Transmission line co-simulation of rolling bearing applications

Dag Fritzson, Jonas Ståhl and Iakov Nakhimovski
SKF Engineering Research Centre, Modelling & Simulation, Göteborg, Sweden
dag.fritzson@skf.com

Abstract

Connectivity, i.e., connecting different types of simulation tools, becomes increasingly important. The objective is to be able to do cost effective simulations of a large system by utilizing existing tools together. The TLM (Transmission Line Modelling) co-simulation technique is stable, does not have numerical problems, and is proven. It has potential for being a co-simulation technique of general usage. The aim of the paper is an evaluation of TLM technique for rolling bearing applications via a study of a grinding spindle model. The conclusion is that the results of TLM and non-TLM simulations are similar for a realistic application with rolling bearings and applicable for engineering purposes. The numerical stability and performance are satisfactory for the considered application.

Keywords: Co-simulation, TLM, bearing simulations, multi-body dynamics

1. Introduction

In the area of modelling and simulation of mechanical systems one can identify many different classes of models and corresponding tools. The specialization leads to different focus for different tools. For instance, consider the differences in focus for the equation-based multi-physics Modelica models [1], general multibody models in MSC.ADAMS [2], models with detailed contact definitions in SKF's BEAST (BEAring Simulation Tool) [3, 4], flexible components as modeled in FE tools [5, 6], etc.

There is no single universal tool that can be used to analyze all kind of problems with maximum efficiency and accuracy. One might say that every tool is optimized for a certain kind of tasks. This has lead to the creation of a large number of different specialized simulation frameworks. Large industrial models often use some special features of the particular framework. This makes the translation of models between different frameworks very complicated and costly.

The need to bring different components into a complete more tightly coupled simulation is therefore justified. This allows higher accuracy and preserves the investments in the components.

Co-simulation is one of the possible techniques of joining existing sub-models into a more complete model. The joining technique is very different depending on the particular application area (see, e.g., surveys in [7, 8, 9]). One method that was earlier used to enable closer interaction between such sub-models in a coupled simulation is transmission lines modelling (TLM). The TLM uses physically motivated time delays to separate the components in time and enable efficient co-simulation. The technique has proven to be stable and was implemented for coupling of different sub-systems [10, 11, 12, 13]. TLM is further described in Section 3.1. Therefore, it has a potential for being a co-simulation technique of general usage. The aim here was to investigate its suitability for rolling bearing applications. The evaluation was performed within the context of SKF BEAST.

2. Application cases

Several different application cases have been studied in order to evaluate the TLM. In the first two examples, two BEAST models were connected to each other through TLM.

2.1 Two bodies

This model was used for fundamental tests in order to investigate the basics of the concept. Two bodies are connected to each other as visualised in Figure 1.

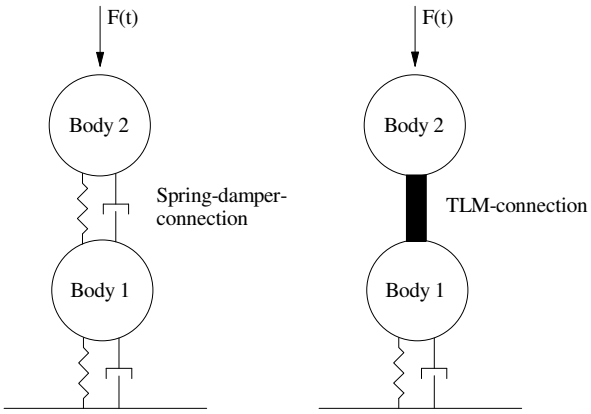

Figure 1. Two bodies connected to each other.

A spring and a damper connect Body 1 to the ground. The connection between the bodies was also initially represented by a spring and a damper, see Figure 1. Both constant and time-dependent forces were applied to Body 2. This complete model was compared to a model where the spring and the damper between the bodies were replaced by a TLM connection; see Figure 1. Same load conditions were applied in the two cases.

Static load cases give identical results. Differences in the dynamic cases were as expected. Since this model was used for fundamental tests only, no further results are presented in this paper.

2.2 Grinding spindle application

A more complex model compared to the previous one is the Grinding spindle application; see Figure 2-Figure 3. The shaft is supported in a housing by two bearings. A detailed model of a DGBB was used for one bearing. A simplified stateless model referred as "SPB" (Simplified Parametric Bearing) was used for the other bearing. The grinding wheel is attached to a hub that is mounted close to the end of the shaft. A circular mass represents a possible balancing unit. A pulley attached to the opposite end of the shaft drives the shaft. The DGBB-model consists of an inner ring, an outer ring, a cage and 11 balls. Figure 4-Figure 6 show all parts of the model separately. Altogether, the grinding spindle model consists of 20 bodies of which 14 represent the DGBB.

The stiffness of the SPB supporting the shaft was kept similar to the DGBB stiffness. The gravity was included.

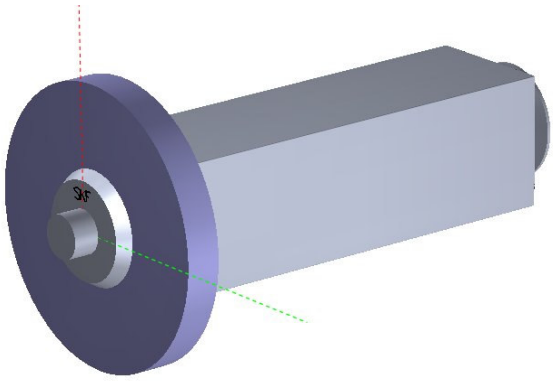

Figure 2. The grinding spindle application, view 1.

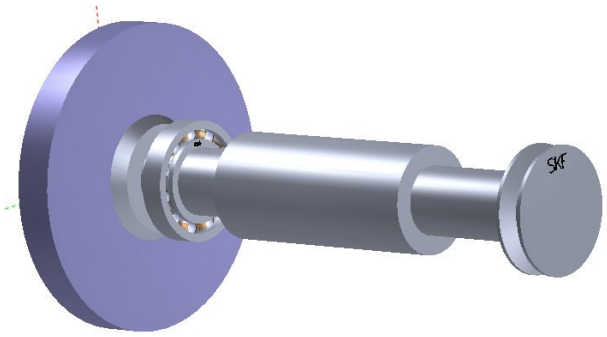

Figure 3. The grinding spindle application, view 2.

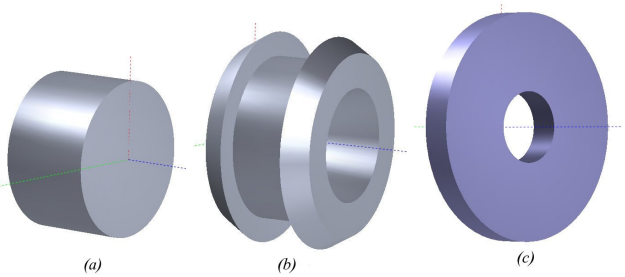

Figure 4. The balancing unit (a), the hub (b), and the grinding wheel (c).

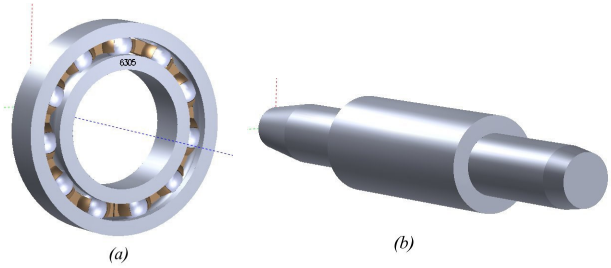

Figure 5. The DGBB (a) and the shaft (b).

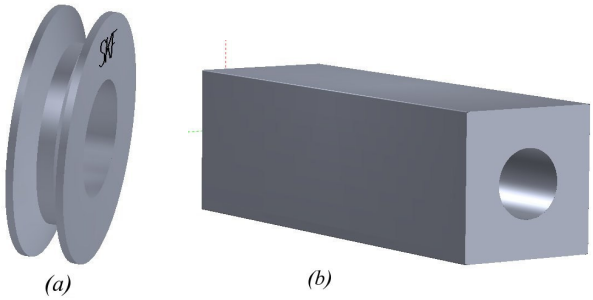

Figure 6. The pulley (a) and the housing (b).

2.2.1 Load conditions

In the studied load cases, the rotational speed of the shaft was 5000 rpm created via rotating "belt-forces" acting on the pulley. The bearings were pre-loaded

axially with 100 N. A radial force of 1000 N was applied to the grinding wheel in order to simulate the grinding force. The loads and the speed were ramped up using a smooth step.

2.2.2 TLM-connections

To study the TLM-concept, some of the connections between the bodies in the original model were replaced by TLM-connections. A schematic view of the models is presented in Figure 7-Figure 8. In this case, both parts on each side of a TLM-connection were modelled with BEAST. However, there are no theoretical restrictions to the choice of software. The bearing model may be analysed with BEAST, and the surrounding parts like the shaft and the housing may be modelled in another multi-body simulation software like MSC.ADAMS. The parameters of the TLM-connections and the stiffness in the non-TLM connections had realistic physical properties. The results from the analyses of the Grinding spindle application are as previously mentioned, presented in Section 4.

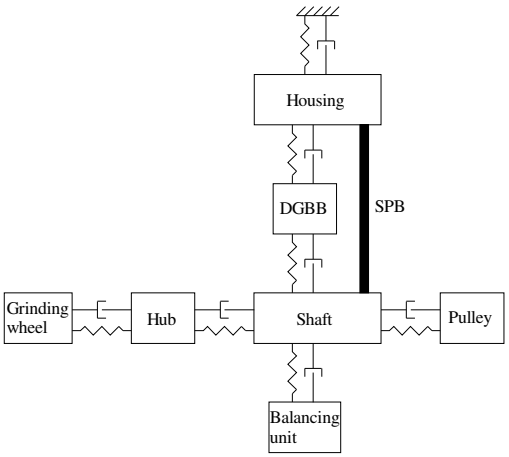

Figure 7. Schematic view of the original model.

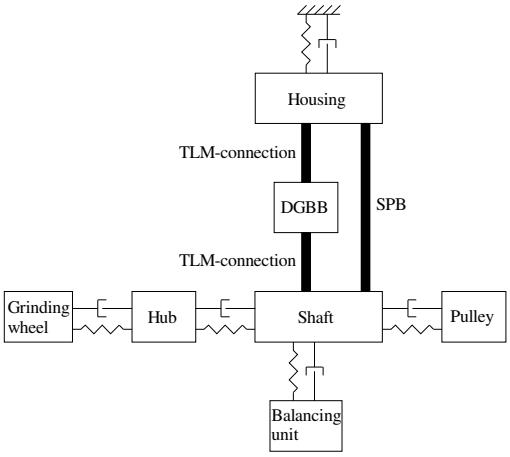

Figure 8. Schematic view of the model with TLM-connections.

3. Co-simulation techniques

Different kinds of co-simulation techniques exist, both the ordinary co-simulation techniques with exchange of state variables, and the Transmission Line Modelling co-simulation. The solution chosen for implementation in this study is the TLM concept. In the long term, the aim is to handle connections between simulation processes in different systems, such as between BEAST and some other multi-body software, by using TLM-technique. In the present investigations, BEAST/BEAST TLM connections were studied.

3.1 Transmission Line Modelling (TLM)

Transmission line modelling (TLM) has been used to model multibody and pneumatic systems with success, see references [11, 14, 15]. Since the TLM technique utilizes distributed simulators it can also be seen as a way to introduce co-simulation in a systematic way.

3.1.1 TLM Basics

Consider the system of two connected masses, see Figure 9.

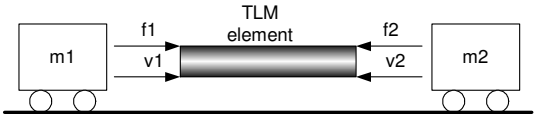

Figure 9. The TLM element communicates the forces and velocities between the masses as energy waves. The direction of the arrows shows the positive direction for energy pulses action on the TLM-element.

The TLM element is a mathematical method to decouple a system into two parts. In the case above, the two mass systems `m1` and `m2` have been decoupled and the TLM element will serve as the communication between these two systems. Some reasoning around how a real physical TLM element would work can motivate the mathematics of the TLM element.

Suppose the TLM element consisted of some elastic medium, of length *L*. A force impulse f_1 on the end of the medium will create a wave in the medium that propagates through the medium and reaches the other end. The time for this wave to reach the other end is the TLM delay time T_{TLM}. The main principle of the TLM element is that any action in one system will be propagated with a certain delay. Therefore, no direct impact (at the same time instance) between the two systems is possible. This decouples the simulation of the two systems.

Apart from the time delay the TLM element must propagate the correct information about the behaviour of the systems at the ends. By examining the PDE of

the wave propagation assuming no energy loss in the TLM element, the TLM interaction can be formulated using the force and velocity at each end, and to form equations for how they interact. These equations turns out to be fairly simple since the details inside the TLM is not important just the "interface" at the ends.

$$\begin{aligned} f_1(t) &= f_2(t-T) + Z\nu_2(t-T) + Z\nu_1(t) \\ f_2(t) &= f_1(t-T) + Z\nu_1(t-T) + Z\nu_2(t) \end{aligned} \quad (1)$$

The most important property with the transmission line is the modelling of the time delay of the signals that enters a TLM element. This time delay can be used to separate the simulation into several parts that can be distributed over several different computers. In this way, it is possible to simulate the parts separately and communicate between the simulations at communication points. Using transmission line elements no numerical error is introduced, as opposed to the multi-rate integration scheme. The errors introduced are due to the separation of the model comes from modelling error instead. The TLM element is symmetric in regard to both separated simulations and no regard needs to be taken to which side is the fastest. The TLM element properties can be determined based on physical considerations only.

In the example above, an elastic medium was used to describe a time delay, but this can of course be anything that provides some reasonable time delay compared to the typical time constants in the simulations. Steel can therefore also be considered as a medium if the simulation time steps are in the same order as the time required for a sound wave to propagate through the steel material.

3.1.2 Benefits of TLM-element

The most obvious reason for using TLM is that it provides a structured way to connect to simulators in a co-simulation set-up. The TLM element does not introduce any numerical problems for the solvers. The price to pay is that a new component has to be introduced in the model. This component behaves as a medium for the communicating "wave" between the simulated processes. This therefore add a model error, but that error is always explicit and well defined compared to errors introduced "under the hood" due to numerical issues in the simulations.

In order to avoid the impact of the model error from the TLM-element it is always important to let the TLM-part represent a natural elasticity in the total system. Then the TLM-element can be tuned to adapt (or model) that elasticity as much as possible, making the model error as small as possible.

3.1.3 TLM-element formulas

Since the TLM element is a continuum, oscillation frequencies can be introduced into the simulation.

$$f_i = \frac{i}{2T_{\mathrm{TLM}}} \quad (2)$$

where T_{TLM} is the time delay factor, and *i* a positive integer. If this cause problems, a damping term can be added in the TLM formulation.

For a simple model of a steel beam, the spring constant of a TLM element can be computed as

$$k = \frac{EA}{L_0} \quad (3)$$

where *E* is Young's modulus, *A* is the cross section area and L_0 the length of the beam. The impedance Z_{F} also has a relation to the spring constant *k*, $Z_{\mathrm{F}} = k \cdot T_{\mathrm{TLM}}$. The impedance factor can then be formulated as a function of the area and length of the beam according to

$$Z_{\mathrm{F}} = \frac{EA\,T_{\mathrm{TLM}}}{L_0} \quad (4)$$

It can be shown that the TLM element also introduces a mass (*parasitic mass*) that can be viewed to be outside the simulated system. The total mass for the combined systems must therefore also include the parasitic mass of the TLM element in order to make, e.g., the energy conservation formulas correct. This mass depends on the impedance factor and the time delay factor

$$m_{\mathrm{p}} = Z_{\mathrm{F}} \cdot T_{\mathrm{TLM}} \quad (5)$$

This implies that if the impedance factor Z_{F} is increased, the parasitic mass will increase if the synchronization delay T_{TLM} is not decreased.

3.1.4 TLM-parameters

The TLM-parameters for the application case, i.e., grinding spindle (see section 2.2), were determined from real physical properties of a chosen connection volume, i.e. the part of the model that will be represented by a TLM-element and applying the formulas above. The chosen shape of the modelled volume depends on application. If a cube with a specific length, L_0, is assumed, then for the volume gives the stiffness according to Eq. (3) is:

$$k = \frac{EL_0^2}{L_0} = EL_0 \quad (6)$$

For rotation stiffness, one way is to assume two springs of stiffness $k/2$ spaced with distance L_0. The time

constant T_{TLM} is computed from the specific length and the speed of sound in the material

$$T_{\mathrm{TLM}} = \frac{L_0}{v} \tag{7}$$

Using steel as the connection medium gives $E = 2.1 \cdot 10^{11}\ \mathrm{Pa}$, $v = 5180\ \mathrm{m/s}$, and $\rho = 7.8 \cdot 10^{3}\ \mathrm{kg/m^3}$.

Calculations like these give approximate values of the stiffness and the time delay of the TLM element. This gives a foundation for choosing these parameters. In the reference case for the analysis of the grinding spindle, the stiffness and time delay parameters were chosen to $k = 1 \cdot 10^{10}\ \mathrm{N/m}$ and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$. Two other cases were analysed where the time delay was halved and doubled respectively, compared to the reference case, see Table 1.

Table 1. Examples of TLM-parameters used in the grinding spindle case. The stiffness was $k = 1 \cdot 10^{10}\ \mathrm{N/m}$.

Case	m_{p} [kg]	Z_{F} [Ns/m]	T_{TLM} [s]
Ref.	1.0	$1.0 \cdot 10^{5}$	$1.0 \cdot 10^{-5}$
Low T_{TLM}	0.25	$0.5 \cdot 10^{5}$	$0.5 \cdot 10^{-5}$
High T_{TLM}	4.0	$2.0 \cdot 10^{5}$	$2.0 \cdot 10^{-5}$

Decreasing the stiffness is one way to decrease the parasitic mass. However, before this is done, the stiffness of the surrounding, serially connected, parts must be estimated in order to avoid decreasing the stiffness of the TLM-connection too much. Another action is to change the delay time. Increasing the delay time makes it possible for the solver to take larger time steps and thus decrease the execution time. However, the parasitic mass is a second order function of the delay time, which has a negative influence on the simulation results. A positive effect is that an increase of the delay time counteracts a decrease of the stiffness.

3.2 Related work

So far, co-simulations gained most acceptance for the mechatronic applications. Developers of control systems routinely utilize co-simulations to test the control algorithms. The most common environment that serves as a coupling framework is SimuLink [16]. Simulation systems that support multibody dynamics simulations like MSC.ADAMS [2] or Dymola [17] provide special modules for running such co-simulations. Other competitive environments like CosiMate [18] are also available. The central issue in such simulations is often real-time performance since the models are eventually used for the hardware-in-the-loop simulations. High Level Architecture (HLA) standard [19] is designed with the vision of distributed simulations in mind. The goal here is to simulate a complete environment with many different actors and tools. The main concerns are not simulation methods but instead protocols that allow individual actors to communicate with the environment. The focus of this paper is different. The intention is to couple transient simulation of mechanical components. The main issues in this case are stability and accuracy of the coupling method. Let us provide a brief survey of the methods used for coupling of dynamics simulations. The intention is to further motivate the choice of the TLM technique for the framework. Otherwise a wider recent survey can be found in [8]. The same paper provides a classification of some of the co-simulation strategies depending on the interface variables exchanged between the sub-models and the co-simulation coordinator. The problem with setting kinematic variables into an existing simulation codes is identified. This motivates the preference for the approaches that compute only reaction forces. The authors also classify the time stepping methods typically used in co-simulations. The parallel time stepping where the compatibility between the sub-systems is achieved at every global time step is identified as the most attractive. Glue code [20, 21] is a co-simulation framework developed for connecting MSC.ADAMS to other codes with discrete or continuous time stepping. It uses quadratic interpolation to pass information about ADAMS model to the other code and quadratic extrapolation for the force coming into the ADAMS model. The latter article introduces a concept of interface mass. The interface mass is simulated in the both coupled codes and gets different reaction forces in different simulations. The reaction forces are communicated between the tools. The coupling framework was successfully used in coupling an ADAMS model to a flight simulator. The problem with this coupling approach is the lack of study on the numerical stability of the method. That leads to the necessity for additional 'tweaking' for each particular case. The paper [22] is mostly focused on fluid-structure interaction problems but the results can be applied for other kinds of co-simulations as well. The paper analyses different numerical coupling methods and a new block-Newton method is proposed. The discussion here is focused on global solution of implicitly coupled differential equations. The introduction of the global iterative solver requires a close interaction with the numerical solvers in the sub-models and may be hard

to implement for existing simulation codes. A similar problem arises for the gluing algorithm presented in [8]. The algorithm relies on the information available at the subsystem interfaces but it requires a special evaluation of the coupling matrices. This, again, may be hard to implement in an existing environment. In [23] a control block, called Boundary Condition Coordinator, is used to minimize the difference in the coupled variables. The mismatch in the conditions does not have any physical meaning and should be seen as a numerical artifact. Additionally the paper does not address the problems related to the use of multi-step implicit solvers for the sub-systems. The TLM technique combines the following attractive properties:

- Very clear and small interface to the sub-models where kinematic data is only read and the response is introduced via an external reaction force and torque.
- Proven unconditional stability of the coupling method.
- Coupling parameters are physically motivated and no additional numerical error is introduced.

The outlined combination of characteristics makes the technique an attractive candidate for a co-simulation.

3.2.1 Example: Co-simulation of a bouncing ball

To illustrate the problem of co-simulation, a co-simulation of a system with two masses according to Figure 10 is carried out. The set-up is a ball (1 kg), bouncing on a floor with mass 100 kg suspended with a spring 1000 N/m. The bounce of the ball is modelled by a Hertz contact, i.e. the acting force on the body when hitting the floor is $10^{10}(-\Delta)^{3/2}$, where Δ is the geometrical intersection.

Figure 11 shows the plots of the simulation where each body is simulated with a separate simulator that is connected using a simple co-simulation technique. The integrator that has integrated the shortest time period will take a step and extrapolate the motion from the other body. The extrapolation uses a Hermite interpolation scheme, i.e. a third order formula interpolating the position and the derivate of the other body in the two previous integration points. The co-simulation was run with two different tolerances (absolute and relative tolerance of the solution vector) of the solver. The topmost plot in Figure 11 represents a tolerance of 10^{-6} and the lower plot represents a tolerance of 10^{-8}. The behaviour of the lower plot emerges because the simulation process for the floor extrapolates the ball position too far, computes the reaction force and moves the floor downwards. When the simulation process for the ball reaches the place where the floor should be, a little time after the extrapolated time that the floor computed, the floor has already started to move downwards. The ball process extrapolates the movement of the floor, and finds that it will not hit the floor this time and continues to fall. The floor process will then extrapolate the position of the ball again and will react too much. At some point the ball will hit the floor and continue to hit the floor several times on the way upwards. This behaviour is very hard to predict, since it did not occur with a lower tolerance, thus rendering this type of co-simulation useless for our purpose.

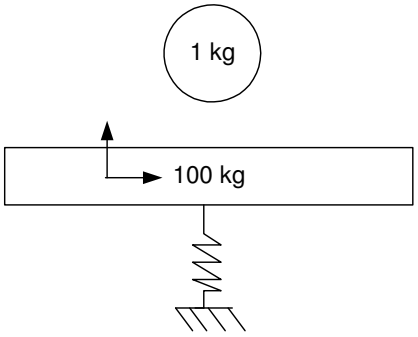

Figure 10. A ball with mass 1 kg is bouncing on a floor (mass of 100 kg) suspended with a spring 1000 N/m. Only the ball is exposed to gravity.

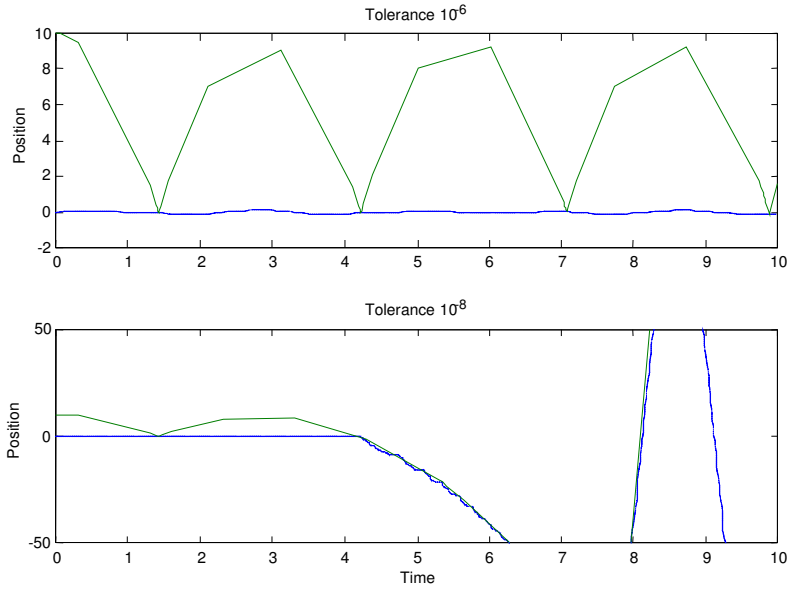

Figure 11. Co-simulation of the two bodies in Figure 17, which are interacting with a Hertz contact between them. The plots show the vertical position of the bodies where the small body (ball) is bouncing with highest altitude for tolerances 10^{-6} and 10^{-6} respectively.

4. Results from grinding spindle application case

A number of different parametric studies have been done for the grinding spindle case, described in section 2.2, and some results are presented here. Different combinations of TLM delay time, stiffness and damping parameters were used. The aim of the studies was a comparison between analyses with and without the TLM-connections. Note, that TLM elements are used in place of springs and dampers. The assumption that exactly the same results should be calculated in both models is misleading. Differences are caused by the presence of parasitic mass, see Eq. (5). However, this may actually be a more relevant description of the physical system, than mass less springs.

In Figure 12-Figure 14 views of the normal force between the inner ring and one of the balls are shown. This force is a fundamental parameter in rolling bearing analysis. The three diagrams represent different TLM-times, each compared with the non-TLM case. The normal forces are varying as the balls are passing through the loaded and unloaded zones. The variation of these zones is solely determined by the rotation of the shaft, which is a motion dominating the vibration of the different bodies in the model. Therefore, no significant difference between the cases could be identified.

The difference between the TLM-time affects as well the size of the time-steps the solver can take. Each TLM case is compared to a non-TLM case where the same time intervals have been used for the maximum time step.

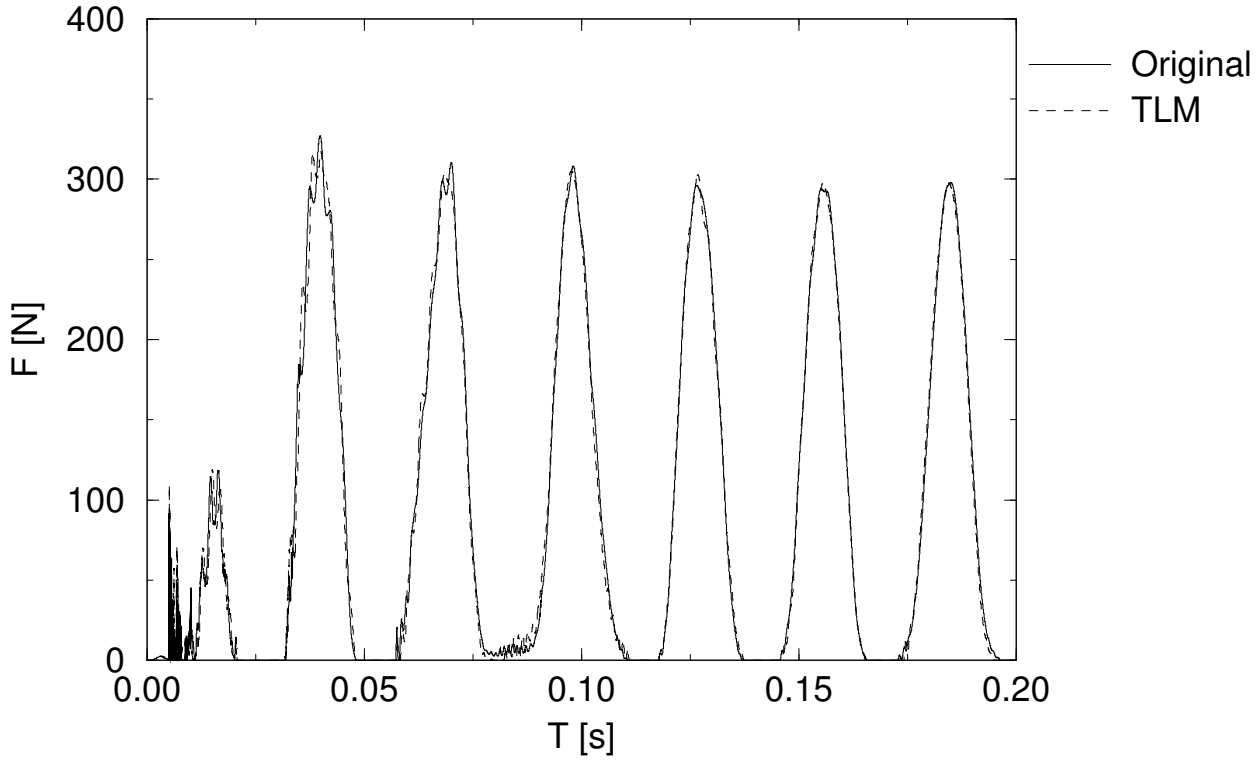

Figure 12. The normal force in the contact between the inner ring and a ball for $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}$ s .

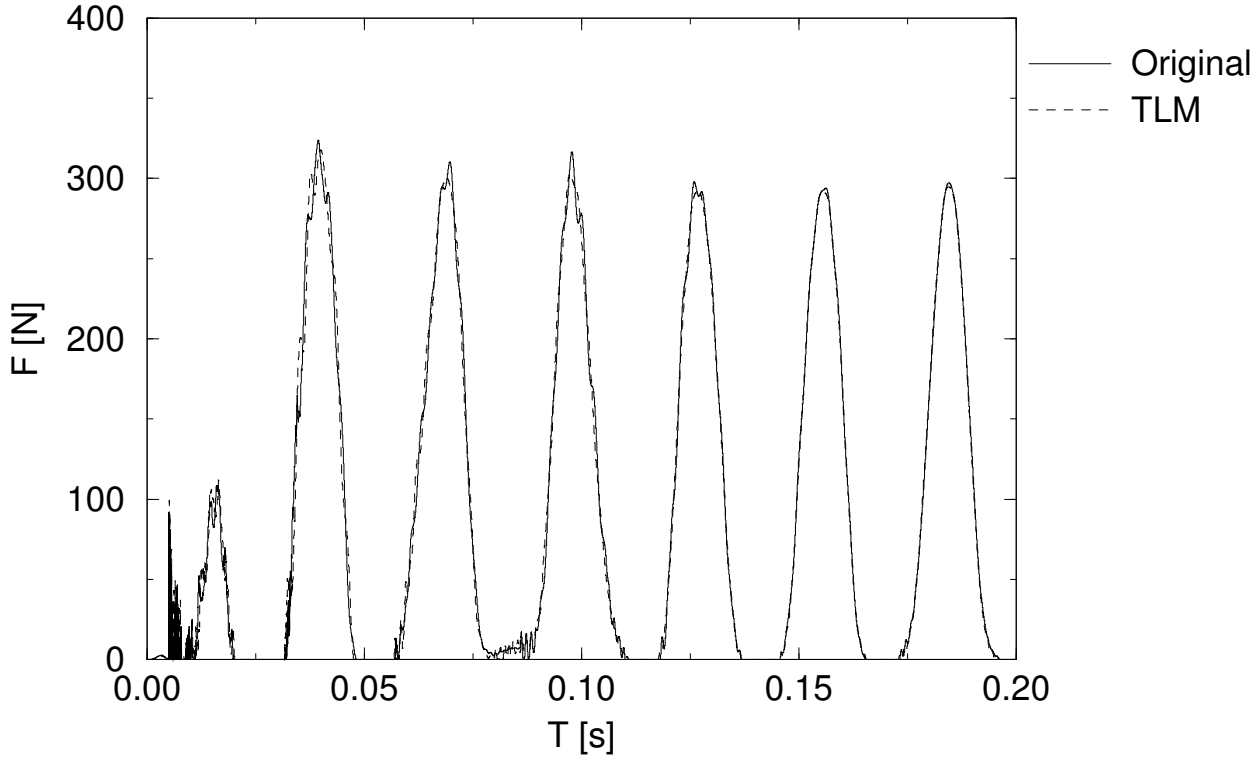

Figure 13. The normal force in the contact between the inner ring and a ball for $T_{\mathrm{TLM}} = 0.5 \cdot 10^{-5}$ s .

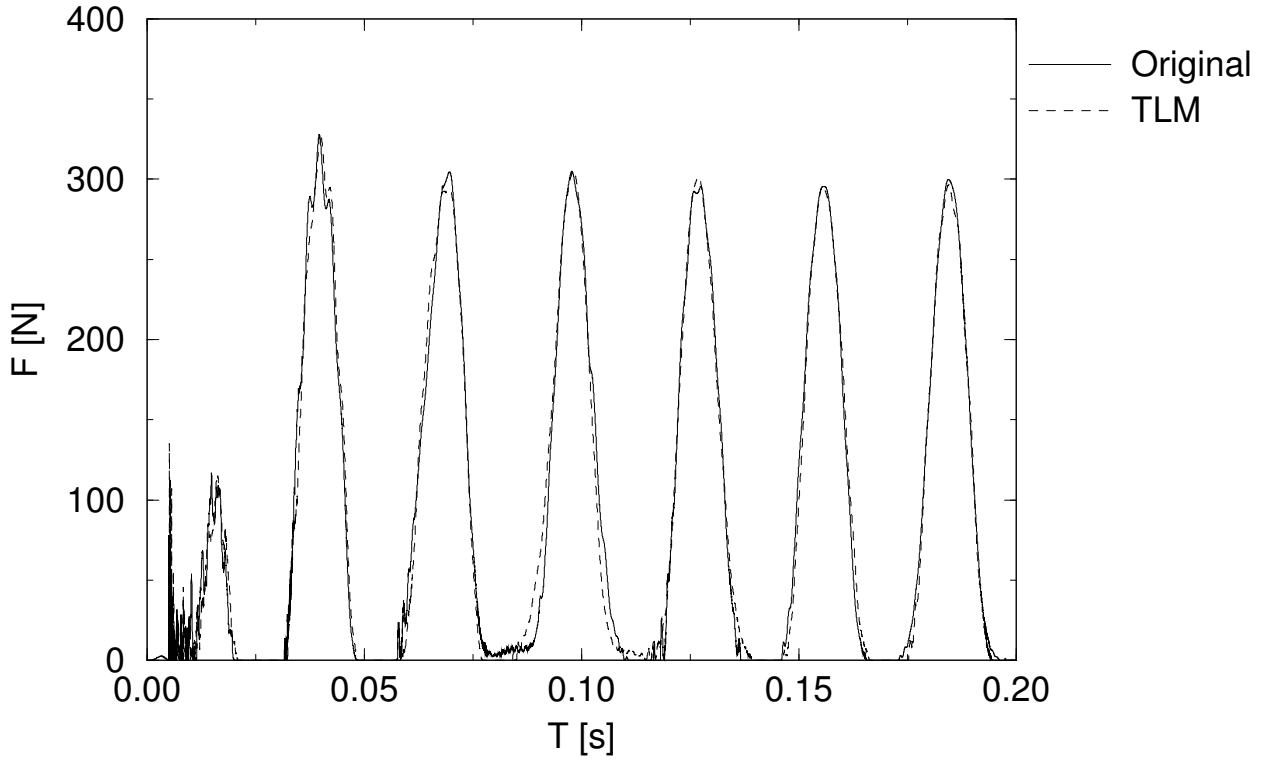

Figure 14. The normal force in the contact between the inner ring and a ball for $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$.

Another set of simulations was performed in order to find out how to choose the TLM parameters and how they influence the results compared to non-TLM simulations. Different combinations of the TLM delay time (T_{TLM}), the stiffness, and the damping were used. The damping parameter was only used for the non-TLM simulations since no corresponding parameter exist in a TLM-connection. The stiffness values were $k = 1 \cdot 10^{9}\ \mathrm{N/m}$ and $k = 1 \cdot 10^{10}\ \mathrm{N/m}$. The TLM delay times were $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}\ \mathrm{s}$ and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$. Finally, the relative damping coefficients used in the non-TLM simulations were $c = 1$, $c = 0.1$ and $c = 0.01$. Here $c = 1$ corresponds to the critical damping for an oscillating system with a mass of 1 kg. In these tests, the non-dimensional tolerance was 10^{-7}, which is a typical value for rolling bearing applications. The parasitic mass varies from 0.1 kg to 4.0 kg for the four different TLM cases.

Here, some diagrams are presented showing the results in terms of the positions of the shaft and the cage, with four graphs shown in the same diagram. The four graphs represent, for a certain combination of TLM delay time and stiffness, three non-TLM simulations (one for each damping coefficient) and one TLM simulation. The legends denoted `c1`, `c01` and `c001` represent the relative damping coefficients $c = 1$, $c = 0.1$ and $c = 0.01$ respectively.

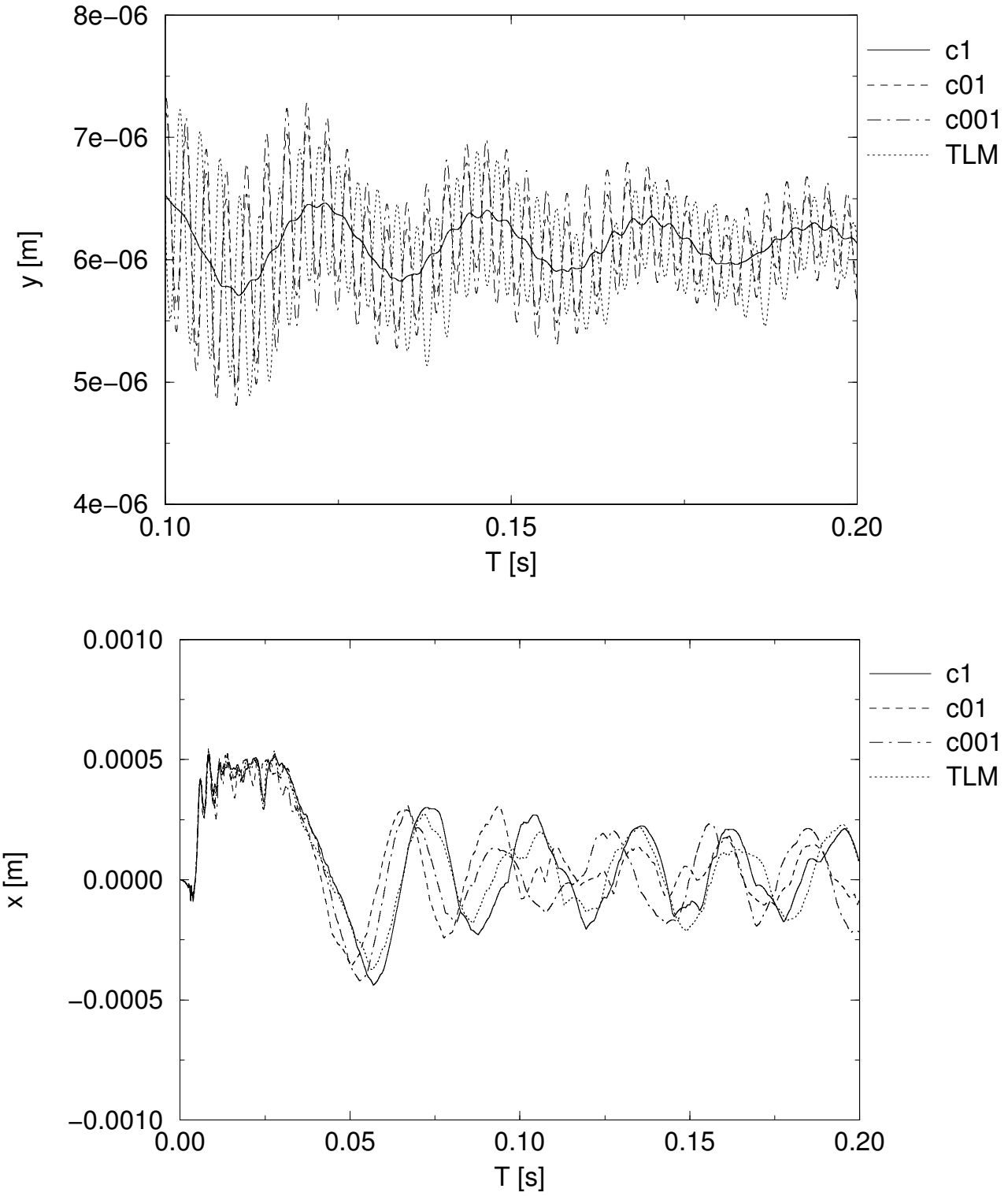

Figure 15. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^9$ N/m and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}$ s .

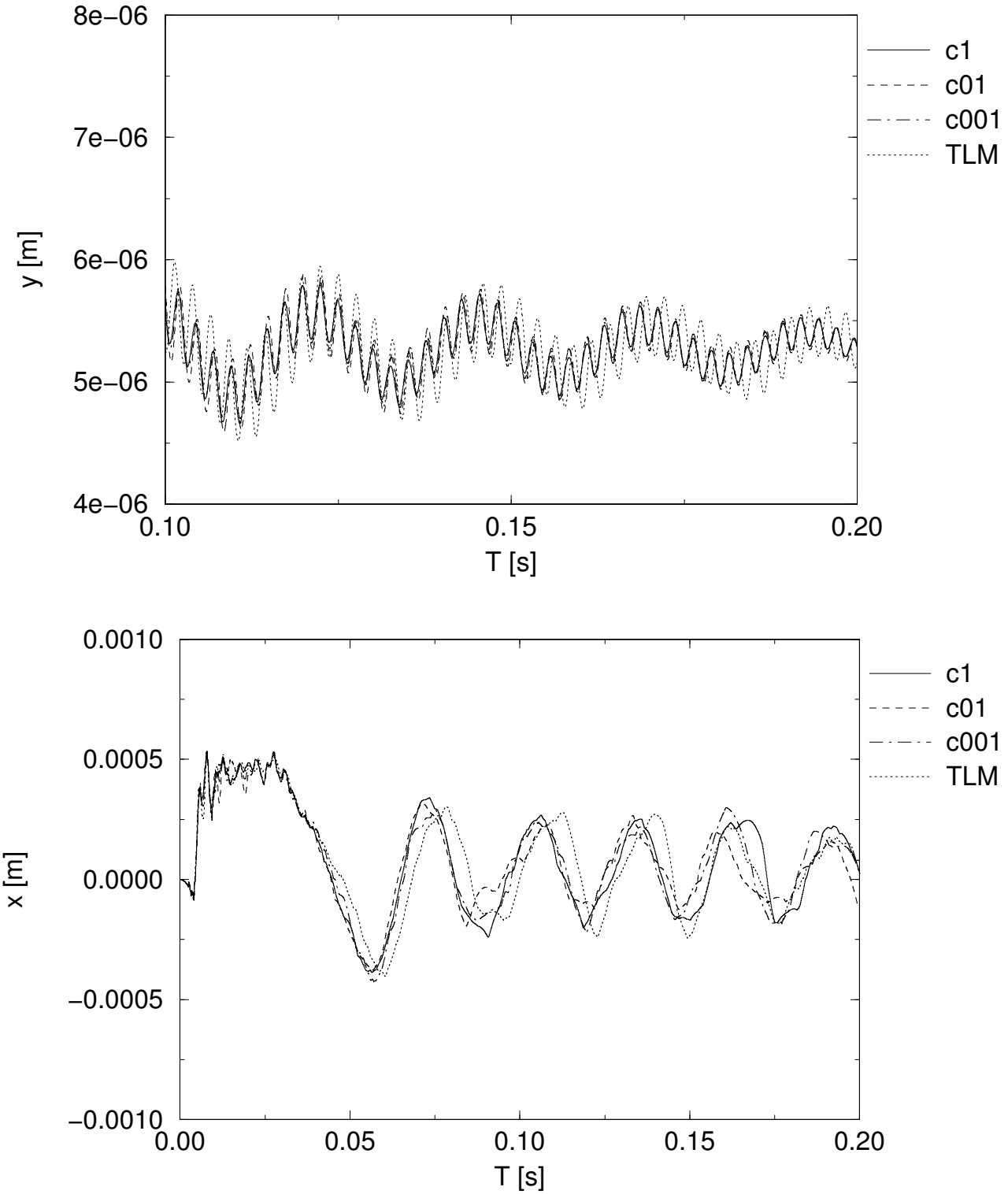

Figure 16. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^{10}$ N/m and $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}$ s .

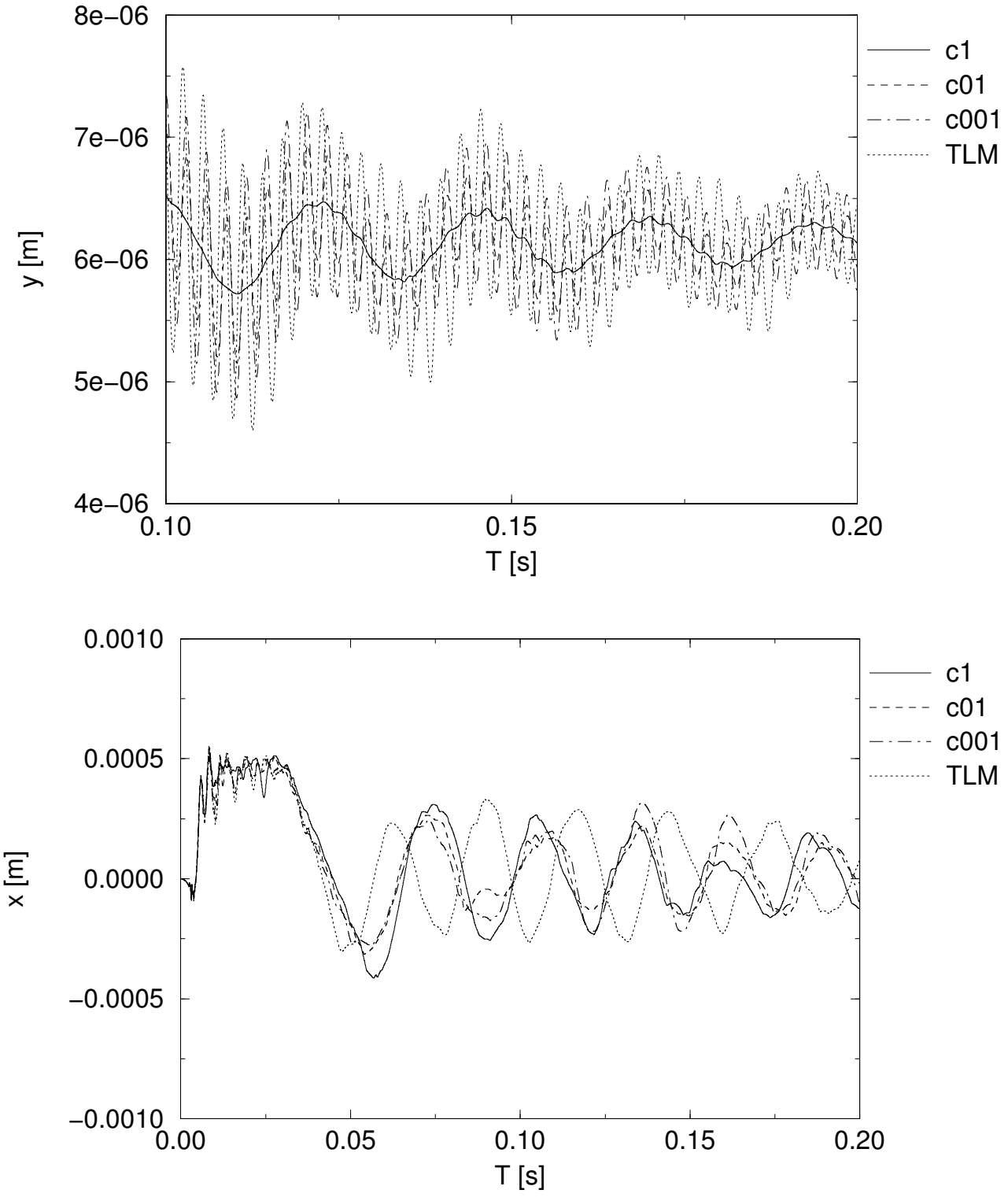

Figure 17. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^9\ \mathrm{N/m}$ and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}\ \mathrm{s}$.

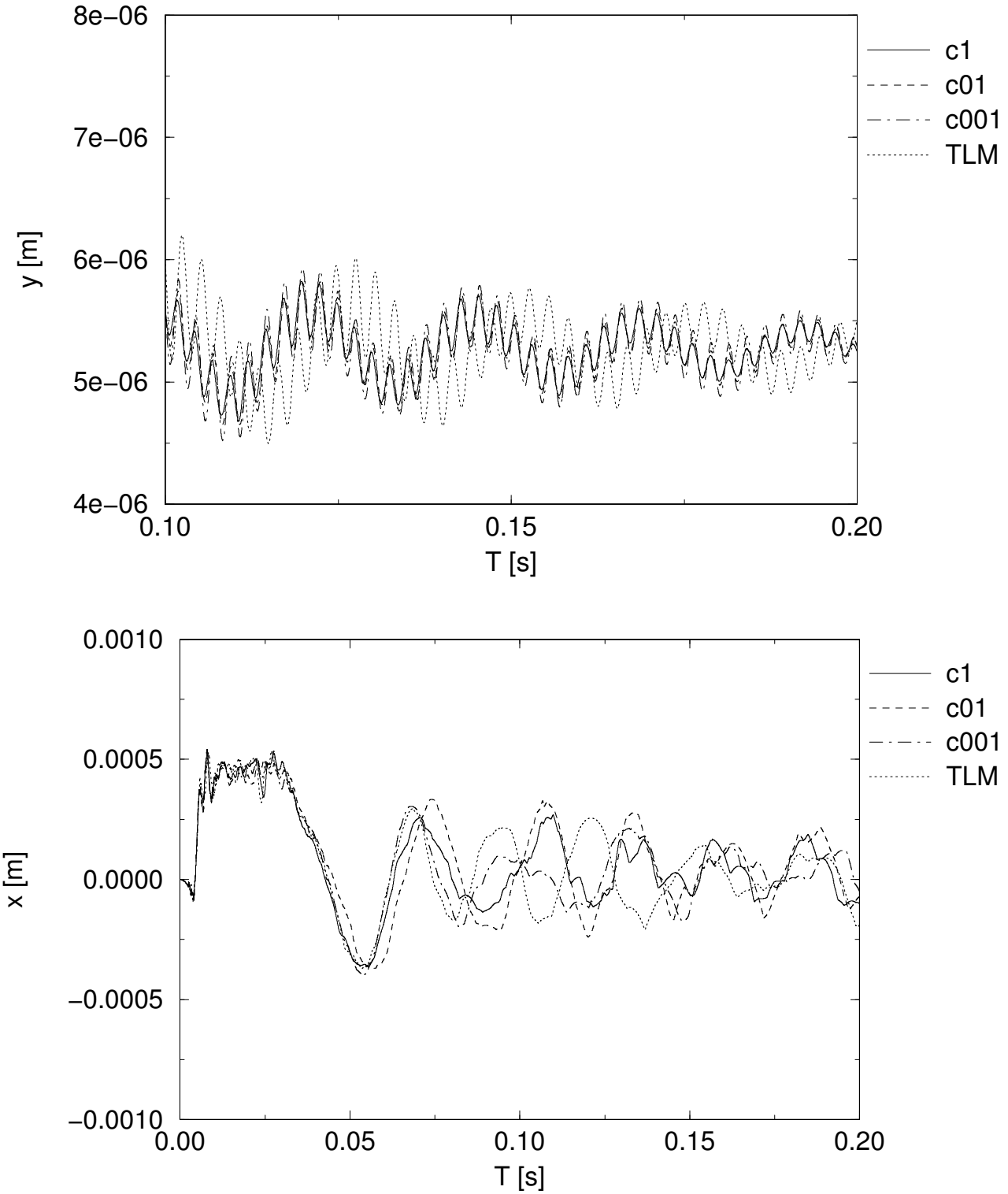

Figure 18. The position of the shaft in the horizontal (y) direction in the upper graph. The lower graph shows the vertical position of the cage. The load case were $k = 1 \cdot 10^{10}$ N/m and $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}$ s.

The case with a stiffness of $k = 1 \cdot 10^{10}$ N/m and a TLM delay time of $T_{\mathrm{TLM}} = 1 \cdot 10^{-5}$ s is referred to as the reference case. Here the results are almost independent of the damping as clearly seen in Figure 16. The agreement between the non-TLM and the TLM analyses is good. If the stiffness is decreased, the agreement is still acceptable, at least if low damping is used. It is also natural that the oscillations increase if the stiffness of the system is decreased. An increased TLM delay time causes larger difference between the TLM analysis and the non-TLM analyses, which is especially visualised in the top diagram in Figure 18. However, this may again be explained with that it is two different systems from a physical point of view. A larger TLM delay time makes the parasitic mass larger and thus increases the difference between the two systems. It might be of some interest to increase the TLM delay time since the solver may then take longer steps and consequently decreases the calculation time. It is then possible to limit the negative influence on the results by decreasing the stiffness since this parameter counteracts the effect of an increased TLM delay time.

5. Numerical performance and stability

In this section the performance and stability of the grinding spindle test case will be analysed in detail.

In any study like this where there is a large number of parameters that might influence the results and a very large number of combinations, one can for practical reasons only investigate a subset of such combinations. A large number of simulations have been done, and below only a few of the most relevant ones are presented.

For practical reasons, all the simulations have been done in serial mode. Running serial instead of parallel BEAST calculations does not affect the analysis. All the necessary measurements that were carried out to approximately predict the behaviour in parallel model, led to this conclusion. For this application, a speedup of a factor 7-11 is expected.

When the whole application or system is simulated not using TLM, it is denoted “BRG+HS”. Using TLM means that two simulation processes communicate. One is the bearing sub-system denoted “BRG TLM” and the other the rest of the grinding spindle sub-system denoted “HS TLM”.

The parasitic mass of the TLM connection, given in the table captions, can be compared with the mass of the spindle shaft 27.8 kg, the housing 48.0 kg, the bearing outer ring 0.51 kg, and the bearing inner ring 0.33 kg.

The solver non-dimensional tolerance on state is $1.0 \cdot 10^{-7}$, which gives dimensional values of $8.334 \cdot 10^{-10}$ [m], $1.0 \cdot 10^{-7}$ [Euler parameters], $1.0 \cdot 10^{-5}$ [m/s], and $1.2 \cdot 10^{-3}$ [rad/s]. The tolerance is typical for a rolling bearing analysis.

Looking at the numerical performance for these TLM cases, the general conclusion is that the performance of the solver is similar for all cases, i.e., the maximum difference in number of RHS calls for the cases was less than 10 %. The case with a stiffness of $k = 1 \cdot 10^{10}$ N/m and a TLM delay time of $T_{\mathrm{TLM}} = 2 \cdot 10^{-5}$ s was the fastest. Consequently, the solver is relative unhindered to do its job. A further increase in $T_{\max}$ will give some performance increase in the order of 10 %.

For the non-TLM cases it is seen that the influence of a double $T_{\max}$ is in the order of 1-15 % fewer RHS calls and thus the solver is also here relative unrestricted and this is the least sensitive parameter. The most sensitive parameter is the damping; for the high stiffness the increase of RHS calls was about 90 % from critical (1.0) to low damping (0.01). Corresponding increase for the low stiffness was about 20 % from critical (1.0) to low damping (0.01).

Comparing the number of RHS calls for the TLM cases with the non-TLM cases, shows that the variation is much larger for the non-TLM cases that have values both lower and higher than the TLM cases. Compared to a typical TLM case, it is from minus 26 % to plus 42 %. It is the difference in physical properties of the models that give these results.

In Table 2-Table 3, two examples of TLM simulations compared with the medium non-TLM damping (0.1) are given. It serves as examples that the numerical performance and stability is good.

Table 2. Case with a stiffness of $k = 1 \cdot 10^{10}\ \mathrm{N/m}$, a damping ratio of 0.1, with $T_{\mathrm{TLM}} = 1.0 \cdot 10^{-5}\ \mathrm{s}$, $T_{\max} = T_{\mathrm{TLM}}\,/\,2.01$, $T_{\mathrm{sync}} = T_{\mathrm{TLM}}\,/\,2$, and $T_{\mathrm{write}} = T_{\mathrm{TLM}}$. Parasitic mass is 1.0 kg.

Description	BRG+HS	BRG TLM	HS TLM
Number of steps	71042	72436	40205
Number of RHS calls	88321	88479	40347
Number of Jacobian calls	1253	1270	671
Failed convergence	4	7	0
Failed error test	3725	3175	1
Unsuccessful RHS calls/step %	10.1	8.51	0.005
Jacobian calls/step %	1.76	1.75	1.67
Time in solver %	0.42	0.0569	0.0122
Time in writing (with extra RHS) %	23.3	43.8	0.466
Time in RHS %	71.0	50.6	3.28
Time in Jacobian %	5.18	3.64	0.00374
Time in TLM com. %	0	1.75	99.45
Mean solver order	3.39	3.17	1.99

Table 3. Case with a stiffness of $k = 1 \cdot 10^{9}\ \mathrm{N/m}$, a damping ratio of 0.1, with $T_{\mathrm{TLM}} = 2.0 \cdot 10^{-5}\ \mathrm{s}$, $T_{\max} = T_{\mathrm{TLM}}\,/\,2.01$, $T_{\mathrm{sync}} = T_{\mathrm{TLM}}\,/\,2$, and $T_{\mathrm{write}} = T_{\mathrm{TLM}}$. Parasitic mass is 0.4 kg.

Description	BRG+HS	BRG TLM	HS TLM
Number of steps	56995	68570	20129
Number of RHS calls	75217	88086	20234
Number of Jacobian calls	1028	1214	336
Failed convergence	6	6	0
Failed error test	3884	3930	2
Unsuccessful RHS calls/step %	13.2	11.1	0.02
Jacobian calls / step %	1.80	1.77	1.67
Time in solver %	0.481	0.0643	0.007
Time in writing (with extra RHS) %	17.2	42.6	0.21
Time in RHS %	76.6	52.7	0.02
Time in Jacobian %	5.71	3.78	0.002
Time in TLM com. %	0	0.837	99.7
Mean solver order	3.49	3.09	3.89

6. Discussion

The TLM delay time computed from physical properties is of the magnitude that it can influence the simulation performance. That means that the TLM input parameters needs to be selected with care, to avoid an increased simulation time. However, an increase can be compensated by that a TLM co-simulation is in fact a parallel computation.

The overall conclusion is that the performance of TLM and non-TLM simulations are similar for the grinding spindle application, and that the numerical stability and performance is very good.

7. Conclusions

Two test cases have been created, one simple two-body model used for debugging and fundamental testing, and one grinding spindle application used for evaluation of the TLM technique for rolling bearing applications.

The conclusion is that the results of TLM and non-TLM simulations are similar for a realistic application with rolling bearings and applicable for engineering purposes. The numerical stability and performance is good.

8. References

[1] Modelica standard. *http://www.modelica.org*.

[2] MSC Software Corporation homepage. *http://www.mscsoftware.com/*.

[3] **Fritzson, D., Stacke, L.-E., and Nordling, P**. (1999) BEAST - a Rolling Bearing Simulation Tool. *Proc. Instn Mech Engrs*, Part K, 213, pp. 63-71, 1999

[4] **Stacke, L.-E. and Fritzson, D**. (2001). Dynamic behaviour of rolling bearings: simulations and experiments. *Proc. Instn Mech Engrs*, Part J, 213, pp. 499-508

[5] ABAQUS homepage. *http://www.abaqus.com/*.

[6] ANSYS solutions homepage. *http://www.ansys.com/*.

[7] **Felippa, C.A., Park, K. C. and Farhat, C.**. (2001). Partitioned analysis of coupled mechanical systems. Invited Plenary Lecture, 4th World Congress in Computational Mechanics, Buenos Aires, Argentina, July 1998, expanded version in *Comp. Meth. Appl. Mechanics and Engineering.*, 190, 3247-3270

[8] **Hulbert, G.M., Ma, Z.-D., and Wang, J.** (2005). Gluing for Dynamic Simulation of Distributed Mechanical Systems. In Jorge A.C. Ambrosio, editor, Advances in *Computational Multibody Systems*, pages 69–94. Springer.

[9] **Veitl, A., Gordon, T., Van De Sand, A., Howell, M., Valasek, M., Vaculin, O., and Steinbauer, P.**. (1999). Methodologies for coupling simulation models and codes in mechatronic system analysis and design. *Vehicle System Dynamics Supplement*, 33:231–243.

[10] **Johns, P.B. and O'Brien, M.** (1980). Use of the transmission-line modeling (t.l.m.) method to solve non-linear lumped networks. *The Radio and Electronic Engineer*, 50(1/2): 59–70, Jan/Feb 1980.

[11] **Krus, P.** (1999). Modelling of Mechanical Systems Using Rigid Bodies and Transmission Line Joints. *Transactions of ASME Journal of Dynamic Systems Measurement and Control.*, Dec 1999.

[12] **Krus, P. and Jansson, A.** (1990). Distributed simulation of hydromechanical systems. In *Third Bath International Fluid Power Workshop*, Bath, UK.

[13] **Pulko, S.H., Mallik, A., Allen, R., and Johns, P.B.** (1990). Automatic Timestepping in TLM Routines for the Modelling of Thermal Diffusion Processes. *Int. Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 3:127-136, 1990.

[14] **Auslander, D.M.** (1968) Distributed System Simulation with Bilateral Delay-Line Models, *Trans ASME J. Basic Eng*, 1968, pp 195-200.

[15] **Krus, P.** (2000). Distributed Modelling for Simulation of Pneumatic Systems. Tech Report, Linköping University, 2000

[16] Mathworks homepage. *http://www.mathworks.com/*.

[17] Dynasim and Dymola homepage. *http://www.dynasim.com/*.

[18] Cosimate homepage. *http://www.tni-world.com/cosimate.asp/*.

[19] **Kuhl, F., Weatherly, R. and Dahmann, J.** (2000). Creating Computer Simulation Systems. An Introduction to the High Level Architecture. Prentice Hall PTR, 2000

[20] **Elliot, A.S.** (2000). A Highly Efficient, General-Purpose Approach for Co-Simulation with ADAMS. MDI Inc, Nov 2000

[21] **Elliot, A.S.** (2002). Status Update on Advanced, General-Purpose Co-Simulation with ADAMS. MDI Inc, May 2002.

[22] **Matthies, H.G. and Steindorf, J.** (2002). Strong coupling methods (revised version). In Stuttgart Multifield Conference, Stuttgart, Germany, April 2002

[23] **Ben Gum, H., Asada, H., He, X.** (2002). Software Development of Co-Simulation of Algebraically Coupled Dynamic Subsystems without Disclosure of Proprietary Subsystem Models, IMECE, New Orleans, Louisiana, Nov 2002