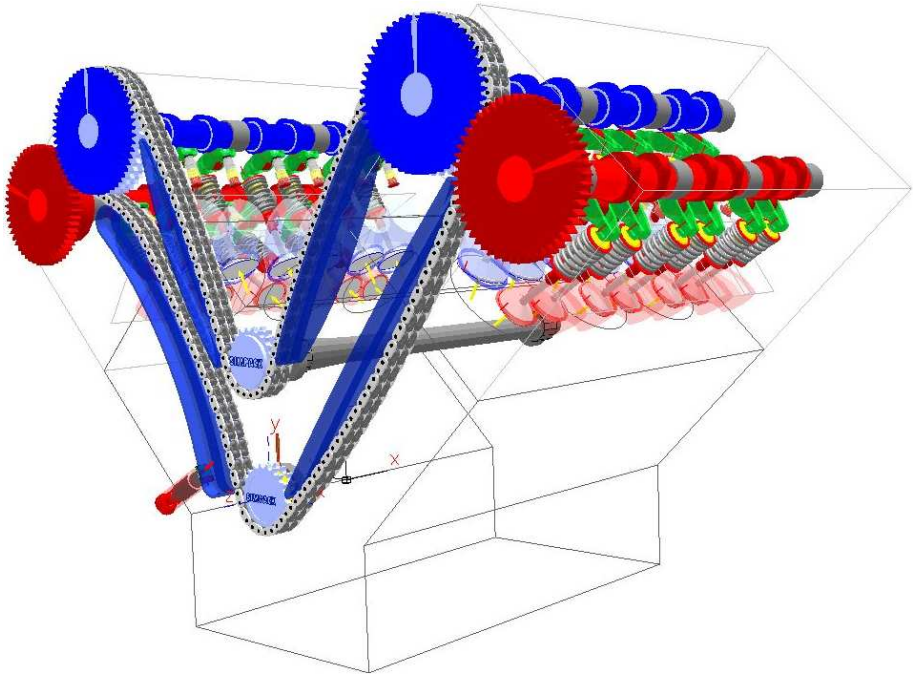

Figure 8: MBS model of a V6 passenger car engine.

Due to the huge dimension of the problem simulation of a full chain cycle takes a few hours on a standard PC. However, we did not experience any numerical stability problems in calculating various engine speeds.

Figure 9 compares simulation results to measurements of the cam shaft deviation motion and the tensioner force at 2500 rounds per minute. In consideration of the high complexity of the problem the agreement appears satisfactory and sufficient to enable the engine designer to optimise the system. Current work on enhancing contact force laws and flexible guide models may further improve the results. Moreover the model will be extended by the crank train to consider the force feedback from the chain drive to the engine.

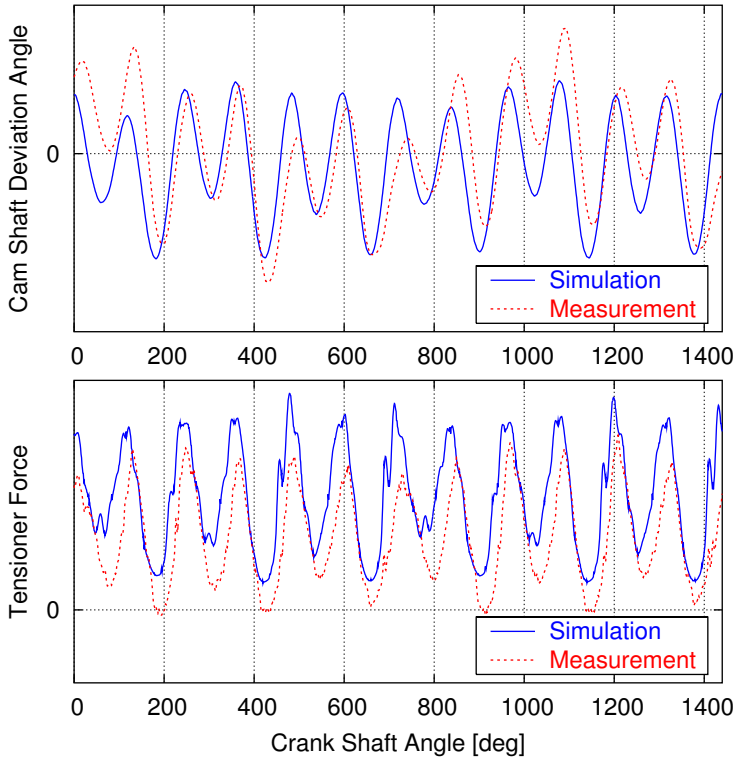

Figure 9: Comparison of measurements and simulation results.

5 SUMMARY

Chain drive simulations are challenging high-end applications of multibody dynamics. A 2D model with relative coordinates that are tailored to the gross motion of the chain along its trajectory allows the reliable and efficient simulation of the highly complex dynamical behaviour of chain drives without numerical damping. Substructure techniques are used to embed the chain drive model in multibody system models of combustion engines.

Standard time integration methods of multibody dynamics become very inefficient in their application to chain drive models because of the models' large dimension and the large number of high frequency solution components. The characteristic sparse and modular structure of the equations of motion is exploited to reduce the numerical effort by several orders of magnitude.

The new approach is applied successfully to the dynamical analysis of a detailed V6 engine MBS model with more than 1500 rigid and 29 flexible bodies. The simulation is performed on standard PC hardware with a computing time of a few hours. The simulation results are in good agreement with measured data for the cam shaft deviation and tensioner force.

Further improvements are expected from more detailed modelling, such as enhanced contact force laws and flexible guides. Moreover, additional features like non-circular wheels and toothed chains are currently being implemented.

Acknowlegements

The authors are indebted to Dr. W. Rulka (Siemens Transportation Systems) for various contributions to the efficient modelling of roller chain drives.

REFERENCES

[1] *Short-pitch transmission precision roller and bush chains, attachments and associated chain sprockets*. International standard ISO 606:2004(E), International Organization for Standardization, 2004.

[2] G. Niemann, H. Winter. *Maschinenelemente Bd. 3*. Springer Verlag, Berlin, Heidelberg, 1983.

[3] M. Arnold, A. Fuchs, and C. Führer. Efficient corrector iteration for DAE time integration in multibody dynamics. Accepted for publication in *Comp. Meth. Appl. Mech. Eng.*.

[4] O. Wallrapp, W. Kortüm. *MEDYNA – ein Mehrkörperprogramm zur Analyse und Auslegung der Dynamik von spurgeführten Fahrzeugen*. VDI-Berichte Nr. 510, 1984.

[5] W. Rulka. *Effiziente Simulation der Dynamik mechatronischer Systeme für industrielle Anwendungen*. PhD thesis, Vienna University of Technology, 1998.

[6] W. Rulka. SIMPACK Conceptual design – Chain drive systems. Internal report, Intec GmbH, Wessling, 2003.

[7] P. Fritz. *Dynamik schnellaufender Kettentriebe*. PhD thesis, Munich University of Technology, 1997.

[8] G. Zachmann. *Exact and fast collision detection*. Diploma thesis, Department of Computer Science, Darmstadt University of Technology, 1994.

[9] W.O. Schiehlen and P. Eberhard. *Technische Dynamik. Modelle für Regelung und Simulation*. B.G. Teubner, Stuttgart Leipzig Wiesbaden, 2nd edition, 2004.

[10] A.A. Shabana. *Computational Dynamics*. John Wiley & Sons, Inc., New York, 2nd edition, 2001.

[11] R.E. Roberson and R. Schwertassek. *Dynamics of Multibody Systems*. Springer-Verlag, Berlin Heidelberg New York, 1988.

[12] M. Arnold. Simulation algorithms and software tools. To appear in G. Mastinu and M. Plöchl, editors, *Road and Off-Road Vehicle System Dynamics Handbook*. Taylor & Francis, London, 2005.

[13] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*. SIAM, Philadelphia, 2nd edition, 1996.

[14] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin Heidelberg New York, 2nd edition, 1996.

[15] E. Eich-Soellner and C. Führer. *Numerical Methods in Multibody Dynamics*. Teubner-Verlag, Stuttgart, 1998.

[16] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I. Nonstiff Problems*. Springer-Verlag, Berlin Heidelberg New York, 2nd edition, 1993.

[17] U. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1998.

[18] M. Arnold and G. Hippmann. Implicit-explicit time integration in multibody dynamics. Accepted for publication in *Proceedings of IDETC/CIE 2005, ASME 2005 International Design Engineering Technical Conferences*, Long Beach, CA, 2005.

[19] R. Kübler and W. Schiehlen. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamical Systems*, 6:93–113, 2000.

[20] A. Veitl and M. Arnold. Coupled simulation of multibody systems and elastic structures. In J.A.C. Ambrósio and W.O. Schiehlen, editors, *Advances in Computational Multibody Dynamics*, pages 635–644, IDMEC/IST Lisbon, Portugal, 1999.

[21] S. Dietz, G. Hippmann, and G. Schupp. Interaction of vehicles and flexible tracks by co-simulation of multibody vehicle systems and finite element track models. In H. True, editor, *The Dynamics of Vehicles on Roads and on Tracks, Proc. of the 17th IAVSD Symposium, Denmark, 20-24 August 2001*, pages 372–384. Supplement to Vehicle System Dynamics, Vol. 37, Swets & Zeitlinger B.V., 2003.

[22] A.R. Curtis, M.J.D. Powell, and J.K. Reid. On the estimation of sparse Jacobian matrices. *J. Inst. Math. Appl.*, 13:117–119, 1974.

[23] M. Arnold, A. Carrarini, A. Heckmann, and G. Hippmann. Simulation techniques for multidisciplinary problems in vehicle system dynamics. In M. Valášek, editor, *Computational Mechanics in Vehicle System Dynamics*, Supplement to Vehicle System Dynamics, Vol. 40, pages 17–36. Taylor & Francis, London, 2004.