THE ARCHIVE OF MECHANICAL ENGINEERING

VOL. LVI 2009 Number 3

DOI: 10.24425/ame.2009.132099

Key words: multibody dynamics, parallel computing, efficiency, commercial multibody solvers

PAWEŁ MALCZYK [*], *JANUSZ FRĄCZEK* [*]

EVALUATION OF PARALLEL EFFICIENCY IN MODELING OF MECHANISMS USING COMMERCIAL MULTIBODY SOLVERS

Parallel computers are becoming more available. The natural way to improve computational efficiency of multibody simulations seems to be parallel processing. Within this work we are trying to estimate the efficiency of parallel computations performed using one of the commercial multibody solver. First, the short theoretical outline is presented to give the overview of modeling issues in multibody dynamics. Next, the experimental part is demonstrated. The series of dynamics analyses are carried out. The test mechanisms with variable number of bodies are used to gather the performance results of the solver. The obtained data allow for estimating the number of bodies which are sufficient to gain benefits from parallel computing as well as their level. The parallel processing profits are taken into account in the case of contact forces present in the system. The performance benefits are indicated, when the multilink belt chain is simulated, in which contact forces are included in the model.

1. Introduction

Modeling and simulation of dynamics of multibody systems is often pursued by engineers and scientists during designing and building new systems in a wide variety of fields. In many situations, prototyped systems can be complex and the considered phenomena include contact forces, friction and flexibility. Frequently, the simulation of mechanical systems is extended to modeling of devices, extended by pneumatic, hydraulic, electric and electronic systems. In modeling and simulation of complex systems the scientists and engineers often use general purpose commercial solvers which perform computations by multibody system method [1, 2, 9]. The available software uses specially developed computer techniques which enables one to significantly

[*] *Institute of Aeronautics and Applied Mechanics, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, ul. Nowowiejska 24, 00-665 Warsaw, Poland. E-mail: pmalczyk@meil.pw.edu.pl, jfraczek@meil.pw.edu.pl*

reduce the cost and time associated with design, construction and testing of prototypes and systems. Currently, available multibody solvers are capable of simulating small and moderate size systems regarding to the degrees of freedom on conventional computers. The computations can be performed even in real time. Large and very large, realistic, complex problems such as e.g. military vehicles, trains and interdisciplinary applications (biomolecular systems) require powerful computers, significant computational efforts and specialized algorithms. In this case, the simulation time (turnaround time) of the modeled system is one of the essential factors, which affect the speed and costs of designing. One approach that allows to significantly reduce the turnaround time is to apply parallel or distributed processing techniques [8, 10]. These methods, which attract attention of research and development and industrial institutes, are becoming more important, regarding to the development of parallel computing standards and variety of hardware architectures such as shared and distributed memory parallel computing systems [3, 8]. In spite of advances in software and hardware, there are not so many general purpose multibody solvers, which allow to exploit benefits from parallel computing.

The objective of this paper is to evaluate the parallel efficiency of the simulations of mechanical systems performed by a representative of the group of general multibody purpose solvers [1, 2]. During numerical experiments, dynamics analyses of mechanical systems with variable number of bodies are considered. The benefits from parallel processing are evaluated, taking into account the number of bodies in the system. The obtained results can be important in the case of estimation of simulation turnaround times for complex multibody systems.

2. Commercial multibody solvers and parallel processing extensions

Currently, there are many commercial and free software packages for the kinematic and dynamic analysis of mechanical systems. The software helps present-day engineers to design devices in a faster and cheaper manner. Usually available multibody codes are developed and optimized for sequential computations. In spite of the availability of parallel computers and the development of specialized parallel algorithms in the literature [4]–[7] (see [11] for detailed overview), not numerous multibody software allows to perform multiprocessor computations.

One of the commercial multibody software, which possess mentioned previously feature, is MSC.ADAMS [1, 2, 13]. Some of the components of the solver were parallelized and make it possible to carry out parallel computations. The multibody solver can be used to perform parallel processing on shared memory computers, however, it does not possess the option of distributed (e.g. cluster) computing. Starting with the 2005 version, the package provides partially parallelized solver, written in C++. Regarding to the parallel part of the solver, the software has been written using POSIX threads [3, 8]. POSIX is the IEEE standard and defines application programming interface with shell and other utilities compatible with Unix-like and many real time operating systems. The solver, which is a multithreaded application, enables carrying out computations within one to eight threads, both batch based and using graphical user interface. In the case of one thread, which is a default option, the computations are entirely sequential.

In connection with rapid advances in architectures of parallel computers, parallel computing is a strategy and current trend in scientific and engineering applications. It is essential to estimate benefits obtained from parallel processing, compared to sequential counterparts. Evaluation of the efficiency measures is often connected with series of laborious experiments. This issue is discussed in next sections. Prior the demonstration of the experimental results, a short theoretical outline of the dynamic analysis of multibody systems in absolute coordinates is presented.

3. Theoretical outline

The governing equations of motion for a mechanical system can be derived in a variety of forms, depending on the type of coordinates being taken for an analysis. A multibody system can be described by means of absolute coordinates, as it is apparent in the considered commercial package. The basic theoretical foundations [1, 2, 13] for the dynamic analysis of multibody systems in global coordinates are presented in this section. The provided description is often used in other general purpose solvers and can be regarded as a representative one.

A vector of absolute coordinates for an unconstrained body i, located in space can be written in the form:

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i^T & \boldsymbol{\varphi}_i^T \end{bmatrix}^T \tag{1}$$

where $\mathbf{r}_i$ – is a vector of Cartesian coordinates of the origin of a reference frame fixed in the body, and $\boldsymbol{\varphi}_i$ – form a vector of Euler angles with respect to global coordinate system. A vector which describes positions and orientations of all n bodies in the mechanical system can be expressed in the compact form as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T & \dots & \mathbf{q}_n^T \end{bmatrix}^T \tag{2}$$

In many mechanisms, the motion of bodies is constrained, because of the presence of conditions imposed on the relative motion between the pairs of bodies (e.g. joints, driving constraints). The restrictions on absolute coordinates can be formed as a nonlinear function with respect to the coordinates and time

$$\mathbf{\Phi}\left(\mathbf{q}, t\right) = \mathbf{0} \tag{3}$$

The relationships (3) impose restrictions on velocities $\dot{\mathbf{q}}$:

$$\dot{\mathbf{\Phi}}\left(\mathbf{q}, t\right) \equiv \mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{\Phi}_t = \mathbf{0} \tag{4}$$

and on accelerations $\ddot{\mathbf{q}}$:

$$\ddot{\mathbf{\Phi}}(\mathbf{q}, t) \equiv \mathbf{\Phi}_{\mathbf{q}}\ddot{\mathbf{q}} + \left(\mathbf{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}\right)_{\mathbf{q}}\dot{\mathbf{q}} + 2\mathbf{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} + \mathbf{\Phi}_{tt} = \mathbf{0} \tag{5}$$

The matrix $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix and plays central role in kinematic and dynamic analysis of multibody systems [9]. The equations of motion of constrained mechanical system can be expressed in a form of Lagrange equations with multipliers.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(L_{\dot{\mathbf{q}}}^T\right) - L_{\mathbf{q}}^T + \mathbf{\Phi}_{\mathbf{q}}^T\lambda = \mathbf{Q}. \tag{6}$$

where $L = T - V$ denotes Lagrangian function (as a difference between kinetic and potential energy), functions $\mathbf{\Phi}_{mx1}$ are constraint equations, $\lambda_{m\times1}$ is a vector of Lagrange multipliers associated with constraints (3) and $\mathbf{Q} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, t)$ are generalized non-potential forces acting on the multibody system. For the convenience of further expressions, the following quantities are introduced

$$\mathbf{u} = \dot{\mathbf{r}} = \begin{bmatrix} \dot{\mathbf{r}}_1^T & \dot{\mathbf{r}}_2^T & \dots & \dot{\mathbf{r}}_n^T \end{bmatrix}^T = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T & \dots & \mathbf{u}_n^T \end{bmatrix}^T \tag{7}$$

$$\varepsilon = \dot{\varphi} = \begin{bmatrix} \dot{\boldsymbol{\varphi}}_{\mathbf{1}}^T & \dot{\boldsymbol{\varphi}}_{\mathbf{2}}^T & \dots & \dot{\boldsymbol{\varphi}}_{\boldsymbol{n}}^T \end{bmatrix}^T = \begin{bmatrix} \varepsilon_1^T & \varepsilon_2^T & \dots & \varepsilon_n^T \end{bmatrix}^T \tag{8}$$

The kinetic energy of a system can be expressed as

$$T = \frac{1}{2}\mathbf{u}^T\mathbf{M}\mathbf{u} + \frac{1}{2}\bar{\omega}^T\bar{\mathbf{J}}\bar{\omega}. \tag{9}$$

where the mass matrix

$$\mathbf{M} = diag\left(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n\right) = diag\left(m_1\mathbf{I}_{3\times3}, m_2\mathbf{I}_{3\times3}, \dots, m_n\mathbf{I}_{3\times3}\right). \tag{10}$$

and the inertia matrix together with angular velocities are defined as

$$\bar{\mathbf{J}} = diag\left(\bar{\mathbf{J}}_1, \bar{\mathbf{J}}_2, \ldots, \bar{\mathbf{J}}_n\right) \quad \bar{\omega} = \left[\begin{array}{cccc} \bar{\omega}_1^T & \bar{\omega}_2^T, & \ldots & \bar{\omega}_n^T \end{array}\right]^T \tag{11}$$

Dashed quantities are expressed in local coordinate systems. The relationship between the angular velocity $\bar{\omega}_i$ and the time derivatives of Euler angles $\dot{\boldsymbol{\varphi}}_i$ is as follows

$$\bar{\omega}_i = \mathbf{E}_i\left(\boldsymbol{\varphi}_i\right)\dot{\boldsymbol{\varphi}}_i = \mathbf{E}_i\left(\boldsymbol{\varphi}_i\right)\boldsymbol{\varepsilon}_i, \text{ where } \mathbf{E}_i\left(\varphi_i\right) = \begin{bmatrix} \sin\phi_i \sin\theta_i & \cos\phi_i & 0 \\ \cos\phi_i \sin\theta_i & -\sin\phi_i & 0 \\ \cos\theta_i & 0 & 1 \end{bmatrix} \tag{12}$$

Let us introduce additional variable $\mathbf{p}$, which denotes a component of generalized momentum with respect to the angular coordinates.

$$\mathbf{p} = \left[\begin{array}{cccc} \mathbf{p}_1^T & \mathbf{p}_2^T & \ldots & \mathbf{p}_n^T \end{array}\right]^T = L_{\boldsymbol{\varepsilon}}^T = \left(\frac{\partial L}{\partial \boldsymbol{\varepsilon}}\right)^T = \left[\begin{array}{cccc} L_{\boldsymbol{\varepsilon}_1}^T & L_{\boldsymbol{\varepsilon}_2}^T & \ldots & L_{\boldsymbol{\varepsilon}_n}^T \end{array}\right]^T \tag{13}$$

It can be proved that

$$\mathbf{p}_i = \left(\frac{\partial L_i}{\partial \boldsymbol{\varepsilon}_i}\right)^T = \mathbf{E}_i^T \bar{\mathbf{J}}_i \mathbf{E}_i \boldsymbol{\varepsilon}_i \tag{14}$$

Denoting forces and torques which externally act on the bodies as

$$\mathbf{F} = \left[\begin{array}{cccc} \mathbf{F}_1^T & \mathbf{F}_2^T & \ldots & \mathbf{F}_n^T \end{array}\right]^T \quad \text{and} \quad \bar{\mathbf{N}} = \left[\begin{array}{cccc} \bar{\mathbf{N}}_1^T & \bar{\mathbf{N}}_2^T & \ldots & \bar{\mathbf{N}}_n^T \end{array}\right]^T, \tag{15}$$

Lagrange equations with multipliers (6) can be finally expressed in the form

$$\mathbf{M}\dot{\mathbf{u}} - L_{\mathbf{r}}^T + \boldsymbol{\Phi}_{\mathbf{r}}^T \lambda - \mathbf{H}_F^T \mathbf{F} = \mathbf{0}_{3n\times 1} \tag{16}$$

$$\dot{\mathbf{p}} - L_{\varphi}^{\mathrm{T}} + \boldsymbol{\Phi}_{\varphi}^T \lambda - \mathbf{H}_N^T \bar{\mathbf{N}} = \mathbf{0}_{3n\times 1} \tag{17}$$

$$\mathbf{p} - L_{\boldsymbol{\varepsilon}}^T = \mathbf{0}_{3n\times 1} \tag{18}$$

$$\mathbf{u} - \dot{\mathbf{r}} = \mathbf{0}_{3n\times 1} \tag{19}$$

$$\boldsymbol{\varepsilon} - \dot{\boldsymbol{\varphi}} = \mathbf{0}_{3n\times 1} \tag{20}$$

where the matrices $\mathbf{H}_F = \mathbf{I}_{3\times3}$ and $\mathbf{H}_N = diag(\mathbf{E}_1, ..., \mathbf{E}_n)$ are projection matrices, associated with transformation of the forces and torques. Apart from equations (16)-(20), the absolute coordinates must satisfy constraint equations (3), (4), (5). In order to be consistent with the description available in the considered solver, let us introduced forces **f** and torques **n**, that are defined by the user.

$$\mathbf{F} - \mathbf{f}(\mathbf{q}, \mathbf{u}, \boldsymbol{\varepsilon}, \mathbf{F}, \bar{\mathbf{N}}, t) = \mathbf{0}_{3n\times1} \tag{21}$$

$$\bar{\mathbf{N}} - \mathbf{n}(\mathbf{q}, \mathbf{u}, \boldsymbol{\varepsilon}, \mathbf{F}, \bar{\mathbf{N}}, t) = \mathbf{0}_{3n\times1} \tag{22}$$

The complete set of equations constitutes equations (16) to (22) and constraints equations (3)-(5) and form the mixed system of differential-algebraic equations of motion. Special approaches for integrating DAE equations of dynamics are used to obtain solutions at discrete instants of time with desired accuracy. Numerical integration procedures, used in the package, are limited not only to the multistep Gear algorithms (GSTIFF, WSTIFF), predictor-corrector (ABAM) and one step Runge-Kutta subroutines, but include also differential index reduction techniques [1, 2, 13]. These approaches have significant influence on the accuracy of dynamics simulations, however, their description is beyond the scope of this paper.

4. Numerical experiments

This section describes in detail conditions under which numerical tests are performed. The computer hardware is specified in subsection 4.1. Construction and parameters of the test mechanisms are presented in subsection 4.2. Subsection 4.3 describes the efficiency measures, which can be used to estimate the profits from sequential and parallel computations in the case of the multibody simulations with rigid bodies. In subsection 4.4, the performance of numerical simulations of a mechanical system with contact forces is considered.

4.1. Hardware

The Sun Server is used to carry out numerical experiments. The parallel computer with shared memory is equipped with two-socket motherboard, in which two quad-core processors are installed. Each processor is a Quad-Core AMD Opteron Processor 2356 (2.3GHz) with 512kB cache L2 per core. Four 2GB ECC DDR-667 memory modules are set in the motherboard. Suse 10.2 Operating System is installed on the computer with standard GNU compilers (Fortran, C, C++) and the multibody software package.

4.2. Test conditions

Test mechanical systems with variable number of bodies are prepared in order to obtain the efficiency measures of the multibody solver (Fig. 1). The collected experimental data concerns turnaround times. The dynamic simulations are performed in the command line (batch data processing) without the graphical user interface. The dataset is provided with body, joint and force definitions. The description of the mechanisms is generated by a separate source code. Apart from the dataset file, a script file is provided. This file contains solver configuration data, e.g. the number of threads used for computations.

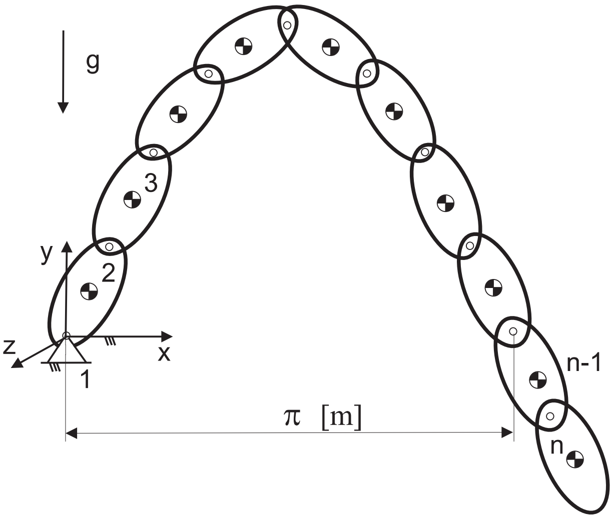

Fig. 1. Test mechanism

Open kinematic chain is depicted in figure 1. The mechanism is used to perform numerical experiments [11]. Each body is connected with its neighbour by a revolute joint. The axes of rotation are perpendicular to the plane of the figure. The bodies are numerated consecutively from the non-movable base body 1 to the end-body n. Each body in the chain has a mass m_i = 1.0 kg and moment of inertia with respect to the body mass center $\mathbf{I}_i$ = diag(1.0)kgm^2. At the initial instant, relative angles in joints are nonzero and the relative velocities are set to zero. The points that define joints and body mass center are located at the sine function with period 2 π. Every half-period of the sine function contains open kinematic chain, which is composed of eight bodies. Ten second dynamic simulations of the mechanical system from figure 1 are considered during the numerical experiments. The mechanism leaves a rest state under the gravity forces. The equations of motion are solved using GSTIFF I3 numerical integration procedure [1, 2, 13].

4.3. Numerical experiment I

Within the numerical tests described in subsections 4.2, series of dynamic simulations of the open chain system are carried out. The execution times (wall-clock times) of the package are gathered to evaluate the speedup. The efficiency measure captures the relative benefit of solving a problem in parallel [8]. Formally, the speedup is defined as the ratio of the serial runtime of the sequential algorithm for solving a problem to the time taken by the parallel algorithm to solve the same problem on the processing elements [8, 10]. Figure 2 depicts turnaround times in terms of the number of bodies in the system. The problem is solved sequentially (one thread) and in parallel from two to eight threads. Figure 3 presents speedups that are obtained from the parallel solver executions.

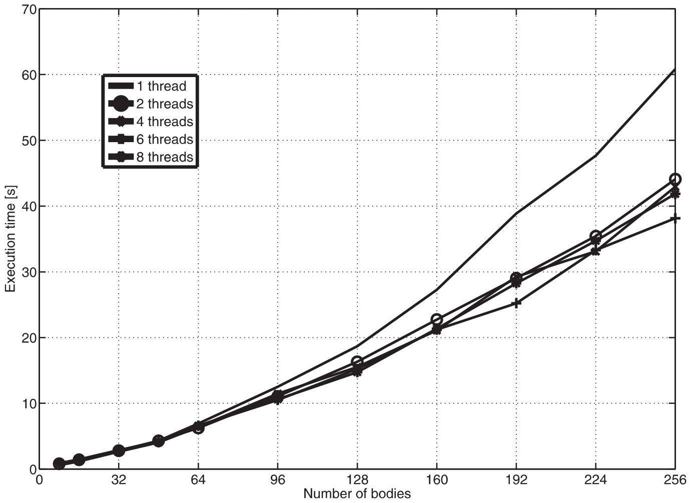

Fig. 2. Execution time

From the figure 2 it follows that the profits achieved from parallel computing are relatively small. The execution time is a nonlinear function in terms of the number of bodies of the mechanical system. This result arises partially from the type of formulation being used for dynamic simulation (global formulation). Moreover, it is worth noting that the dynamic simulations for moderate-sized multibody systems can be carried out even in real time. The evaluated speedup in figure 3 is also a nonlinear function in terms of number of bodies in the system. The benefits are observed only for

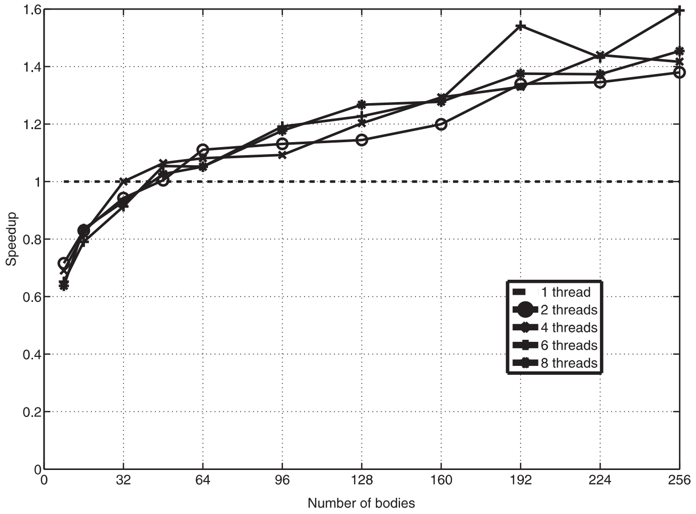

Fig. 3. Speedup

systems with more than 32 degrees of freedom. During the numerical tests, the maximal registered speedup is below 1.6. The moderate profits can be explained as an implication of partially parallelized solver, which concerns only the construction of the Jacobians [1, 2].

An important issue, which should be raised in case of parallel computing, is the scalability of computations. Figure 4 presents the execution time characteristics as a function of the number of threads which are used for computations.

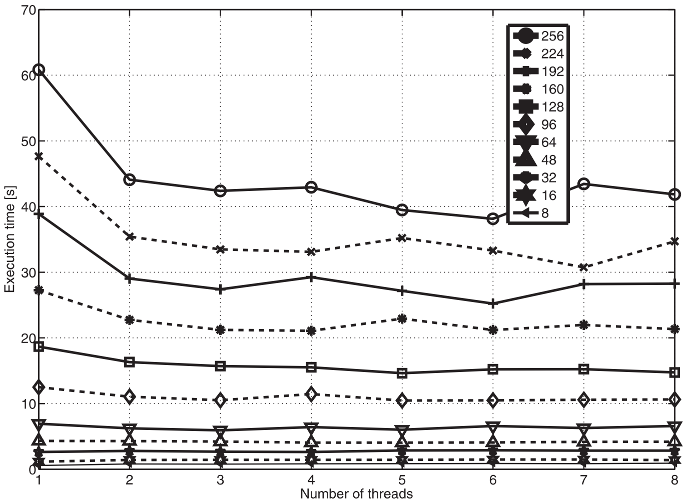

Fig. 4. Scalability of computations

From figure 4, it appears that the profits from parallel computing are observed for large number of bodies. For moderate-sized multibody systems, adding more threads does not cause significant efficiency growth, however, it can help to balance the computational load over the available cores. For dynamic simulations of systems with fewer than 96 bodies, the curves are nearly constant, which means no benefits from parallel processing.

4.4. Numerical experiment II

Apart from the series of dynamic simulations of mechanical multi-rigid systems, which are presented in subsection 4.3, the numerical experiment with contact forces is considered in this subsection. The described model is taken from the reference [1]. The model from figure 5 consists of two spheres that differ in size. The 100-link belt is stretched over the spheres. Gravity is turned off, and the torque is applied to both areas. The model includes 200 three-dimensional contact forces, which act between the spheres and every element of the chain.

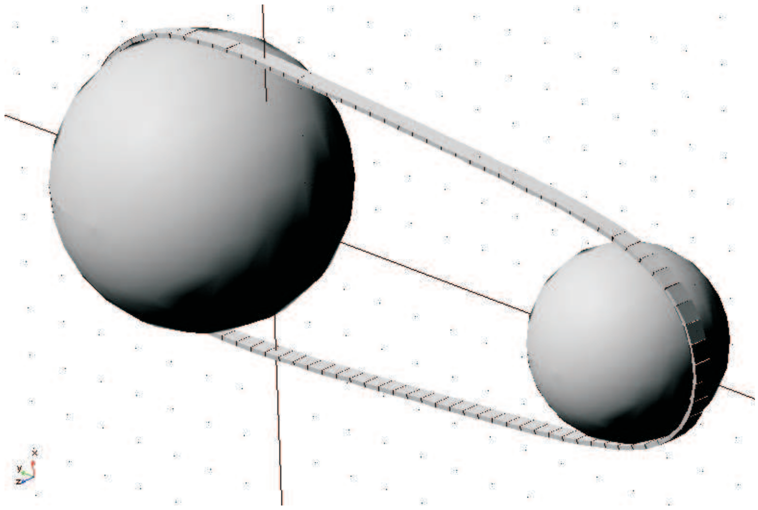

Fig. 5. Belt mechanism

Dynamic simulation of the belt mechanism is carried out over 1.0-second period of time. The integration procedure GSTIFF is taken for simulation purposes. The minimal integration step is set to 10^{-16} s. Figure 6 depicts the dependence of the speedup and the turnaround time on the number of threads used for computations.

From figure 6 it follows that the maximal obtained speedup amount to about 3 for six threads working in parallel. Additional threads do not improve significantly the computational efficiency. The obtained result is a moderate

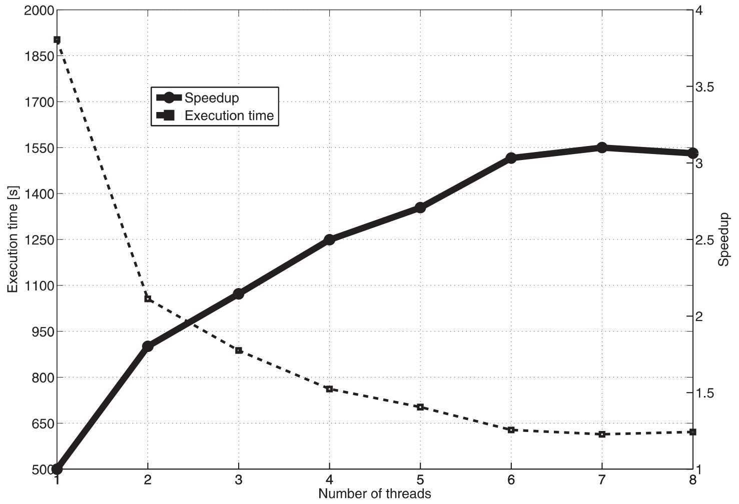

Fig. 6. Parallel performance in case of contact forces

profit, taking into account the presence of eight cores in the parallel computer. The efficiency measures from this test can be transferred and generalized for a variety of belts, chains, track vehicles, which are often analyzed in engineering practice.

5. Summary and conclusions

Theoretical outline of the multibody modeling issues has been presented in the paper on the basis of one of the commercial multibody package. The overview can be regarded as a representative one, because many multibody solvers employ global formulations for the dynamic analysis. Parallelism and parallel computing abilities are demonstrated and investigated for the multibody package. Two numerical experiments are carried out to indicate the performance measures. For the dynamic analysis of the open chain system, the speedup at the level of 1.6 is obtained, in turn, the analysis of the multilink belt with contact forces gives the speedup at the level of 3 for eight cores in the parallel computer. The demonstrated performance results can be important in the case of anticipation of the simulation turnaround times for complex multibody systems.

Parallel computing is not just a strategy for achieving high performance, but also a way to analyze more and more complex multibody systems. There are commercial multibody software packages which make it possible to perform parallel computations, however, the obtained benefits are relatively small, taking into account the current stage of development of the dynamic algorithms.

This work was supported by the European Union in the framework of European Social Fund through the Warsaw University of Technology Development Programme, and by the Institute of Aeronautics and Applied Mechanics statutory funds.

The work was co-financed by the European Regional Development Fund within the framework of the 1. priority axis of the Innovative Economy Operational Programme, 2007-2013 – PROTEUS under grant number PO IG.01.02.01-00-014/08-00.

Manuscript received by Editorial Board, May 05, 2009;
final version, August 31, 2009.

REFERENCES

[1] ADAMS MSCSoftware Knowledge Base, www.mscsoftware.com.

[2] ADAMS Solver, ADAMS View Training Guide, 2005.

[3] Butenhof D.: Programming with POSIX Threads, Addison-Wesley, Boston, 1997.

[4] Critchley J. H., Anderson K. S.: A Parallel Logarithmic Order Algorithm for General Multibody System Dynamics, Multibody System Dynamics, 2004, Vol. 12, pp. 75-93.

[5] Critchley J. H., Binani A., Anderson K. S.: Design and Implementation of an Efficient Multibody Divide and Conquer Algorithm. Proceedings of the ASME 2007 International Design Engineering Technical Conferences, 2007, Las Vegas, Nevada, USA.

[6] Featherstone R.: A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 1: Basic algorithm. International Journal of Robotics Research, 1999, Vol. 18, pp. 867-875.

[7] Featherstone R.: A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 2: Trees, loops, and accuracy. International Journal of Robotics Research, 1999, Vol. 18, 876-892.

[8] Grama A., Gupta A., Karypis G., Kumar V.: Introduction to Parallel Computing, Second Edition, Addison Wesley, 2003.

[9] Haug E. J.: Computer Aided Kinematics and Dynamics of Mechanical Systems. Volume I: Basic Methods, Allyn and Bacon, 1989.

[10] Mattson T. G., Sanders B. A., Massingill B. L.: Patterns for Parallel Programming, Addison-Wesley, Boston, 2005.

[11] Malczyk P., Frączek J.: Cluster Computing of Mechanisms Dynamics using Recursive Formulation, Multibody System Dynamics, 2008, Vol. 20(2), pp. 177-196.

[12] Malczyk P., Frączek J.: Ocena efektywności obliczeń równoległych w modelowaniu mechanizmów z wykorzystaniem pakietów komercyjnych, Proceeding of the XXI Polish Conference on Theory of Machines and Mechanisms, Bielsko Biała-Szczyrk, 22-25 September 2008, pp. 231-238, (in Polish).

[13] Wojtyra M., Frączek J.: Metoda układów wieloczłonowych w dynamice mechanizmów. Ćwiczenia z zastosowaniem programu ADAMS. Wydawnictwa Politechniki Warszawskiej, 2007 (in Polish).

Ocena efektywności obliczeń równoległych w modelowaniu mechanizmów z zastosowaniem pakietów komercyjnych

Streszczenie

W związku z szybkim rozwojem komputerów równoległych naturalną drogą poprawienia wydajności obliczeń symulacyjnych staje się wykorzystanie możliwości przetwarzania równoległego. W ramach niniejszej pracy podjęto próbę oceny efektywności obliczeń równoległych, w przypadku obliczeń jednym z pakietów komercyjnych do modelowania układów mechanicznych. Początkowo zaprezentowano podstawy teoretyczne metody układów wieloczłonowych. Następnie zademonstrowano wyniki eksperymentalne. W pracy zrealizowano obliczenia dynamiki testowych układów mechanicznych, o zmiennej liczbie członów, podczas których zarejestrowano czasy obliczeń sekwencyjnych i równoległych. Uzyskane charakterystyki wydajnościowe pozwoliły oszacować liczbę ogniw, przy której otrzymuje się zyski z obliczeń równoległych, a także wskazać ich poziom. W zakresie badań uwzględniono potencjalne korzyści ze stosowania przetwarzania równoległego w przypadku sił kontaktu, obecnych w modelu, na przykładzie symulacji wieloczłonowego mechanizmu łańcuchowego.