High-performance simulations of continuously variable transmission dynamics

S.G. Orlov[1], N.B. Melnikova[1,2], Yu.G. Ispolov[1], N.N. Shabrov[1]

[1]Peter the Great St. Petersburg Polytechnic University, [2]National Research University ITMO

The paper describes a parallel computational model simulating continuously variable transmission (CVT) dynamics. A specific feature of the CVT model is the combination of relatively small problem sizes (about 1000 unknowns) and high computational costs (up to several weeks of sequential code computations for a thirty seconds simulation period). The main source of computational complexity here is the calculation of non-linear contact forces acting between transmission parts at each step of the explicit time integration procedure. Below we analyze simulation workflow and runtime load distribution among the application modules. Based on the profiling data, we present a task parallel multithreaded implementation of the model over shared memory and discuss steps towards further parallelization.

1. Introduction

The present paper focuses on the design and implementation of the high-performance computational model of a continuously variable transmission (CVT) dynamics. CVT has been extensively used in the automotive industry for last decades. Numerical simulations of CVT dynamics are commonly employed in the engineering practice at the design stage. However, at the high level of models complexity and realism, simulations become very time-consuming. Depending on the model complexity, it can take from several weeks to several months of sequential code computations for a thirty seconds model time period.

There have been developed a number of CVT models with different levels of realism and complexity. These include: (a) primitive low-dimension chain models [1]; (b) discrete multi-body mechanical models [2,3]; (c) continuous FEM models [4]. An important requirement for chain models is the ability to represent each pin and plate separately, hence low-dimension chain models similar to [1] were not considered in our research. Our chain models differ from those described in [2,3] in that they are more detailed and consider 3D motion of chain. Continuous FEM models are extremely high-weighted due to a large number of contact interactions in the system; their application is, in fact, limited to detailed static simulations of local contact interactions between typical CVT parts [4, 5] coupled to a multi-body system analysis [5].

Our original CVT model has been specially designed in order to keep an optimal balance between model realism and computational costs. Reliability of this model has been validated during more then ten years of successful application in the real industrial environment. However, even this relatively light-weighted model is quite time-consuming: for a time step $h = 2{,}1\cdot 10^{-8}$ sec, model time $T = 1$ sec, serial mode simulation takes about 60 hours on CPU Intel G2130 @3.2 GHz.

Parallel implementation of the CVT model has naturally become our next step in building a really fast application. Our final goal is reducing typical computational times by 100 times, to get simulation done within 30-40 minutes. At the current working stage, a task based OpenMP parallelization has been performed for the most time-consuming portion of the code (see Sections 4-5 for details).

The rest of the paper is organized as follows: Section 2 introduces the basic principles of CVT work; Section 3 describes mathematical foundations of our CVT model, Section 4 considers numerical implementation issues and serial performance analysis, Section 5 presents parallel benchmarking results, and, finally, Section 6 summarises conclusions and future work directions.

2. CVT mechanics basics

A schematic view of the CVT is presented in Figure 1. Two pairs of conic sheaves (pulleys) are installed on two shafts (driving and driven) and connected with a steel chain. The shafts are on elastic supports. One pulley in each pair is able to move along its axis, the other one is fixed. The axial motion of the pulleys provides an infinite number of gear ratios in a fixed diapason. Besides, there is a hydraulic system controlling the moving sheaves, and a stabilization system for CVT gear ratio.

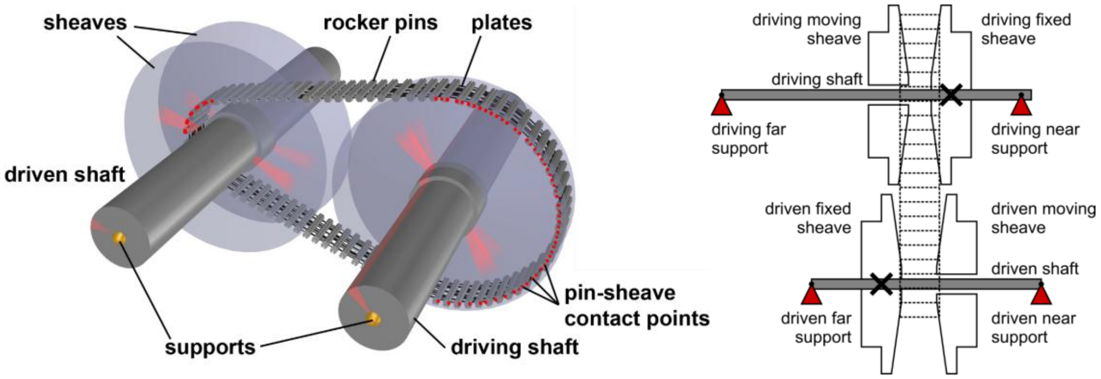

Figure 1. A schematic view of continuously variable transmission

The chain consists of rocker pins and plates (Figure 2). The driving torque is transmitted due to friction forces in contact points between pins and sheaves.

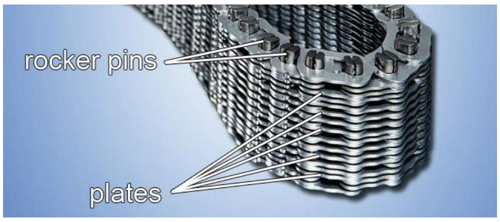

Figure 2. CVT chain: pairs of contacting pins connected with plates

3. Mathematical model of CVT dynamics

The dynamics of a holonomic system of elastic bodies can be described using standard mathematical formalism of Lagrangian mechanics [6]. In application to our CVT model, this can be written in the following form:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q, \qquad (1)$$

Where $L = T + \Pi$ is Lagrangian (sum of kinetic and potential energies),
$T = T(q, \dot{q})$ is the kinetic energy, $\Pi = \Pi(q)$ is the potential energy.

$Q = -\dfrac{\partial \Phi}{\partial \dot{q}} + Q^{applied}$ is the vector of generalized forces, $\Phi(q, \dot{q}) = 0$ is the Rayleigh dissipative function, and $Q^{applied}$ is the applied forces vector.

Substitution of potential and kinetic energies expressions into (1) gives us a system of ordinary differential equations with a non-linear right hand side:

$$A\ddot{q} = F(t, q, \dot{q}) \qquad (2),$$

Where $A = A(t, q)$ is a symmetric and positive-definite inertia matrix and the right-hand side vector is a sum of elastic, damping and external forces.

CVT chain consists of many separate solid bodies, which are rocker pins and plates (Figure 2); the total number of them is over 1000, and there are many contact interactions between these bodies. Besides, it’s essential to take local deformations of the bodies into account in a mathematical model in order to correctly predict the behaviour of the entire system.

Mathematical expressions for calculating potential and kinetic energies for the parts of our CVT model have been published in [7]. In short, we suppose that pins, pulleys and shafts have inertia, and plates do not. Plates' masses are added to the pins connected by those plates (see (Figure 3a for a cross-sectional view of pins connected by plates). The pins experience extension and bending; we neglect pins torsion. The plates (Figure 3c,d) work mostly for extension, but also for bending and torsion. The shafts work mostly for tension, and, in a less degree, for extension and bending. Two halves of each pin roll over each other (Figure 3b). Pin halves contacts and contacts between pin ends and pulleys are modelled with Herz theory of contact [8].

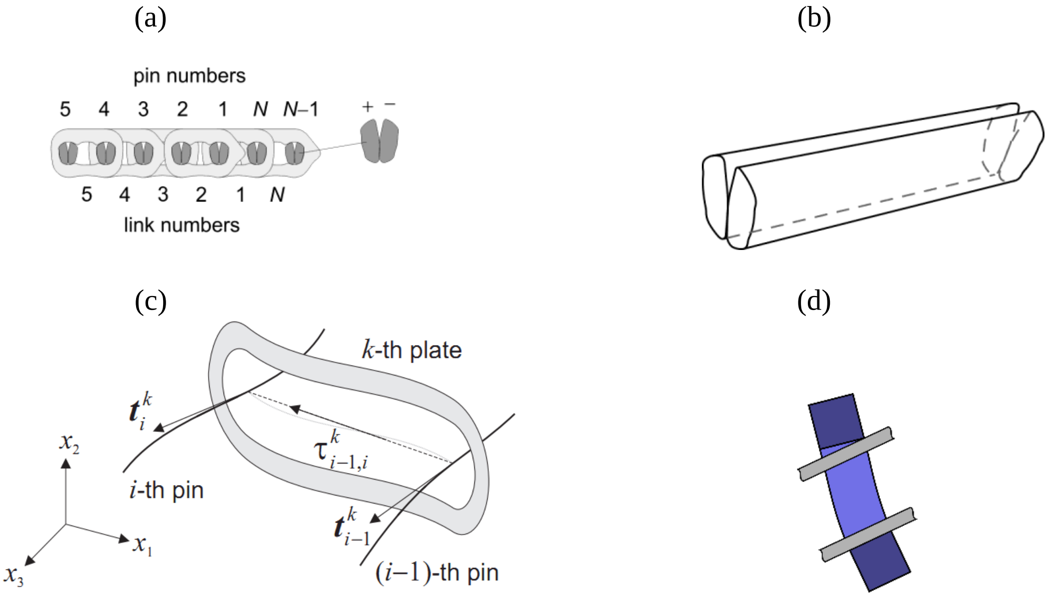

Figure 3. (a) Cross-sectional view of pins connected with plates; (b) two pin halves rolling over each other; (c) a chain plate; (d) bending of a chain plane and independent rotations of pins

4. Numerical implementation

4.1 Time integration scheme

The well-known four-stage Runge-Kutta time integration scheme was used to integrate ODE system (2). The scheme is explicit; it is stable in the region on plane (hλ) shown in Figure 4 (here λ is the maximal eigenvalue) [9]. The highest frequencies in the model correspond to axial tension of the pins; the values typically are about 10^7-10^8 rad/sec, which requires time integration step value less then 10^{-7}-10^{-8} sec.

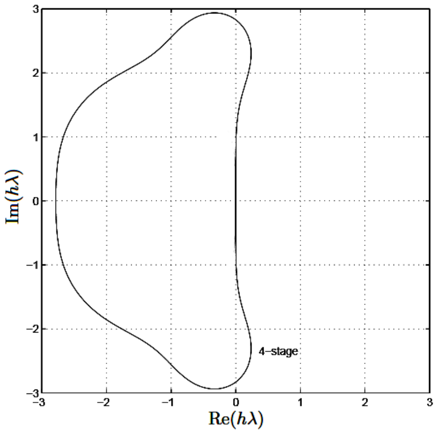

Figure 4. Explicit RK4 time integration scheme – stability region

4.2 Software implementation

Functionality of our CVT dynamics simulator currently includes:

- Building various CVT models from a set of standard parts in graphical pre-processor;
- The computational core: integrating ODE systems + generalized eigenvalue problem solution;
- Visualization utilities (2D, 3D, video recording);
- Building plots and spectrograms;
- Logging user actions;
- Help system.

The code has been implemented in object-oriented approach in C++. Graphical user interface employs Qt5 library functionality. The application is cross-platform and can run both on Windows and Linux. Other software technologies used in the code are:

- V8 (JavaScript implementation)
- Doxygen (for developers' and users' documentation systems)
- om (in-house object model)
- cmake (meta-building system)

4.3 Work load distribution in a serial mode

Both serial and parallel simulations were run on a computational node of the supercomputing cluster TESLA in the Laboratory of Virtual Reality (St. Petersburg State Polytechnic University). The node is equipped with a six-cored Intel Xeon CPU X5660 2.80GHz, 8 GB RAM with 3 memory channels and 500 GB of disk storage. Operative system is Linux Ubuntu, C++ compiler version is GNU v.4.5.1 with OpenMP v.3.0.

According to the profiling results for a sequential code, the "heaviest" computational module (58% of computational time) is the one calculating contact forces between pins and plates ("ChainForces" module in Figure 5). Chain-pulley contact forces calculation takes 18% of computational time. Inertia matrix decomposition and constraints elimination take 7% and 5% of time, correspondingly. Computation of other forces (shafts support, pulley control forces et cetera) takes about 2% of computational time. "Rk4Step" in Figure 5 denotes a load portion taken by the whole simulation (100%, naturally). The rest 9% of the global elapse time are not occupied with the specified modules; they are spent for I/O, memory operations and other overheads.

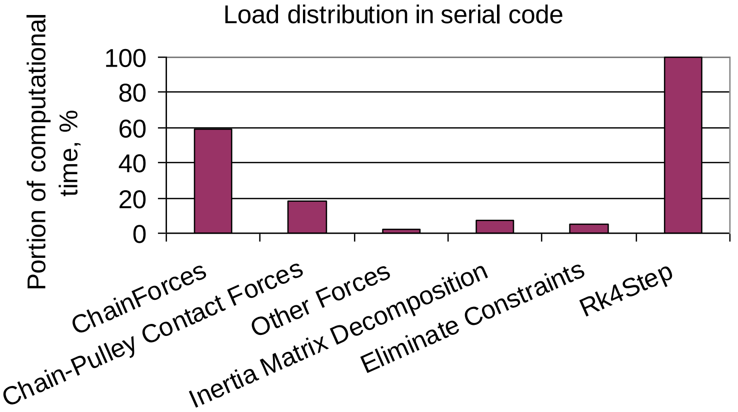

Figure 5. Load distribution between the computational modules in the serial mode

4.4 Sources of parallelism

Logical workflow of the simulation is shown in Figure 6. The modules within the group of concurrent tasks are totally independent and, in principle, can be run simultaneously. The workflow steps connected by arrows are logically sequential and transfer their results to the next element in the computational sequence. After each time integration step, all contact pairs in the model are checked for gap opening: if a normal reaction in some contact pair becomes negative, the gap is open and the contact pair will not be active in the next integration step.

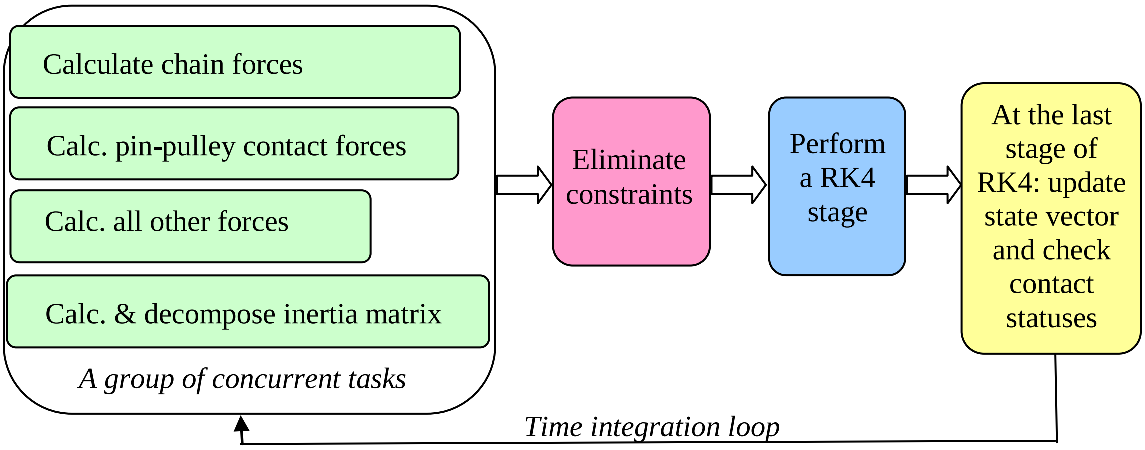
Calculate chain forces
Calc. pin-pulley contact forces
Calc. all other forces
Calc. & decompose inertia matrix
A group of concurrent tasks
Eliminate constraints
Perform a RK4 stage
At the last stage of RK4: update state vector and check contact statuses
Time integration loop

Figure 6. Logical workflow of the simulation within one RK4 step: concurrent and sequential modules

Besides task concurrency, there is a second, even more important, source of parallelism based on data locality naturally resulting from the locality of contact interactions between transmission parts. Typically total number of chain pins *nPins* = 80 ÷ 100; each pin interacts with a set of 20-30 plates and with two neighboring pins. Computation of contact forces between a pin and a pulley is totally independent from neighboring pins. Hereby, one pin (with a set of calculations associated with this pin) can represent a grain of parallelism in this application.

4.5 OpenMP parallelization algorithm

Locality of interactions in a CVT chain has been used in the OpenMP parallelization of the chain forces calculation module. The chain was split into *np* parts, where *np* is the number of cores; *np* varied from 1 to 6. Globally, simulation was controlled by a master thread which invoked a group of parallel threads each time the "chain forces" module was called, namely four times per time step in RK4 method (Figure 7).

```
…//initialize simulation
for(i=0;i<nsteps;i++){ //time integration loop
   for(k=0;k<4;k++){ //a loop over RK4-stages
#pragma omp parallel sections //fork threads
   {
#pragma omp section
   {
   CalcChainForces(0, nPins/np);
   }
#pragma omp section
   {
    CalcChainForces(nPins/np+1, 2*nPins/np);
   }
…
   }// join threads
…/* Do the rest of work: calculate other forces, update
and decompose inertia matrix, eliminate constraints,
perform one RK4 stage */
   }//End of RK4-stage loop. Update state vector. Check
contact statuses.
}//end of time integration loop
…
```

Figure 7. OpenMP parallelization of the chain forces calculation module

Parallel efficiency benchmarking results will be discussed further in Section 5.2.

5. Simulation results

5.1 Test case description and basic dynamics analysis results

The test case simulated dynamics of a CVT for 1.118 sec model time period. Time integration step was constant and equaled to 10^{-8} sec. The results of simulations have been visualized and analyzed in the in-house post-processing component of our software system.

Local behavior of CVT components, as well as global characteristics (such as efficiency) can be observed through many postprocessor values. Figure 8 shows some examples (plate and link tension force, pin axial force, chain torsion angle, support force and CVT efficiency).

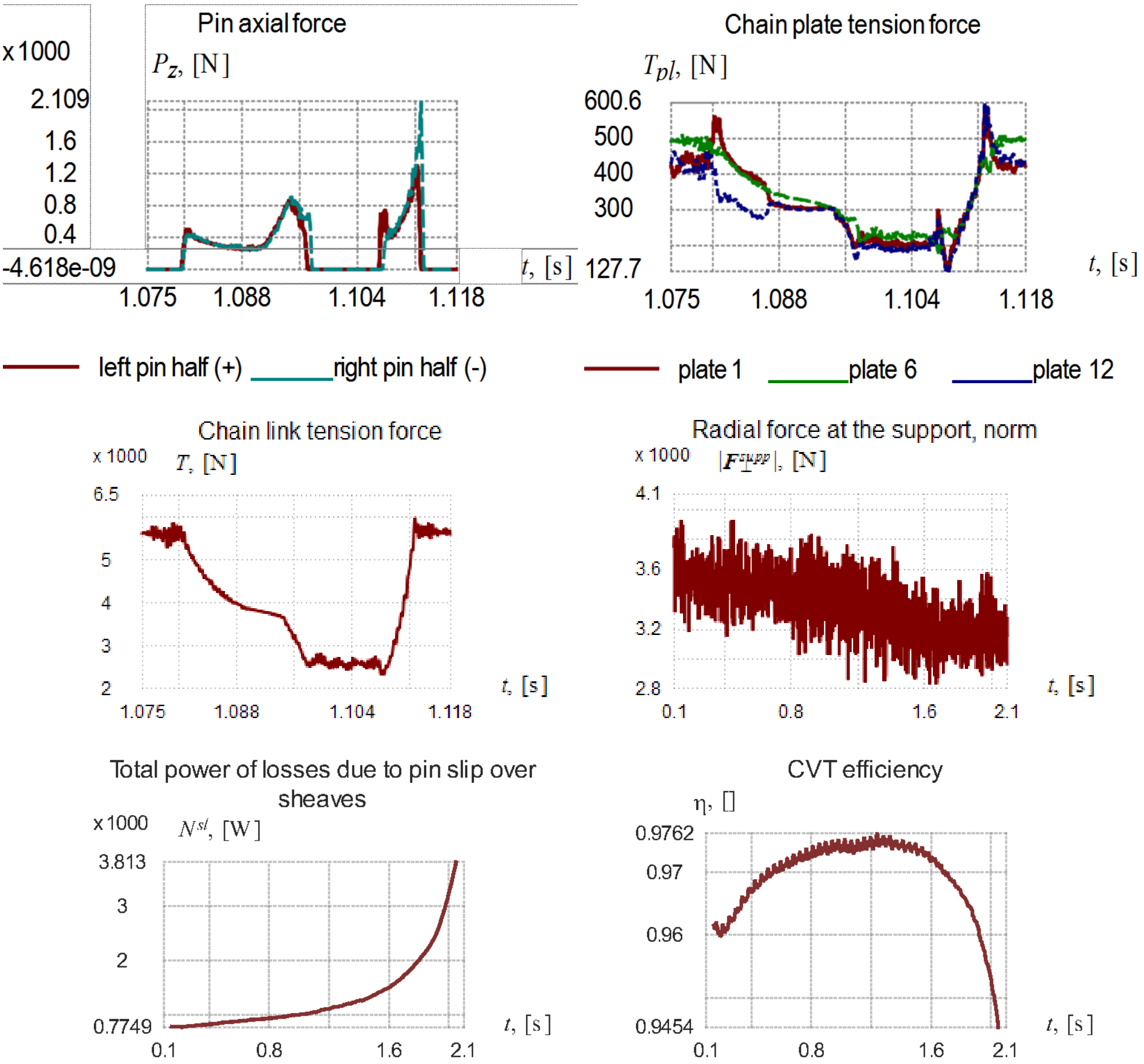

Figure 8. Simulation results: time history plots for selected chain parameters

5.2 Parallel performance benchmarking

In parallel performance tests, model simulation period was 0.1 sec; time integration step was 10-8 sec. In serial mode, this test takes about an hour on Intel Xeon X5660 (see Table 1 for computational times).

OpenMP parallelization reduced computational time almost twice (in 1.9 times, to be precise), when using all 6 cores (see Figure 9b, Table 1). The parallel portion of the code (chain forces calculation) scales almost perfectly, producing 1.8, 3.5 and 5.5 times acceleration on two, four and six cores, correspondingly. The sequential portion of the code naturally slows down the whole program.

According to Amdahl's law, the maximal speedup of the code with a 42% portion of sequential work on six cores would equal to 1.94:

$$\frac{42\% + 58\%/6}{100\%} = 1.94$$

We have obtained speedup=1.9; hence, the OpenMP parallelization of the chain forces calculation module was almost ideal, with minor synchronization and memory bandwidth losses.

Our next optimization steps will include parallelization of the module computing chain-pulley contact forces and inertia matrix decomposition module.

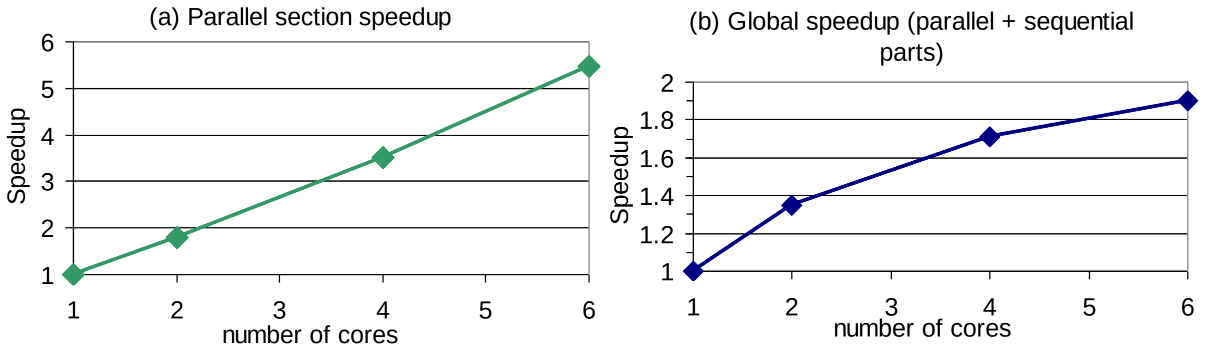

Figure 9. (a) Speedup of the parallel section of the code (chain forces calculation module); (b) Speedup of the whole CVT code, including its serial part

Table 1. Parallel simulation time, minutes

Number of cores	**1**	**2**	**4**	**6**
Simulation time, min.	51	38	30	27

6. Conclusions and future work

We have developed a discrete model of continuously variable transmission dynamics and implemented it in the object-oriented software tool. The model has been successfully validated during more then ten years of its application in the real industrial environment. The most time-consuming portion of the code has been parallelized on a 6-cored CPU with OpenMP in the task-parallel paradigm. The obtained code speedup (1.9) is close to the maximal speedup (1.94) estimated by the Amdahl's law for the code with a 42% portion of sequential calculations.

Our next steps will naturally include parallelization of the rest of modules and exploiting functional concurrency of the workflow, according to Figure 6. Then, in order to obtain a speedup of about 100, we are going to implement MPI parallelization with local data storage (one pin per one process) and organize ring communications between processes (pins). A latency of 1 microsecond will hopefully allow hiding MPI communications behind computations: according to our benchmarks, a portion of serial computations per one pin takes 6 microseconds at each time integration step on Intel Xeon CPU X5660 2.80GHz. Additionally, we will perform OpenMP parallelization of the nested loops over plates and study the efficiency of this hybrid parallelization.

Acknowledgements

The authors thank Russian Fund of Fundamental Research for financial support under the frame of Grant № 13-07-12077.

References

1. M. Bullinger, F. Pfeiffer, and H. Ulbrich, Elastic modelling of bodies and contacts in continuous variable transmissions, Multibody Syst. Dyn. 13, 175–194 (2005).
2. L. Neumann, H. Ulbrich, and F. Pfeiffer, New model of a CVT rocker-pin chain with exact joint kinematics, J. Comput. Nonlinear Dyn. 1(2), 143–149 (2006).
3. L. Neumann, H. Ulbrich, and F. Pfeiffer, Optimization of the joint geometry of a rocker pin chain, Machine Dyn. Problems 29(4), 97–108 (2005).
4. M.T. Lates, R.G. Velicu, R. Papuc. Multiscale modeling of chain-guide contact by using tests and FEM. 11th World Congress on Computational Mechanics WCCM XI (2014).
5. Th. Geier, M. Foerg, R. Zander, and H. Ulbrich, F. Pfeiffer, A. Brandsma, A. van der Velde, Simulation of a push belt CVT considering uni- and bilateral constraints, Journal of Applied Mathematics and Mechanics 86(10), pp. 795–806 (2006).
6. V. Arnold, Mathematical Methods of Classical Mechanics, 2nd edition (Springer, New York, 1989), Chap. 3.
7. N. Shabrov, Yu. Ispolov, S. Orlov. Simulations of continuously variable transmission dynamics. Journal of Applied Mathematics and Mechanics 94(11), pp. 917-922. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. DOI 10.1002/zamm.201300249 (2014).
8. K. Johnson, Contact Mechanics (Cambridge University Press, Cambridge, 1987).
9. E. Hairer, G. Wanner. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, p.17. Springer-Verlag Berlin Heidelberg (1996).